

Ubuntu Server

Contents

1	In this documentation	3
1.1	Getting started	3
1.2	Networking	3
1.3	Security	3
1.4	Managing your system	4
1.5	Data and storage	4
1.6	Web and mail services	4
1.7	Virtualisation and containers	4
1.8	Performance	5
2	How this documentation is organized	6
3	Project and community	7
3.1	Ubuntu Server tutorial	7
3.2	Ubuntu Server how-to guides	75
3.3	Ubuntu Server reference	768
3.4	Ubuntu Server explanation guides	840
3.5	Contribute to this documentation	1073

Ubuntu Server is a version of the Ubuntu operating system designed and engineered as a backbone for the internet.

Ubuntu Server brings economic and technical scalability to your data center, public or private. Whether you want to deploy an OpenStack cloud, a Kubernetes cluster or a 50,000-node render farm, Ubuntu Server delivers the best value scale-out performance available.

This documentation always targets the latest LTS version of Ubuntu. When there are differences towards older LTS releases, they are noted in the guide.

1. In this documentation

1.1. Getting started

The tutorial walks you through a fresh installation and the first steps of operating an Ubuntu Server.

- **Installation:** [System requirements](#) (page 826) • [Basic installation](#) (page 8) • [Ubuntu Pro subscription](#) (page 73)
- **System basics:** [Managing software](#) (page 47) • [Customizing package files](#) (page 956) • [Third party repository usage](#) (page 950)
- **The Ubuntu command line:** [Welcome to the terminal](#) (page 12) • [The command line in depth](#) (page 32) • [Command-line cheat sheet](#) (page 819)

1.2. Networking

Configure network interfaces, provide name and address services, share resources, and synchronize time.

- **Understanding networks:** [Key concepts](#) (page 913) • [About Netplan](#) (page 917) • [Time synchronization](#) (page 937)
- **Configuration:** [Using Netplan](#) (page 917) • [Domain Name System \(DNS\)](#) (page 299) • [Dynamic Host Configuration Protocol \(DHCP\)](#) (page 331)
- **Securing your network:** [Install OpenSSH](#) (page 235) • [Firewalls](#) (page 163) • [DNS Security Extensions \(DNSSEC\)](#) (page 848)
- **High throughput:** [Open vSwitch with DPDK](#) (page 322) • [About DPDK](#) (page 929)
- **Sharing:** [Files with NFS](#) (page 350) • [Printers with CUPS](#) (page 412)
- **With Windows systems:** [Samba](#) (page 359) • [Windows Active Directory](#) (page 395)

1.3. Security

Protect your system with access controls, firewalls, encrypted communications, and network authentication. For more general information about security in Ubuntu, refer also to the [Ubuntu Security documentation](#)¹.

- **Understanding security:** [Introduction](#) (page 840) • [Security suggestions](#) (page 842)
- **Basic security:** [User management](#) (page 154) • [AppArmor](#) (page 170)
- **Authentication:** [Install Kerberos](#) (page 183) • [Set up SSSD with LDAP](#) (page 214) • [Smart cards](#) (page 223)
- **Cryptography:** [Cryptographic libraries](#) (page 858) • [About certificates](#) (page 901) • [OpenSSL](#) (page 864)
- **VPNs:** [OpenVPN](#) (page 254) • [WireGuard VPN](#) (page 273)

¹ <https://documentation.ubuntu.com/security/>

1.4. Managing your system

Install, update, upgrade, and troubleshoot the software packages running on your system.

- **Managing software:** [Package management](#) (page 416) • [etckeeper](#) (page 557) • [Get older package versions using snapshots](#) (page 438)
- **Update considerations:** [Automatic updates](#) (page 423) • [Phased updates](#) (page 963) • [Testing updates in advance](#) (page 966)
- **Upgrade Ubuntu version:** [Release upgrades](#) (page 435) • [Upgrade a virtual machine](#) (page 1013)
- **Troubleshooting:** [Report a bug](#) (page 444) • [Debugging with debuginfod](#) (page 746) • [Kernel crash dump](#) (page 752)

1.5. Data and storage

Store and manage data using directory services, databases, logical volumes, and backup tools.

- **Databases:** [MySQL](#) (page 498) • [PostgreSQL](#) (page 505)
- **Directory services:** [Introduction to OpenLDAP](#) (page 969) • [Install OpenLDAP](#) (page 449) • [OpenLDAP replication](#) (page 464)
- **Device mapper multipathing:** [Intro to multipathing](#) (page 973) • [Multipath configuration options and overview](#) (page 978)
- **Storage:** [About LVM](#) (page 971) • [Manage logical volumes](#) (page 510) • [Sharing data storage with iSCSI](#) (page 513)
- **Version control:** [gitolite](#) (page 559)

1.6. Web and mail services

Serve web content, proxy traffic, write server-side applications, and handle email.

- **Web servers:** [About web servers](#) (page 1001) • [Apache2](#) (page 589) • [nginx](#) (page 598)
- **Squid proxy server:** [About Squid proxy](#) (page 1002) • [Install a Squid proxy server](#) (page 584)
- **Mail services:** [Postfix](#) (page 564) • [Exim4](#) (page 572) • [Dovecot](#) (page 581)

1.7. Virtualisation and containers

Run workloads in virtual machines and containers using lightweight and full-virtualisation tooling.

- **Understand virtualization:** [Intro to virtualisation](#) (page 1004)
- **Virtual machines:** [Multipass](#) (page 633) • [QEMU](#) (page 641) • [QEMU microvm](#) (page 1007)
- **VM tooling:** [libvirt and virsh](#) (page 660) • [virt-manager](#) (page 673) • [Nested virtualization](#) (page 680)

- **Containers:** [Docker for system admins](#) (page 687) • [Docker storage, networking, and logging](#) (page 1018) • [Run rocks on your server](#) (page 699)
- **Deployment:** [Find cloud images](#) (page 1025) • [About cloud-init](#) (page 1024) • [LXD](#) (page 625)
- **Confidential computing:** [AMD SEV](#) (page 647) • [Intel TDX](#) (page 649)

1.8. Performance

Set up GPU resources and optimize system performance.

- **GPU resources:** [Install Nvidia drivers](#) (page 611) • [GPU virtualization](#) (page 618)
- **Performance:** • [CPU P-states](#) (page 1057) • [hwloc](#) (page 1052)
- **Optimization:** [CPU power management](#) (page 1061) • [TuneD](#) (page 1067) • [Profile-Guided Optimization](#) (page 1046)
- **High availability:** [Introduction to High Availability](#) (page 1027) • [Set up DRBD](#) (page 708)
 - [Migrate from crmsh to pcs](#) (page 827)

2. How this documentation is organized

This documentation uses the [Diátaxis documentation structure](https://diataxis.fr/)².

- The [Tutorial](#) (page 7) takes you step-by-step through installing and setting up your first Ubuntu Server system.
- [How-to guides](#) (page 75) assume basic familiarity with Ubuntu Server and walk you through specific tasks.
- [Reference](#) (page 768) provides system requirements, a glossary, and other technical specifications.
- [Explanation](#) (page 840) includes topic overviews, background and context, and detailed discussion of key concepts.

² <https://diataxis.fr/>

3. Project and community

Ubuntu Server is a member of the Ubuntu family. It's an open source project that welcomes community projects, contributions, suggestions, fixes and constructive feedback.

- [Read our Code of Conduct](#)³
- [Get community support](#)⁴
- [Join the Discourse forum](#)⁵
- [Chat to us on Matrix](#)⁶
- [Contribute to this documentation](#) (page 1073)

If you find any errors or have suggestions for improvements, please use the “Give feedback” button at the top of every page. This will take you to GitHub where you can share your comments or let us know about bugs with any page.

Releases

[Download Ubuntu Server](#)⁷

[Ubuntu Release Notes](#)⁸

PDF versions of this documentation

[Current PDF](#)⁹ (for Ubuntu 20.04 LTS onward)

[Ubuntu 18.04 LTS PDF](#)¹⁰ (and earlier)

Commercial support

Thinking about using Ubuntu Server for your next project? [Get in touch!](#)¹¹

3.1. Ubuntu Server tutorial

This tutorial should be a good place to start learning about Ubuntu Server in general, how it works, and what it's capable of.

In our “Getting started” tutorial you will learn how to set up an Ubuntu Server; from installing using a bootable USB device, to navigating the Server installer menu.

³ <https://ubuntu.com/community/docs/ethos/code-of-conduct>

⁴ <https://ubuntu.com/support/community-support>

⁵ <https://discourse.ubuntu.com/c/project/server/17>

⁶ <https://matrix.to/#/#server:ubuntu.com>

⁷ <https://ubuntu.com/download/server>

⁸ <https://documentation.ubuntu.com/release-notes/>

⁹ https://ubuntu.com/server/docs/_/downloads/en/latest/pdf/

¹⁰ <https://assets.ubuntu.com/v1/8f8ea0cf-18-04-serverguide.pdf>

¹¹ <https://ubuntu.com/server/contact-us?product=server>

3.1.1. Getting started

- [Basic installation](#) (page 8)

The Server installer

The Ubuntu installer has its own documentation. For more guidance on installing Ubuntu Server with the installer, you can also see these guides in the Ubuntu installer documentation (note: these pages will redirect you).

- [Operating the Server installer](#)¹²
- [Screen-by-screen installer guide](#)¹³
- [Configuring storage](#)¹⁴
- [Troubleshooting the installer](#)¹⁵

3.1.2. Next steps

Once your Ubuntu Server is up and running, you may be interested in this collection of related tutorials and topics that will help you learn more about how it works and what's available to you. These tutorials can be viewed in any order.

- [Welcome to the terminal](#) (page 12)
- [The command line in depth](#) (page 32)
- [Managing software in Ubuntu Server](#) (page 47)
- [Attach your Ubuntu Pro subscription](#) (page 73)

If you have a specific goal, and are already familiar with Ubuntu Server, take a look at our [How-to guides](#) (page 75). These have more in-depth detail and can be applied to a broader set of applications.

Take a look at our [Reference section](#) (page 768) when you need to determine what commands are available, and how to interact with various tools.

Finally, for a better understanding of how Ubuntu Server works and how it can be used and configured, our [Explanation section](#) (page 840) enables you to expand your knowledge of the operating system and additional software.

Basic installation

This chapter provides an overview of how to install Ubuntu Server Edition. You can also refer to this guide on [how to operate the installer](#)¹⁶ for more information on using the installer, and to this [screen-by-screen reference guide](#)¹⁷ for more information about each of the installer screens.

¹² <https://canonical-subiquity.readthedocs-hosted.com/en/latest/tutorial/operate-server-installer.html>

¹³ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/tutorial/screen-by-screen.html>

¹⁴ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/howto/configure-storage.html>

¹⁵ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/howto/report-bugs.html>

¹⁶ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/tutorial/operate-server-installer.html>

¹⁷ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/tutorial/screen-by-screen.html>

Preparing to install

This section explains various aspects to consider before starting the installation.

System requirements

Ubuntu Server Edition provides a common, minimalist base for a variety of server applications, such as file/print services, web hosting, email hosting, etc. This version supports four 64-bit architectures:

- amd64 (Intel/*AMD* 64-bit)
- arm64 (64-bit ARM)
- ppc64el (POWER8 and POWER9)
- s390x (IBM Z and LinuxONE)

The recommended minimal system requirements for this tutorial are:

- RAM: 2 *GB* or more
- Disk: 5 GB or more

If you are looking for more general system requirements, *refer to this page* (page 826).

Perform a system backup

Before installing Ubuntu Server Edition you should make sure all data on the system is backed up.

If this is not the first time an operating system has been installed on your computer, it is likely you will need to re-partition your disk to make room for Ubuntu.

Any time you partition your disk, you should be prepared to lose all data on the disk should you make a mistake or something goes wrong during partitioning. The programs used in installation are quite reliable, most have seen years of use, but they also perform destructive actions.

Download the server ISO

You can obtain the amd64 server download from <https://releases.ubuntu.com/>. Select the version you wish to install and select the “server install image” download. Note that the server download includes the installer.

There are platform specific how-to guides for installations on:

- *s390x LPAR* (page 124)
- *z/VM* (page 99)
- *ppc64el* (page 92)

Create a bootable USB

There are many ways to boot the installer but the simplest and most common way is to [create a bootable USB stick](#)¹⁸ to boot from ([tutorials for other operating systems](#)¹⁹ are also available).

Boot the installer

Plug the USB stick into the system to be installed and start it.

Most computers will automatically boot from USB or DVD, though in some cases manufacturers disable this feature to improve boot times. If you don't see the boot message and the "Welcome" screen which should appear after it, you will need to set your computer to boot from the install media.

There should be an on-screen message when the computer starts telling you what key to press for settings or a boot menu. Depending on the manufacturer, this could be Escape, F2, F10 or F12. Simply restart your computer and hold down this key until the boot menu appears, then select the drive with the Ubuntu install media.

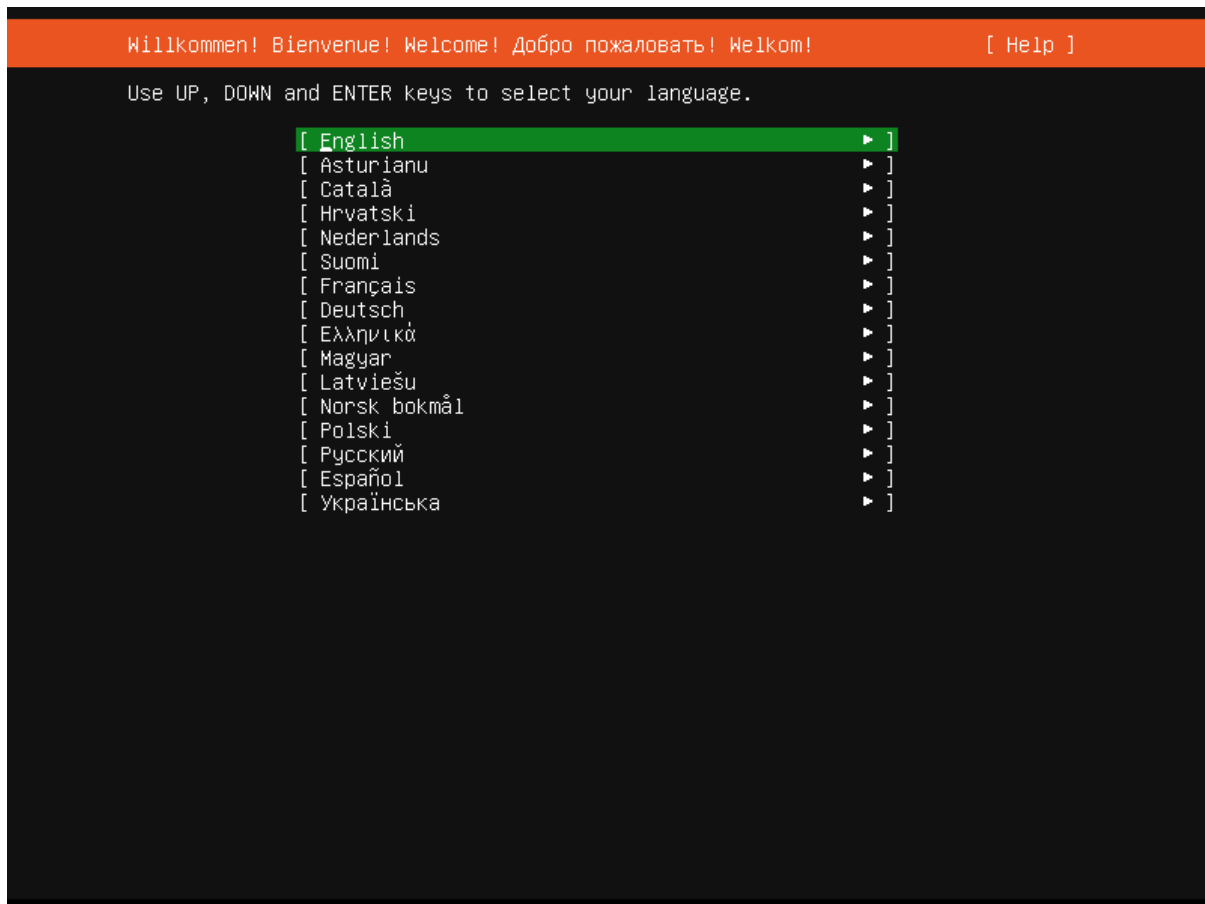
If you are still having problems, check out the [Ubuntu Community documentation on booting from CD/DVD](#)²⁰.

After a few moments, the installer will start in its language selection screen.

¹⁸ <https://documentation.ubuntu.com/desktop/en/latest/how-to/create-a-bootable-usb-stick/>

¹⁹ <https://ubuntu.com/search?q=%22create+a+bootable+USB+stick%22>

²⁰ <https://help.ubuntu.com/community/BootFromCD>



Using the installer

The installer is designed to be easy to use and have sensible defaults so for a first install you can mostly just accept the defaults for the most straightforward install:

- Choose your language
- Update the installer (if offered)
- Select your keyboard layout
- Do not configure networking (the installer attempts to configure wired network interfaces via [DHCP](#), but you can continue without networking if this fails)
- Do not configure a proxy or custom mirror unless you have to in your network
- For storage, leave “use an entire disk” checked, and choose a disk to install to, then select “Done” on the configuration screen and confirm the install
- Enter a username, hostname and password
- On the SSH and snap screens, select “Done”
- You will now see log messages as the install is completed
- Select restart when this is complete, and log in using the username and password provided

If you need more details, however, the Ubuntu installer has [its own documentation](#)²¹, including a [screen-by-screen guide](#)²².

Welcome to the terminal

On Ubuntu Server, the terminal is the primary way of interacting with your system. It's easier to use than you might think, and learning it gives you much more control over your system (including on your Ubuntu Desktop). In this tutorial, you will learn how to use the terminal to navigate your Ubuntu system, find and work with files, and where to find help if you get stuck.

Set up the virtual machine

To avoid making changes to your computer we will set up a virtual machine (VM), which provides us with a safe environment to run the commands in. [Multipass](#)²³ is great for quickly creating Ubuntu virtual machines, so we will use that. It is available on Windows, Mac and Linux, so you don't need to be already running Ubuntu on your computer to follow this tutorial.

Prerequisites

- The Multipass VM needs **5 GB of disk space**, and **1 GB of memory** available.

Install Multipass

On Ubuntu

You can install Multipass directly from [the Multipass page](#)²⁴ in the online snap store (make sure to select the "latest/stable" version from the dropdown menu next to the install button).

On other operating systems

If you're on Windows, Mac, or another Linux operating system, Multipass can be installed [using these instructions](#)²⁵.

²¹ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/index.html>

²² <https://canonical-subiquity.readthedocs-hosted.com/en/latest/tutorial/screen-by-screen.html>

²³ <https://documentation.ubuntu.com/multipass/latest/>

²⁴ <https://snapcraft.io/multipass>

²⁵ <https://documentation.ubuntu.com/multipass/stable/how-to-guides/install-multipass/>

Open a terminal window

Now that Multipass is installed, we need a terminal window to run our commands in. How you open one depends on your operating system.

On Ubuntu

Press `Ctrl + Alt + T` together and a terminal window will open. You can also search for “Terminal” in your application launcher.

On Windows

PowerShell is the most reliable option for working with Multipass on Windows.

1. Press the Windows key.
2. Type PowerShell.
3. Right-click it and select **Run as Administrator** — this helps avoid permission issues when managing virtual machines.

If you have Windows Terminal installed from the Microsoft Store, that works just as well and offers a nicer experience with tab support.

To confirm everything is working, type `multipass version` and press Enter. If you see version numbers, you’re ready to continue.

On macOS

1. Press `⌘ + Space` to open Spotlight.
2. Type Terminal and press Enter.

If you use iTerm2, that works perfectly well too.

To confirm everything is working, type `multipass version` and press Enter. If you see version numbers, you’re ready to continue.

Create the virtual machine

Once you have installed Multipass and opened your terminal, we can launch a new VM using the Ubuntu 26.04 LTS release (codename “resolute”), and let’s give our VM the name `tutorial` using the following command in our terminal window:

```
$ multipass launch resolute --name tutorial
```

It may take a little time, depending on the speed of your internet connection. Multipass downloads the most recent daily **image** and creates the VM for us.

Note:

An Ubuntu **image** is a collection of files we need to install and run Ubuntu. We don't need to specify "server" or "desktop" anywhere in our command, because the image is the same for both. The only difference between Ubuntu Server and Ubuntu Desktop is the subset of software packages we use from the Ubuntu Archive.

Once the above command has finished running, we can access the VM by running:

```
$ multipass shell tutorial
```

We will get a "Welcome to Ubuntu" message. Notice that when we run this command, the terminal username changes to ubuntu and the *hostname* changes to tutorial:

```
ubuntu@tutorial:~ $
```

This shows that we are inside the VM, and this is where we will run all our commands. Now that our setup is complete, let's carry on with the tutorial!

Understanding the terminal

In our new terminal window, we see a **shell prompt** (sometimes called a command prompt), which is constructed as your-username@your-hostname and followed by a dollar sign (\$). This is where we input commands to the computer.

```
your-name@your-machine-name:~$ Input: The commands you type
```

```
Output: The results after running the command
```

The output – the results of running your commands – is shown separately, underneath the shell prompt.

Tip:

You can use your mouse to copy text *into* the terminal by holding the *left mouse button* and dragging your mouse to highlight the text you want to copy. Then, release the left button.

Move your mouse pointer to the terminal window and press the *middle mouse button* (or click the *right mouse button* and select *paste*).

The **Command Line Interface (CLI)** is the language we use in our terminal to interact with our system. The CLI gives you far more control over your system than the **Graphical User Interface (GUI)** that you may be more familiar with.

In software documentation, you will often see instructions to "run" or "execute" a command (or a set of commands). What this means is: type the command into your terminal at the **shell prompt** and press Enter on your keyboard. We did this already when we ran the `multipass launch` command to create our VM.

The command is sent to the **shell**, which interprets the command and sends it to the operating system, which runs the command. It took a little time for the tutorial instance to be

created, but eventually the shell showed output like this:

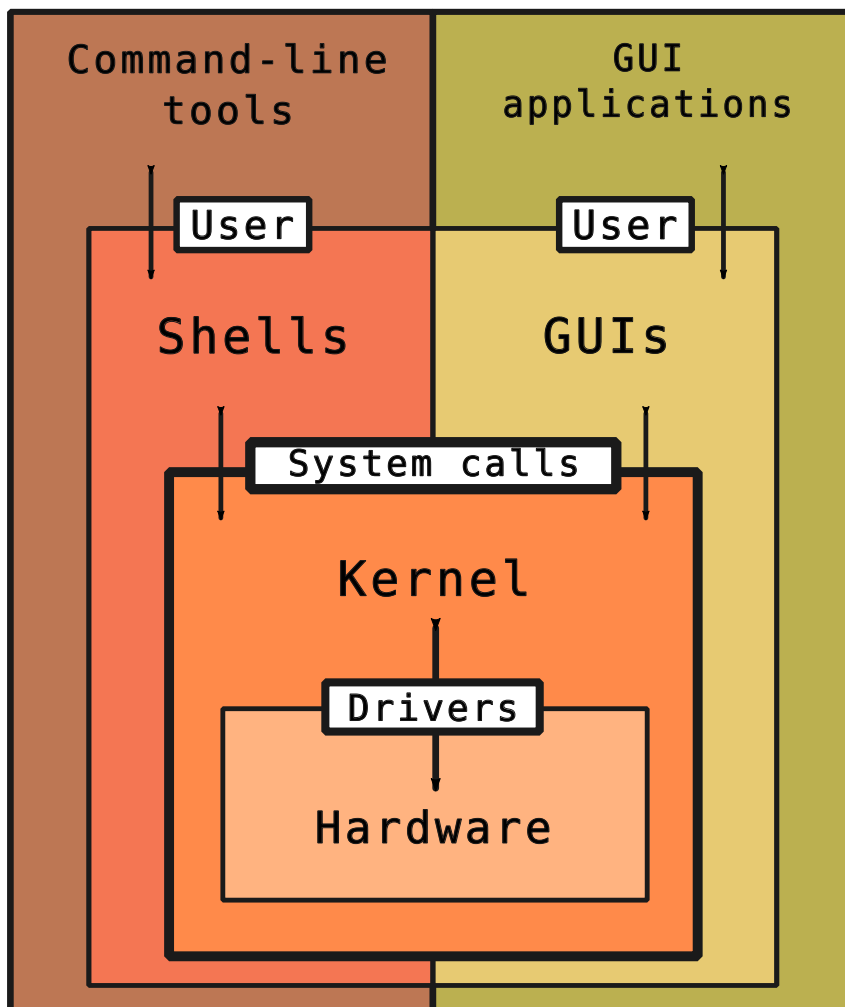
```
user@computer:~$ multipass launch --name tutorial
Launched: tutorial
```

So we know that the VM called tutorial was created successfully!

The shell

You may be wondering: “why we need a shell at all? Why can’t we just tell the operating system what to do using our Desktop environment?”

This is a good question. The simple answer is that while you *can* use the Desktop’s Graphical User Interface (GUI) to interact with your computer, the shell is much more resource-efficient and faster, since it doesn’t need to handle computationally expensive graphics. Servers rarely even have a GUI installed for precisely this reason.



For simple tasks, like renaming a single file, there’s little difference between using your GUI and the CLI to complete it. However, if you need to rename 1,000 files to include the date they were created in the name, this would be a tiresome and time-consuming task to do in the GUI – lots of clicking and typing. Using the CLI, you can pull in the date automatically, set the pattern you want the names to follow, and have all 1,000 files renamed instantly.

The real power and efficiency of the shell can be felt when you need to do something many times. You can even automate routine or large/complex tasks with shell scripts and command chaining – more on that in the next tutorial, though!

The virtual machine provided by Multipass has no Graphical User Interface (GUI). It **abstracts** away all the details of virtualization, which is quite complicated, so we don't have to worry about them. This means we can just open a shell inside our Multipass VM **instance**, which we did when we ran the `multipass shell tutorial` command:

After we ran the command, our terminal window shows us a welcome message like this:

```
ubuntu@tutorial:~$ multipass shell tutorial

Welcome to Ubuntu 26.04 LTS (GNU/Linux 7.0.0-15-generic x86_64)

 * Documentation:  https://docs.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri May 29 17:22:28 BST 2026

System load:  0.17               Processes:            110
Usage of /:   56.8% of 3.70GB    Users logged in:     0
Memory usage: 22%               IPv4 address for ens3: 10.234.118.66
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
10 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

You've probably noticed that your shell prompt has changed, and now says `ubuntu@tutorial:~$` – `tutorial` is the name of our virtual machine (the hostname), and `ubuntu` is the default username in a Multipass virtual machine. This is how we know we're now playing in our virtual sandbox, and not in our physical computer.

Other ways to use commands

As we've now seen, CLI commands are the instructions that we give the computer by typing them into the terminal. In our documentation, commands are formatted like this.

Sometimes, we might want to run a whole set of commands at once. For convenience, we can save them in a file called a **shell script**, in the order we want the shell to execute them. Then, we just need to tell the shell to run the script, and it executes the commands in the script by itself.

As you become more familiar with the command line, you can also begin to define your own commands built from other commands. These are called **aliases**, and they can be very handy shortcuts for small sets of commands you want to run often.

Navigating the filesystem

The **filesystem** is the structure your operating system uses to organize all files and directories on the computer. It's like the folder structure you see on a desktop computer's file manager, except we'll be navigating it using text commands in the CLI.

There are some CLI commands that are so useful, and that we'll use so often, they'll become baked into our memory. First, let's try this one:

```
ubuntu@tutorial:~$ ls
```

Nothing happened, right?

The `ls` command ("list show") shows us a list of all the files in our current folder. This is a brand new virtual machine, though, and we haven't created anything yet, so there's nothing to show! Let's fix that by creating a few files using the `touch` command:

```
ubuntu@tutorial:~$ touch my-first-file.txt my-second-file.txt  
my-third-file.txt
```

Now if we run `ls` again it should show them to us:

```
ubuntu@tutorial:~$ ls  
  
my-first-file.txt my-second-file.txt my-third-file.txt
```

We can also create folders to help keep all our files organized using the `mkdir` command:

```
ubuntu@tutorial:~$ mkdir my-folder
```

Let's run `ls` again:

```
ubuntu@tutorial:~$ ls  
  
my-first-file.txt my-folder my-second-file.txt my-third-file.txt
```

There is our new folder!

The `mkdir` command is short for "make directory". A directory is exactly the same as a folder, but it abbreviates to the conveniently short `dir` so it's used more often. We'll use them interchangeably.

Let's now move the three files we created into `my-folder`, so we can keep our current folder nice and tidy. We can do this using the `mv` command.

Since it doesn't know which files to move, we must give some **arguments** to the move (`mv`) command. Actually, we already did this with our `touch` and `mkdir` commands earlier:

```
touch FILE(S)  
touch <file-name> # Create one file  
touch <file-name-1> <file-name-2> # Create more than one file  
  
mkdir FOLDER(S)
```

(continues on next page)

(continued from previous page)

```
mkdir <folder-name> # To create one folder
mkdir <folder-name-1> <folder-name-2> # To create more than one folder
```

The `mv` command is similar, but takes more arguments:

```
mv FILE(S) FOLDER
mv <file-name> <folder-name> # Move specific file to this folder
mv <file-name-1> <file-name-2> <folder-name> # Move all the named files to this folder
```

You can move any number of files you want. The names of the files you want to move always come right after the `mv` command. The folder you want to move those files to is *always* at the end. That's to say, these are **positional arguments** – the order of where they appear in the command is important.

Note:

It's also worth highlighting the common convention command `[foo] <bar>` for arguments that you'll see in most CLI documentation.

- Square brackets (`[ ]`): anything inside is *optional*, so the command will run without it. This is common for **flags** that change or extend the behaviour of a command, as we'll see shortly.
- Side angle brackets (`< >`): this is a *required* placeholder, where you should substitute your own information. In both cases, you exclude the brackets when you run the command – the brackets exist to show you what's required and what's optional.

If you have a lot of files to move, though, there's a handy shortcut.

```
ubuntu@tutorial:~$ mv *.txt my-folder
```

The wildcard (*) symbol tells the `mv` command to look for *all files* whose name matches the pattern. In this case, files with *any* name, but ending with the `.txt` file extension. Let's check this did what we expected:

```
ubuntu@tutorial:~$ ls

my-folder
```

Nice and tidy! But how do we know that the files have been moved, and not just deleted?

Here, we can use the `cd` command ("change directory"), to move to `my-folder` and take a peek inside:

```
ubuntu@tutorial:~$ cd my-folder/
```

```
ubuntu@tutorial:~/my-folder$ ls

my-first-file.txt my-second-file.txt my-third-file.txt
```

Did you notice how the shell prompt changed to `ubuntu@tutorial:~/my-folder$` after we used the `cd` command? This is our biggest navigation help! The tilde (~) symbol signifies

our home directory (located at `/home/your_user_name`, which was our starting point. All the directories and subdirectories we create can trace a path back to this home directory.

Let's illustrate how this works by moving around a bit more. Start by making a subdirectory:

```
ubuntu@tutorial:~/my-folder$ mkdir second-level
```

Another helpful feature of the command line is "tab completion". If a command, directory, or file already exists, we can start typing it and then press the TAB key on the keyboard to complete it. Let's navigate to the directory we just made by typing `cd s`, and then press TAB. The shell automatically fills in the rest!

```
ubuntu@tutorial:~/my-folder$ cd second-level/
```

And then let's make another subdirectory, and change to it:

```
ubuntu@tutorial:~/my-folder/second-level$ mkdir third-level
```

```
ubuntu@tutorial:~/my-folder/second-level$ cd third-level
```

We're getting quite far into the subdirectory structure now. Let's use the `pwd` command to double check exactly where we are:

```
ubuntu@tutorial:~/my-folder/second-level/third-level$ pwd  
/home/ubuntu/my-folder/second-level/third-level
```

This command shows us our "present working directory" (sometimes people refer to it as the "current working directory"), and shows us the path we took to get to where currently are!

So, now we're here, how do we get back again?

You guessed it – with our `cd` command:

```
ubuntu@tutorial:~/my-folder/second-level/third-level$ cd ..
```

Using `cd ..` will take us back up one directory – in our case, from `third-level` back to the `second-level`. The `..` dots represent one folder level, and we could go back any number of levels we like using forward slashes (`/`) as separators between the dots. For example, starting from `third-level`, using `cd ../../` would take us back two levels, to `my-folder`, and `cd ../../../` takes us back three levels, to our home directory.

If we know we want to go back to the home directory, though, we could also use `cd ~` to skip back there directly.

```
ubuntu@tutorial:~/my-folder/second-level$ cd ~
```

```
ubuntu@tutorial:~$ pwd  
/home/ubuntu
```

And now we're back home again!

To keep things tidy, let's remove the directories we created using the `rmdir` ("remove directory") command:

```
ubuntu@tutorial:~$ rmdir my-folder
```

But wait! This returns an error message: `rmdir: failed to remove 'my-folder/': Directory not empty`. It turns out that `rmdir` is a safe option, because it will *only* let you delete a directory if it is empty. This prevents you from accidentally deleting files you didn't know were in the folder.

One option we have is to go back to our second-level directory, and from there, run `rmdir third-level`. Then, change to the `my-folder` directory and delete the second-level directory using `rmdir second-level`. Then, finally, we can return home and run the `rmdir my-folder` command again, and it will work.

A faster (but less safe) option, is to use `rm` ("remove") instead.

Since we're in a sandbox, and we know there's nothing in this entire tree of folders we want to save, we can go ahead. Here, we're going to use the `-r` flag to *recursively* delete the `my-folder` dir, and all its subdirectories.

```
ubuntu@tutorial:~$ rm -r my-folder
```

Now if we run `ls` again, we'll see nothing gets returned, because our home directory is empty once more!

The `rm` command won't let you delete a directory without the `-r` flag, but it also won't stop you (or ask if you're *really, truly* sure) when deleting a folder and all its contents – so you should be cautious when using this command on a real system, to ensure you're really sure you're not going to delete something by accident that you really want to keep.

Working with files

We've already seen how to create empty files. Let's look at how to copy, move, and remove files.

Since our directory is empty again, let's create a new file to play with using the `touch` command:

```
ubuntu@tutorial:~$ touch file1.txt
```

If we run `ls`, we'll see it is there:

```
ubuntu@tutorial:~$ ls  
file1.txt
```

To copy a file, we use `cp` ("copy"), followed by the original file's name, and then the new file's name. This is another case of using *positional arguments*:

```
ubuntu@tutorial:~$ cp file1.txt file2.txt
```

Since we're not renaming or otherwise touching file1.txt, we should now have two files:

```
ubuntu@tutorial:~$ ls  
file1.txt file2.txt
```

Before we can see how to move a file, we need to have somewhere to move the file to, so let's quickly create a directory to test this out using `mkdir new-folder`.

Now we can move file2.txt to our new-folder using the `mv` ("move") command, which takes two arguments:

```
mv <original-location/file-name> <new-location/file-name>
```

In our case, that translates to:

```
ubuntu@tutorial:~$ mv file2.txt new-folder/file2.txt
```

Both of the arguments here can take a **path**, but if we're in the same directory as the file is located in, we don't need to specify it. This becomes really powerful when you have to handle a lot of files in subdirectories.

```
ubuntu@tutorial:~$ ls  
file1.txt new-folder
```

If we `cd` to our new-folder we can see our file has been moved with the same name. But – what if we don't want to keep the same name? It turns out `mv` can help us with that too.

```
ubuntu@tutorial:~/new-folder$ mv file2.txt file2-renamed.txt
```

We can also combine both moving the file and renaming it into one command.

```
ubuntu@tutorial:~/new-folder$ mv file2-renamed.txt ../file2.txt
```

Then if we `cd ..` back to the home directory, and run `ls` again, we'll see that our file has been moved back:

```
ubuntu@tutorial:~$ ls  
file1.txt file2.txt new-folder
```

So what if we want more copies of the same file, but in our new-folder? For that, we can use `cp` again, but give the directory as the destination:

```
ubuntu@tutorial:~$ cp file1.txt new-folder/
```

Double check with `ls` and `cd` that this has worked. You should now have file1.txt in both the home directory, and in new-folder.

Note:

Be aware that the terminal will not check for, or warn you about, collisions between names. If you tell it to rename a file where a file of that name already exists, it will gleefully overwrite the original!

The `cp` command is not just for files. We can make backup copies of a whole directory (and all its contents) using the `-a` (or `--archive`) flag. This flag preserves all the file permissions:

```
ubuntu@tutorial:~$ cp -a new-folder/ new-folder-backup/
```

Take a look around your `new-folder-backup` directory to make sure it all matches the original, then `cd ..` back to the home directory.

Let's say we no longer want `file.txt`. We can use the `rm` command to remove it, the way we did before. This time, we don't need the `-r` flag because we're deleting a single file.

```
ubuntu@tutorial:~$ rm file1.txt
```

Warning:

Remember, there is no recycle bin in the Linux terminal – once you `rm` a file, it is gone. On your real system, double-check what you are deleting before running this command.

Understanding file attributes

In Linux, everything is represented as a file – including directories, devices, and network connections. When we use `ls -l`, the very first character of each line tells us what type of file it is.

Let's look at a few examples:

```
ubuntu@tutorial:~$ ls -l
-rw-rw-r-- 1 ubuntu ubuntu    0 Apr 21 17:46 file2.txt
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 21 18:00 new-folder
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 21 18:00 new-folder-backup
```

The output looks quite complicated at first, so let's break it down using the first row of the output.

Column	What it is	Description
-rw-rw-r--	Permissions	The type of file and who can read/write/execute it
1	Links	The number of distinct filesystem paths that refer to this file or directory (called “hard links”)
ubuntu	Owner	The username of the person who owns the file
ubuntu	Group	The name of the group that has specific permissions
0	Size	The size of the file in bytes
Apr 21 17:46	Times-tamp	The date and time the file was last modified
file2.txt	Name	The actual name of the file or directory

Fields like the file name, size, and timestamp, are quite intuitive once you know that this is what the fields represent. Permissions, owner and group are a bit less intuitive, so let's examine those in more detail.

Users, groups, and permissions

Linux is a **multi-user system** – it is designed to be used by multiple people at once, and it carefully controls who can read, modify, or run any given file. Understanding users, groups, and permissions is essential for working safely on a Linux system.

Every file on a Linux system has an **owner** (a user account) and a **group** associated with it. Permissions grant different types of access to owners and groups.

The nine characters in the permissions field show the different levels of access granted for each file. They are split into three groups of three:

```
-   rwx  r-x  r-x
|       |    |
|       |    +----- Other (everyone else): read + execute
|       +----- Group: read + execute
+----- Owner: read + write + execute
File type: regular file
```

Each group of three characters can contain:

- **r – Read:** view the file’s contents
- **w – Write:** modify or delete the file
- **x – Execute:** run the file as a program (or, for directories, enter them with `cd`)
- **-** – the permission is not granted

In our own example, we can see this means that for `file2.txt`, the permissions are:


```
-  rw-  rw-  r--
|  |    |    |
|  |    |    |----- Other (everyone else): read only
|  |    |    |----- Group: read + write (no execute)
|  |    |    |----- Owner: read + write (no execute)
|  |    |    |----- File type: regular file
```

The first character on each line is the **file type indicator**, which tells us what type of file it is:

Sym- bol	Type	Description
-	Regular file	An ordinary file: documents, scripts, binaries, images, etc.
d	Directory	A file that is a list of other files
l	Symbolic link (or symlink)	A pointer (shortcut) to another file
c	Character device	A device file for hardware that handles data one character at a time (e.g., a keyboard)
b	Block device	A device file for hardware that handles buffered data in blocks (e.g., a hard disk)
s	Socket	A file used for inter-process communication
p	Named pipe	Allows processes to communicate without network socket semantics

Changing file permissions

We can change a file's permissions using `chmod` ("change file mode"). There are two ways to use it.

Symbolic notation

This method uses letters to specify who to change:

- u – user (owner)
- g – group
- o – other

And what to change:

- + – add a permission
- - – remove a permission

Let's test method out first by creating a script file:

```
ubuntu@tutorial:~$ touch symbolic-test.sh
```

```
ubuntu@tutorial:~$ ls -l symbolic-test.sh
-rw-rw-r-- 1 ubuntu ubuntu 0 Apr 23 10:52 symbolic-test.sh
```

And then let's make it executable by us (the owner):

```
ubuntu@tutorial:~$ chmod u+x symbolic-test.sh
```

```
ubuntu@tutorial:~$ ls -l symbolic-test.sh
-rwxrw-r-- 1 ubuntu ubuntu 0 Feb 12 2024 myscript.sh
```

Note how the permissions for the owner have change from "read and write only" (rw-) to "read, write and execute" (rwx).

Numeric notation

This method uses a three-digit number where each digit is calculated by adding: read (4) + write (2) + execute (1):

```
ubuntu@tutorial:~$ touch numeric-test.sh
```

```
ubuntu@tutorial:~$ ls -l numeric-test.sh
-rw-rw-r-- 1 ubuntu ubuntu 0 Apr 23 15:58 numeric-test.sh
```

Then let's use the numeric notation to change the permissions:

```
ubuntu@tutorial:~$ chmod 755 numeric-test.sh
```

```
ubuntu@tutorial:~$ ls -l numeric-test.sh
-rwxr-xr-x 1 ubuntu ubuntu 0 Apr 23 15:58 numeric-test.sh
```

From this we can see that 755 means: the owner has read, write and execute (7) privileges, the group has read and execute (5), while all others have read and execute (5).

Some common permission combinations:

Num ber	Sym- bolic	Meaning
644	rw- r-- r--	Owner can read/write; others can only read. Suitable for most files.
755	rxr- xr- x	Owner can do everything; others can read and execute. Suitable for scripts and directories.
700	rx- --- --	Only the owner has any access.
600	rw- --- ---	Owner can read/write; no one else has any access.
777	rxr- wxrw	Full access to <i>everyone</i> . This is a dangerous permission to grant , but may be useful for debugging permission-related problems or on temporary files. Please avoid using this one unless you really know what you're doing, and if you do see it, you should probably change it to one of the less permissive combinations.

When we create a new file, it gets a default set of permissions. The `umask` ("user file-creation mode mask") controls what those defaults are. We can check the current `umask` with:

```
ubuntu@tutorial:~$ umask  
  
0002
```

The `umask` works by subtracting permissions from the maximum (777 for directories, 666 for files). A `umask` of 022 results in files being created with permissions of 644 and directories with 755 by default.

Temporary admin privileges with `sudo`

Most day-to-day work on Ubuntu is done as a regular user, without administrative privileges. When we do need to run a command as an administrator, we prefix it with `sudo` ("superuser do"), which temporarily grants elevated privileges for that one command. For example, if we want to update the software on our computer, we would run these two commands in sequence:

```
ubuntu@tutorial:~$ sudo apt update
```

```
ubuntu@tutorial:~$ sudo apt upgrade
```

When we run a `sudo` command, we are asked for our password to confirm it is us.

The sudo group on Ubuntu is special: any user account added to it gains the ability to run commands with admin privileges. Groups are listed in `/etc/group`, and a user can belong to many groups at once. We can inspect who is in the sudo group with:

```
ubuntu@tutorial:~$ grep sudo /etc/group  
  
sudo:x:27:ubuntu
```

Each line of the `/etc/group` file follows this format:

```
group_name:password:GID:user_list
```

Where:

- `group_name` is the human-readable name of the group (what you see when you run `ls -l`)
- `password` is always shown as `x` in modern Linux systems - the actual (encrypted) group password is stored in the secure `/etc/passwd` file.
- `GID` is the group ID, which is a unique numerical integer assigned to the group.
- `user_list` is a comma-separated list of usernames that are members of the group.

The Filesystem Hierarchy Standard

Now that we know how to move around the filesystem, it's worth understanding how it is organised. Most Linux distributions, including Ubuntu, follow the **Filesystem Hierarchy Standard** (FHS), which defines where different types of files and directories belong. This is maintained by the Linux Foundation, and it means the structure of any conforming system is predictable and consistent.

All files and directories live under the **root directory**, represented by a single forward slash `/`. Even files stored on separate physical disks appear as part of this single tree.

Let's take a look at the top level:

```
ubuntu@tutorial:~$ ls /  
  
bin  dev  home  lib64  media  opt   root  sbin  srv  tmp  usr  var  
boot  etc  lib   mnt    proc  run   snap  sys
```

Here is what the most important directories contain:

Directory	Purpose
/bin	Essential command binaries (e.g., <code>ls</code> , <code>cp</code> , <code>mv</code>)
/sbin	System binaries (e.g. <code>iptables</code> , <code>reboot</code>)
/boot	Static files needed to boot the system, including the Linux kernel and boot-loader files
/dev	Device files (e.g. <code>/dev/sda</code> and <code>/dev/null</code>)
/etc	System configuration files (e.g. <code>/etc/group</code> and <code>/etc/password</code>)
/home	User home directories (optional, but standard)
/lib	Essential shared libraries (needed by the binaries in <code>/bin</code> and <code>/sbin</code>)
/media	Mount points for temporarily mounted removable media (like USBs or CDs)
/mnt	Mount points for manually mounted filesystems (kept separate to avoid clashes)
/root	The home directory for the root user specifically
/srv	Service data, for services provided by the system
/tmp	Temporary files (cleared on reboot)
/usr	Most user programs, libraries, and documentation
/var	Variable data: logs, caches, spool files

This is not a complete list, by any means! If you'd like more information about the Standard and other directories contained within it, take a look at [the official specification](#)²⁶.

Exploring deeper

Let's browse a few of these. The `/etc` directory contains configuration files for the system and its services. Some important files there include:

- `/etc/passwd` – the user account database
- `/etc/shadow` – encrypted passwords (readable only by root)
- `/etc/fstab` – filesystems to be mounted automatically at startup
- `/etc/group` – group account information

The `/var/log` directory holds the system's log files – a useful place to look when diagnosing problems:

```
ubuntu@tutorial:~$ ls /var/log
```

The `tree` command can be used to visualise directory structures in a tree-style listing. It's not installed by default, so let's install it:

²⁶ <https://specifications.freedesktop.org/fhs/latest/>

```
ubuntu@tutorial:~$ sudo apt install tree -y
```

The `-L` flag can be used to set the depth you want to view. To see the top level (or level 1) of the root filesystem (`/`):

```
ubuntu@tutorial:~$ tree -L 1 /
```

To go one level deeper, to level 2:

```
ubuntu@tutorial:~$ tree -L 2 /
```

What happens if you just run `tree` by itself, with no arguments?

```
ubuntu@tutorial:~$ tree

.
├── file2.txt
├── new-folder
│   └── file1.txt
├── new-folder-backup
│   └── file1.txt
├── numeric-test.sh
└── symbolic.sh

3 directories, 5 files
```

This is a much more convenient way to see what files each directory contains.

A note about flags

So far we've been using short flags, with a single dash (`-`) followed by a single letter, like with `ls -l`.

Most flags also have a long version, where you'll see a double dash (`--`) with a full word. If a flag has both a long and short version, they can be used interchangeably.

It's possible to combine flags together. For example, the `-l` and `-a` flags ("long list", "all") would be combined into `ls -la` gives us a long listing that also includes hidden files, or those whose names start with a `.` character:

```
ubuntu@tutorial:~$ ls -la
```

```
drwxr-x--- 6 ubuntu ubuntu 4096 Apr 23 17:26 .
drwxr-xr-x 3 root root 4096 Apr 16 14:15 ..
-rw----- 1 ubuntu ubuntu 936 Apr 23 13:00 .bash_history
-rw-r--r-- 1 ubuntu ubuntu 220 Mar 31 2024 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Mar 31 2024 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Apr 16 14:15 .cache
-rw----- 1 ubuntu ubuntu 20 Apr 23 17:26 .lessht
-rw-r--r-- 1 ubuntu ubuntu 807 Mar 31 2024 .profile
drwx----- 2 ubuntu ubuntu 4096 Apr 16 14:15 .ssh
-rw-r--r-- 1 ubuntu ubuntu 0 Apr 23 11:05 .sudo_as_admin_successful
-rw-rw-r-- 1 ubuntu ubuntu 0 Apr 23 15:47 file2.txt
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 23 15:53 new-folder
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 23 15:53 new-folder-backup
-rwxr-xr-x 1 ubuntu ubuntu 0 Apr 23 15:58 numeric-test.sh
-rwxrw-r-- 1 ubuntu ubuntu 0 Apr 23 15:54 symbolic.sh
```

This brings us onto one of the most important skills when you are learning the terminal – how to find help.

Getting help

Linux comes with a comprehensive built-in manual, and learning to use it helps us explore on our own. Every command in Linux has a **manual page** that describes what it does, and lists all the option flags available.

Most documentation uses the short version of a flag when demonstrating a command – mostly for convenience. However, if you are ever unsure what a flag does, you can look up the command’s manual page (often called a “manpage” for short).

For example, you can invoke the manual page for `ls` using the `man ls` command.

```
ubuntu@tutorial:~$ man ls
```

This opens the manual page for `ls`. We can navigate it with the following keys on our keyboard:

Key	Action
Enter	Move down one line
Space	Move down one page
g	Move to the top
G	Move to the bottom
h	Display navigation help
q	Quit

You can also scroll up and down using the mouse wheel, or the up and down keys.

Let’s try reading the man page for `man` itself:

```
ubuntu@tutorial:~$ man man
```

The Linux manual is divided into numbered sections – the number appears in the page title, e.g., LS(1). Section 1 means “executable programs and shell commands”. We can search within a man page by pressing / followed by a search term, then Enter.

Manual pages are also rendered online and can be [viewed in your browser](https://manpages.ubuntu.com/)²⁷.

Quick help with `-help`

Most commands also accept a `--help` flag that prints a brief summary of the available options:

```
ubuntu@tutorial:~$ ls --help
```

This is often quicker than opening the full manual page when we just want to check a specific flag.

Searching for a command

If we know what we want to do but don’t know which command to use, `man -k` searches the man page descriptions for a keyword:

```
ubuntu@tutorial:~$ man -k copy
```

This lists every command whose description mentions “copy”, which can help us find the right tool for the job.

Next steps

This is the end of our introduction to the terminal! We have covered a lot of ground: launching a virtual machine, navigating the filesystem, working with files and directories, understanding file types and the filesystem hierarchy, managing permissions, and finding help. All the commands we covered here (as well as some extra ones) can be found in our reference [Command-line cheat sheet](#) (page 819).

If you wish, you can continue to explore the file system on your virtual machine, or proceed straight to the next tutorial where you can use the same VM.

However, if you’re done for now and want to exit the VM, just type “exit”.

```
ubuntu@tutorial:~$ exit
```

This will take you out of the VM and back to your live machine. Then, you can run the following commands to delete the VM and remove it completely from your machine:

```
ubuntu@tutorial:~$ multipass delete tutorial > multipass purge
```

²⁷ <https://manpages.ubuntu.com/>

Next steps

If you wish to continue learning, we recommend following the [The command line in depth](#) (page 32) tutorial next. This tutorial will teach you the more advanced ways to interact with the Command Line Interface; how data flows, how input and output can be used more efficiently, and how to use variables.

The command line in depth

In [Welcome to the terminal](#) (page 12), we learned how to navigate the filesystem, work with files and directories, and manage permissions on the command line. Now we'll go deeper into how to handle data flowing between programs, how to redirect that data to files and other programs, and how **shell variables** and **expansion** make our commands *even more* powerful.

If you are re-using a virtual machine from a different tutorial, skip directly to [Standard streams](#) (page 34).

Set up the virtual machine

To avoid making changes to your computer we will set up a virtual machine (VM), which provides us with a safe environment to run the commands in. [Multipass²⁸](#) is great for quickly creating Ubuntu virtual machines, so we will use that. It is available on Windows, Mac and Linux, so you don't need to be already running Ubuntu on your computer to follow this tutorial.

Prerequisites

- The Multipass VM needs **5 GB of disk space**, and **1 GB of memory** available.

Install Multipass

On Ubuntu

You can install Multipass directly from [the Multipass page²⁹](#) in the online snap store (make sure to select the "latest/stable" version from the dropdown menu next to the install button).

²⁸ <https://documentation.ubuntu.com/multipass/latest/>

²⁹ <https://snapcraft.io/multipass>

On other operating systems

If you're on Windows, Mac, or another Linux operating system, Multipass can be installed using [these instructions](#)³⁰.

Open a terminal window

Now that Multipass is installed, we need a terminal window to run our commands in. How you open one depends on your operating system.

On Ubuntu

Press `Ctrl + Alt + T` together and a terminal window will open. You can also search for "Terminal" in your application launcher.

On Windows

PowerShell is the most reliable option for working with Multipass on Windows.

1. Press the Windows key.
2. Type PowerShell.
3. Right-click it and select **Run as Administrator** — this helps avoid permission issues when managing virtual machines.

If you have Windows Terminal installed from the Microsoft Store, that works just as well and offers a nicer experience with tab support.

To confirm everything is working, type `multipass version` and press Enter. If you see version numbers, you're ready to continue.

On macOS

1. Press `⌘ + Space` to open Spotlight.
2. Type Terminal and press Enter.

If you use iTerm2, that works perfectly well too.

To confirm everything is working, type `multipass version` and press Enter. If you see version numbers, you're ready to continue.

³⁰ <https://documentation.ubuntu.com/multipass/stable/how-to-guides/install-multipass/>

Create the virtual machine

Once you have installed Multipass and opened your terminal, we can launch a new VM using the Ubuntu 26.04 LTS release (codename “resolute”), and let’s give our VM the name `tutorial` using the following command in our terminal window:

```
$ multipass launch resolute --name tutorial
```

It may take a little time, depending on the speed of your internet connection. Multipass downloads the most recent daily **image** and creates the VM for us.

Note:

An Ubuntu **image** is a collection of files we need to install and run Ubuntu. We don’t need to specify “server” or “desktop” anywhere in our command, because the image is the same for both. The only difference between Ubuntu Server and Ubuntu Desktop is the subset of software packages we use from the Ubuntu Archive.

Once the above command has finished running, we can access the VM by running:

```
$ multipass shell tutorial
```

We will get a “Welcome to Ubuntu” message. Notice that when we run this command, the terminal username changes to `ubuntu` and the *hostname* changes to `tutorial`:

```
ubuntu@tutorial:~ $
```

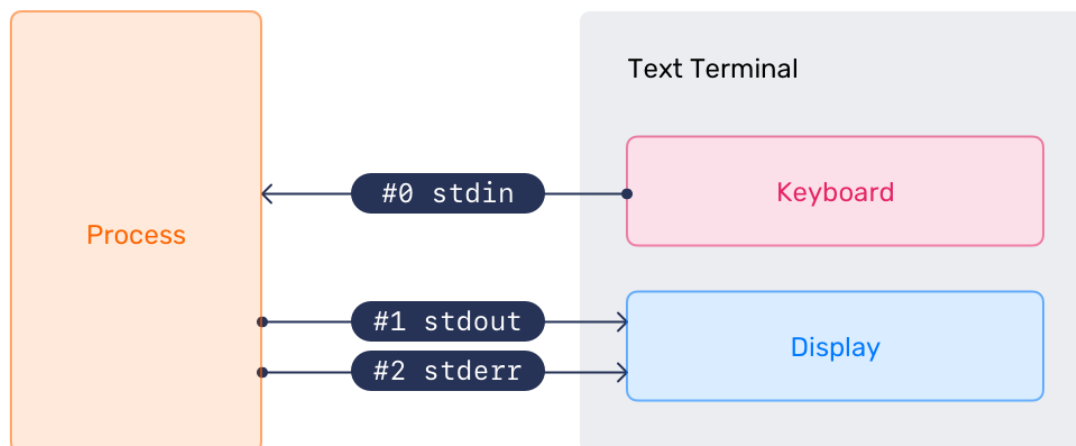
This shows that we are inside the VM, and this is where we will run all our commands. Now that our setup is complete, let’s carry on with the tutorial!

Standard streams

Now that we can work with files, we’re ready to understand something fundamental about how programs work in the terminal: **standard streams**.

Every program that runs has three open communication channels by default:

- **Standard input (stdin)** – the channel through which the program receives input. By default, this is your keyboard.
- **Standard output (stdout)** – the channel through which the program sends its results. By default, this is your terminal screen.
- **Standard error (stderr)** – the channel through which the program sends error messages. This also appears on your terminal screen by default, but it is a separate channel from stdout.



Each stream has a numerical identifier called a **file descriptor**:

- 0 – stdin
- 1 – stdout
- 2 – stderr

You may be wondering: “why three separate channels? Why not just one?”.

The answer is that separating normal output from error output gives us much more flexibility. We can save the output of a command to a file while still seeing errors on screen – or silence errors entirely while saving results. We’ll see exactly how in the next sections.

Redirecting output

Normally, a command’s output goes straight to our terminal screen. We can tell the shell to send it somewhere else instead – like a file.

To redirect output to a file, we use the `>` operator followed by a filename. Let’s redirect the output of `ls` to a file called `file-list.txt`:

```
ubuntu@tutorial:~$ ls -la > file-list.txt
```

No output appears on screen – it has all gone into a new file called `file-list.txt` instead.

Viewing file contents

Now that `file-list.txt` exists and has some content in it, let’s look at the commands we can use to see what the file contains. There are a few to choose from, and which one is most convenient depends on how much of the file we want to see.

To print the full contents of a file to the screen, we can use `cat` (“concatenate”):

```
ubuntu@tutorial:~$ cat file-list.txt

total 36
drwxr-x--- 4 ubuntu ubuntu 4096 May  1 09:25 .
drwxr-xr-x 3 root  root  4096 Apr 16 14:15 ..
-rw----- 1 ubuntu ubuntu 1623 Apr 23 17:33 .bash_history
-rw-r--r-- 1 ubuntu ubuntu  220 Mar 31  2024 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Mar 31  2024 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Apr 16 14:15 .cache
-rw----- 1 ubuntu ubuntu   20 Apr 23 17:33 .lessht
-rw-r--r-- 1 ubuntu ubuntu  807 Mar 31  2024 .profile
drwx----- 2 ubuntu ubuntu 4096 Apr 16 14:15 .ssh
-rw-r--r-- 1 ubuntu ubuntu    0 Apr 23 11:05 .sudo_as_admin_successful
-rw-rw-r-- 1 ubuntu ubuntu    0 May  1 09:25 file-list.txt
```

You may see more lines or fewer lines here, depending on whether you ran through the [Welcome to the terminal](#) (page 12) tutorial with this VM or not.

For longer files, `less` is more comfortable – with `less` we can scroll through the outputs one page at a time. Let's look at `/etc/passwd`, which stores user account information and is long enough to scroll through:

```
ubuntu@tutorial:~$ less /etc/passwd
```

Press Space to move down a page, and Q to quit when you are done.

If we only want to see the first lines of a file, we can use `head`:

```
ubuntu@tutorial:~$ head /etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/spool/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

By default, `head` shows the first 10 lines but we can request a different number using the `-n` flag. For example, `head -n 5 /etc/passwd` shows only the first 5 lines.

Or we can see the last few lines with `tail`, which is used the same way as `head`:

```
ubuntu@tutorial:~$ tail -n 3 /etc/passwd

fwupd-refresh:x:990:990:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
polkitd:x:989:989:User for polkitd:/:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
```

`tail` is especially useful for examining log files, where the most recent entries are always at the bottom.

Warning:

One thing to be careful about: we know that `file-list.txt` is empty because it didn't exist before, but if `file-list.txt` had already contained something, `>` would have overwritten it completely. To *add* things to the end of an existing file rather than replacing it, we use `>>` (**append** mode):

Let's make our file a bit longer by using `>>` to add extra contents:

```
ubuntu@tutorial:~$ ls -l /tmp >> file-list.txt
```

Now `file-list.txt` contains the original home directory listing followed by the `/tmp` listing. We can check on this by using `cat file-list.txt` again.

Sometimes we want a command to run **silently**, i.e. with its output discarded entirely. We can redirect stdout to `/dev/null`, a special file that throws away everything written to it:

```
ubuntu@tutorial:~$ ping -c 2 localhost > /dev/null
```

We can also redirect stderr (file descriptor 2) independently from stdout. This is useful for capturing errors to a log file while still seeing normal output on screen:

```
ubuntu@tutorial:~$ ls /nonexistent 2> errors.txt
```

Checking on this file shows us the error that would otherwise have been printed to the screen:

```
ubuntu@tutorial:~$ cat errors.txt
ls: cannot access '/nonexistent': No such file or directory
```

We can also merge stderr into stdout using `2>&1`, which means "send file descriptor 2 to wherever file descriptor 1 is currently going":

```
ubuntu@tutorial:~$ ls /nonexistent >> output.txt 2>&1
```

By doing this, both normal output and error messages will now appear in `output.txt`.

Redirecting input

We can go the other direction too! Instead of a command reading from the keyboard, we can feed it input from a file using the `<` operator.

Let's use `wc` ("word count"), which counts lines, words, and characters in its input. Without redirection, we would type text and press `Ctrl + D` to signal the end of input. Instead, we can feed it a file directly:

```
ubuntu@tutorial:~$ wc -l < /etc/passwd
```

```
33
```

Once again, you might see a different value here, depending on whether this is a new VM or you're reusing the one from the previous tutorial.

The `-l` flag tells `wc` to count lines only. When we use `<`, the file name does not appear in the output – `wc` sees the file's contents as if they had been typed at the keyboard.

Here-documents

There is a related form called a **here-document**, written as `<<`. It lets us type multi-line input directly in the terminal, ending when we type a delimiter we choose ourselves (often EOF is used as this marker – short for “end of file”). Let's test this by sending a greeting into a new file called `greeting.txt`:

```
ubuntu@tutorial:~$ cat > greeting.txt << EOF > Hello from the terminal! >
This text is going into a file. > EOF
```

The shell reads everything we type until it sees EOF on a line by itself, then passes it all as input to the command on the left. Let's confirm the file was created correctly:

```
ubuntu@tutorial:~$ cat greeting.txt

Hello from the terminal!
This text is going into a file.
```

Pipes

Pipes are one of the most powerful ideas in the Unix philosophy: the output of one command becomes the input of the next. We connect them with the `|` operator (the “pipe” character).

For example, `ls -la /usr/bin` produces many lines of output – more than can fit on screen, and you can only see the end of the output. Scrolling back up is often very inconvenient, if you can even do it at all (some terminals can't!). To make it more convenient, we can pipe the output to `less` so we can page through it with the space key:

```
ubuntu@tutorial:~$ ls -la /usr/bin | less
```

Press `Q` to quit `less` when you are done.

We can also pipe to `grep` to filter the output, keeping only lines that match a pattern. Let's find all currently set environment variables whose names contain the word `LANG`:

```
ubuntu@tutorial:~$ env | grep LANG

LANG=C.UTF-8
```

Pipes can be chained together to build a **pipeline** – each command passes its output along to the next. For example, to see the 3 most recently modified files in `/usr/bin`, we can sort by modification time and then cut the output down with `head`:

```
ubuntu@tutorial:~$ ls -lt /usr/bin | head -n 3

total 121540
-rwxr-xr-x 1 root root      174328 Apr 30 13:32 kmod
lrwxrwxrwx 1 root root         4 Apr 30 13:32 lsmod -> kmod
```

Each command in the pipeline only ever sees the output of the command immediately before it – the shell connects them all behind the scenes.

Chaining commands

Pipes pass data from one command to another and run all of their commands in parallel. This allows commands to begin working on long streams of data while their source is still producing more. However, there are scenarios where you might want a serial process, in which case you would use **command chaining** instead. Chaining is different: it controls the *order* in which commands run, and whether the next command runs at all.

The simplest way to chain commands is to separate them with a semicolon (;). The shell runs each command in turn, regardless of whether the previous one succeeded, with the output of each command shown on separate lines:

```
ubuntu@tutorial:~$ ls; pwd; whoami

errors.txt  file-list.txt  greeting.txt  output.txt
/home/ubuntu
ubuntu
```

Note:

On Linux, processes declare whether they have “succeeded” or not with an **exit code**. An exit code (or “exit status”) is a number from 0–255 that a process gives to its parent upon completion. “Success” is defined by exiting with code 0, with *all* other values indicating different types of failures. These might be specific to a particular program, but the generic failure status will usually be exit code 1. You can use `exit n` to immediately exit with code `n`.

Often we want the second command to run *only* if the first one succeeded. We use `&&` for this:

```
ubuntu@tutorial:~$ cd /tmp && ls

snap-private-tmp  systemd-private-...
```

If `cd /tmp` had failed (which can happen in a script when the directory does not exist), `ls` would not run at all.

The opposite of `&&` is `||`, which runs the second command only if the first one *failed*:


```
ubuntu@tutorial:~/tmp$ cd /nonexistent || echo "Could not change directory"

-bash: cd: /nonexistent: No such file or directory
Could not change directory
```

We can combine `&&` and `||` to produce a simple success-or-failure message:

```
ubuntu@tutorial:~/tmp$ ping -c1 localhost && echo "Host is up" || echo "Host
is unreachable"

PING localhost (:::1) 56 data bytes
[...]
Host is up
```

A few other operators are useful to know. Try each of these out, both with and without the operators, to see how they change the behaviour for yourself:

Background execution; `&`

Runs a command in the background. The shell prompt returns immediately, and the command continues running. Let's compare the difference in running `sleep 10` and then `sleep 10 &`. The `sleep` command doesn't actually *do* anything, it just makes the system wait for the specified number of seconds. So, in this example our prompt disappears for 10 seconds and you can't interact with the terminal during that time. If we *background* it, using `sleep 10 &`, our command prompt returns immediately and we can continue to use the terminal.

When we use the ampersand (`&`) to background the process, it returns something like:

```
[1] 168658
```

This number is the **process ID (PID)**; a unique number identifying every process running on your machine. When the background process has finished running, you'll see a message returned that looks like this, so you know it's done:

```
[1]+  Done                  sleep 10
```

Command grouping; `{ }`

Groups commands together so they share the same output redirection or run as a unit. E.g. to create a quick log file you could run:

```
ubuntu@tutorial:~/tmp$ { echo "System Report"; date; uptime; } >
report.txt
```

Subshell execution; `()`

Run commands in a subshell, where the commands are forgotten the moment they finish running. This is useful for isolating side effects like a `cd` – the directory change does not affect the current shell.

```
ubuntu@tutorial:/tmp$ (export NAME="Alice"; echo "Hello $NAME") > echo  
"Name is: $NAME"
```

Line continuation; \

Continue a long command on the next line, which helps a lot with command readability. You'll see this often in documentation.

Let's return home before continuing:

```
ubuntu@tutorial:/tmp$ cd ~
```

Shell environment variables

Before we look at shell expansion, we need to understand **shell variables**, because expansion builds heavily on them.

Think of a variable as a labelled box: we give it a name, put a value inside, and later retrieve that value using the name of the box. The shell uses variables to store all kinds of information – including a great deal about the environment we're working in.

Built-in variables

The shell already has a number of variables set up for us. To see every variable currently set in our environment, we can run:

```
ubuntu@tutorial:~$ env
```

We can print the value of any variable using echo with a \$ in front of the variable name:

```
ubuntu@tutorial:~$ echo $SHELL  
/bin/bash
```

```
ubuntu@tutorial:~$ echo $USER  
ubuntu
```

```
ubuntu@tutorial:~$ echo $HOME  
/home/ubuntu
```

One particularly important variable is PATH. It tells the shell which directories to search when we type a command name:

```
ubuntu@tutorial:~$ echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
```

Each directory is separated by a colon (:). When we type a command like `ls`, the shell searches through each of these directories in order until it finds a program with that name. This is why we can run commands without typing their full path. If a command lives outside these directories, we have to give its full path explicitly.

Setting and unsetting variables

We can create our own variables by assigning a name and a value. By convention, user-defined variable names are uppercase:

```
ubuntu@tutorial:~$ GREETING="Hello, world!"
```

```
ubuntu@tutorial:~$ echo $GREETING
Hello, world!
```

To remove a variable, we use `unset`:

```
ubuntu@tutorial:~$ unset GREETING
```

```
ubuntu@tutorial:~$ echo $GREETING
```

This time we get no output, because the variable no longer exists.

Exporting variables to subshells

Variables we create in the shell are only available in the current shell session. If a new shell process is started (for example, when running a script), it will not have access to our variables unless we **export** them.

Let's demonstrate. First, we set a variable and try to access it from a subshell:

```
ubuntu@tutorial:~$ MY_VAR="hello"
```

```
ubuntu@tutorial:~$ bash -c 'echo $MY_VAR'
```

Nothing is printed – the subshell has no knowledge of `MY_VAR`. Now let's export it first:

```
ubuntu@tutorial:~$ export MY_VAR="hello"
```

```
ubuntu@tutorial:~$ bash -c 'echo $MY_VAR'
```

```
hello
```

`export` makes the variable part of the environment that is passed down to any child processes we start.

Special read-only variables

Some variables are set by the shell itself and can only be read, never assigned. Two of the most useful are:

- `$?` – the exit status of the last command. 0 means success; any non-zero value means something went wrong.
- `$$` – the Process ID (PID) of the current shell.

Let's see `$?` in action. A successful command returns 0:

```
ubuntu@tutorial:~$ ls /etc/passwd
/etc/passwd
```

```
ubuntu@tutorial:~$ echo $?
0
```

And a failing command returns a non-zero value:

```
ubuntu@tutorial:~$ ls /nonexistent
ls: cannot access '/nonexistent': No such file or directory
```

```
ubuntu@tutorial:~$ echo $?
2
```

This is especially useful when we build scripts, where we can check whether the previous command succeeded before deciding what to do next.

Shell expansion

When we press Enter on a command, the shell does not send our text directly to the program. First, it processes the command line and **expands** certain patterns into their final values. Understanding expansion explains a lot of seemingly “magic” shell behaviour.

Tilde expansion

We have already used tilde expansion without thinking about it. When the shell sees ~ at the start of a word, it replaces it with the value of \$HOME:

```
ubuntu@tutorial:~$ echo ~  
/home/ubuntu
```

```
ubuntu@tutorial:~$ echo ~/my-file.txt  
/home/ubuntu/my-file.txt
```

Variable expansion

We have seen this already: \$VARIABLE_NAME (or \${VARIABLE_NAME}) is replaced with the variable's value before the command runs. The braces form is important when the variable name might be confused with the characters that follow it:

```
ubuntu@tutorial:~$ DAY="Monday"
```

```
ubuntu@tutorial:~$ echo "Today is ${DAY}!"  
Today is Monday!
```

Without the braces, the shell would try to expand \$DAY! as a single variable name and find nothing.

It's also important to note that this variable expansion applies *everywhere* in the line, even with the command name itself:

```
ubuntu@tutorial:~$ CMD="echo"
```

```
ubuntu@tutorial:~$ $CMD "It's dynamic!"  
It's dynamic!
```

This can be useful in, say, a script that runs one of two very similar commands with the same argument, so you don't have to re-type everything twice in your conditional.

Command substitution

Command substitution lets us use the *output of a command* as part of another command. We wrap the inner command in `$(...)`:

```
ubuntu@tutorial:~$ echo "Today is $(date)"
Today is Fri Apr 24 10:00:00 BST 2026
```

The shell runs `date` first, captures its output, and substitutes it into the outer command before running it.

Arithmetic expansion

For integer arithmetic, we use `$((...))`:

```
ubuntu@tutorial:~$ echo $((5 + 3))
8
```

```
ubuntu@tutorial:~$ X=10
```

```
ubuntu@tutorial:~$ echo $((X * 2))
20
```

The available Arithmetic operators are:

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Integer division (/)
- Modulo/remainder (%)
- Exponentiation (**)

Brace expansion

Brace expansion generates a list of strings from a pattern. Unlike the other types of expansion, it does not involve variables or filenames – it is a purely text-based generation mechanism:

```
ubuntu@tutorial:~$ echo sp{in,an,ar}
spin span spar
```

Ranges work too, using `..`:

```
ubuntu@tutorial:~$ echo {1..10}

1 2 3 4 5 6 7 8 9 10
```

Brace expansion is particularly useful for creating multiple files or directories in one command. For example, to create a directory for each month of the year:

```
ubuntu@tutorial:~$ mkdir {01..12}-reports
```

```
ubuntu@tutorial:~$ ls

01-reports 02-reports 03-reports 04-reports 05-reports 06-reports
07-reports 08-reports 09-reports 10-reports 11-reports 12-reports
```

We can even clean them up in one command, too:

```
ubuntu@tutorial:~$ rm -r {01..12}-reports
```

Wildcard expansion

Wildcard expansion (also called **globbing**) lets us match filenames by pattern. We touched on this in [Welcome to the terminal](#) (page 12) when we used `mv *.txt my-folder` to move all text files at once.

The three main wildcards are:

*

Matches any sequence of characters (including none), so as we saw previously, `*.txt` matches all files ending in `.txt`.

?

Matches any single character. For example, `file?.txt` matches `file1.txt` and `fileA.txt`, but not `file10.txt`.

[...]

Matches any one of the characters inside the brackets. `file[123].txt` matches `file1.txt`, `file2.txt`, and `file3.txt` only.

Character classes like `[:digit:]` and `[:alpha:]` can be used inside brackets to match whole categories of characters.

Let's see wildcards in action. To list every file in `/usr/bin` whose name starts with `m`:

```
ubuntu@tutorial:~$ ls /usr/bin/m*

/usr/bin/make /usr/bin/man /usr/bin/md5sum [...]
```

Wildcards are expanded by the shell *before* the command runs, so the program receives the resulting list of matching filenames – it never sees the `*` character itself.

Success!

This is the end of our journey around the terminal, and some of the powerful things we can do with it, but it's certainly not all there is to know!

If you wish, you can continue to explore the file system on your virtual machine.

Once you are finished and want to exit the VM, just type “exit”.

```
ubuntu@tutorial:~$ exit
```

This will take you out of the VM and back to your live machine. Then, you can run the following commands to delete the VM and remove it completely from your machine:

```
ubuntu@tutorial:~$ multipass delete tutorial
```

```
ubuntu@tutorial:~$ multipass purge
```

Next steps

If you wish to continue learning, we recommend following the [Managing your software](#) (page 47) tutorial next. This tutorial will teach you all the different ways to handle installing, removing and customizing software on your Ubuntu machine, as well as how to safely keep it up-to-date.

Managing your software

If you are new to Ubuntu, you may be wondering what to do after installation. After all, Ubuntu is endlessly customizable according to your needs. There are two types of software found in Ubuntu: **Debian packages** (debs) and **snaps** – we will focus mainly on debs in this tutorial!

To help you get the most from your Ubuntu experience, this tutorial will walk you through managing the software on your Ubuntu machine. This tutorial can be completed using either Ubuntu Server or Ubuntu Desktop.

If you are re-using a virtual machine from a different tutorial, skip directly to [Updating the system with APT](#) (page 49).

Set up the virtual machine

To avoid making changes to your computer we will set up a virtual machine (VM), which provides us with a safe environment to run the commands in. [Multipass](#)³¹ is great for quickly creating Ubuntu virtual machines, so we will use that. It is available on Windows, Mac and Linux, so you don't need to be already running Ubuntu on your computer to follow this tutorial.

³¹ <https://documentation.ubuntu.com/multipass/latest/>

Prerequisites

- The Multipass VM needs **5 GB of disk space**, and **1 GB of memory** available.

Install Multipass

On Ubuntu

You can install Multipass directly from [the Multipass page³²](#) in the online snap store (make sure to select the “latest/stable” version from the dropdown menu next to the install button).

On other operating systems

If you’re on Windows, Mac, or another Linux operating system, Multipass can be installed [using these instructions³³](#).

Open a terminal window

Now that Multipass is installed, we need a terminal window to run our commands in. How you open one depends on your operating system.

On Ubuntu

Press **Ctrl + Alt + T** together and a terminal window will open. You can also search for “Terminal” in your application launcher.

On Windows

PowerShell is the most reliable option for working with Multipass on Windows.

1. Press the Windows key.
2. Type PowerShell.
3. Right-click it and select **Run as Administrator** — this helps avoid permission issues when managing virtual machines.

If you have Windows Terminal installed from the Microsoft Store, that works just as well and offers a nicer experience with tab support.

To confirm everything is working, type `multipass version` and press Enter. If you see version numbers, you’re ready to continue.

³² <https://snapcraft.io/multipass>

³³ <https://documentation.ubuntu.com/multipass/stable/how-to-guides/install-multipass/>

On macOS

1. Press `⌘ + Space` to open Spotlight.
2. Type `Terminal` and press `Enter`.

If you use `iTerm2`, that works perfectly well too.

To confirm everything is working, type `multipass version` and press `Enter`. If you see version numbers, you're ready to continue.

Create the virtual machine

Once you have installed Multipass and opened your terminal, we can launch a new VM using the Ubuntu 26.04 LTS release (codename "resolute"), and let's give our VM the name `tutorial` using the following command in our terminal window:

```
$ multipass launch resolute --name tutorial
```

It may take a little time, depending on the speed of your internet connection. Multipass downloads the most recent daily **image** and creates the VM for us.

Note:

An Ubuntu **image** is a collection of files we need to install and run Ubuntu. We don't need to specify "server" or "desktop" anywhere in our command, because the image is the same for both. The only difference between Ubuntu Server and Ubuntu Desktop is the subset of software packages we use from the Ubuntu Archive.

Once the above command has finished running, we can access the VM by running:

```
$ multipass shell tutorial
```

We will get a "Welcome to Ubuntu" message. Notice that when we run this command, the terminal username changes to `ubuntu` and the `hostname` changes to `tutorial`:

```
ubuntu@tutorial:~ $
```

This shows that we are inside the VM, and this is where we will run all our commands. Now that our setup is complete, let's carry on with the tutorial!

Updating the system with APT

The first thing we always want to do with a new system (whether a VM, container, bare metal, or cloud instance) is to make sure we have the latest versions of all the pre-installed software.

Debian packages, commonly referred to as **debs**, are the standard software package format in Ubuntu. They can be identified by the `.deb` file extension.

Every Linux distribution has their own preferred **package manager** for installing, updating and removing packages. In Ubuntu, the default package manager is [Advanced Packaging](#)

Tool³⁴ (or APT, for short).

APT handles all of your system software (and other deb packages). It provides an interface to the Ubuntu Archive *repository*, so it can access both the **database** of all the packages available in Ubuntu, and the means to handle the **packages** themselves.

There are two APT commands we need to update our system: update and upgrade, which we will always run in that order.

apt update

The apt update command is about the **database**. Any bug fixes in a package (or new versions since your last update) will be stored in the metadata about that package in the database (the **package index**).

When we run the update command it updates the APT database on our machine, fetching the newest available metadata from the package index:

```
$ sudo apt update

Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
19 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

As we can see, it checks (“hits”) the various archives (**pockets**) that updates can come from for the 24.04 LTS release (noble-security, noble, noble-updates and noble-backports – remember these, as we’ll come back to them later). It has found some packages that can be upgraded to newer versions. If we want to see which packages those are, we can run the command hinted in the output:

```
$ apt list --upgradable
```

The output tells us:

- the package name and where the update will come from (e.g. base-files/noble-updates),
- the most up-to-date package version available (e.g. 13ubuntu10.1)
- the hardware version the update is for (e.g. amd64), and
- what package version is currently installed (e.g. 13ubuntu10)

The specific packages included in this list changes over time, so the exact packages shown will be different, but the output will be structured like this:

³⁴ <https://wiki.debian.org/AptCLI>

```
bind9-dnssutils/resolute-updates,resolute-security 1:9.20.18-1ubuntu2.1 amd64
[upgradable from: 1:9.20.18-1ubuntu2]
bind9-host/resolute-updates,resolute-security 1:9.20.18-1ubuntu2.1 amd64 [upgradable
from: 1:9.20.18-1ubuntu2]
bind9-libs/resolute-updates,resolute-security 1:9.20.18-1ubuntu2.1 amd64 [upgradable
from: 1:9.20.18-1ubuntu2]
bpftool/resolute-updates 7.7.0+7.0.0-22.22 amd64 [upgradable from: 7.7.0+7.0.0-15.
15]
libgcrypt20/resolute-updates,resolute-security 1.12.0-2ubuntu0.1 amd64 [upgradable
from: 1.12.0-2]
[...]
```

apt upgrade

The `apt upgrade` command is about the **packages** on your system. It looks at the metadata in the package index we just updated, finds the packages with available upgrades, and lists them for us. Once we've checked the proposed upgrade and are happy to proceed, it will then install the newer versions for us.

After we have updated the database (which we did by running `apt update`) we can then upgrade the packages to their newest versions by running:

```
$ sudo apt upgrade
```

When we run this command, it will ask us to confirm if the summary of proposed changes that will be made to our system is what we want.

Let's type `Y`, then press `Enter` to confirm that yes, we do want that, and then the upgrade will proceed. This may take a few minutes.

Tip:

You can use the `-y` flag, which is a shorthand for `--assume-yes`. If we ran the command `sudo apt upgrade -y` it would proceed with the upgrade without asking us to confirm. Shorthand versions of flags are common – for most packages, you can check which flags are equivalent using [the manual pages](https://manpages.ubuntu.com/)³⁵ or using the `man` command, as we'll see later.

³⁵ <https://manpages.ubuntu.com/>

In the output, we'll see where `apt upgrade` is fetching the upgrade from for each package. For example:

```
Get:1 http://archive.ubuntu.com/ubuntu resolute-updates/main amd64 libgcrypt20 amd64
1.12.0-2ubuntu0.1 [670 kB]
```

APT combines the various elements; the package name (`libgcrypt20`), version (`1.21.0-2ubuntu0.1`), source (`resolute-updates/main`), etc into a single URL that it can use for the download. The package is then unpacked, and the upgrade applied to the system.

Note:

These commands only upgrade the packages for the release of Ubuntu that we are using (26.04 LTS). If we wanted to upgrade the entire system to the next release of Ubuntu (e.g. from 22.04 LTS to 24.04 LTS), we would use the `do-release-upgrade` command. See this guide on [how to upgrade your release](#) (page 435) for more information.

It's important to know that `apt upgrade` will only handle packages that can be straightforwardly upgraded. If the package has **dependency** issues (i.e., the version you have “depends” on other packages that also need to be added, upgraded or removed), you would need to use `sudo apt dist-upgrade` instead. The `dist-upgrade` command is able to resolve conflicts between package versions, but it *could* end up removing some packages – so although `apt upgrade` is safe to use unattended (in a script, for example), you should only use `dist-upgrade` when you can pay attention to it.

Searching with APT

Now we're up-to-date, we can start exploring! As with any other database, we can search the list of available packages using APT in order to find software. Let's say that we want to find a web server, for example. We can run the following command:

```
$ apt search webserver
```

This will return us a long list of all “webserver” packages it can find. But some of the descriptions don't actually contain the text “webserver” – like in this section of the list:

```
inotify-tools/resolute 4.25.9.0-1 amd64
  command-line programs providing a simple interface to inotify

ipcalc/resolute 0.51-1build1 all
  parameter calculator for IPv4 addresses

iwatch/resolute 0.2.2-12 all
  realtime filesystem monitoring program using inotify
```

We can use `apt show` to inspect the description and summary details of any package, so let's take a closer look at `ipcalc` from our list:

```
$ apt show ipcalc
```

The summary has been replaced with [...] for brevity, but we can see that the text “webserver” is in the long description of the “Description” field.

```
Package: ipcalc
Version: 0.51-1build1
[...]
APT-Sources: http://archive.ubuntu.com/ubuntu resolute/universe amd64 Packages
Description: parameter calculator for IPv4 addresses
  ipcalc takes an IPv4 address and netmask and calculates the resulting
  broadcast, network, Cisco wildcard mask, and host range. By giving a
  second netmask, you can design sub- and supernetworks. It is also
  intended to be a teaching tool and presents the results as
  easy-to-understand binary values.
.
Originally, ipcalc was intended for use from the shell prompt, but a
CGI wrapper is provided to enable colorful HTML display through a
webserver.
```

In many places, you will see reference to `apt-get` and `apt-cache` instead of `apt`. Historically, the *database* part of APT was accessed using `apt-cache` (e.g. `apt-cache show ipcalc`), and the *packages* part of APT used `apt-get` (e.g. `apt-get install ipcalc`).

APT has recently been streamlined, so although it uses `apt-get` and `apt-cache` “behind the scenes” (and these commands do still work), we don’t need to worry about remembering which command to use – we can use the more convenient `apt` directly. To find out more about these packages and how to use them (or indeed, any package in Ubuntu!) we can refer to the manual pages.

Run `man apt`, `man apt-get` or `man apt-cache` in the terminal to access the manuals for these packages on the command line, or view the same content in the [online manual pages](#)³⁶.

Installing deb packages

For the examples for this section, we’re going to use the popular web server package, [Apache2](#)³⁷.

APT gives us a lot of details about what will be included in the installation, and it’s always important to understand the implications of a command *before* we run it. We’ll be taking a close look at the details APT gives us, so we need to be careful in this section.

When we run a command that asks us “Do you want to continue? [Y/n]”, make sure to type N for “no” and then press Enter unless instructed otherwise – this will let us see the output of the commands without making changes that then need to be undone.

Installing deb packages using APT is done using the `apt install` command. We can install either a single package, or a list of packages at once, by including their names in a space-separated list after the `install` command, in this format:

```
sudo apt install <package 1> <package 2> <package 3>
```

³⁶ <https://manpages.ubuntu.com>

³⁷ <https://httpd.apache.org/>

About sudo

We've seen the `sudo` prefix in a couple of commands already, and you may be wondering what that's about. In Linux, system tasks (like installing software) need elevated administrator permissions. These permissions are often called "root access", and a user with root access is called a "root user".

However, it can be dangerous to operate your machine as a root user – since root access gives you full system control the whole time, it allows you to change or delete important system files. It's very easy to accidentally break your system in root mode!

Instead, we use `sudo` (which is short for superuser do). This command is a safety feature that grants regular users *temporary* (per command) admin privileges to make system changes. It's still important for us to always understand what a command does before we run it, but using `sudo` means we purposefully limit any potential mistakes to a single command.

About dependencies

As we hinted earlier, packages often come with **dependencies** – other packages that *your* package needs so it can function. Sometimes, a package might depend on a specific version of another package. If a package has dependencies, then installing a package via `apt` will also install any dependencies, which ensures the software can function properly.

APT tells us how it will resolve any dependency conflicts or issues when we run the `install` command. Let's try this for ourselves, but remember, we **don't** want to proceed with the install yet, so let's type `N` when it asks us if we want to continue:

```
$ sudo apt install apache2
```

The output should be similar to the below. It tells us:

- which packages we have but don't need (we'll talk about that in the "auto-remove" section),
- additional packages that will be installed (these are our dependencies),
- suggested packages (which we'll discuss in the next section), and
- a summary of which *new* packages will be present on the system after the install is done (which in this case is `apache2` itself, and all its dependencies).

```
Installing:
  apache2

Installing dependencies:
  apache2-bin      libapr1t64          libaprutil1t64
  apache2-data     libaprutil1-dbd-sqlite3  liblua5.4-0
  apache2-utils    libaprutil1-ldap      ssl-cert

Suggested packages:
  apache2-doc  apache2-suexec-pristine | apache2-suexec-custom  www-browser

Summary:
  Upgrading: 0, Installing: 10, Removing: 0, Not Upgrading: 0
  Download size: 2116 kB
  Space needed: 8218 kB / 1328 MB available

Continue? [Y/n]
```

Let's try and make sense of this output.

Types of dependencies

The relationship between a package and any other packages follows the [Debian policy on binary dependencies](#)³⁸, which we'll briefly look at here. The most common ones you might come across are: depends, recommends, and suggests (although there are others!), so we'll take a look at these three.

- **depends:** Absolutely required, the package won't work without it. If we try to remove a package that is depended on by another, both will be removed!
- **recommends:** Strongly dependent, but not absolutely necessary (which means the package will work better with it, but can still function without it)
- **suggests:** Not needed, but may enhance the usefulness of the package in some way.

We can see, using `apt show`, exactly which packages fall into each of these categories. Let's use Apache2 as our example again:

```
$ apt show apache2
```

If we look only at the sections on dependencies, we can see that `ssl-cert` is a recommended package:

```
[...]
Installing dependencies:
  apache2-bin      libapr1t64          libaprutil1t64
  apache2-data     libaprutil1-dbd-sqlite3  liblua5.4-0
  apache2-utils    libaprutil1-ldap      ssl-cert
[...]
```

³⁸ <https://www.debian.org/doc/debian-policy/ch-relationships.html#binary-dependencies-depends-recommends-suggests-enhances>

In Ubuntu, the default configuration of `apt install` is set to install recommended packages alongside depends, so when we ran the `apt install apache2` command, `ssl-cert` was included in the proposed packages to be installed (even though it's only recommended, not strictly needed).

We can override this behavior by passing the `--no-install-recommends` flag to our command, like this:

```
$ sudo apt install apache2 --no-install-recommends
```

Then the output becomes the following (type N at the prompt again to avoid installing for now):

```
[...]
Installing dependencies:
  apache2-bin  apache2-utils  libaprutil1-dbd-sqlite3  libaprutil1t64
  apache2-data  libapr1t64      libaprutil1-ldap        liblua5.4-0

Suggested packages:
  apache2-doc  apache2-suexec-pristine | apache2-suexec-custom  www-browser

Recommended packages:
  ssl-cert

Summary:
  Upgrading: 0, Installing: 9, Removing: 0, Not Upgrading: 0
  Download size: 2098 kB
  Space needed: 8149 kB / 1328 MB available
[...]
```

Now, we see that `ssl-cert` is only mentioned as a recommended package, but is excluded from the list of packages to be installed.

There is a second flag we could pass – the `--install-suggests` flag. This will not only install the strict dependencies and recommended packages, but *also* the suggested packages. From our previous output, it doesn't look like too much, right? It's only four additional packages.

But actually, if we run this command:

```
$ sudo apt install apache2 --install-suggests
```

By comparing the amount of space needed with “suggests” included, to the previous output, we can see that there's a considerable increase! Sometimes, including suggested packages will cause the package to not be installed due to a lack of space on the system:

```
[...]
Installing dependencies:
  apache2-bin      apache2-utils      libaprutil1-ldap
  apache2-data     chromium-browser   libaprutil1t64
  apache2-doc      libapr1t64         liblua5.4-0
  apache2-suexec-pristine  libaprutil1-dbd-sqlite3  ssl-cert

Summary:
  Upgrading: 0, Installing: 13, Removing: 0, Not Upgrading: 0
  Download size: 6078 kB
  Space needed: 34.1 MB / 1328 MB available
```

This is because each of these suggested packages also comes with their own lists of dependencies, including suggested packages, all of which would *also* be installed. It's perhaps clear to see why this is not the default setting!

What if we remove a dependency?

We'll go into more detail about removing packages later, but for now, let's see what happens if we remove a required *dependency*. First, we should (finally!) install the `apache2` package. Let's run the following command again, but this time when we are asked whether we want to continue, let's press `Y` and then `Enter` to confirm, and `APT` will install the package:

```
$ sudo apt install apache2
```

One of the required dependencies is the `apache2-data` package. Let's try to remove it using `apt remove`:

```
$ sudo apt remove apache2-data
```

Once again, `apt` won't proceed without confirmation, so we get the following output – let's take a look before we choose anything:

```
[...]
The following packages were automatically installed and are no longer required:
  apache2-bin      libapr1t64      libaprutil1-ldap  liblua5.4-0
  apache2-utils   libaprutil1-dbd-sqlite3  libaprutil1t64    ssl-cert
Use 'sudo apt autoremove' to remove them.

REMOVING:
  apache2  apache2-data

Summary:
  Upgrading: 0, Installing: 0, Removing: 2, Not Upgrading: 0
  Freed space: 1349 kB

Continue? [Y/n]
```

Let's break this down a little bit, because there are some subtle differences here that we want to understand before we proceed.

- “The following packages were automatically installed and are no longer required”

These were other dependencies that apache2 needed, but none of them depend upon apache2-data, so even if we remove apache2 and apache2-data they would still be functional – they just aren’t used by any other installed packages...and so have no reason to be there anymore. They won’t be removed, APT is helpfully telling us so we’re aware of them.

- “The following packages will be REMOVED”

These are the packages that will be removed directly - we’ve told APT we want to remove apache2-data, so we expect that to be included, but it will also remove apache2 itself! This is because apache2-data is a required dependency, and apache2 won’t function *at all* without it.

Let’s now choose Y to confirm we want to remove this dependency.

Warning:

Removing dependencies can, at worst, cause a system to become unusable – you should always be careful when doing so. If you remove a dependency that is part of a chain, the removals will cascade up the chain as each dependency and the package that depends on it are removed. You can end up removing more than you originally anticipated!

Auto-remove dependencies

So, we have removed the apache2 and apache2-data packages, but the other dependencies that were installed alongside apache2 are still there. The output of our remove command gave us the hint about how to deal with these redundant packages – the autoremove command:

```
$ sudo apt autoremove
```

When we run this command, apt once again gives us a summary of the operation we requested, but let’s choose N for now when it asks if we want to continue:

```
[...]
REMOVING:
  apache2-bin      libapr1t64          libaprutil1-ldap    liblua5.4-0
  apache2-utils    libaprutil1-dbd-sqlite3  libaprutil1t64      ssl-cert

Summary:
  Upgrading: 0, Installing: 0, Removing: 8, Not Upgrading: 0
  Freed space: 6869 kB

Continue? [Y/n]
```

You may be wondering why we don’t need to specify any packages when we call the autoremove command – after all, we’ve just been dealing with packages related to apache2. This is because apt will check all the packages on your system. It examines the dependency tree, and if the original reason for the package to be installed no longer exists (i.e., it isn’t needed by anything), it will be flagged for auto-removal.

But!

We might, in the future, uninstall Apache2 without uninstalling the redundant packages at the time. We might have found another use for `ssl-cert`, perhaps in a script that makes use of SSL certificates. So how can we keep the `ssl-cert` package, even though it's flagged for auto-removal?

We can solve this problem, and un-flag the `ssl-cert` package for removal, by *manually* installing it:

```
$ sudo apt install ssl-cert
```

This sets `ssl-cert` to **manually installed**. We might well wonder “why didn't APT didn't ask us to confirm anything this time?”. In this case, it's because `ssl-cert` is already present on the system so APT doesn't need to install anything new.

```
[...]
ssl-cert is already the newest version (1.1.3ubuntu2).
ssl-cert set to manually installed.
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
```

If the `ssl-cert` package is manually installed on our system, by us, then apt knows the package is wanted, and we can see that it has been removed from the auto-remove list so our next autoremove will not uninstall it. Let's test this, just to make sure!

```
$ sudo apt autoremove
```

This time we'll select Y when prompted, and then we can run `apt list ssl-cert` to quickly see if our `ssl-cert` package is still on the system:

```
$ apt list ssl-cert

ssl-cert/resolute,now 1.1.3ubuntu2 all [installed]
```

If you're curious, you can also run `apt list apache2` to see how the output differs for a package that was once installed and then removed!

Anyway, we're not quite finished with the Apache2 package, so let's reinstall it:

```
$ sudo apt install apache2
```

And this time select Y to confirm when it asks.

Customize configuration

In general, the default package configuration should just work well, and work “out of the box” when it’s installed. But it’s almost inevitable that, sooner or later, we’ll want to customize the package so that it better fits our own purposes.

Before we try to customize the package, we should probably look at what files are included in it. We can check this using `dpkg(1)`³⁹, which is the Debian package manager. Although APT is now more commonly used for basic package handling, `dpkg` retains some really helpful commands for examining files and finding out package information. It’s installed by default on Ubuntu systems so we can use it directly:

```
$ dpkg --getfiles ssl-cert
```

This gives us the following list of files and their directory structure (the end of the list is truncated for brevity):

```
/.
/etc
/etc/ssl
/etc/ssl/certs
/etc/ssl/private
/usr
/usr/lib
/usr/lib/systemd
[...]
/usr/share/man
/usr/share/man/man8
/usr/share/man/man8/make-ssl-cert.8.gz
/usr/share/ssl-cert
/usr/share/ssl-cert/ssleay.cnf
[...]
```

If we find a file but we’re not sure what package it comes from, `dpkg` can help us there too! Let’s use the example of one of the configuration files from the previous output: `/usr/share/ssl-cert/ssleay.cnf` and do a search for it using `dpkg`:

```
$ dpkg --getowner /usr/share/ssl-cert/ssleay.cnf
ssl-cert: /usr/share/ssl-cert/ssleay.cnf
```

Although this is an obvious case, because we already know the source of this file, the `dpkg` search function is really useful for tracking down the sources of files we don’t know about!

³⁹ <https://manpages.ubuntu.com/manpages/resolute/man1/dpkg.1.html>

Conffiles

Most of a package's configuration is handled through [configuration files](#)⁴⁰ (often known as **conffiles**). Conffiles often contain things like file paths, logs and debugging configuration, kernel parameters (which can be changed to optimize system performance), access control, and other configuration settings. The actual parameters available will vary from one package to another.

Package conffiles are different from all other files delivered in a package. A package may have any number of conffiles (including none!). Conffiles are explicitly marked by the package maintainer during development to protect local configuration from being overwritten during upgrades so that your changes are saved. This is not the case for any other types of files – changes you make to regular files in that package *will be overwritten* during an upgrade.

How upgrades are handled

Since a conffile can be changed by us, we might end up with conflicts when the package maintainer changes those same files. Therefore, it's important to understand how such conflicts are handled.

We can show the four possible upgrade scenarios using the following table. What happens during an upgrade depends on whether the conffile on our system has been changed by us ("changed/not changed by user"), and whether the version's default content has been changed by the package maintainer ("changed/not changed by maintainer"):

The conffile is...	not changed by maintainer	changed by maintainer
...changed by user	Keep user's changes	Ask user
...not changed by user	No changes to make	Apply changes from update

So we can see that if we do make changes to a conffile, APT will never overwrite our changes without asking us first.

Identifying conffiles

Out of the list of files in a package, how do we know which ones are the conffiles?

After all, they are not marked by any particular file extension, and although they are often found in the `/etc/` directory, they don't *have* to be there. As we saw before, the only thing conffiles have in common is that the package maintainer decided to mark them as such.

But that's our clue! So once more, `dpkg` can come to our rescue. The following command will show us (`--show`) the subset of files in the `apache2` package that have been marked as "Conffiles" (`-f='${Conffiles}\n'`) by the maintainer and shows each on a new line (`\n`) in the output:

```
$ dpkg-query --show -f='${Conffiles}\n' apache2
```

⁴⁰ <https://www.debian.org/doc/debian-policy/ap-pkg-conffiles.html#automatic-handling-of-configuration-files-by-dpkg>

If you want to understand more about what this command does, you can refer to the manual page by typing `man dpkg-query --show`, and it will talk you through all the options.

Unlike `dpkg --getfiles`, `dpkg-query` *also* gives us a string of letters and numbers. This string is known as the “**MD5 checksum**” or “**MD5 hash**”.

```
/etc/apache2/apache2.conf 354c9e6d2b88a0a3e0548f853840674c
/etc/apache2/conf-available/charset.conf e6fbb8adf631932851d6cc522c1e48d7
/etc/apache2/conf-available/localized-error-pages.conf
f542d267bfce7815f9453eb1476e5f73
/etc/apache2/conf-available/other-vhosts-access-log.conf
2cad303fc4221d6b0068a8b37597b9fb
/etc/apache2/conf-available/security.conf 332668933023a463046fa90d9b057193
[...]
```

We can see the checksum of a specific file by running this command, which returns us the checksum followed by the file and its location:

```
$ md5sum /etc/apache2/apache2.conf

354c9e6d2b88a0a3e0548f853840674c /etc/apache2/apache2.conf
```

You might well be wondering “why do we care about that?” since they match (in this example).

The checksum is like a fingerprint - it’s unique for every *version* of a file, so any time the file is changed it will get a new checksum – which allows us to see **if a file has been changed** from the default.

Verifying checksums

Let’s set up a situation so we can poke a bit at this idea. We can start by making some changes to a conf file. In Apache2, the main conf file is `/etc/apache2/apache2.conf`, so let’s use that. In a situation where we are setting up a new web server, we might reasonably want to increase the `LogLevel` from “warn” to “debug” to get more debugging messages, so let’s run this command and use `sed` to make that change in the conf file:

```
$ sudo sed -e 's/LogLevel warn/LogLevel debug/' -i /etc/apache2/apache2.conf
```

We won’t be prompted to confirm if we want to make these changes – but we do need root access so we use `sudo` in our command. As we hinted in the section about `sudo`, the fact that we can make these changes without needing to confirm is why it can be so easy to break your system when you’re operating as root! Try running the command without the `sudo`, and you will get a “permission denied” error.

Next, we’ll restart our Apache2 server so that we can activate our configuration changes:

```
$ sudo systemctl restart apache2
```

Now if we run the `md5sum` command again, we can see the hash changed:

```
ubuntu@tutorial:~$ md5sum /etc/apache2/apache2.conf
1109a77001754a836fb4a1378f740702  /etc/apache2/apache2.conf
```

This works great if we know that there's a file we changed, but what about if someone else tampered with a file, and we don't know which one? In that case, we can use:

```
$ dpkg --verify apache2
```

This will verify the checksums of the files on our system against those held in the package index for apache2, and return a rather strange looking result if (or when) it finds a mismatch:

```
??5?????? c /etc/apache2/apache2.conf
```

Which is exactly what we were expecting to see, since we know we changed this file.

But what if something else was messed with...something that shouldn't be, and something not changed by us? Let's make a "silly" change to a different file to test this – in this case, changing all instances of the word "warning" to "silly" in a random package file:

```
$ sudo sed -e 's/warning/silly/' -i /usr/sbin/a2enmod
```

And then run the verification again with:

```
$ dpkg --verify apache2
??5?????? c /etc/apache2/apache2.conf
??5??????  /usr/sbin/a2enmod
```

Note:

You might have noticed there's a "c" next to the top line but not the bottom – the "c" shows the file is a conffile.

dpkg can tell that the file has been changed, but won't tell us what the change was. However, since the file in question is not a conffile, we know that the change *won't be preserved* if we upgrade the package. This means that we can overwrite the changes and restore the default package content by "reinstalling" Apache2:

```
$ sudo apt install --reinstall apache2
```

By using the `--reinstall` flag, we can force apt to re-unpack all of the default content. If we then verify once more...

```
$ dpkg --verify apache2
??5?????? c /etc/apache2/apache2.conf
```


...so we can see that our change to the conf file has been preserved because the checksums are different, but the `a2enmod` file isn't listed anymore because it has been restored to the default. Phew!

Note:

We can use `sudo apt install <package>` to upgrade an installed package, but this will only upgrade to the latest version. In our case, we were already on the latest version of Apache2, so we needed to force APT to re-unpack the content to overwrite our "silly" changes.

Removing packages

Since we have just reinstalled the Apache2 package, we know it is in good shape. But what if we decide we're done with it and just want to remove it? Then we can run:

```
$ sudo apt remove apache2

The following packages were automatically installed and are no longer required:
  apache2-bin  apache2-utils  libaprutil1-dbd-sqlite3  libaprutil1t64
  apache2-data  libapr1t64      libaprutil1-ldap        liblua5.4-0
Use 'sudo apt autoremove' to remove them.

REMOVING:
  apache2

Summary:
  Upgrading: 0, Installing: 0, Removing: 1, Not Upgrading: 0
  Freed space: 471 kB

Continue? [Y/n]
```

Let's type Y to proceed.

As before, we see that the dependencies will still be there even when `apache2` has been removed. Let's check with `dpkg`...

```
$ dpkg --getfiles apache2
```

...and see what else might be left behind...

```
/etc
/etc/apache2
/etc/apache2/apache2.conf
/etc/apache2/conf-available
/etc/apache2/conf-available/charset.conf
/etc/apache2/conf-available/localized-error-pages.conf
/etc/apache2/conf-available/other-vhosts-access-log.conf
/etc/apache2/conf-available/security.conf
/etc/apache2/conf-available/serve-cgi-bin.conf
/etc/apache2/conf-enabled
/etc/apache2/envvars
/etc/apache2/magic
/etc/apache2/mods-available
/etc/apache2/mods-available/access_compat.load
/etc/apache2/mods-available/actions.conf
[...]
```

This looks suspiciously like the list of conffiles we saw earlier, right?

Also removing configuration

As it turns out, removing a package doesn't automatically remove the conffiles. But – this is intentional, for our convenience.

By leaving the conffiles in place, if we decide to reinstall `apache2` again in the future, we don't need to spend time setting up all our configuration again.

Let's see the difference in installing `apache2` after it has been installed (and removed) compared to the first time we installed it:

```
$ sudo apt install apache2
```

Notice that it did not ask us to confirm if we wanted to proceed this time. Why not? As we saw earlier, the "Y/n" confirmation is shown when there are dependencies, and we know that `Apache2` *has* dependencies.

...Ah! But this time, we didn't run `autoremove` when we uninstalled `Apache2`, so the dependencies are still installed on our system. This means that when we ask `apt` to install `apache2` now, there is nothing missing and we are getting *exactly* what we are asking for.

Since the dependencies and conffiles are still there, we can use our former config immediately. It even retains the changes we made before, which we can verify by looking at the checksum again:

```
$ md5sum /etc/apache2/apache2.conf
```

Removing and purging

What if we decide that we don't want the changed conffiles? Perhaps we want to go back to the default installation, or we know we won't want to use the package ever again – how can we ensure that all the conffiles are removed at the same time as we remove the package?

In that case, we can use the `--purge` option of the `remove` command:

```
$ sudo apt remove --purge apache2

[...]
The following packages were automatically installed and are no longer required:
  apache2-bin  apache2-utils  libaprutil1-dbd-sqlite3  libaprutil1t64
  apache2-data  libapr1t64      libaprutil1-ldap        liblua5.4-0
Use 'sudo apt autoremove' to remove them.

REMOVING:
  apache2*

Summary:
  Upgrading: 0, Installing: 0, Removing: 1, Not Upgrading: 0
  Space needed: 0 B / 1318 MB available

Continue? [Y/n]
```

If we look very carefully, we see a little asterisk (*) in the output.

```
REMOVING:
  apache2*
```

This tiny indicator tells us that the package will be removed AND purged. However, it still does not remove the dependencies (or the conffiles of those dependencies).

Let's type Y again to confirm we want to proceed. Then, once the removal is complete, we can check the list once more:

```
$ dpkg --getfiles apache2

dpkg-query: package 'apache2' is not installed
Use dpkg --get-contents (= dpkg-deb --get-contents) to list archive files contents.
```

This time the output is very different!

Note:

We could also use the `dpkg-query --show -f='${Conffiles}\n' apache2` command from earlier, and `dpkg-query` will find no packages matching `apache2`.

There are other ways to change package files. If you would like to read more, check out our [guide to changing package files](#) (page 956).

What else is on our system?

As we saw earlier, we can search the APT package database for keywords using `apt search <keyword>` to find software we might want to install. We can also see all the packages we already have using `apt list`, although it can be easier to navigate and more informative if we use `dpkg -l` instead – then we can use the up and down arrow keys on our keyboard to scroll (or press `Q` to return to our terminal prompt).

For every package, we can see what versions of it exist in the database:

```
$ apt policy apache2
```

This will return a summary of all the versions that exist on our particular Ubuntu release, ordered by “most recent” first:

```
apache2:
  Installed: (none)
  Candidate: 2.4.66-2ubuntu2.1
  Version table:
     2.4.66-2ubuntu2.1 500
        500 http://archive.ubuntu.com/ubuntu resolute-updates/main amd64 Packages
        500 http://security.ubuntu.com/ubuntu resolute-security/main amd64 Packages
     2.4.66-2ubuntu2 500
        500 http://archive.ubuntu.com/ubuntu resolute/main amd64 Packages
```

We know that Apache2 isn't installed right now, because we removed and purged it, which is why the installed version shows as “none”:

```
Installed: (none)
```

If we were to install the default package, we would get this one:

```
Candidate: 2.4.66-2ubuntu2.1
```

Under each version we are also shown the **source**. The newest version (2.4.66-2ubuntu2.1) comes from `resolute-updates (main)` and `resolute-security (main)`. The *original* version (2.4.66-2ubuntu2) comes from `resolute (main)`. This tells us that this was the version released with the with 26.04 LTS (Resolute Raccoon).

Installing older package versions

We can install specific older versions if we want to, for example, to satisfy dependency requirements of another package. We can do that by specifying the package name and version:

```
sudo apt install <package>=<version>
```

However, this can be tricky and often leads to conflicts in dependency versions as APT always wants to install the most recent version. We can see an example of this if we run the following command:

```
$ sudo apt install apache2=2.4.66-2ubuntu2
```

APT warns us that the version of apache2 we want to install depends on earlier versions of the dependencies, but it helpfully tells us which dependency versions we need to successfully install the package we want.

```
[...]
Solving dependencies... Error!
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

Unsatisfied dependencies:
  apache2 : Depends: apache2-bin (= 2.4.66-2ubuntu2) but 2.4.66-2ubuntu2.1 is to be
installed
              Depends: apache2-data (= 2.4.66-2ubuntu2) but 2.4.66-2ubuntu2.1 is to be
installed
              Depends: apache2-utils (= 2.4.66-2ubuntu2) but 2.4.66-2ubuntu2.1 is to be
installed
Error: Unable to satisfy dependencies. Reached two conflicting assignments:
 1. apache2:amd64=2.4.66-2ubuntu2 is selected for install
 2. apache2:amd64=2.4.66-2ubuntu2 Depends apache2-bin (= 2.4.66-2ubuntu2)
    but none of the choices are installable:
    - apache2-bin:amd64=2.4.66-2ubuntu2 is not selected for install
```

So, all we need to do is first install the dependencies, and then run the install command again. Remember that we can install multiple packages at once by separating them with spaces:

```
$ sudo apt install apache2-bin=2.4.66-2ubuntu2 \ >
  apache2-data=2.4.66-2ubuntu2 \ > apache2-utils=2.4.66-2ubuntu2 \ >
  apache2=2.4.66-2ubuntu2
```

In this case we're also breaking the command over multiple lines using backslashes (\) to make it easier to read, but it will still be run as a single command.

APT will warn us that we are downgrading the package, but let us press Y to confirm (when prompted), and it will go ahead and downgrade us anyway. Let's run the following command again to get confirmation that we're running on an older version:

```
$ apt policy apache2
```

```
apache2:
  Installed: 2.4.66-2ubuntu2
  Candidate: 2.4.66-2ubuntu2.1
  Version table:
    2.4.66-2ubuntu2.1 500
        500 http://archive.ubuntu.com/ubuntu resolute-updates/main amd64 Packag>
        500 http://security.ubuntu.com/ubuntu resolute-security/main amd64 Pack>
    *** 2.4.66-2ubuntu2 500
        500 http://archive.ubuntu.com/ubuntu resolute/main amd64 Packages
        100 /var/lib/dpkg/status
lines 1-10/10 (END)
```

Where do packages come from?

You may be wondering by now “where exactly do all these packages come from?”. We’ve spotted a few sources very briefly throughout this tutorial, but haven’t paid direct attention to them yet. Let’s take a little time now to define what we mean by all these different sources that APT can pull packages from.

The source behind APT is the **Ubuntu Package Archive**. This Archive splits into many layers, each with its own terminology. The different terminology is quite confusing at first, but we’ve seen a few of the terms already. So if we take a look, layer-by-layer, we’ll see not just what all the terms mean, but how they all fit together.

Let’s have a quick overview with this diagram. The general flow is that the Archive splits into **Ubuntu series**. Each series is split up into **pockets**, and then each pocket contains four **components**. If we tried to show all of this on one diagram, it would be quite extensive, so let’s take a look through a single path.

Series

The **series** is a set of packages that are released with a specific version of Ubuntu – they’re usually referred to by their codename (e.g., mantic, noble and oracular in our diagram). Each version of Ubuntu may have multiple releases. For example, an LTS will have an initial release when it launches (e.g. 24.04 LTS), and then “subsequent point releases” (e.g. 24.04.1 LTS) – these are all part of the same series (noble).

In practice, people often use the term “Ubuntu release” and “Ubuntu series” interchangeably.

Pockets

Every Ubuntu series (noble, jammy, etc) is split into **pockets**, which are related to the software development/release lifecycle:

- **-release** contains the packages as they are at release time.
- **-proposed** contains package updates while they are being tested.
- Once an update is released, they come from either **-security** or **-updates** depending on whether they are a security-related update or not.

- And **-backports**, which contains packages that were not available at release time.

This is why earlier, we saw that some updates came from `resolute-updates` or `resolute-security`. These refer to updates and security updates from the noble series (respectively). Pockets are usually appended to the end of the series, and it's quite common to see the hyphen (-) included when referring to pockets.

Remember – the original version of the `apache2` package we saw came from `resolute`. The `-release` pocket only includes the software that was part of the original LTS release, and so it takes the name of the Ubuntu series by default (i.e., the `-release` pocket is implied).

Components

Each pocket is split into four **components**, depending on whether the packages they contain are *open source* or *closed source*, and whether they are officially supported by Canonical or are maintained by the Ubuntu Community:

	Open source	Closed source
Officially supported	main	restricted
Community supported	universe	multiverse

- **main** contains the open-source packages that are officially supported by Canonical. These packages are either installed on every Ubuntu machine, or are very widely used for various types of systems and use-cases.
- **universe** holds all other open-source packages in Ubuntu, which are typically maintained by the Debian and Ubuntu communities, but may also include additional security coverage from Canonical under [Ubuntu Pro](https://ubuntu.com/pro)⁴¹, which is available free for personal use on up to five machines.
- **restricted** contains the packages that are officially supported by Canonical but are not available under a completely free license.
- **multiverse** contains community-maintained proprietary software – these packages are completely unsupported by Canonical.

If you would like more information about the Ubuntu release process, how packages are produced, or to learn more about the sort of terminology you might come across, you may be interested in the [Ubuntu Project documentation](https://documentation.ubuntu.com/project/)⁴², which is a great resource containing all this information (and much more!).

⁴¹ <https://ubuntu.com/pro>

⁴² <https://documentation.ubuntu.com/project/>

Installing a .deb file

Although APT is the preferred way to install packages on your system, due to its ability to handle dependencies and keep software up-to-date, not every package is available in the APT repository – especially if they are so old they are no longer maintained, or conversely, are the newest version still in development!

We can install .deb files that aren't in the APT repository using `dpkg` – all we need is to download the .deb file, and we can run a command like this to install it:

```
sudo dpkg -i <file-name.deb>
```

But – APT is helpful here too. Even if we get a .deb file that isn't from the Ubuntu Archive, we can still install it with APT so that if there are dependencies that can be resolved automatically from the Archive – they will be!

```
sudo apt install ./file-name.deb
```

If we ever do want to install a .deb file, APT is definitely the most convenient way to do it. We may still need to handle *some* dependencies manually, but now we have the knowledge to be able to do that.

Luckily, most of the packages you will ever need are likely to be found through APT. If it's not, it's worth checking if the software is available as a **snap** instead.

Snaps

Snaps are a newer, self-contained software format that were developed to be a more portable and easy-to-use alternative to debs. They come with all their dependencies pre-bundled so that there is no need for a package management tool to track dependencies, and they run inside sandboxed environments that limit their interactions with the rest of the system.

Instead of **versions** as we have them in debs, snaps use the concept of **channels**⁴³ to define which release of a snap is installed.

By default, snaps are kept automatically up-to-date, so we don't need to remember to update and upgrade them. There are times on a live system, such as a server in a production environment, where we might not want to have updates automatically applied. In those cases, we can **turn off automatic updates**⁴⁴ and refresh the system snaps when it's convenient (for example, during a maintenance window).

If you would like to try out snaps, we recommend the excellent **quickstart tour**⁴⁵ tutorial in the snap documentation. Feel free to continue using the VM we've been using in this tutorial while exploring!

⁴³ <https://snapcraft.io/docs/channels>

⁴⁴ <https://snapcraft.io/docs/managing-updates>

⁴⁵ <https://snapcraft.io/docs/get-started>

Completion!

Once you are finished and want to leave the tutorial, you can run:

```
$ exit
```

This will take you out of the VM and back to your live machine. Then, you can run the following commands to delete the VM and remove it completely from your machine:

```
$ multipass delete tutorial
```

```
$ multipass purge
```

Summary

Congratulations, we made it to the end! We've covered a lot of material in this tutorial, so let's do a quick recap of what we've learned:

Finding, installing and removing packages

- How to update and upgrade all our system's software with APT:
 - `sudo apt update && sudo apt upgrade`
- How to search for software using keywords or strings:
 - `apt search <keyword>` or `apt search "some content string"`
- How to see the description of a package, including what dependencies it has:
 - `apt show <package name>`

Or how to check what package versions are available:

- `apt policy <package>`
- How to install packages...
 - `sudo apt install <package1> <package2>`
- How to see all the files a package contains
 - `dpkg --getfiles <package>`
- How to find out what package a file belongs to:
 - `dpkg --get-selections <path/to/file>`
- ...And how to remove packages again! As well as the difference between removing and purging.
 - `sudo apt remove <package>`
- We even learned how to downgrade to older versions of APT packages, and all about APT sources.

Customizing package configuration

- How to find the conffiles in a package:

- `dpkg-query --show -f='${Conffiles}\n' <package>`
- How to see if package files have been changed:
 - `dpkg --verify <package>`
- ...And if a non-conffile has been changed by accident, we can fix it with:
 - `sudo apt install --reinstall <package>`
- We know that our changes to conffiles are always safely preserved, while changes to non-conffiles are reverted at the next upgrade or security fix.
- Importantly, we know how to verify checksums with `md5sum` or similar tools, which helps us to more safely build packages from source.
- And finally, we learned about snaps!

Attach your Ubuntu Pro subscription

Attaching the [Ubuntu Pro](#)⁴⁶ subscription to Ubuntu brings you the [enterprise lifecycle](#)⁴⁷, including [Linux kernel livepatching](#)⁴⁸, access to [FIPS-validated packages](#)⁴⁹, and [compliance with security profiles](#)⁵⁰ such as CIS or DISA-STIG. This is not required for Ubuntu Pro instances [through public clouds](#)⁵¹ such as [AWS](#)⁵², [Azure](#)⁵³ or [GCP](#)⁵⁴, since these are automatically attached from launch.

Note:

Subscriptions are not just for enterprise customers. Anyone can get a [personal subscription](#)⁵⁵ for free on up to 5 machines, or 50 if you are an [official Ubuntu Community member](#)⁵⁶.

⁵⁵ <https://ubuntu.com/pro>

⁵⁶ <https://wiki.ubuntu.com/Membership>

The following instructions explain how to attach your subscription to your Ubuntu systems.

⁴⁶ <https://ubuntu.com/pro>

⁴⁷ <https://ubuntu.com/about/release-cycle>

⁴⁸ <https://ubuntu.com/security/livepatch>

⁴⁹ <https://documentation.ubuntu.com/security/compliance/fips/>

⁵⁰ <https://ubuntu.com/security/security-standards>

⁵¹ <https://ubuntu.com/cloud/public-cloud>

⁵² <https://ubuntu.com/aws/pro>

⁵³ <https://ubuntu.com/azure/pro>

⁵⁴ <https://ubuntu.com/gcp/pro>

Step 1: Install the Ubuntu Pro Client

This step is necessary for Ubuntu Pro users or holders of personal subscriptions. If you are an Ubuntu Pro user through a public cloud offering, your subscription is already attached and you can skip these instructions.

First, make sure that you have the latest version of the Ubuntu Pro Client running. The package used to access the Pro Client (pro) is `ubuntu-pro-client`:

```
$ sudo apt update && sudo apt install ubuntu-pro-client
```

Most systems will have `ubuntu-pro-client` installed by default. If that is the case, this install command will upgrade the package to the latest version.

Step 2: Attach your subscription

To attach your machine to a subscription, run the following command in your terminal:

```
$ sudo pro attach
```

Note:

For users who need to attach many systems or automate the process, as in many enterprise scenarios, there are parameters to the attach command to help you. Please check the Ubuntu Pro documentation for [attaching your system with a configuration file](https://documentation.ubuntu.com/pro-client/en/latest/howtoguides/how_to_attach_with_config_file/)⁵⁷

⁵⁷ https://documentation.ubuntu.com/pro-client/en/latest/howtoguides/how_to_attach_with_config_file/

After running the command, you should see output like this, giving you a link and a code:

```
ubuntu@test:~$ sudo pro attach

Initiating attach operation...

Please sign in to your Ubuntu Pro account at this link:
https://ubuntu.com/pro/attach
And provide the following code: H31JIV
```

Open the link without closing your terminal window.

In the field that asks you to enter your code, copy and paste the code shown in the terminal. Then, choose which subscription you want to attach to. By default, the Free Personal Token will be selected.

If you have a paid subscription and want to attach to a different token, you may want to log in first so that your additional tokens will appear.

Once you have pasted your code and chosen the subscription you want to attach your machine to, click on the “Submit” button.

The attach process will then continue in the terminal window, and you should eventually be presented with the following message:

```
Attaching the machine...
Enabling Ubuntu Pro: ESM Apps
Ubuntu Pro: ESM Apps enabled
Enabling Ubuntu Pro: ESM Infra
Ubuntu Pro: ESM Infra enabled
Enabling Livepatch
Livepatch enabled
This machine is now attached to 'Ubuntu Pro - free personal subscription'
```

When the machine has successfully been attached, you will also see a summary of which services are enabled and information about your subscription.

Available services can be enabled or disabled on the command line with `pro enable <service name>` and `pro disable <service name>` after you have attached.

Next steps

- For more information about the Ubuntu Pro Client, you can [read our documentation](#)⁵⁸.
- For a guided tour through the most commonly-used commands available through the Ubuntu Pro Client, [check out this tutorial](#)⁵⁹.

3.2. Ubuntu Server how-to guides

If you have a specific goal, but are already familiar with Ubuntu Server, our how-to guides have more in-depth detail than our tutorials and can be applied to a broader set of applications. They'll help you achieve an end result but may require you to understand and adapt the steps to fit your specific requirements.

3.2.1. Server installation

Our list of [installation guides](#) (page 75) contains installation instructions for a variety of architecture-specific and advanced setups. For a general installation, or if you're just getting started with Ubuntu, you may prefer to use our [basic installation](#) (page 8) tutorial.

Server installation

If you are new to Ubuntu, we recommend our [basic installation](#) (page 8) tutorial to get you started.

⁵⁸ <https://documentation.ubuntu.com/pro-client/en/latest/>

⁵⁹ https://documentation.ubuntu.com/pro-client/en/latest/tutorials/basic_commands/

Automatic install

The Ubuntu Installer has its own documentation for automatic (or “hands off”) installations. These guides from the Ubuntu Installer documentation are available for automatic installations.

- [Introduction to Automated Server installer](#)⁶⁰
- [Autoinstall quickstart](#)⁶¹
- [Autoinstall quickstart on s390x](#)⁶²

Advanced install

This list of guides contains installation instructions for architecture-specific and more advanced setups. Select your preferred architecture to see which guides are available.

amd64

- [Netboot install](#) (page 77)

arm64

- [Netboot install](#) (page 81)
- [Choose between the arm64 and arm64+largemem installer options](#) (page 85)

ppc64el

- [Netboot install](#) (page 86)
- [Virtual CD-ROM and Petitboot install](#) (page 92)

s390x

- [Install via z/VM](#) (page 99)
- [Non-interactive IBM z/VM autoinstall](#) (page 116)
- [Install via LPAR](#) (page 124)
- [Non-interactive IBM Z LPAR autoinstall](#) (page 142)

⁶⁰ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/intro-to-autoinstall.html>

⁶¹ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/howto/autoinstall-quickstart.html>

⁶² <https://canonical-subiquity.readthedocs-hosted.com/en/latest/howto/autoinstall-quickstart-s390x.html>

See also

- Reference: [System requirements](#) (page 826)

How to netboot the server installer on amd64

[amd64](#) systems boot in either UEFI or legacy [BIOS](#) mode, and many systems can be configured to boot in either mode. The precise details depend on the system [firmware](#), but both modes usually support the “Preboot eXecution Environment” (PXE) specification, which allows the provisioning of a [bootloader](#) over the network.

Steps needed

The process for network booting the live server installer is similar for both modes and goes like this:

1. The to-be-installed machine boots, and is directed to network boot.
2. The [DHCP](#)/BOOTP server tells the machine its network configuration and where to get the bootloader.
3. The machine’s firmware downloads the bootloader over TFTP and executes it.
4. The bootloader downloads configuration, also over TFTP, telling it where to download the kernel, RAM Disk and kernel command line to use.
5. The RAM Disk looks at the kernel command line to learn how to configure the network and where to download the server ISO from.
6. The RAM Disk downloads the ISO and mounts it as a loop device.
7. From this point on the install follows the same path as if the ISO were on a local block device.

The difference between UEFI and legacy modes is that in UEFI mode the bootloader is an [EFI](#) executable, signed so that it is accepted by Secure Boot, and in legacy mode it is [PXELINUX](#)⁶³. Most DHCP/BOOTP servers can be configured to serve the right bootloader to a particular machine.

Configure DHCP/BOOTP and TFTP

There are several implementations of the DHCP/BOOTP and TFTP protocols available. This document describes how to configure [dnsmasq](#) to perform both of these roles.

- Install dnsmasq with:

```
$ sudo apt install dnsmasq
```

- Put something like this in `/etc/dnsmasq.d/pxe.conf`:

⁶³ <https://wiki.syslinux.org/wiki/index.php?title=PXELINUX>

```
interface=<your interface>,lo
bind.interfaces
dhcp-range=<your interface>,192.168.0.100,192.168.0.200
dhcp-boot=pxelinux.0
dhcp-match=set:efi-x86_64,option:client-arch,7
dhcp-boot=tag:efi-x86_64,bootx64.efi
enable-tftp
tftp-root=/srv/tftp
```

Note:

This assumes several things about your network; read `man dnsmasq` or the default `/etc/dnsmasq.conf` for many more options.

- Restart dnsmasq with:

```
$ sudo systemctl restart dnsmasq.service
```

- Install TFTP

```
$ sudo apt install tftpd-hpa
```

If the installation is successful, check that the corresponding TFTP service is active using this command:

```
$ systemctl status tftpd-hpa.service
```

It is expected to show **active (running)** in the output messages. We will also assume your TFTP root path is `/srv/tftp` for the remainder of this guide.

Mode-independent set up

- Download the latest live server ISO for the release you want to install:

```
$ wget http://cdimage.ubuntu.com/ubuntu-server/noble/daily-live/current/noble-live-server-amd64.iso
```

- Mount it:

```
$ sudo mount noble-live-server-amd64.iso /mnt
```

- Copy the kernel and initrd from it to the TFTP directory served by dnsmasq:

```
$ sudo cp /mnt/casper/{vmlinuz,initrd} /srv/tftp/
```

Set up the files for UEFI booting

- Copy the signed shim binary into place:

```
$ apt download shim-signed > dpkg-deb --fsys-tarfile shim-signed*deb |
tar x ./usr/lib/shim/shimx64.efi.signed.latest -0 | sudo tee
/srv/tftp/bootx64.efi >/dev/null
```

- Copy the signed GRUB binary into place:

```
$ apt download grub-efi-amd64-signed > dpkg-deb --fsys-tarfile
grub-efi-amd64-signed*deb | tar x
./usr/lib/grub/x86_64-efi-signed/grubnetx64.efi.signed -0 | sudo tee
/srv/tftp/grubx64.efi >/dev/null
```

- GRUB also needs a font to be available over TFTP:

```
$ apt download grub-common > dpkg-deb --fsys-tarfile grub-common*deb |
tar x ./usr/share/grub/unicode.pf2 -0 | sudo tee /srv/tftp/unicode.pf2
>/dev/null
```

- Create /srv/tftp/grub/grub.cfg that contains:

```
sudo mkdir -p /srv/tftp/grub/

sudo tee /srv/tftp/grub/grub.cfg >/dev/null <<EOF
set default="0"
set timeout=-1

if loadfont unicode ; then
    set gfxmode=auto
    set locale_dir=$prefix/locale
    set lang=en_US
fi
terminal_output gfxterm

set menu_color_normal=white/black
set menu_color_highlight=black/light-gray
if background_color 44,0,30; then
    clear
fi

function gfxmode {
    set gfxpayload="${1}"
    if [ "${1}" = "keep" ]; then
        set vt_handoff=vt.handoff=7
    else
        set vt_handoff=
    fi
}

set linux_gfx_mode=keep

export linux_gfx_mode
```

(continues on next page)

(continued from previous page)

```
menuentry 'Ubuntu 24.04' {
    gfxmode $linux_gfx_mode
    linux /vmlinuz $vt_handoff quiet splash root=/dev/ram0 ramdisk_size=1500000 cloud-
config-url=/dev/null ip=dhcp url=http://cdimage.ubuntu.com/ubuntu-server/noble/daily-
live/current/noble-live-server-amd64.iso
    initrd /initrd
}
EOF
```

Set up the files for legacy boot

- Download pxelinux.0 and put it into place:

```
$ wget http://archive.ubuntu.com/ubuntu/dists/focal/main/installer-
amd64/current/legacy-images/netboot/pxelinux.0 > mkdir -p /srv/tftp > mv
pxelinux.0 /srv/tftp/
```

- Ensure the syslinux-common package is installed, then:

```
$ cp /usr/lib/syslinux/modules/bios/ldlinux.c32 /srv/tftp/
```

- Create /srv/tftp/pxelinux.cfg/default containing:

```
DEFAULT install
LABEL install
    KERNEL vmlinuz
    INITRD initrd
    APPEND root=/dev/ram0 ramdisk_size=1500000 cloud-config-url=/dev/null ip=dhcp
url=http://cdimage.ubuntu.com/ubuntu-server/noble/daily-live/current/noble-live-
server-amd64.iso
```

Note:

Setting `cloud-config-url=/dev/null` on the kernel command line prevents cloud-init from downloading the ISO twice.

As you can see, this downloads the ISO from Ubuntu's servers. You may want to host it somewhere on your infrastructure and change the URL to match.

This configuration is very simple. PXELINUX has many, many options, and you can [consult its documentation](https://wiki.syslinux.org/wiki/index.php?title=PXELINUX)⁶⁴ for more.

⁶⁴ <https://wiki.syslinux.org/wiki/index.php?title=PXELINUX>

Netboot the server installer via UEFI PXE on ARM (aarch64, arm64) and x86_64 (amd64)

This document describes the steps to install a system via netbooting and the live server installer (Subiquity) in UEFI mode with Ubuntu 20.04 (or later).

This process applies to both arm64 and [amd64](#) architectures and is based on [this Ubuntu Discourse post for legacy mode netbooting](#)⁶⁵ (UEFI's predecessor). Focal (20.04, 20.04.5) and Groovy (20.10) have been tested with this method.

Configure TFTP

This article assumes that you have set up your Trivial File Transfer Protocol (TFTP) (and/or *DHCP*/bootp if necessary, depending on your LAN configuration) by following [the netbooting method for legacy mode](#)⁶⁶. Alternatively, you can set up your own TFTP server as follows if your *DNS* and DHCP are already configured:

```
$ sudo apt install tftpd-hpa
```

After installation, check that the TFTP service is active:

```
$ systemctl status tftpd-hpa.service
```

The output should show **active (running)**. This article assumes your TFTP root path is `/srv/tftp`.

How to serve files

You can **skip this entire section** by using [this non-official tool](#)⁶⁷. The tool will set up your TFTP server to serve the necessary files for netbooting.

Necessary files

The following files are needed for this process:

- Ubuntu live server image:
 - For arm64 architectures, the image name has the suffix `-arm64`. For example, `ubuntu-20.04.5-live-server-arm64.iso`.
 - For amd64 architectures, the image name has the suffix `-amd64`. For example, `ubuntu-20.04.5-live-server-amd64.iso`.
- GRUB *EFI* binary (and the corresponding `grub.cfg` text file):
 - For arm64 architectures, this is called `grubnetaa64.efi.signed`.
 - For amd64 architectures, this is called `grubnetx64.efi.signed`.

⁶⁵ <https://discourse.ubuntu.com/t/netbooting-the-live-server-installer/14510>

⁶⁶ <https://discourse.ubuntu.com/t/netbooting-the-live-server-installer/14510>

⁶⁷ <https://github.com/dannf/ubuntu-server-netboot>

- `initrd` extracted from your target Ubuntu live server image (use `hwe-initrd` instead if you want to boot with the HWE kernel).
- `vmlinux` extracted from your target Ubuntu live server image (use `hwe-vmlinux` instead if you want to boot with the HWE kernel).

Examples

In the following sections, we will use an arm64 image as an example. This means the following files are used:

- Ubuntu 20.04.5 live server image [ubuntu-20.04.5-live-server-arm64.iso](#)⁶⁸
- GRUB EFI binary [grubnetaa64.efi.signed](#)⁶⁹
- `initrd` extracted from `ubuntu-20.04.5-live-server-arm64.iso`
- `vmlinux` extracted from `ubuntu-20.04.5-live-server-arm64.iso`

Replace the corresponding files if you want to work on an amd64 image. For example, your files may be:

- Ubuntu 20.04.5 live server image [ubuntu-20.04.5-live-server-amd64.iso](#)⁷⁰
- GRUB EFI binary [grubnetx64.efi.signed](#)⁷¹
- `initrd` extracted from `ubuntu-20.04.5-live-server-amd64.iso`
- `vmlinux` extracted from `ubuntu-20.04.5-live-server-amd64.iso`

Download and serve the GRUB EFI binary

The GRUB binary redirects the download path to the target files via `grub.cfg`. For more information about the Preboot Execution Environment (PXE) process and why this binary is required, see [the netbooting instructions for legacy mode](#)⁷².

```
$ sudo wget https://ports.ubuntu.com/ubuntu-ports/dists/focal/main/uefi/
grub2-arm64/current/grubnetaa64.efi.signed -O
/srv/tftp/grubnetaa64.efi.signed
```

Note:

Change **the archive distribution's name** from `Focal` to your target distribution name if needed.

⁶⁸ <https://cdimage.ubuntu.com/ubuntu/releases/20.04.5/release/ubuntu-20.04.5-live-server-arm64.iso>

⁶⁹ <https://ports.ubuntu.com/ubuntu-ports/dists/focal/main/uefi/grub2-arm64/current/grubnetaa64.efi.signed>

⁷⁰ <https://old-releases.ubuntu.com/releases/20.04.5/ubuntu-20.04.5-live-server-amd64.iso>

⁷¹ <http://archive.ubuntu.com/ubuntu/dists/focal/main/uefi/grub2-amd64/current/grubnetx64.efi.signed>

⁷² <https://discourse.ubuntu.com/t/netbooting-the-live-server-installer/14510>

Download and serve more files

Download an Ubuntu ARM server ISO to fetch the installer, such as the [20.04.5 live server arm64 ISO⁷³](#). The “live” prefix is required because the files needed are only available in the live version.

Mount the ISO and copy the target files we need over to the TFTP folder:

```
$ sudo mount ./ubuntu-20.04.5-live-server-arm64.iso /mnt > sudo mkdir /srv/tftp/grub /srv/tftp/casper > sudo cp /mnt/boot/grub/grub.cfg /srv/tftp/grub/ > sudo cp /mnt/casper/initrd /srv/tftp/casper/ > sudo cp /mnt/casper/vmlinuz /srv/tftp/casper/
```

So, the TFTP root folder should look like this now:

```
$ find /srv/tftp/

/srv/tftp/
/srv/tftp/grub
/srv/tftp/grub/grub.cfg
/srv/tftp/grubnetaa64.efi.signed
/srv/tftp/casper
/srv/tftp/casper/initrd
/srv/tftp/casper/vmlinuz
```

Customize the GRUB menu to install the target image by fetching it directly over the internet:

```
$ sudo chmod +w /srv/tftp/grub/grub.cfg
```

```
$ sudo vi /srv/tftp/grub/grub.cfg
```

Add a new boot entry at the top-level menu (not in any submenu):

```
menuentry "Install Ubuntu Server (Focal 20.04.5) (Pull the iso from web)" {
    set gfxpayload=keep
    linux /casper/vmlinuz url=http://cdimage.ubuntu.com/ubuntu/releases/20.04.5/
release/ubuntu-20.04.5-live-server-arm64.iso only-ubiquity ip=dhcp ---
    initrd /casper/initrd
}
```

In this configuration:

- `ip=dhcp` configures DHCP management for the lab environment
- `url` points to your target image download URL

Change these values according to your scenario.

When successful, you will enter the GRUB menu of the ephemeral live prompt. Select the entry you added to `grub.cfg` (Install Ubuntu Server (Focal 20.04.5) (Pull the ISO from web) in this example). After the ISO downloads, the Subiquity welcome message will appear.

⁷³ <http://cdimage.ubuntu.com/ubuntu/releases/20.04.5/release/ubuntu-20.04.5-live-server-arm64.iso>

Appendix

Check the serving file names

Verify that the target file name for linux and initrd is correct. The default initrd binary file name for 20.04.5 is `initrd`, while for 20.10 it is `initrd.lz`. Serving incorrect file names is a common troubleshooting issue.

Booting screenshots

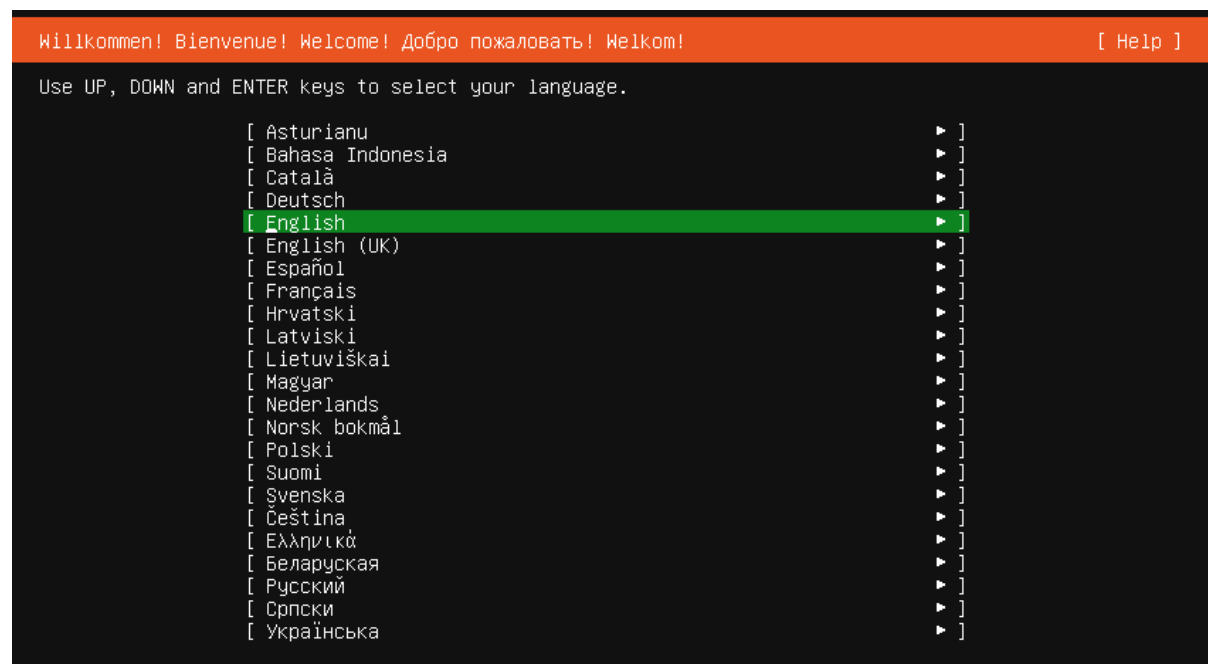
With a correct setup, `grub.cfg` redirects the process to an ephemeral environment where your target image is downloaded and assigned in the GRUB entry. When you access the console or monitor of your target machine, you will see a screen like this:

```

Listening on LPF/ens3/02:56:e6:08:2d:e3
Sending on   LPF/ens3/02:56:e6:08:2d:e3
Sending on   Socket/fallback
DHCPDISCOVER on ens3 to 255.255.255.255 port 67 interval 3 (xid=0x83e74430)
DHCPOFFER of 10.228.68.73 from 10.228.68.5
DHCPREQUEST for 10.228.68.73 on ens3 to 255.255.255.255 port 67 (xid=0x3044e783)
DHCPACK of 10.228.68.73 from 10.228.68.5 (xid=0x83e74430)
bound to 10.228.68.73 -- renewal in 286 seconds.
Connecting to 10.228.68.91 (10.228.68.91:80)
ubuntu-20.04.1-live- 0% | 462k 0:34:48 ETA
ubuntu-20.04.1-live- 2% | 22.2M 0:01:20 ETA
ubuntu-20.04.1-live- 4% * 39.7M 0:01:05 ETA
ubuntu-20.04.1-live- 6% ** 59.6M 0:00:57 ETA
ubuntu-20.04.1-live- 8% ** 77.7M 0:00:53 ETA
ubuntu-20.04.1-live- 10% *** 97.7M 0:00:50 ETA
ubuntu-20.04.1-live- 12% *** 114M 0:00:49 ETA
ubuntu-20.04.1-live- 14% **** 132M 0:00:47 ETA
ubuntu-20.04.1-live- 16% ***** 149M 0:00:45 ETA
ubuntu-20.04.1-live- 18% ***** 166M 0:00:45 ETA
ubuntu-20.04.1-live- 20% ***** 183M 0:00:43 ETA
ubuntu-20.04.1-live- 21% ***** 200M 0:00:42 ETA
ubuntu-20.04.1-live- 23% ***** 214M 0:00:42 ETA

```

Wait for the download to complete. If you see this Subiquity welcome page, the installer successfully launched via your UEFI PXE setup. Congratulations!



Choose between the arm64 and arm64+largemem installer options

From 22.04.4 onward, Ubuntu will provide both 4k and 64k page size kernel ISOs for ARM servers.

The default **arm64** ISO will still use a 4k page size kernel, while the new 64k page size kernel ISO is named **arm64+largemem**.

- [arm64 4k ISO download](#)⁷⁴
- [arm64+largemem ISO download](#)⁷⁵

The default arm64 (4k) option

The 4k page size is the default in our arm64 ISO. It is suitable for workloads with many small processes, or environments with tight memory constraints. Typical use cases include (but are not limited to):

- Web servers
- Embedded devices
- General purpose/build systems

The arm64+largemem (64k) option

Our new **arm64+largemem** ISO includes a kernel with 64k page size. A larger page size can increase throughput, but comes at the cost of increased memory use, making this option more suitable for servers with plenty of memory. Typical use cases for this ISO include:

- Machine learning
- Databases with many large entries
- *High performance computing*
- etc.

Note:

It is possible to switch between these kernel options after installation by installing the other kernel alternative, rebooting, and selecting the new kernel from the GRUB menu.

⁷⁴ <https://cdimage.ubuntu.com/releases/22.04/release/ubuntu-22.04.5-live-server-arm64.iso>

⁷⁵ <https://cdimage.ubuntu.com/releases/22.04/release/ubuntu-22.04.5-live-server-arm64+largemem.iso>

Switching kernels post-installation

To switch between the two kernels after the initial installation you can run the following commands, replacing `<desired-kernel>` with `linux-generic-64k` when swapping to 64k, or `linux-generic` when swapping to the default 4k kernel:

```
$ sudo apt update
```

```
$ sudo apt install <desired-kernel>
```

```
$ sudo reboot
```

Upon reboot you will be greeted with the GRUB menu (you may need to hold down the Shift key during the reboot for it to appear). Select "Advanced Options for Ubuntu", then select your desired kernel to boot into Ubuntu.

To permanently change the default to your `<desired-flavour>`, replace `<desired-flavour>` with `generic` or `generic-64k` and then run the following command:

```
$ echo "GRUB_FLAVOUR_ORDER=<desired-flavour>" | sudo tee  
/etc/default/grub.d/local-order.cfg
```

To apply your change run:

```
$ sudo update-grub
```

Future boots will automatically use your new desired kernel flavour. You can verify this by rebooting using:

```
$ sudo reboot
```

And then running the following command to display the active kernel:

```
$ uname -r
```

Netboot the live server installer on IBM Power (ppc64el) with Petitboot

Open a terminal window on your workstation and make sure the `ipmitool` package is installed.

Verify if you can reach the *BMC* of the IBM Power system via `ipmitool` with a simple `ipmitool` call like:

```
$ ipmitool -I lanplus -H Power9Box -U <user> -P <password> power status  
  
Chassis Power is off
```

or:

```
$ ipmitool -I lanplus -H Power9Box -U <user> -P <password> fru print 47
```

```
Product Name       : OpenPOWER Firmware
Product Version    : open-power-SUPERMICRO-P9DSU-V2.12-20190404-prod
Product Extra      : op-build-1b9269e
Product Extra      : buildroot-2018.11.3-12-g222837a
Product Extra      : skiboot-v6.0.19
Product Extra      : hostboot-c00d44a-pb1307d7
Product Extra      : occ-8fa3854
Product Extra      : linux-4.19.30-openpower1-p22d1df8
Product Extra      : petitboot-v1.7.5-p8f5fc86
```

or:

```
$ ipmitool -I lanplus -H Power9Box -U <user> -P <password> sol info
```

```
Set in progress      : set-complete
Enabled              : true
Force Encryption     : false
Force Authentication : false
Privilege Level       : OPERATOR
Character Accumulate Level (ms) : 0
Character Send Threshold : 0
Retry Count          : 0
Retry Interval (ms)   : 0
Volatile Bit Rate (kbps) : 115.2
Non-Volatile Bit Rate (kbps) : 115.2
Payload Channel       : 1 (0x01)
Payload Port          : 623
```

Activate serial-over-LAN

Open a second terminal and activate serial-over-LAN (SOL), so that you have two terminal windows open:

- 1) to control the BMC via IPMI
- 2) for the serial-over-LAN console

Activate serial-over-LAN:

```
$ ipmitool -I lanplus -H Power9Box -U <user> -P <password> sol activate
```

```
...
```

Power the system on in the 'control terminal' and watch the SOL console:

```
$ ipmitool -I lanplus -H Power9Box -U <user> -P <password> power on
```

```
...
```

It takes some time to see the first lines in the SOL console:


```
[SOL Session operational. Use ~? for help]
--== Welcome to Hostboot ==--

2.77131|secure|SecureROM valid - enabling functionality
3.15860|secure|Booting in secure mode.
5.59684|Booting from SBE side 0 on master proc=00050000
5.60502|ISTEP 6. 5 - host_init_fsi
5.87228|ISTEP 6. 6 - host_set_ipl_parms
6.11032|ISTEP 6. 7 - host_discover_targets
6.67868|HWAS|PRESENT> DIMM[03]=A0A0000000000000
6.67870|HWAS|PRESENT> Proc[05]=8800000000000000
6.67871|HWAS|PRESENT> Core[07]=3FFF0C33FFC30000
6.98988|ISTEP 6. 8 - host_update_master_tpm
7.22711|SECURE|Security Access Bit> 0xC000000000000000
7.22711|SECURE|Secure Mode Disable (via Jumper)> 0x0000000000000000
7.22731|ISTEP 6. 9 - host_gard
7.43353|HWAS|FUNCTIONAL> DIMM[03]=A0A0000000000000
7.43354|HWAS|FUNCTIONAL> Proc[05]=8800000000000000
7.43356|HWAS|FUNCTIONAL> Core[07]=3FFF0C33FFC30000
7.44509|ISTEP 6.10 - host_revert_sbe_mcs_setup
...
```

After a moment the system reaches the Petitboot screen:

```
Petitboot (v1.7.5-p8f5fc86) 9006-12P 1302NXA

[Network: enP2p1s0f0 / 0c:c4:7a:87:04:d8]
Execute
netboot enP2p1s0f0 (pxelinux.0)
[CD/DVD: sr0 / 2019-10-17-13-35-12-00]
Install Ubuntu Server
[Disk: sda2 / 295f571b-b731-4ebb-b752-60aadc80fc1b]
Ubuntu, with Linux 5.4.0-14-generic (recovery mode)
Ubuntu, with Linux 5.4.0-14-generic
Ubuntu

System information
System configuration
System status log
Language
Rescan devices
Retrieve config from URL
Plugins (0)

*Exit to shell

Enter=accept, e=edit, n=new, x=exit, l=language, g=log, h=help
```

Select “*Exit to shell”.

Note:

Make sure you really watch the SOL, since the Petitboot screen (above) has a time out (usually 10 or 30 seconds) and afterwards it automatically proceeds and tries to boot from the configured devices (usually disk). This can be prevented by just navigating in Petitboot. The petitboot shell is small Linux based OS:

```
...
Exiting petitboot. Type 'exit' to return.
You may run 'pb-sos' to gather diagnostic data
```

Note:

In case one needs to gather system details and diagnostic data for IBM support, this can be done here by running 'pb-sos' (see msg).

Download the ISO

Now download the 'live-server' ISO image (notice that 'focal-live-server-ppc64el.iso' uses the liver server installer, Subiquity, while 'focal-server-s390x.iso' uses d-i).

Again, for certain web locations a proxy needs to be used:

```
/ # export http_proxy=http://squid.proxy:3128 # in case a proxy is required
/ #
/ # wget http://cdimage.ubuntu.com/ubuntu-server/daily-live/current/focal-live-server-ppc64el.iso
Connecting to <proxy-ip>:3128 (<proxy-ip>:3128)
focal-live-server-pp 100% |.....| 922M 0:00:00 ETA
```

Next we need to loop-back mount the ISO:

```
/ # mkdir iso
/ # mount -o loop focal-live-server-ppc64el.iso iso
```

Or, in case auto-detect of type iso9660 is not supported or not working, you should explicitly specify the 'iso9660' type:

```
/ # mount -t iso9660 -o loop focal-live-server-ppc64el.iso iso
```

Now load kernel and initrd from the loop-back mount, specify any needed kernel parameters and get it going:

```
/ # kexec -l ./iso/casper/vmlinuz --initrd=./iso/casper/initrd.gz --append="ip=dhcp
url=http://cdimage.ubuntu.com/ubuntu-server/daily-live/current/focal-live-server-ppc64el.
iso http_proxy=http://squid.proxy:3128 --- quiet"
/ # kexec -e
The system is going down NOW!
Sent SIGTERM to all processes
```

(continues on next page)

(continued from previous page)

```
Sent SIGKILL to all processes
...
```

Note:

In order to boot with and install the HWE kernel (if available), substitute `vmlinux` with `vmlinux-hwe` in the first `kexec` line.

The system now performs the initial boot of the installer:

```
[ 1200.687004] kexec_core: Starting new kernel
[ 1277.493883374,5] OPAL: Switch to big-endian OS
[ 1280.465061219,5] OPAL: Switch to little-endian OS
ln: /tmp/mountroot-fail-hooks.d//scripts/init-premount/lvm2: No such file or directory
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Listening on LPF/enP2p1s0f3/0c:c4:7a:87:04:db
Sending on LPF/enP2p1s0f3/0c:c4:7a:87:04:db
Listening on LPF/enP2p1s0f2/0c:c4:7a:87:04:da
Sending on LPF/enP2p1s0f2/0c:c4:7a:87:04:da
Listening on LPF/enP2p1s0f1/0c:c4:7a:87:04:d9
Sending on LPF/enP2p1s0f1/0c:c4:7a:87:04:d9
Listening on LPF/enP2p1s0f0/0c:c4:7a:87:04:d8
Sending on LPF/enP2p1s0f0/0c:c4:7a:87:04:d8
Sending on Socket/fallback
DHCPDISCOVER on enP2p1s0f3 to 255.255.255.255 port 67 interval 3
(xid=0x8d5704c)
DHCPDISCOVER on enP2p1s0f2 to 255.255.255.255 port 67 interval 3
(xid=0x94b25b28)
DHCPDISCOVER on enP2p1s0f1 to 255.255.255.255 port 67 interval 3
(xid=0x4edd0558)
DHCPDISCOVER on enP2p1s0f0 to 255.255.255.255 port 67 interval 3
(xid=0x61c90d28)
DHCPOFFER of 10.245.71.102 from 10.245.71.3
DHCPREQUEST for 10.245.71.102 on enP2p1s0f0 to 255.255.255.255 port 67
(xid=0x280dc961)
DHCPACK of 10.245.71.102 from 10.245.71.3 (xid=0x61c90d28)
bound to 10.245.71.102 -- renewal in 236 seconds.
Connecting to 91.189.89.11:3128 (91.189.89.11:3128)
focal-live-server-pp 1% | 14.0M 0:01:04 ETA
focal-live-server-pp 4% | * 45.1M 0:00:38 ETA
focal-live-server-pp 8% | ** 76.7M 0:00:33 ETA
focal-live-server-pp 11% | *** 105M 0:00:31 ETA
focal-live-server-pp 14% | **** 133M 0:00:29 ETA
focal-live-server-pp 17% | ***** 163M 0:00:27 ETA
focal-live-server-pp 20% | 190M 0:00:26 ETA
focal-live-server-pp 24% | 222M 0:00:25 ETA
focal-live-server-pp 27% | 253M 0:00:23 ETA
focal-live-server-pp 30% | 283M 0:00:22 ETA
focal-live-server-pp 34% | 315M 0:00:21 ETA
focal-live-server-pp 37% | 343M 0:00:20 ETA
focal-live-server-pp 39% | 367M 0:00:19 ETA
focal-live-server-pp 42% | 392M 0:00:18 ETA
```

(continues on next page)

(continued from previous page)

```
focal-live-server-pp 45% |*****| 420M 0:00:17 ETA
focal-live-server-pp 48% |*****| 451M 0:00:16 ETA
focal-live-server-pp 52% |*****| 482M 0:00:15 ETA
focal-live-server-pp 55% |*****| 514M 0:00:14 ETA
focal-live-server-pp 59% |*****| 546M 0:00:13 ETA
focal-live-server-pp 62% |*****| 578M 0:00:11 ETA
focal-live-server-pp 65% |*****| 607M 0:00:10 ETA
focal-live-server-pp 69% |*****| 637M 0:00:09 ETA
focal-live-server-pp 72% |*****| 669M 0:00:08 ETA
focal-live-server-pp 75% |*****| 700M 0:00:07 ETA
focal-live-server-pp 79% |*****| 729M 0:00:06 ETA
focal-live-server-pp 82% |*****| 758M 0:00:05 ETA
focal-live-server-pp 85% |*****| 789M 0:00:04 ETA
focal-live-server-pp 88% |*****| 817M 0:00:03 ETA
focal-live-server-pp 91% |*****| 842M 0:00:02 ETA
focal-live-server-pp 93% |*****| 867M 0:00:01 ETA
focal-live-server-pp 97% |*****| 897M 0:00:00 ETA
focal-live-server-pp 100% |*****| 922M 0:00:00 ETA
mount: mounting /cow on /root/cow failed: No such file or directory
Connecting to plymouth: Connection refused
passwd: password expiry information changed.
[ 47.202736] /dev/loop3: Can't open blockdev
[ 52.672550] cloud-init[3759]: Cloud-init v. 20.1-10-g71af48df-0ubuntu1 running
'init-local' at Wed, 18 Mar 2020 15:18:07 +0000. Up 51.87 seconds.
...
```

Once it has completed, you will reach the initial Subiquity installer screen:

```

Willkommen! Bienvenue! Welcome! Добро пожаловать! Welkom
Use UP, DOWN and ENTER keys to select your language.

[ English ]
[ Asturianu ]
[ Català ]
[ Hrvatski ]
[ Nederlands ]
[ Suomi ]
[ Français ]
[ Deutsch ]
[ Ελληνικά ]
[ Magyar ]
[ Latviešu ]
[ Norsk bokmål ]
[ Polski ]
[ Русский ]
[ Español ]
[ Українська ]
```

From this point, you can follow the normal Subiquity installation. For more details, refer to the [Subiquity installer documentation](https://canonical-subiquity.readthedocs-hosted.com/en/latest/index.html)⁷⁶.

⁷⁶ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/index.html>

How to start a live server installation on IBM Power (ppc64el) with a virtual CD-ROM and Petitboot

Note:

Not all IBM Power machines come with the capability of installing via a virtual CD-ROM! However, it is also possible to *boot the installer over the network* (page 86).

A separate system (ideally in the same network, because of `ipmitool`) is needed to host the ppc64el ISO image file, which is later used as the virtual CD-ROM.

Install `ipmitool` and Samba

Log in to this separate host and make sure that the `ipmitool` package is installed:

```
$ sudo apt install ipmitool
```

as well as Samba:

```
$ sudo apt install samba
```

Configure Samba

Next, set up and configure Samba:

```
$ sudo touch /etc/samba/smb.conf && sudo tee -a /etc/samba/smb.conf <<EOF
[winshare]
path=/var/winshare
browseable = yes
read only = no
guest ok = yes
EOF
```

And do a quick verification that the required lines are present:

```
$ tail -n 5 /etc/samba/smb.conf
[winshare]
path=/var/winshare
browseable = yes
read only = no
guest ok = yes
```

Optional step

For downloading the image you may have to use a proxy server:

```
$ sudo touch ~/.wgetrc && sudo tee -a ~/.wgetrc <<EOF > use_proxy=yes >  
http_proxy=squid.proxy:3128 > https_proxy=squid.proxy:3128 > EOF
```

Download the ISO image

The ISO image needs to be downloaded now:

```
$ wget http://cdimage.ubuntu.com/ubuntu/releases/focal/release/ubuntu-20.04-  
live-server-ppc64el.iso --directory-prefix=/var/winshare
```

The proxy can also be passed over as a wget argument, like this:

```
$ wget -e use_proxy=yes -e http_proxy=squid.proxy:3128 http://cdimage.  
ubuntu.com/ubuntu/releases/focal/release/ubuntu-20.04-live-server-ppc64el.iso  
--directory-prefix=/var/winshare
```

Change the file mode of the ISO image file:

```
$ sudo chmod -R 755 /var/winshare/
```

```
$ ls -l /var/winshare/  
-rwxr-xr-x 1 ubuntu ubuntu 972500992 Mar 23 08:02 focal-live-server-ppc64el.iso
```

Restart and check Samba

Next we need to restart and check the Samba service:

```
$ sudo service smbd restart
```

```
$ sudo service smbd status
```

```
❏ smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor
   preset: ena
   Active: active (running) since Tue 2020-02-04 15:17:12 UTC; 4s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Main PID: 6198 (smbd)
    Status: "smbd: ready to serve connections..."
     Tasks: 4 (limit: 19660)
    CGroup: /system.slice/smbd.service
            └─6198 /usr/sbin/smbd --foreground --no-process-group
            └─6214 /usr/sbin/smbd --foreground --no-process-group
            └─6215 /usr/sbin/smbd --foreground --no-process-group
            └─6220 /usr/sbin/smbd --foreground --no-process-group
Feb 04 15:17:12 host systemd[1]: Starting Samba SMB Daemon...
Feb 04 15:17:12 host systemd[1]: Started Samba SMB Daemon.
```

Test Samba share:

```
ubuntu@host:~$ smbclient -L localhost

WARNING: The "syslog" option is deprecated
Enter WORKGROUP\ubuntu's password:
  Sharename      Type            Comment
  -----
  print$         Disk            Printer Drivers
  winshare       Disk
  IPC$           IPC             IPC Service (host server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.
  Server                Comment
  -----
  Workgroup              Master
  -----
  WORKGROUP              host
```

Get the IP address of the Samba host:

```
$ ip -4 -brief address show

lo                UNKNOWN        127.0.0.1/8
ibmveth2          UNKNOWN        10.245.246.42/24
```

Optional step

Additional testing to make certain the Samba share is accessible from remote:

```
user@workstation:~$ mkdir -p /tmp/test
```

```
user@workstation:~$ sudo mount -t cifs -o
```

```
username=guest,password=guest //10.245.246.42/winshare /tmp/test/
```

```
user@workstation:~$ ls -la /tmp/test/
```

```
total 1014784
drwxr-xr-x  2 root root      0 May  4 15:46 .
drwxrwxrwt 18 root root    420 May  4 19:25 ..
-rwxr-xr-x  1 root root 1038249984 May  3 19:37 ubuntu-20.04-live-server-ppc64el.iso
```

Now use a browser and navigate to the [BMC](#) of the Power system, which should be installed (let's assume the BMC's IP address is 10.245.246.247):

```
firefox http://10.245.246.247/
```

Log into the BMC, then find and select:

Virtual Media → CD-ROM

Enter the IP address of the Samba share:

10.245.246.42

and the path to the Samba share:

\winshare\focal-live-server-ppc64el.iso

Click "Save and Mount". Make sure that the virtual CD-ROM is really properly mounted!

CD-ROM Image:

This option allows you to share a CD-ROM image over a Windows Share with a maximum size of 4.7GB. This image will be emulated to the host as USB device.

Device 1 There is an iso file mounted.

Device 2 No disk emulation set.

Device 3 No disk emulation set.

<Refresh Status>

Share host: 10.245.246.42

Path to image: \winshare\focal-live-server-ppc64el.iso

(continues on next page)

(continued from previous page)

```
User (optional):
Password (optional):
<Save> <Mount> <Unmount>
```

Note:

It's important that you see a status like this:

```
Device 1 There is an iso file mounted
```

Then the virtual CD-ROM is properly mounted and you will see the boot/install from CD-ROM entry in Petitboot:

```
[CD/DVD: sr0 / 2020-03-23-08-02-42-00]
Install Ubuntu Server
```

Boot into the Petitboot loader

Now use `ipmitool` to boot the system into the Petitboot loader:

```
$ ipmitool -I lanplus -H 10.245.246.247 -U ADMIN -P <password> power status
```

```
$ ipmitool -I lanplus -H 10.245.246.247 -U ADMIN -P <password> sol activate
```

```
$ ipmitool -I lanplus -H 10.245.246.247 -U ADMIN -P <password> power on

Chassis Power Control: Up/On
```

And reach the Petitboot screen:

```
Petitboot (v1.7.5-p8f5fc86)                                9006-12C B0S0026
-----
[Network: enP2p1s0f0 / ac:1f:6b:09:c0:52]
  execute
  netboot enP2p1s0f0 (pxelinux.0)

System information
System configuration
System status log
Language
Rescan devices
Retrieve config from URL
*Plugins (0)
Exit to shell
-----
Enter=accept, e=edit, n=new, x=exit, l=language, g=log, h=help
Default boot cancelled
```

Make sure that booting from CD-ROM is enabled:

Petitboot (v1.7.5-p8f5fc86)

9006-12C B0S0026

```
[Network: enP2p1s0f0 / ac:1f:6b:09:c0:52]
Execute
netboot enP2p1s0f0 (pxelinux.0)
[Disk: sda2 / ebdb022b-96b2-4f4f-ae63-69300ded13f4]
  Ubuntu, with Linux 5.4.0-12-generic (recovery mode)
  Ubuntu, with Linux 5.4.0-12-generic
  Ubuntu

System information
System configuration
System status log
Language
Rescan devices
Retrieve config from URL
*Plugins (0)
Exit to shell
```

```
Enter=accept, e=edit, n=new, x=exit, l=language, g=log, h=help
[sda3] Processing new Disk device
```

Petitboot System Configuration

```
Autoboot:      ( ) Disabled
               (*) Enabled

Boot Order:    (0) Any CD/DVD device
               (1) disk: sda2 [uuid: ebdb022b-96b2-4f4f-ae63-69300ded13f4]
               (2) net:  enP2p1s0f0 [mac: ac:1f:6b:09:c0:52]

               [   Add Device   ]
               [ Clear & Boot Any ]
               [      Clear     ]

Timeout:       30    seconds

Network:       (*) DHCP on all active interfaces
               ( ) DHCP on a specific interface
               ( ) Static IP configuration
```

```
tab=next, shift+tab=previous, x=exit, h=help
```

Petitboot System Configuration

```
Network:       (*) DHCP on all active interfaces
               ( ) DHCP on a specific interface
               ( ) Static IP configuration

DNS Server(s): (eg. 192.168.0.2)
               (if not provided by DHCP server)
```

(continues on next page)

(continued from previous page)

```

HTTP Proxy:
HTTPS Proxy:

Disk R/W:      ( ) Prevent all writes to disk
                (*) Allow bootloader scripts to modify disks

Boot console:  (*) /dev/hvc0 [IPMI / Serial]
                ( ) /dev/tty1 [VGA]
                Current interface: /dev/hvc0

                [ OK ] [ Help ] [ Cancel ]

```

tab=next, shift+tab=previous, x=exit, h=help

Now select the 'Install Ubuntu Server' entry below the CD/DVD entry:

```
[CD/DVD: sr0 / 2020-03-23-08-02-42-00]
* Install Ubuntu Server
```

And let Petitboot boot from the (virtual) CD-ROM image:

```
Sent SIGKILL to all processes
[ 119.355371] kexec_core: Starting new kernel
[ 194.483947394,5] OPAL: Switch to big-endian OS
[ 197.454615202,5] OPAL: Switch to little-endian OS
```

The initial Subiquity installer screen will show up in the console:

[illegible]

From this point, you can follow the normal Subiquity installation. For more details, refer to the [Subiquity installer documentation](#)⁷⁷.

⁷⁷ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/index.html>

Interactive live server installation on IBM z/VM (s390x)

Doing an interactive (manual) live installation as described here - meaning without specifying a parmfile - has been supported in Ubuntu Server since LTS 20.04.5 ("Focal").

The following guide assumes that a z/VM guest has been defined, and that it is able to either reach the public cdimage.ubuntu.com server or an internal FTP or HTTP server that hosts an Ubuntu Server 20.04 installer image, like this [20.04 \(a.k.a. Focal\) daily live image](http://cdimage.ubuntu.com/ubuntu/releases/20.04.5/release/ubuntu-20.04.5-live-server-s390x.iso)⁷⁸

Find a place to download the installer image:

```
user@workstation:~$ wget http://cdimage.ubuntu.com/ubuntu/releases/20.04.5/
release/ubuntu-20.04.5-live-server-s390x.iso

--2020-08-08 16:01:52--
http://cdimage.ubuntu.com/ubuntu/releases/20.04.5/release/ubuntu-20.04.5-live-
server-s390x.iso
Resolving cdimage.ubuntu.com (cdimage.ubuntu.com)... 2001:67c:1560:8001::1d,
2001:67c:1360:8001::27, 2001:67c:1360:8001::28, ...
Connecting to cdimage.ubuntu.com
(cdimage.ubuntu.com)|2001:67c:1560:8001::1d|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 705628160 (673M) [application/x-iso9660-image]
Saving to: 'ubuntu-20.04.5-live-server-s390x.iso'

ubuntu-20.04.5-live 100%[=====] 672.94M  37.1MB/s   in
17s

2020-08-08 16:02:10 (38.8 MB/s) - 'ubuntu-20.04.5-live-server-s390x.iso' saved
[705628160/705628160]
```

Now loop-back mount the ISO to extract four files that are needed for a z/VM guest installation:

```
user@workstation:~$ mkdir iso
```

```
user@workstation:~$ sudo mount -o loop ubuntu-20.04.5-live-server-s390x.iso
iso
```

```
user@workstation:~$ ls -1
./iso/boot/{ubuntu.exec,parmfile.*,kernel.u*,initrd.u*}

./iso/boot/initrd.ubuntu
./iso/boot/kernel.ubuntu
./iso/boot/parmfile.ubuntu
./iso/boot/ubuntu.exec
```

Now transfer these four files to your z/VM guest (for example to its 'A' file mode), using either the 3270 terminal emulator or FTP.

Then log on to the z/VM guest that you want to use for the installation. In this example it will be guest '10.222.111.24'.

⁷⁸ <http://cdimage.ubuntu.com/ubuntu/releases/20.04.5/release/ubuntu-20.04.5-live-server-s390x.iso>

Execute the ubuntu REXX script to kick-off the installation:

```
ubuntu
00: 0000004 FILES PURGED
00: RDR FILE 0125 SENT FROM 10.222.111.24 PUN WAS 0125 RECS 101K CPY 001 A NOHOLD NO
KEEP
00: RDR FILE 0129 SENT FROM 10.222.111.24 PUN WAS 0129 RECS 0001 CPY 001 A NOHOLD NO
KEEP
00: RDR FILE 0133 SENT FROM 10.222.111.24 PUN WAS 0133 RECS 334K CPY 001 A NOHOLD NO
KEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
01: HCPGSP2627I The virtual machine is placed in CP mode due to a SIGP initial CPU reset
from CPU 00.
02: HCPGSP2627I The virtual machine is placed in CP mode due to a SIGP initial CPU reset
from CPU 00.
03: HCPGSP2627I The virtual machine is placed in CP mode due to a SIGP initial CPU reset
from CPU 00.
- 0.390935| Initramfs unpacking failed: Decoding failed
Unable to find a medium container a live file system
```

In the usual case that no parmfile was configured, the installation system now offers to interactively configure the basic network:

```
Attempt interactive netboot from a URL?
yes no (default yes): yes
Available qeth devices:
0.0.0600 0.0.0603
zdev to activate (comma separated, optional): 0600
QETH device 0.0.0600:0.0.0601:0.0.0602 configured
Two methods available for IP configuration:
* static: for static IP configuration
* dhcp: for automatic IP configuration
static dhcp (default 'dhcp'): static
ip: 10.222.111.24
gateway (default 10.222.111.1): .
dns (default .):
vlan id (optional):
http://cdimage.ubuntu.com/ubuntu/releases/20.04.5/release/ubuntu-20.04.5-live-server-
s390x.iso (default)
url: ftp://10.11.12.2:21/ubuntu-live-server-20.04.5/ubuntu-20.04.5-live-server-s390x.iso
http_proxy (optional):
```

Ensure that the same version of the ISO image that was used to extract the installer files – kernel and initrd – is referenced at the ‘url:’ setting. It can be at a different location, for example directly referencing the public `cdimage.ubuntu.com` server: `http://cdimage.ubuntu.com/ubuntu/releases/20.04.5/release/ubuntu-20.04.5-live-server-s390x.iso`

The boot-up of the live-server installation now completes:

```
Configuring networking...
QETH device 0.0.0600:0.0.0601:0.0.0602 already configured
IP-Config: enc600 hardware address 02:28:0a:00:00:39 mtu 1500
IP-Config: enc600 guessed broadcast address 10.222.111255
IP-Config: enc600 complete:
address: 10.222.111.24 broadcast: 10.222.111255 netmask: 255.255.255.0
```

(continues on next page)

(continued from previous page)

```
gateway: 10.222.111.1      dns0      : 10.222.111.1      dns1      : 0.0.0.0

rootserver: 0.0.0.0 rootpath:
filename :
Connecting to 10.11.12.2:21 (10.11.12.2:21)
focal-live-server-s 5% !*                                ! 35.9M 0:00:17 ETA
focal-live-server-s 19% !*****                            ! 129M 0:00:08 ETA
focal-live-server-s 33% !*****                            ! 225M 0:00:05 ETA
focal-live-server-s 49% !*****                            ! 330M 0:00:04 ETA
focal-live-server-s 60% !*****                            ! 403M 0:00:03 ETA
focal-live-server-s 76% !*****                            ! 506M 0:00:01 ETA
focal-live-server-s 89% !*****                            ! 594M 0:00:00 ETA
focal-live-server-s 100% !*****                            ! 663M 0:00:00 ETA
passwd: password expiry information changed.
QETH device 0.0.0600:0.0.0601:0.0.0602 already configured
no search or nameservers found in /run/net-enc600.conf /run/net-*.conf /run/net6
-*.conf
- 594.766372| /dev/loop3: Can't open blockdev
- 595.610434| systemd-1|: multi-user.target: Job getty.target/start deleted to
break ordering cycle starting with multi-user.target/start
- 0;1;31m SKIP -0m| Ordering cycle found, skipping -0;1;39mLogin Prompts -0m
- 595.623027| systemd-1|: Failed unmounting /cdrom.
- 0;1;31mFAILED -0m| Failed unmounting -0;1;39m/cdrom -0m.

- 598.973538| cloud-init-1256|: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1 runnin
g 'init-local' at Thu, 04 Jun 2020 12:06:46 +0000. Up 598.72 seconds.
- 599.829069| cloud-init-1288|: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1 runnin
g 'init' at Thu, 04 Jun 2020 12:06:47 +0000. Up 599.64 seconds.
- 599.829182| cloud-init-1288|: ci-info: ++++++Net device info+++++
- 599.829218| cloud-init-1288|: ci-info: +-----+-----+-----+
+-----+-----+-----+
- 599.829255| cloud-init-1288|: ci-info: ! Device ! Up !           Address
! Mask ! Scope ! Hw-Address !
- 599.829292| cloud-init-1288|: ci-info: +-----+-----+-----+
+-----+-----+-----+
- 599.829333| cloud-init-1288|: ci-info: ! enc600 ! True !           10.222.111.24
! 255.255.255.0 ! global ! 02:28:0a:00:00:39 !
- 599.829376| cloud-init-1288|: ci-info: ! enc600 ! True ! fe80::28:aff:fe00:3
/64 ! . ! link ! 02:28:0a:00:00:39 !
- 599.829416| cloud-init-1288|: ci-info: ! lo ! True !           127.0.0.1
! 255.0.0.0 ! host ! . !
- 599.829606| cloud-init-1288|: ci-info: ! lo ! True !           ::1/128
! . ! host ! . !
- 599.829684| cloud-init-1288|: ci-info: +-----+-----+-----+
+-----+-----+-----+
- 599.829721| cloud-init-1288|: ci-info: ++++++Route IP
v4 info+++++
- 599.829754| cloud-init-1288|: ci-info: +-----+-----+-----+
+-----+-----+-----+
- 599.829789| cloud-init-1288|: ci-info: ! Route ! Destination ! Gateway
! Genmask ! Interface ! Flags !
- 599.829822| cloud-init-1288|: ci-info: +-----+-----+-----+
+-----+-----+-----+
- 599.829858| cloud-init-1288|: ci-info: ! 0 ! 0.0.0.0 ! 10.222.111.1
! 0.0.0.0 ! enc600 ! UG !
```

(continues on next page)

(continued from previous page)

```

- 599.829896| cloud-init-1288|: ci-info: ! 1 ! 10.222.1110 ! 0.0.0.0
! 255.255.255.0 ! enc600 ! U !
- 599.829930| cloud-init-1288|: ci-info: +-----+-----+-----+-----+
+-----+-----+
- 599.829962| cloud-init-1288|: ci-info: ++++++Route IPv6 info++++
+++++
- 599.829998| cloud-init-1288|: ci-info: +-----+-----+-----+-----+
-----+-----+
- 599.830031| cloud-init-1288|: ci-info: ! Route ! Destination ! Gateway ! Inte
rface ! Flags !
- 599.830064| cloud-init-1288|: ci-info: +-----+-----+-----+-----+
-----+-----+
- 599.830096| cloud-init-1288|: ci-info: ! 1 ! fe80::/64 ! :: ! en
c600 ! U !
- 599.830131| cloud-init-1288|: ci-info: ! 3 ! local ! :: ! en
c600 ! U !
- 599.830164| cloud-init-1288|: ci-info: ! 4 ! ff00::/8 ! :: ! en
c600 ! U !
- 599.830212| cloud-init-1288|: ci-info: +-----+-----+-----+-----+
-----+-----+
- 601.077953| cloud-init-1288|: Generating public/private rsa key pair.
- 601.078101| cloud-init-1288|: Your identification has been saved in /etc/ssh/
ssh_host_rsa_key
- 601.078136| cloud-init-1288|: Your public key has been saved in /etc/ssh/ssh
host_rsa_key.pub
- 601.078170| cloud-init-1288|: The key fingerprint is:
- 601.078203| cloud-init-1288|: SHA256:kHtkABZwk8AE80fy0KPzTRcYpht4iXdZmJ37Cgi3
fJ0 root@ubuntu-server
- 601.078236| cloud-init-1288|: The key's randomart image is:
- 601.078274| cloud-init-1288|: +----RSA 3072|----+
- 601.078307| cloud-init-1288|: !o+*B++*.. !
- 601.078340| cloud-init-1288|: ! o.X+=+=+ !
- 601.078373| cloud-init-1288|: ! +.0.= oo !
- 601.078406| cloud-init-1288|: ! ++.+.=o !
- 601.078439| cloud-init-1288|: ! *.=.oSo !
- 601.078471| cloud-init-1288|: ! = +.E . !
- 601.078503| cloud-init-1288|: ! . . . !
- 601.078537| cloud-init-1288|: ! . !
- 601.078570| cloud-init-1288|: ! !
- 601.078602| cloud-init-1288|: +-----SHA256|-----+
- 601.078635| cloud-init-1288|: Generating public/private dsa key pair.
- 601.078671| cloud-init-1288|: Your identification has been saved in /etc/ssh/
ssh_host_dsa_key
- 601.078704| cloud-init-1288|: Your public key has been saved in /etc/ssh/ssh_
host_dsa_key.pub
- 601.078736| cloud-init-1288|: The key fingerprint is:
- 601.078767| cloud-init-1288|: SHA256:ZBNykvVYZVhKJeL+PWKpsdUcn21yiceX/DboXQd
Pq0 root@ubuntu-server
- 601.078800| cloud-init-1288|: The key's randomart image is:
- 601.078835| cloud-init-1288|: +----DSA 1024|----+
- 601.078867| cloud-init-1288|: ! o++...+=+o !
- 601.078899| cloud-init-1288|: ! .+...+.. !
- 601.078932| cloud-init-1288|: ! +. + o o!
- 601.078964| cloud-init-1288|: ! o..o = oo.!
- 601.078996| cloud-init-1288|: ! S. =..o++!
- 601.079029| cloud-init-1288|: ! o *.*=!

```

(continues on next page)

(continued from previous page)

```

- 601.079061| cloud-init-1288|: !          o .o.B.*!
- 601.079094| cloud-init-1288|: !          = .oEo.!
- 601.079135| cloud-init-1288|: !          .+          !
- 601.079167| cloud-init-1288|: +-----SHA256|-----+
- 601.079199| cloud-init-1288|: Generating public/private ecdsa key pair.
- 601.079231| cloud-init-1288|: Your identification has been saved in /etc/ssh/
ssh_host_ecdsa_key
- 601.079263| cloud-init-1288|: Your public key has been saved in /etc/ssh/ssh_
host_ecdsa_key.pub
- 601.079295| cloud-init-1288|: The key fingerprint is:
- 601.079327| cloud-init-1288|: SHA256:Bitar9fVHUH2FnYVSJJnldprdAcM5Est0dmRWFTU
i8k root@ubuntu-server
- 601.079362| cloud-init-1288|: The key's randomart image is:
- 601.079394| cloud-init-1288|: +----ECDSA 256|---+
- 601.079426| cloud-init-1288|: !          o**0%&!
- 601.079458| cloud-init-1288|: !          o.OB+=!
- 601.079491| cloud-init-1288|: !          .      B *o+!
- 601.079525| cloud-init-1288|: !          o      . E.=o!
- 601.079557| cloud-init-1288|: !          o . S      ....+!
- 601.079589| cloud-init-1288|: !          o o .      . . .o !
- 601.079621| cloud-init-1288|: !          .      ..      . !
- 601.079653| cloud-init-1288|: !          ..      .      !
- 601.079685| cloud-init-1288|: !          ..      .      !
- 601.079717| cloud-init-1288|: +-----SHA256|-----+
- 601.079748| cloud-init-1288|: Generating public/private ed25519 key pair.
- 601.079782| cloud-init-1288|: Your identification has been saved in /etc/ssh/
ssh_host_ed25519_key
- 601.079814| cloud-init-1288|: Your public key has been saved in /etc/ssh/ssh_
host_ed25519_key.pub
- 601.079847| cloud-init-1288|: The key fingerprint is:
- 601.079879| cloud-init-1288|: SHA256:yWsZ/5+7u7D3SIcd7HYnyajXyewnt5nQ+ZI3So3b
eN8 root@ubuntu-server
- 601.079911| cloud-init-1288|: The key's randomart image is:
- 601.079942| cloud-init-1288|: +---ED25519 256|---+
- 601.079974| cloud-init-1288|: !          !
- 601.080010| cloud-init-1288|: !          !
- 601.080042| cloud-init-1288|: !          !
- 601.080076| cloud-init-1288|: !          . .      . !
- 601.080107| cloud-init-1288|: !          S          o !
- 601.080139| cloud-init-1288|: !          =          o=++!
- 601.080179| cloud-init-1288|: !          + . o**$=!
- 601.080210| cloud-init-1288|: !          .      oo+&B%!
- 601.080244| cloud-init-1288|: !          ..o*/E!
- 601.080289| cloud-init-1288|: +-----SHA256|-----+
- 612.293731| cloud-init-2027|: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1 runnin
g 'modules:config' at Thu, 04 Jun 2020 12:06:59 +0000. Up 612.11 seconds.
- 612.293866| cloud-init-2027|: Set the following 'random' passwords
- 612.293940| cloud-init-2027|: installer:wgYsAPzYQbFYqU2X2hYm
ci-info: no authorized SSH keys fingerprints found for user installer.
<14>Jun  4 12:07:00 ec2:
<14>Jun  4 12:07:00 ec2: #####
#####
<14>Jun  4 12:07:00 ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
<14>Jun  4 12:07:00 ec2: 1024 SHA256:ZBNyksVVYVZhKJeL+PWKpsdUcn21yiceX/DboXQdPq0
root@ubuntu-server (DSA)
<14>Jun  4 12:07:00 ec2: 256 SHA256:Bitar9fVHUH2FnYVSJJnldprdAcM5Est0dmRWFTUi8k

```

(continues on next page)

(continued from previous page)

```

root@ubuntu-server (ECDSA)
<14>Jun  4 12:07:00 ec2: 256 SHA256:yWsZ/5+7u7D3SIcd7HYnyajXyeWnt5nQ+ZI3So3beN8
root@ubuntu-server (ED25519)
<14>Jun  4 12:07:00 ec2: 3072 SHA256:kHtKABZwk8AE80fy0KPzTrCYpht4iXdZmJ37Cgi3fJ0
root@ubuntu-server (RSA)
<14>Jun  4 12:07:00 ec2: -----END SSH HOST KEY FINGERPRINTS-----
<14>Jun  4 12:07:00 ec2: #####
#####
-----BEGIN SSH HOST KEY KEYS-----
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIXM6t1/
35ot/aPI59ThIJBzg+qGJJ17+1ZVHfzMEDbsTwpM7e9pstPZUM7W1IHWqDvLQDBm/hGg4u8ZGEqmIMI=
root@ubuntu-server
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIN7QtU+en+RGruj2zuxWgkMqLmh+35/GR/OEOD16k4nA
root@ubuntu-server
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDjKT7iUAvSjkUqI1l3fHysE+Gj7uLwGgGjYh639px
kchEbbS3V48eROY9BmDISEHfjYXGY2wEH0tGJjNRR0GJhZJVNR+qAqJBioj9d/TwXEgwLP8eAy9aVtJB
K1rIylnMQltx/SIhgiymjHLCtKlVoIS4l0frT9FiF54Qi/JeJlwGJIW3W2XgcY90DT0Q5g3PSmLZ8KTR
imTf9Fy7WJEPa08b3fimYwsz9enuS/gECEUGV3M1MvrzpaQju27NUEOpSMZHR62IMxGvIjYIu3dUkAzM
MBdwxHdLMQ8rI8PehyHdiFr6g2Ifxoy5QLmb3hISKlq/R6pLLeXbb748gN2i8WCvK0AEGfa/kJDW3RNU
VYd+ACBBzyhVbiw7W1CQW/ohik3wyosUyi9nJq2Iq0A7kkGH+1XoYq/e4/MoqxhIK/oaiudYAKaCWmP1
r/fBa3hlf0f7mVHvxA3tWZc2wYUxFPTmePvpydP2PSctHMHgboaHrGIY2CdSgg8SUDPKrOE= root@ub
untu-server
-----END SSH HOST KEY KEYS-----
- 612.877357| cloud-init-2045|: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1 runnin
g 'modules:final' at Thu, 04 Jun 2020 12:07:00 +0000. Up 612.79 seconds.
- 612.877426| cloud-init-2045|: ci-info: no authorized SSH keys fingerprints fo
und for user installer.
- 612.877468| cloud-init-2045|: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1 finish
ed at Thu, 04 Jun 2020 12:07:00 +0000. DataSource DataSourceNoCloud --seed=/var/l
ib/cloud/seed/nocloud|--dsmode=net|. Up 612.87 seconds
- 612.877509| cloud-init-2045|: Welcome to Ubuntu Server InstallerÜ
- 612.877551| cloud-init-2045|: Above you will find SSH host keys and a random
password set for the `installer` user. You can use these credentials to ssh-in a
nd complete the installation. If you provided SSH keys in the cloud-init datasou
rce, they were also provisioned to the installer user.
- 612.877634| cloud-init-2045|: If you have access to the graphical console, li
ke TTY1 or HMC ASCII terminal you can complete the installation there too.

```

It is possible to connect to the installer over the network, which might allow the use of a more capable terminal.

To connect, SSH to installer@10.222.111.24.

The password you should use is "KRuXtz5dURAYPkcjcUvA".

The host key fingerprints are:

```

RSA      SHA256:3IvYMKU05LQSKBx0VZUJMzdtXpz3RJL3dEQsg3UWc54
ECDSA    SHA256:xd1xnkBpn49DUbuP8uWro2mu1GM4MtnqR2WEWg1fS3o
ED25519  SHA256:Hk3+/4+X7NJBHL6/e/6xHhNXsbHBsOvt6i8YEFUepko

```

Ubuntu Focal Fossa (development branch) ubuntu-server ttyS0

The next step is to remotely connect to the install system and to proceed with the Subiquity installer.

Notice that at the end of the installer boot-up process, all necessary data is provided to pro-

ceed with running the installer in a remote SSH shell. The command to execute locally is:

```
user@workstation:~$ ssh installer@10.222.111.24
```

A temporary random password for the installation was created and shared as well, which should be used without the leading and trailing double quotes:

```
"KRuXtz5dURAYPkcjcUvA"
```

```
user@workstation:~$ ssh installer@10.222.111.24
```

```
The authenticity of host '10.222.111.24 (10.222.111.24)' can't be established.  
ECDSA key fingerprint is  
SHA256:xd1xnkBpn49DUbuP8uWro2mu1GM4MtnqR2WEWg1fS3o.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '10.222.111.24' (ECDSA) to the list of known hosts.  
installer@10.222.111.24's password: KRuXtz5dURAYPkcjcUvA
```

You may now temporarily see some login messages like these:

```
Welcome to Ubuntu Focal Fossa (development branch) (GNU/Linux 5.4.0-42-generic s390x)
```

```
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/pro
```

```
System information as of Wed Jun  3 17:32:10 UTC 2020
```

```
System load:   0.0      Memory usage: 2%   Processes:      146  
Usage of /home: unknown  Swap usage:   0%   Users logged in: 0
```

```
0 updates can be installed immediately.  
0 of these updates are security updates.
```

```
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
```

Eventually, the initial Subiquity installer screen appears:

```
=====
Willkommen! Bienvenue! Welcome! ????? ??????????!! Welkom!
=====
Use UP, DOWN and ENTER keys to select your language.

[ English                > ]
[ Asturianu              > ]
[ Catal                 > ]
[ Hrvatski               > ]
[ Nederlands             > ]
[ Suomi                 > ]
[ Fran  ais              > ]
[ Deutsch                > ]
```

(continues on next page)

(continued from previous page)

```
[ Magyar                > ]
[ Latviešu              > ]
[ Norsk bokmål          > ]
[ Polski                > ]
[ Español               > ]
```

From this point, you can follow the normal Subiquity installation. For more details, refer to the [Subiquity installer documentation](https://canonical-subiquity.readthedocs-hosted.com/en/latest/index.html)⁷⁹.

```
=====
Keyboard configuration
=====
Please select your keyboard layout below, or select "Identify keyboard" to
detect your layout automatically.

Layout:  [ English (US)                v ]

Variant: [ English (US)                v ]

[ Identify keyboard ]

[ Done      ]
[ Back      ]
```

```
=====
Zdev setup
=====
ID                                ONLINE  NAMES                                ^
generic-ccw
0.0.0009                          >
0.0.000c                          >
0.0.000d                          >
0.0.000e                          >

dasd-eckd
0.0.0190                          >
0.0.0191                          >
0.0.019d                          >
0.0.019e                          >
0.0.0200                          >
0.0.0300                          >
0.0.0400                          >
0.0.0592                          >
v
```

(continues on next page)

⁷⁹ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/index.html>

(continued from previous page)

```
[ Continue ]
[ Back    ]
```

- If the list is long, hit the End key that will automatically scroll you down to the bottom of the Z devices list and screen.

```
=====
Zdev setup
=====
ID                ONLINE  NAMES                ^
generic-ccw
0.0.0009          >
0.0.000c          >
0.0.000d          >
0.0.000e          >

dasd-eckd
0.0.0190          >
0.0.0191          >
0.0.019d          >
0.0.019e          >
0.0.0200          > < (close)
0.0.0300          > Enable
0.0.0400          > Disable
0.0.0592          >
                                     v

[ Continue ]
[ Back    ]
```

```
=====
Zdev setup
=====
ID                ONLINE  NAMES                ^
generic-ccw
0.0.0009          >
0.0.000c          >
0.0.000d          >
0.0.000e          >

dasd-eckd
0.0.0190          >
0.0.0191          >
0.0.019d          >
0.0.019e          >
0.0.0200          online  dasda >
0.0.0300          >
0.0.0400          >
0.0.0592          >
                                     v

[ Continue ]
[ Back    ]
```

```
=====
Zdev setup
=====

dasd-eckd
0.0.0190 >
0.0.0191 >
0.0.019d >
0.0.019e >
0.0.0200 online dasda >
0.0.0300 >
0.0.0400 >
0.0.0592 >

qeth
0.0.0600:0.0.0601:0.0.0602 enc600 >
0.0.0603:0.0.0604:0.0.0605 >

dasd-eckd
0.0.1607 >

[ Continue ]
[ Back ]
```

```
=====
Network connections
=====

Configure at least one interface this server can use to talk to other
machines, and which preferably provides sufficient access for updates.

NAME    TYPE  NOTES
[ enc600 eth - > ]
static 10.222.111.24/24
02:28:0a:00:00:39 / Unknown Vendor / Unknown Model

[ Create bond > ]

[ Done ]
[ Back ]
```

```
=====
Configure proxy
=====

If this system requires a proxy to connect to the internet, enter its
details here.

Proxy address:
    If you need to use a HTTP proxy to access the outside world,
    (continues on next page)
```

(continued from previous page)

enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of "http://[[user][:pass]@host[:port]]/".

[Done]

[Back]

```
=====
Configure Ubuntu archive mirror
```

If you use an alternative mirror for Ubuntu, enter its details here.

```
Mirror address: http://ports.ubuntu.com/ubuntu-ports
You may provide an archive mirror that will be used instead
of the default.
```

[Done]
[Back]

```
=====
Guided storage configuration
```

Configure a guided storage layout, or create a custom one:

(X) Use an entire disk

```
[ 0X0200      local disk 6.876G      v ]
```

```
[ ] Set up this disk as an LVM group
```

```
[ ] Encrypt the LVM group with LUKS
```

Passphrase:

(continues on next page)

(continued from previous page)

Confirm passphrase:

() Custom storage layout

[Done]
[Back]

Storage configuration

FILE SYSTEM SUMMARY

MOUNT POINT	SIZE	TYPE	DEVICE TYPE
[/	6.875G	new ext4	new partition of local disk >]

AVAILABLE DEVICES

No available devices

[Create software RAID (md) >]
[Create volume group (LVM) >]

USED DEVICES

[Done]
[Reset]
[Back]

Storage configuration

FILE SYSTEM SUMMARY

Confirm destructive action

Selecting Continue below will begin the installation process and result in the loss of data on the disks selected to be formatted.

You will not be able to return to this or a previous screen once the installation has started.

Are you sure you want to continue?

[No]
[Continue]

[Reset]
[Back]

```
=====
Profile setup
=====
Enter the username and password you will use to log in to the system. You
can configure SSH access on the next screen but a password is still needed
for sudo.

Your name: Ed Example

Your server's name: 10.222.111.24
                   The name it uses when it talks to other computers.

Pick a username: ubuntu

Choose a password: *******

Confirm your password: *******

[ Done ]
```

```
=====
SSH Setup
=====
You can choose to install the OpenSSH server package to enable secure remote
access to your server.

[ ] Install OpenSSH server

Import SSH identity: [ No v ]
                   You can import your SSH keys from Github or Launchpad.

Import Username:

[X] Allow password authentication over SSH

[ Done ]
[ Back ]
```

- It's a nice and convenient new feature to add the user's SSH keys during the installation to the system, since that makes the system login password-less on the initial login!

```
=====
SSH Setup
=====
You can choose to install the OpenSSH server package to enable secure remote
access to your server.
```

(continues on next page)

(continued from previous page)

[X] Install OpenSSH server

Import SSH identity: [from Launchpad v]
You can import your SSH keys from Github or Launchpad.

Launchpad Username: user
Enter your Launchpad username.

[X] Allow password authentication over SSH

[Done]
[Back]

SSH Setup

You can choose to install the OpenSSH server package to enable secure remote access to your server.

Confirm SSH keys

Keys with the following fingerprints were fetched. Do you want to use them?

2048 SHA256:joGsdfW7NbJRkg17sRyXaegoR0iZEdDWdR9HpbC2KIw user@W520 |
(RSA)
521 SHA256:T3JzxvB6K1GzXJpP5NFgX4yXvk0jhHgVbw01F7/fZ2c
frank.heimes@canonical.com (ECDSA)

[Yes]
[No]

[Done]
[Back]

Featured Server Snaps

These are popular snaps in server environments. Select or deselect with SPACE, press ENTER to see more details of the package, publisher and versions available.

[]	kata-containers	Lightweight virtual machines that seamlessly plug into >
[]	docker	Docker container runtime >
[]	mosquitto	Eclipse Mosquitto MQTT broker >
[]	etcd	Resilient key-value store by CoreOS >
[]	stress-ng	A tool to load, stress test and benchmark a computer s >

(continues on next page)

(continued from previous page)

```
[ ] sabnzbd      SABnzbd >
[ ] wormhole    get things from one computer to another, safely >
[ ] slcli       Python based SoftLayer API Tool. >
[ ] doctl       DigitalOcean command line tool >
[ ] keepalived  High availability VRRP/BFD and load-balancing for Linu >
[ ] juju        Simple, secure and stable devops. Juju keeps complexit >
```

```
[ Done      ]
[ Back      ]
```

```
=====
Install complete!
=====
```

```

      configuring raid (mdadm) service ^
      installing kernel
      setting up swap
      apply networking config
      writing etc/fstab
      configuring multipath
      updating packages on target system
      configuring pollinate user-agent on target
      updating initramfs configuration
finalizing installation
      running 'curtin hook'
      curtin command hook
      executing late commands
final system configuration
      configuring cloud-init
      installing openssh-server | v
```

[View full log]

```
=====
Installation complete!
=====
```

```

      Finished install! ^
      apply networking config
      writing etc/fstab
      configuring multipath
      updating packages on target system
      configuring pollinate user-agent on target
      updating initramfs configuration
      finalizing installation
      running 'curtin hook'
      curtin command hook
      executing late commands
final system configuration
      configuring cloud-init
      installing openssh-server
      restoring apt configuration
      downloading and installing security updates v
```

(continues on next page)

(continued from previous page)

[View full log]
[Reboot]

Installation complete!

=====

```
----- Finished install! -----
  apply networking config
  writing etc/fstab
  configuring multipath
  updating packages on target system
  configuring pollinate user-agent on target
  updating initramfs configuration
finalizing installation
  running 'curtin hook'
  curtin command hook
  executing late commands
final system configuration
  configuring cloud-init
  installing openssh-server
  restoring apt configuration
  downloading and installing security updates
```

[Connection to 10.222.111.24 closed by remote host.

[Rebooting...]

Connection to 10.222.111.24 closed.

user@workstation:~\$

Type reset to clear the screen and to revert it back to the defaults.

Now remove the old host key, since the system got a new one during the installation:

```
user@workstation:~$ ssh-keygen -f "/home/user/.ssh/known_hosts" -R
"10.222.111.24"
```

```
# Host 10.222.111.24 found: line 159
/home/user/.ssh/known_hosts updated.
Original contents retained as /home/user/.ssh/known_hosts.old
```

And finally login to the newly installed z/VM guest:

```
user@workstation:~$ ssh ubuntu@10.222.111.24
```

```
Warning: Permanently added the ECDSA host key for IP address
'10.222.111.24' to the list of known hosts.
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-42-generic s390x)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro
```

```
System information as of Wed 03 Jun 2020 05:50:05 PM UTC
```

```
System load: 0.08          Memory usage: 2%    Processes:      157
Usage of /:  18.7% of 6.70GB Swap usage:   0%    Users logged in: 0
```

```
0 updates can be installed immediately.
0 of these updates are security updates.
```

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the individual files
in /usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
```

```
To run a command as administrator (user "root"), use `sudo <command>`.
See `man sudo_root` for details.
```

```
ubuntu@10.222.111.24:~$ uptime
```

```
17:50:09 up 1 min,  1 user,  load average: 0.08, 0.11, 0.05
```

```
ubuntu@10.222.111.24:~$ lsb_release -a
```

```
No LSB modules are available.
Distributor ID:      Ubuntu
Description: Ubuntu 20.04.5 LTS
Release:            20.04
Codename:           focal
```

```
ubuntu@10.222.111.24:~$ uname -a
```

```
Linux 10.222.111.24 5.4.0-42-generic #30-Ubuntu SMP Wed Aug 05 16:57:22 UTC 2020
s390x s390x s390x GNU/Linux
```

```
ubuntu@10.222.111.24:~$ exit
```

```
logout
Connection to 10.222.111.24 closed.
```

```
Done!
```

Non-interactive IBM z/VM autoinstall (s390x)

This non-interactive installation uses ‘autoinstall’, which can be considered the successor to the Debian installer (d-i) and preseed on Ubuntu. This is a detailed step-by-step guide, including output and logs (which are partially a bit shortened, as indicated by ‘...’, to limit the size of this document).

The example z/VM guest here uses a direct-access storage device (*DASD*) and is connected to a regular (non-VLAN) network.

For a zFCP and a VLAN network example, please see the [non-interactive IBM LPAR \(s390x\) installation using autoinstall](#) (page 142) guide.

- Start with the preparation of the (FTP) install server (if it doesn’t already exist).

```
user@local:~$ ssh admin@installserver.local
admin@installserver:~$ mkdir -p /srv/ftp/ubuntu-daily-live-server-20.04
admin@installserver:~$ wget http://cdimage.ubuntu.com/ubuntu-server/focal/daily-live/
current/focal-live-server-s390x.iso --directory-prefix=/srv/ftp/ubuntu-daily-live-
server-20.04
--2020-06-26 12:18:48-- http://cdimage.ubuntu.com/ubuntu-server/focal/daily-live/
current/focal-live-server-s390x.iso
Resolving cdimage.ubuntu.com (cdimage.ubuntu.com)... 2001:67c:1560:8001::1d,
2001:67c:1360:8001::28, 2001:67c:1360:8001::27,
...
Connecting to cdimage.ubuntu.com (cdimage.ubuntu.com)|2001:67c:1560:8001::1d|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 700952576 (668M) [application/x-iso9660-image]
Saving to: ‘focal-live-server-s390x.iso’

focal-live-server-s 100%[=====>] 668.48M  2.73MB/s   in 4m 54s

2020-06-26 12:23:42 (2.27 MB/s) - ‘focal-live-server-s390x.iso’ saved [700952576/
700952576]
admin@installserver:~$
```

- The ISO image needs to be extracted now. Since files in the boot folder need to be modified, loopback mount is not an option here:

```
admin@installserver:~$ cd /srv/ftp/ubuntu-daily-live-server-20.04
admin@installserver:/srv/ftp/ubuntu-daily-live-server-20.04$ mkdir iso
admin@installserver:/srv/ftp/ubuntu-daily-live-server-20.04$ sudo mount -o loop ./
focal-live-server-s390x.iso ./iso
[sudo] password for admin:
mount: /home/user/iso-test/iso: WARNING: device write-protected, mounted read-only.
admin@installserver:/srv/ftp/ubuntu-daily-live-server-20.04$ ls -l
total 684530
-rw-rw-r-- 1 user user 700952576 Jun 26 10:12 focal-live-server-s390x.iso
dr-xr-xr-x 10 root root 2048 Jun 26 10:12 iso
admin@installserver:/srv/ftp/ubuntu-daily-live-server-20.04$
```

- Now make sure an FTP server is running in the *installserver* with */srv/ftp* as ftp-server root (as used in this example).
- Next, prepare an *autoinstall* (HTTP) server. This hosts the configuration data for the non-interactive installation.

```
admin@installserver:/srv/ftp/ubuntu-daily-live-server-20.04$ mkdir -p /srv/www/
autoinstall/zvmguest
admin@installserver:/srv/ftp/ubuntu-daily-live-server-20.04$ cd /srv/www/autoinstall/
zvmguest
admin@installserver:/srv/www/autoinstall/zvmguest$
admin@installserver:/srv/www/autoinstall/zvmguest$ echo "instance-id: $(uuidgen ||
openssl rand -base64 8)" > meta-data
admin@installserver:/srv/www/autoinstall/zvmguest$ cat meta-data
instance-id: 2c2215fb-6a38-417f-b72f-376b1cc44f01
admin@installserver:/srv/www/autoinstall/zvmguest$
```

```
admin@installserver:/srv/www/autoinstall/zvmguest$ vi user-data
admin@installserver:/srv/www/autoinstall/zvmguest$ cat user-data
```

```
#cloud-config
autoinstall:
  version: 1
  refresh-installer:
    update: yes
  reporting:
    builtin:
      type: print
  apt:
    preserve_sources_list: false
    primary:
      - arches: [amd64, i386]
        uri: http://archive.ubuntu.com/ubuntu
      - arches: [default]
        uri: http://ports.ubuntu.com/ubuntu-ports
  keyboard:
    layout: en
    variant: us
  locale: en_US
  identity:
    hostname: zvmguest
    password:
"$6$ebJ1f8wxED22bTL4F46P0"
    username: ubuntu
  user-data:
    timezone: America/Boston
    users:
      - name: ubuntu
        password:
"$6$KwuxED22bTL4F46P0"
        lock_passwd: false
  early-commands:
    - touch /tmp/lets_activate_the_s390x_devices
    - chzdev dasd -e 1f00
    - touch /tmp/s390x_devices_activation_done
  network:
    version: 2
    ethernet:
      enc600:
        addresses: [10.11.12.23/24]
        gateway4: 10.11.12.1
        nameservers:
          addresses: [10.11.12.1]
```

(continues on next page)

(continued from previous page)

```
ssh:
  install-server: true
  allow-pw: true
  authorized-keys: ['ssh-rsa meQwtZ user@workstation # ssh-import-id lp:user']
admin@installserver:~$
```

- For s390x installations, the early-commands section is the interesting part:

```
early-commands:
- touch /tmp/lets_activate_the_s390x_devices
- chzdev dasd -e 1f00
- touch /tmp/s390x_devices_activation_done
```

The first and last early-commands are optional; they only frame and indicate the real s390x command activation.

In this particular example a single *DASD ECKD* disk with the address 1f00 is enabled. zFCP disk storage can be enabled via their host (*host-bus-adapters*) addresses, for example *e000* (chzdev zfc -e e000) and *e100* (chzdev zfc -e e000). These have certain Logical Unit Numbers (LUNs) assigned, which are all automatically discovered and activated by chzdev zfc-lun -e --online. Activation of a QETH device would look like this: chzdev qeth -e 0600.

- For more details about the autoinstall config options, please have a look at the [autoinstall reference](#)⁸⁰ and [autoinstall schema](#)⁸¹ page.
- Now make sure a HTTP server is running with /srv/www as web-server root (in this particular example).
- Log in to your z/VM system using your preferred 3270 client – for example x3270 or c3270.
- Transfer the installer kernel, initrd, parmfile and exec file to the z/VM system that is used for the installation. Put these files (for example) on File Mode (Fm) A (a.k.a disk A):

```
listfiles
UBUNTU  EXEC    A1
KERNEL  UBUNTU  A1
INITRD   UBUNTU  A1
PARMFILE UBUNTU  A1
```

- Now specify the necessary autoinstall parameters in the parmfile:

```
xedit PARMFILE UBUNTU A
  PARMFILE UBUNTU 01 F 80 Trunc=80 Size=3 Line=0 Col=1 Alt=0
00000 * * * Top of File * * *
00001 ip=10.11.12.23::10.11.12.1:255.255.255.0:zvmguest:enc600:none:10.11.12.1
00002 url=ftp://installserver.local:21/ubuntu-daily-live-server-20.04/focal-live-ser
ve-ser
00003 ver-s390x.iso autoinstall ds=nocloud-net;s=http://installserver.local:80
00004 /autoinstall/zvmguest/ --- quiet
00005 * * * End of File * * *
```

⁸⁰ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/reference/autoinstall-reference.html>

⁸¹ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/reference/autoinstall-schema.html>

Note:

In case of any issues hitting the 80-character-per-line limit of the file, you can write parameters across two lines as long as there are no unwanted white spaces. To view all 80 characters in one line, disable the prefix area on the left. “prefix off | on” will be your friend – use it in the command area.

- You can now start the z/VM installation by executing the UBUNTU REXX script with UBUNTU.
- Now monitor the initial program load (IPL) – a.k.a. the boot-up process – of the install system. This is quite crucial, because during this process a temporary installation password is generated and displayed. The line(s) look similar to this:

```
...
|37.487141| cloud-init-1873|: Set the following 'random' passwords
|37.487176| cloud-init-1873|: installer: **i7UFdP8fhiVVMme3qqH8**
...
```

This password is needed for remotely connecting to the installer via SSH in the next step.

- So, start the REXX script:

```
UBUNTU
00: 00000004 FILES PURGED
00: RDR FILE 1254 SENT FROM zvmguest  PUN WAS 1254 RECS 102K CPY
001 A NOHOLD NO
KEEP
00: RDR FILE 1258 SENT FROM zvmguest  PUN WAS 1258 RECS 0003 CPY
001 A NOHOLD NO
KEEP
00: RDR FILE 1262 SENT FROM zvmguest  PUN WAS 1262 RECS 303K CPY
001 A NOHOLD NO
KEEP
00: 00000003 FILES CHANGED
00: 00000003 FILES CHANGED
01: HCPGSP2627I The virtual machine is placed in CP mode due to a SIGP
initial C
PU reset from CPU 00.
02: HCPGSP2627I The virtual machine is placed in CP mode due to a SIGP
initial C
PU reset from CPU 00.
03: HCPGSP2627I The virtual machine is placed in CP mode due to a SIGP
initial C
PU reset from CPU 00.
- 0.403380| Initramfs unpacking failed: Decoding failed
ln: /tmp/mountroot-fail-hooks.d//scripts/init-premount/lvm2: No such file or dir
ectory
QETH device 0.0.0600:0.0.0601:0.0.0602 configured
IP-Config: enc600 hardware address 02:28:0b:00:00:51 mtu 1500
IP-Config: enc600 guessed broadcast address 10.11.12.255
IP-Config: enc600 complete:
  address: 10.11.12.23    broadcast: 10.210.210.255    netmask: 255.255.255.0

gateway: 10.11.12.1    dns0      : 10.11.12.1    dns1      : 0.0.0.0
```

(continues on next page)

(continued from previous page)

```

host      : zvmguest
rootserver: 0.0.0.0 rootpath:
filename  :
Connecting to installserver.local:21 (10.11.12.2:21)
focal-live-server-s3  2% !                               ! 15.2M  0:00:48 ETA
focal-live-server-s3  16% !*****                         ! 126M  0:00:09 ETA
focal-live-server-s3  31% !*****                         ! 236M  0:00:06 ETA
focal-live-server-s3  46% !*****                         ! 347M  0:00:04 ETA
focal-live-server-s3  60% !*****                         ! 456M  0:00:03 ETA
focal-live-server-s3  74% !*****                         ! 563M  0:00:02 ETA
focal-live-server-s3  88% !*****                         ! 667M  0:00:00 ETA
focal-live-server-s3 100% !*****                         ! 752M  0:00:00 ETA
mount: mounting /cow on /root/cow failed: No such file or directory
Connecting to plymouth: Connection refused
passwd: password expiry information changed.
- 16.748137| /dev/loop3: Can't open blockdev
- 17.908156| systemd-1|: Failed unmounting /cdrom.
- -0;1;31mFAILED -0m| Failed unmounting -0;1;39m/cdrom -0m.
- -0;32m OK -0m| Listening on -0;1;39mJournal Socket -0m.
      Mounting -0;1;39mHuge Pages File System -0m...
      Mounting -0;1;39mPOSIX Message Queue File System -0m...
      Mounting -0;1;39mKernel Debug File System -0m...
      Starting -0;1;39mJournal Service -0m...
...
[ 61.190916] cloud-init[2076]: Cloud-init v. 20.1-10-g71af48df-0ubuntu5 running
'modules:final' at Fri, 26 Jun 2020 11:02:01 +0000. Up 61.09 seconds.
[ 61.191002] cloud-init[2076]: ci-info: no authorized SSH keys fingerprints fo
und for user installer.
[ 61.191071] cloud-init[2076]: Cloud-init v. 20.1-10-g71af48df-0ubuntu5 finished at
Fri, 26 Jun 2020 11:02:01 +0000.
Datasource DataSourceNoCloudNet [seed=cmdline,
/var/lib/cloud/seed/nocloud,
http://installserver.local:80/autoinstall/zvmguest/]
[dsmode=net]. Up 61.18 seconds
[ 61.191136] cloud-init[2076]: Welcome to Ubuntu Server Installer!
[ 61.191202] cloud-init[2076]: Above you will find SSH host keys and a random
password set for the `installer` user. You can use these credentials to ssh-in
and complete the installation. If you provided SSH keys in the cloud-init datasource,
they were also provisioned to the installer user.
[ 61.191273] cloud-init[2076]: If you have access to the graphical console,
like TTY1 or HMC ASCII terminal you can complete the installation there too.

It is possible to connect to the installer over the network, which
might allow the use of a more capable terminal.

To connect, SSH to installer@10.11.12.23.

The password you should use is 'i7UFdP8fhiVVMme3qqH8'.

The host key fingerprints are:

RSA      SHA256:rBXLeUke3D4gKdsruKEajHjocxc9hr3PI
ECDSA    SHA256:KZZYFswtKxFXWQPuQS9Qp0BUoS6RHswis
ED25519  SHA256:s+5tZfagx0zffc6gYRGW3t1KcBH6f+Vt0

Ubuntu 20.04 LTS ubuntu-server scli_line0

```

- At short notice, you can even log in to the system with the user 'installer' and the temporary password that was given at the end of the boot-up process (see above) of the installation system:

```
user@workstation:~$ ssh installer@zvmguest

The authenticity of host 'zvmguest (10.11.12.23)' can't be established.
ECDSA key fingerprint is SHA256:0/dU/D8jJAEGQcbqKGE9La24IRxUPLpzzs5li9F6Vvk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'zvmguest,10.11.12.23' (ECDSA) to the list of known hosts.
installer@zvmguest's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-37-generic s390x)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Jun 26 11:08:18 UTC 2020

System load:      1.25      Memory usage: 4%    Processes:       192
Usage of /home:  unknown   Swap usage:  0%    Users logged in: 0

0 updates can be installed immediately.
0 of these updates are security updates.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

the installer running on /dev/tty1 will perform the autoinstall

press enter to start a shell
```

- Please note that it informs you about a currently-running autoinstall process:

```
the installer running on /dev/tty1 will perform the autoinstall
```

- Nevertheless, we can quickly check some things – although, only until the autoinstall process is finished and the post-install reboot has been triggered:

```
root@ubuntu-server:/# ls -l /tmp/lets_activate_the_s390x_devices
-rw-r--r-- 1 root root 0 Jun 26 11:08 /tmp/lets_activate_the_s390x_devices
-rw-r--r-- 1 root root 0 Jun 26 11:09 /tmp/s390x_devices_activation_done

root@ubuntu-server:/# lszdev | grep yes
dasd-eckd      0.0.1f00                      yes  yes
qeth           0.0.0600:0.0.0601:0.0.0602   yes  no   enc600
```

(continues on next page)

(continued from previous page)

```
root@ubuntu-server:/#
```

- If you wait long enough, you'll see that the remote session gets closed:

```
root@ubuntu-server:/# Connection to zvmguest closed by remote host.
Connection to zvmguest closed.
user@workstation:~$
```

- As well as at the console:

```
ubuntu-server login:

[[0;1;31mFAILED[0m] Failed unmounting [0;1;39m/cdrom[0m.
[ 169.161139] sd-umoun[15600]: Failed to unmount /oldroot: Device or resource busy
[ 169.161550] sd-umoun[15601]: Failed to unmount /oldroot/cdrom: Device or resource busy
[ 169.168118] shutdown[1]: Failed to finalize file systems, loop devices, ignoring
Total: 282 Selected: 0

Command:
```

- ...and that the post-install reboot got triggered:

```
Message
Mounting [0;1;39mKernel Configuration File System[0m...
Starting [0;1;39mApply Kernel Variables[0m...
[[0;32m OK [0m] Finished [0;1;39mRemount Root and Kernel File Systems[0m.
[[0;32m OK [0m] Finished [0;1;39mUncomplicated firewall[0m.
[[0;32m OK [0m] Mounted [0;1;39mFUSE Control File System[0m.
[[0;32m OK [0m] Mounted [0;1;39mKernel Configuration File System[0m.
...
[ 35.378928] cloud-init[2565]: Cloud-init v. 20.1-10-g71af48df-0ubuntu5 running
'modules:final' at Fri, 26 Jun 2020 11:10:44 +0000. Up 35.29 seconds.
[ 35.378978] cloud-init[2565]: Cloud-init v. 20.1-10-g71af48df-0ubuntu5 finished
at Fri, 26 Jun 2020 11:10:44 +0000. Datasource DataSourceNone. Up 35.37 seconds
[ 35.379008] cloud-init[2565]: 2020-06-26 11:10:44,359 - cc_final_message.py[W
ARNING]: Used fallback datasource
[[0;32m OK [0m] Finished [0;1;39mExecute cloud user/final scripts[0m.
[[0;32m OK [0m] Reached target [0;1;39mCloud-init target[0m.

zvmguest login:
```

- With the completion of the reboot the autoinstall is finished and the z/VM guest is ready to use:

```
user@workstation:~$ ssh-keygen -f "/home/user/.ssh/known_hosts" -R "zvmguest"
# Host zvmguest found: line 163
/home/user/.ssh/known_hosts updated.
Original contents retained as /home/user/.ssh/known_hosts.old
user@workstation:~$ ssh ubuntu@zvmguest
The authenticity of host 'zvmguest (10.11.12.23)' can't be established.
ECDSA key fingerprint is SHA256:iGtCArEg+ZnoZlgt0kvmyy0gPY8UEI+f7zoISOF+m/0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'zvmguest,10.11.12.23' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-39-generic s390x)
```

(continues on next page)

(continued from previous page)

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

System information as of Fri 26 Jun 2020 11:12:23 AM UTC

System load: 0.21           Memory usage: 3%   Processes:      189
Usage of /:  28.2% of 30.88GB Swap usage:   0%   Users logged in: 0

10 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY,
to the extent permitted by applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@zvmguest:~$ uptime
 11:12:35 up 2 min,  1 user,  load average: 0.18, 0.17, 0.08
ubuntu@zvmguest:~$ lsb_release -a
No LSB modules are available.
Distributor ID:      Ubuntu
Description:         Ubuntu 20.04 LTS
Release:             20.04
Codename:            focal
ubuntu@zvmguest:~$ uname -a
Linux zvmguest 5.4.0-39-generic #43-Ubuntu SMP Fri Jun 19 10:27:17
UTC 2020 s390x s390x s390x
GNU/Linux
ubuntu@zvmguest:~$ lsdev | grep yes
dasd-eckd      0.0.1f00                      yes  yes
qeth          0.0.0600:0.0.0601:0.0.0602    yes  yes  enc600
ubuntu@zvmguest:~$ exit
logout
Connection to zvmguest closed.
user@workstation:~$
```

Some closing notes

- It's always best to use the latest installer and autoinstall components: either make sure the installer gets updated to the latest level, or just use a current daily live-server image.
- The ISO image specified with the kernel parameters needs to fit in the boot folder. Its kernel and initrd are specified in the 'Load from Removable Media and Server' task at the *hardware management console (HMC)*.
- In addition to activating disk storage resources in early-commands, other devices like OS-A/QETH can be added and activated there, too. This is not needed for the basic network

device, as specified in the kernel parameters that are used for the installation (that one is automatically handled).

- If everything is properly set up – FTP server for the image, HTTP server for the autoinstall config files – the installation can be as quick as 2 to 3 minutes (depending on the complexity of the autoinstall YAML file).
- There is a simple way of generating a sample autoinstall YAML file: one can perform an interactive Subiquity installation, grab the file `/var/log/installer/autoinstall-user-data`, and use it as an example – but beware that the `early-commands` entries to activate the s390x-specific devices need to be added manually!

Interactive live server installation on IBM Z LPAR (s390x)

Doing an interactive (manual) live installation as described here - meaning without specifying a parmfile - has been supported in Ubuntu Server since LTS 20.04.5 ('Focal').

The following guide assumes that an FTP server to host the installation files is in place, which can be used by the 'Load from Removable Media and Server' task of the *Hardware Management Console (HMC)*.

Download and mount the ISO

Download the 'focal daily live image' (later 20.04.5 image)⁸².

Now loop-back mount the ISO to extract the four files that are needed for the installation:

```
user@workstation:~$ mkdir iso
```

```
user@workstation:~$ sudo mount -o loop ubuntu-20.04.5-live-server-s390x.iso iso
```

```
user@workstation:~$ ls -l  
./iso/boot/{ubuntu.exec,parmfile.*,kernel.u*,initrd.u*}$  
./iso/boot/initrd.ubuntu > ./iso/boot/kernel.ubuntu >  
./iso/boot/parmfile.ubuntu > ./iso/boot/ubuntu.exec
```

Make the files available via your FTP server.

- Open the IBM Z HMC and navigate to 'Systems Management' on your machine.
- Select the LPAR that you are going to install Ubuntu Server on. In this example we use LPAR s1lp11.
- Now select menu: 'Recovery' → 'Load from Removable Media or Server' task.
- Fill out the 'Load from Removable Media or Server' form as follows (adapt the settings to your particular installation environment):

⁸² <http://cdimage.ubuntu.com/ubuntu/releases/20.04.5/release/ubuntu-20.04.5-live-server-s390x.iso>

```
Load from Removable Media, or Server - <machine>:s1lp11
Use this task to load operating system software or utility programs
from a CD / DVD-ROM or a server that can be accessed using FTP.
Select the source of the software:
o Hardware Management Console CD / DVD-ROM
o Hardware Management Console CD / DVD-ROM and assign for operating system use
o Hardware Management Console USB flash memory drive
o Hardware Management Console USB flash memory drive and assign for operating system
use
* FTP Source
  Host computer:      install-server
  User ID:           ftpuser
  Password:          *****
  Account (optional):
  File location (optional): ubuntu-live-server-20.04.5/boot
```

You may need to adjust the file's location according to your install server environment.

- Confirm the entered data:

```
Load from Removable Media or Server - Select Software to Install -
<machine>:s1lp11
Select the software to install.
Select  Name                                     Description
*  ubuntu-live-server-20.04.5/boot/ubuntu.ins    Ubuntu for IBM Z (default kernel)
```

- Confirm again that jobs might be cancelled if proceeding:

```
Load from Removable Media or Server Task Confirmation -
<machine>:s1lp11
Load will cause jobs to be cancelled.
Do you want to continue with this task?
ACT33501
```

- And confirm a last time that it's understood that the task is disruptive:

```
Disruptive Task Confirmation : Load from Removable Media or Server -
<machine>:s1lp11

Attention: The Load from Removable Media or Server task is disruptive.

Executing the Load from Removable Media or Server task may
adversely affect the objects listed below. Review the confirmation text
for each object before continuing with the Load from Removable Media
or Server task.

Objects that will be affected by the Load from Removable Media or
Server task

System Name      Type      OS Name      Status      Confirmation Text
<machine>:s1lp11 Image      Operating    Load from Removable Media
or Server causes operations to be disrupted, since the target is
currently in use and operating normally.

Do you want to execute the Load from Removable Media or Server task?
```

- The 'Load from Removable media or Server' task is now executed:

```
Load from Removable media or Server Progress - P00B8F67:S1LPB
Turn on context sensitive help.
Function duration time:      00:55:00
Elapsed time: 00:00:04
Select      Object Name      Status
*           <machine> s1lp11  Please wait while the image is being loaded.
```

- This may take a moment, but you will soon see:

```
Load from Removable media or Server Progress - <machine>:s1lp11
Function duration time:      00:55:00
Elapsed time: 00:00:21
Select      Object Name      Status
*           <machine> s1lp11  Success
```

- Close the 'Load from Removable media or Server' task and open the console a.k.a. 'Operating System Messages' instead. If no parmfile was configured or provided, you will find the following lines in the 'Operating System Messages' task:

```
Operating System Messages - <machine>:s1lp11

Message

Unable to find a medium container a live file system
Attempt interactive netboot from a URL?
yes no (default yes):
```

- By default, you will now see the interactive network configuration menu (again, only if no parmfile was prepared with sufficient network configuration information).
- Proceed with the interactive network configuration – in this case in a VLAN environment:

```
Unable to find a medium container a live file system
Attempt interactive netboot from a URL?
yes no (default yes):
yes
Available qeth devices:
0.0.c000 0.0.c003 0.0.c006 0.0.c009 0.0.c00c 0.0.c00f
zdev to activate (comma separated, optional):
0.0.c000
QETH device 0.0.c000:0.0.c001:0.0.c002 configured
Two methods available for IP configuration:
* static: for static IP configuration
* dhcp: for automatic IP configuration
static dhcp (default 'dhcp'):
static
ip:
10.222.111.11
gateway (default 10.222.111.1):
10.222.111.1
dns (default 10.222.111.1):
10.222.111.1
vlan id (optional):
1234
http://cdimage.ubuntu.com/ubuntu/releases/20.04.5/release/ubuntu-20.04.5-live-server-
```

(continues on next page)

(continued from previous page)

```
s390x.iso (default)
url:
ftp://10.11.12.2:21/ubuntu-live-server-20.04.5/ubuntu-20.04.5-live-server-s390x.iso
http_proxy (optional):
```

- After the last interactive step here (that this is about an optional proxy configuration), the installer will complete its boot-up process:

```
Configuring networking...
IP-Config: encc000.1234 hardware address 3e:00:10:55:00:ff mtu 1500
IP-Config: encc000.1234 guessed broadcast address 10.222.111.255
IP-Config: encc000.1234 complete:
address: 10.222.111.11 broadcast: 10.222.111.255 netmask: 255.255.255.0

gateway: 10.222.111.1 dns0 : 10.222.111.1 dns1 : 0.0.0.0

rootserver: 0.0.0.0 rootpath:
filename :
Connecting to 10.11.12.2:21 (10.11.12.2:21)
focal-live-server-s 10% |***| 72.9M 0:00:08 ETA
focal-live-server-s 25% |*****| 168M 0:00:05 ETA
focal-live-server-s 42% |*****| 279M 0:00:04 ETA
focal-live-server-s 58% |*****| 390M 0:00:02 ETA
focal-live-server-s 75% |*****| 501M 0:00:01 ETA
focal-live-server-s 89% |*****| 595M 0:00:00 ETA
focal-live-server-s 99% |*****| 662M 0:00:00 ETA
focal-live-server-s 100% |*****| 663M 0:00:00 ETA
ip: RTNETLINK answers: File exists
no search or nameservers found in /run/net-encc000.1234.conf / run/net-*.conf /run/
net6-*.conf
[ 399.808930] /dev/loop3: Can't open blockdev
[[0;1;31m SKIP [0m] Ordering cycle found, skipping [0;1;39mLogin Prompts[0m
[ 401.547705] systemd[1]: multi-user.target: Job getty.target/start deleted to
break ordering cycle starting with multi-user.target/start
[ 406.241972] cloud-init[1321]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1 running
'init-local' at Wed, 03 Jun 2020 17:07:39 +0000. Up 406.00 seconds.
[ 407.025557] cloud-init[1348]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1 running
'init' at Wed, 03 Jun 2020 17:07:40 +0000. Up 406.87 seconds.
[ 407.025618] cloud-init[1348]: ci-info: +++Net device info+++
[ 407.025658] cloud-init[1348]: ci-info: +-----+
+-----+
[ 407.025696] cloud-init[1348]: ci-info: | Device | Up | Address
| Mask | Scope | Hw-Address |
[ 407.025731] cloud-init[1348]: ci-info: +-----+
+-----+
[ 407.025766] cloud-init[1348]: ci-info: | encc000 | True | fe80::3ca7:10f
f:fea5:c69e/64 | link | 72:5d:0d:09:ea:76 |
[ 407.025802] cloud-init[1348]: ci-info: | encc000.1234 | True | 10.245.
236.11 | 255.255.255.0 | global | 72:5d:0d:09:ea:76 |
[ 407.025837] cloud-init[1348]: ci-info: | encc000.1234 | True | fe80::3ca7:10f
f:fea5:c69e/64 | link | 72:5d:0d:09:ea:76 |
[ 407.025874] cloud-init[1348]: ci-info: | lo | True | 127.0
.0.1 | 255.0.0.0 | host | . |
[ 407.025909] cloud-init[1348]: ci-info: | lo | True | ::1/
128 | . | host | . |
[ 407.025944] cloud-init[1348]: ci-info: +-----+
+-----+
```

(continues on next page)

(continued from previous page)

```

-----+-----+-----+-----+
[ 407.025982] cloud-init[1348]: ci-info: ++++++Route I
Pv4 info+++++
[ 407.026017] cloud-init[1348]: ci-info: +-----+-----+-----+
-----+-----+-----+
[ 407.026072] cloud-init[1348]: ci-info: | Route | Destination | Gateway
| Genmask | Interface | Flags |
[ 407.026107] cloud-init[1348]: ci-info: +-----+-----+-----+
-----+-----+-----+
[ 407.026141] cloud-init[1348]: ci-info: | 0 | 0.0.0.0 | 10.222.111.1
| 0.0.0.0 | encc000.1234 | UG |
[ 407.026176] cloud-init[1348]: ci-info: | 1 | 10.222.111.0 | 0.0.0.0
| 255.255.255.0 | encc000.1234 | U |
[ 407.026212] cloud-init[1348]: ci-info: +-----+-----+-----+
-----+-----+-----+
[ 407.026246] cloud-init[1348]: ci-info: ++++++Route IPv6 info+++
+++++
[ 407.026280] cloud-init[1348]: ci-info: +-----+-----+-----+
-----+-----+
[ 407.026315] cloud-init[1348]: ci-info: | Route | Destination | Gateway | Int
erface | Flags |
[ 407.026355] cloud-init[1348]: ci-info: +-----+-----+-----+
-----+-----+
[ 407.026390] cloud-init[1348]: ci-info: | 1 | fe80::/64 | :: | en
cc000 | U |
[ 407.026424] cloud-init[1348]: ci-info: | 2 | fe80::/64 | :: | encc
000.1234 | U |
[ 407.026458] cloud-init[1348]: ci-info: | 4 | local | :: | en
cc000 | U |
[ 407.026495] cloud-init[1348]: ci-info: | 5 | local | :: | encc
000.1234 | U |
[ 407.026531] cloud-init[1348]: ci-info: | 6 | ff00::/8 | :: | en
cc000 | U |
[ 407.026566] cloud-init[1348]: ci-info: | 7 | ff00::/8 | :: | encc
000.1234 | U |
[ 407.026600] cloud-init[1348]: ci-info: +-----+-----+-----+
-----+-----+
[ 407.883058] cloud-init[1348]: Generating public/private rsa key pair.
[ 407.883117] cloud-init[1348]: Your identification has been saved in /etc/ssh/
ssh_host_rsa_key
[ 407.883154] cloud-init[1348]: Your public key has been saved in /etc/ssh/ssh_
host_rsa_key.pub
[ 407.883190] cloud-init[1348]: The key fingerprint is:
[ 407.883232] cloud-init[1348]: SHA256:KX5cHC4YL9dXpvhnP6eSfS+J/zmKgg9zdLEzaEb+
RTA root@ubuntu-server
[ 407.883267] cloud-init[1348]: The key's randomart image is:
[ 407.883302] cloud-init[1348]: +---[RSA 3072]---+
[ 407.883338] cloud-init[1348]: | . E.. |
[ 407.883374] cloud-init[1348]: | o . o |
[ 407.883408] cloud-init[1348]: | . . = +o. |
[ 407.883443] cloud-init[1348]: | + =ooo++ |
[ 407.883478] cloud-init[1348]: | + S *.o. |
[ 407.883512] cloud-init[1348]: | . = o o. |
[ 407.883546] cloud-init[1348]: | .o+o ..+o. |
[ 407.883579] cloud-init[1348]: | o=.. =o+++|
[ 407.883613] cloud-init[1348]: | .... ++*0|

```

(continues on next page)

(continued from previous page)

```
[ 407.883648] cloud-init[1348]: +----[SHA256]-----+
[ 407.883682] cloud-init[1348]: Generating public/private dsa key pair.
[ 407.883716] cloud-init[1348]: Your identification has been saved in /etc/ssh/
ssh_host_dsa_key
[ 407.883750] cloud-init[1348]: Your public key has been saved in /etc/ssh/ssh_
host_dsa_key.pub
[ 407.883784] cloud-init[1348]: The key fingerprint is:
[ 407.883817] cloud-init[1348]: SHA256:xu3vLG1BReKDy3DsuMZc/lg5y/+nhzlEmLDk/qFZ
Am0 root@ubuntu-server
[ 407.883851] cloud-init[1348]: The key's randomart image is:
[ 407.883905] cloud-init[1348]: +---[DSA 1024]-----+
[ 407.883941] cloud-init[1348]: |                ..o|
[ 407.883975] cloud-init[1348]: |                o. o o |
[ 407.884008] cloud-init[1348]: |                +.o+o+ |
[ 407.884042] cloud-init[1348]: |                ...E*oo.. |
[ 407.884076] cloud-init[1348]: |                S+o =.. |
[ 407.884112] cloud-init[1348]: |                . +o+oo.o |
[ 407.884145] cloud-init[1348]: |                **+o*o |
[ 407.884179] cloud-init[1348]: |                .oo.*+oo|
[ 407.884212] cloud-init[1348]: |                .+ ===|
[ 407.884246] cloud-init[1348]: +----[SHA256]-----+
[ 407.884280] cloud-init[1348]: Generating public/private ecdsa key pair.
[ 407.884315] cloud-init[1348]: Your identification has been saved in /etc/ssh/
ssh_host_ecdsa_key
[ 407.884352] cloud-init[1348]: Your public key has been saved in /etc/ssh/ssh_
host_ecdsa_key.pub
[ 407.884388] cloud-init[1348]: The key fingerprint is:
[ 407.884422] cloud-init[1348]: SHA256:P+hBF3fj/pu6+0KaywUYii3Lyuc09Za9/a2elCDO
gdE root@ubuntu-server
[ 407.884456] cloud-init[1348]: The key's randomart image is:
[ 407.884490] cloud-init[1348]: +---[ECDSA 256]----+
[ 407.884524] cloud-init[1348]: |                |
[ 407.884558] cloud-init[1348]: |                . |
[ 407.884591] cloud-init[1348]: |                ..E . o |
[ 407.884625] cloud-init[1348]: |                o .ooo o . |
[ 407.884660] cloud-init[1348]: |                o +S.+.. |
[ 407.884694] cloud-init[1348]: |                . +..*oo.+ . |
[ 407.884728] cloud-init[1348]: |                = o+=.+.+ |
[ 407.884762] cloud-init[1348]: |                . o.....++o oo|
[ 407.884795] cloud-init[1348]: |                oo. . +.*@*+|
[ 407.884829] cloud-init[1348]: +----[SHA256]-----+
[ 407.884862] cloud-init[1348]: Generating public/private ed25519 key pair.
[ 407.884896] cloud-init[1348]: Your identification has been saved in /etc/ssh/
ssh_host_ed25519_key
[ 407.884930] cloud-init[1348]: Your public key has been saved in /etc/ssh/ssh_
host_ed25519_key.pub
[ 407.884966] cloud-init[1348]: The key fingerprint is:
[ 407.884999] cloud-init[1348]: SHA256:CbZpkR9eFHuB1sCDZwSdSdwJzy9FpsIWRIyc9ers
hZ0 root@ubuntu-server
[ 407.885033] cloud-init[1348]: The key's randomart image is:
[ 407.885066] cloud-init[1348]: +---[ED25519 256]---+
[ 407.885100] cloud-init[1348]: |                ../%X..o |
[ 407.885133] cloud-init[1348]: |                .o&*+= |
[ 407.885167] cloud-init[1348]: |                = .+*. * . |
[ 407.885200] cloud-init[1348]: |                . B = + o |
[ 407.885238] cloud-init[1348]: |                + S . . . |
```

(continues on next page)

(continued from previous page)

```
[ 407.885274] cloud-init[1348]: |      .   o o o   |
[ 407.885308] cloud-init[1348]: |      + E      |
[ 407.885345] cloud-init[1348]: |      . .      |
[ 407.885378] cloud-init[1348]: |      .        |
[ 407.885420] cloud-init[1348]: +-----[SHA256]-----+
[ 418.521933] cloud-init[2185]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1 running
'modules:config' at Wed, 03 Jun 2020 17:07:52 +0000. Up 418.40 seconds.
[ 418.522012] cloud-init[2185]: Set the following 'random' passwords
[ 418.522053] cloud-init[2185]: installer:C7BZrW76s4mJzmpf4eUy
ci-info: no authorized SSH keys fingerprints found for user installer.
<14>Jun  3 17:07:52 ec2:
<14>Jun  3 17:07:52 ec2: #####
#####
<14>Jun  3 17:07:52 ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
<14>Jun  3 17:07:52 ec2: 1024 SHA256:xu3vLG1BReKdy3DsuMZc/lg5y/+nhzlEmLDk/qFZAm0
root@ubuntu-server (DSA)
<14>Jun  3 17:07:52 ec2: 256 SHA256:P+hBF3fj/pu6+0KaywUYi3Lyuc09Za9/a2elCD0gdE
root@ubuntu-server (ECDSA)
<14>Jun  3 17:07:52 ec2: 256 SHA256:CbZpkR9eFHuB1sCDZwSdSdwJzy9FpsIWRIyc9ershZ0
root@ubuntu-server (ED25519)
<14>Jun  3 17:07:52 ec2: 3072 SHA256:KX5cHC4YL9dXpvhnP6eSfS+J/zmKgg9zdLEzaEb+RTA
root@ubuntu-server (RSA)
<14>Jun  3 17:07:52 ec2: -----END SSH HOST KEY FINGERPRINTS-----
<14>Jun  3 17:07:52 ec2: #####
#####
-----BEGIN SSH HOST KEY KEYS-----
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBC2zp4Fq
r1+NJOIEQIISbX+EzeJ6ucXSLi2xEvurgwq8iMYT6yYOXB0Pc/XzeFa6vBCDZk3SSSW6Lq83y7VmdRQ=
root@ubuntu-server
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJFzgips94nJNoR4QumiyqlJoSLZ48P+NVrd7zgD5k4T
root@ubuntu-server
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQChKo060715FAjd6ImK7qZbWnL/cpgQ2A2gQEgFN0+1
joF/41ygxuw5aG0IQ0bWFpV9jDsMF5z4qHKzX8tFCpKC0s4uR8QBxh1dDm4wcwgcgAfVLqh7S4/R9Sqa
IFnkCzxThhNeMarcRrutY0mIzspmCg/QvFE1wrXJzl+RtOJ7GiuHHqpm76fX+6ZF1BYhkA87dXQiID2R
yUubSXKCG0NtZlgSzPqD3GB+HxRHHHLT5/Xq+njPq8jIUpqSoHtkBupsyVmcd9gDbz6vng2PuBHwZP9X
17QtyOwxddxk4xIXaTup4g8bH1oF/czswqVxNdfB7XqzROFUOD9rMIB+DwBihsMH1kRik4wwLi6IH4hu
xrykKvfb1xcZe65kR42oDI7JbBwxvxGrOKx8DrEXnBpOWozS0IDm2ZPh3ci/0uCJ4LTIItByyCfAe/gyR
5si4SkmXrIXf5BnErZRgyJnfxKXmsFaSh7wf15w6GmsgzyD9sI2jES9+4By32ZzY0ldPi0s= root@ub
untu-server
-----END SSH HOST KEY KEYS-----
[ 418.872320] cloud-init[2203]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1 running
'modules:final' at Wed, 03 Jun 2020 17:07:52 +0000. Up 418.79 seconds.
[ 418.872385] cloud-init[2203]: ci-info: no authorized SSH keys fingerprints fo
und for user installer.
[ 418.872433] cloud-init[2203]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1 finish
ed at Wed, 03 Jun 2020 17:07:52 +0000. DataSource DataSourceNoCloud [seed=/var/li
b/cloud/seed/nocloud][dsmode=net]. Up 418.86 seconds
[ 418.872484] cloud-init[2203]: Welcome to Ubuntu Server Installer!
[ 418.872529] cloud-init[2203]: Above you will find SSH host keys and a random
password set for the 'installer' user. You can use these credentials to ssh-in a
nd complete the installation. If you provided SSH keys in the cloud-init datasou
rce, they were also provisioned to the installer user.
[ 418.872578] cloud-init[2203]: If you have access to the graphical console, li
ke TTY1 or HMC ASCII terminal you can complete the installation there too.
```

It is possible to connect to the installer over the network, which

(continues on next page)

(continued from previous page)

might allow the use of a more capable terminal.

To connect, SSH to installer@10.222.111.11.

The password you should use is "C7BZrW76s4mJzmpf4eUy".

The host key fingerprints are:

```
RSA      SHA256:KX5cHC4YL9dXpvhnP6eSfS+J/zmKgg9zdLEzaEb+RTA
ECDSA    SHA256:P+hBF3fj/pu6+0KaywUYii3Lyuc09Za9/a2eLCD0gdE
ED25519  SHA256:CbZpkR9eFHuB1sCDZwSdSdwJzy9FpsIWRIyc9ershZ0
```

```
Ubuntu Focal Fossa (development branch) ubuntu-server scli_line0
ubuntu-server login:
```

- At this point you can proceed with the regular installation either by using 'Recovery' → 'Integrated ASCII Console' or with a remote SSH session.
- If the 'Integrated ASCII Console' was opened (you can hit F3 to refresh the task), the initial Subiquity installation screen is presented, which looks like this:

```
=====
Willkommen! Bienvenue! Welcome! ????? ?????????? Welkom!           [ Help ]
=====
Use UP, DOWN and ENTER keys to select your language.
[ English                               > ]
[ Asturianu                             > ]
[ Catal                                 > ]
[ Hrvatski                              > ]
[ Nederlands                            > ]
[ Suomi                                 > ]
[ Fran  ais                             > ]
[ Deutsch                                > ]
[ Magyar                                 > ]
[ Latvie  u                             > ]
[ Norsk bokm  l                         > ]
[ Polski                                 > ]
[ Espa  ol                              > ]
```

- Since the user experience is nicer in a remote SSH session, we recommend using that. However, with certain network environments it's just not possible to use a remote shell, and the 'Integrated ASCII Console' will be the only option.

Note:

At the end of the installer boot-up process, **all** necessary information is provided to proceed with a remote shell.

- The command to execute locally is:

```
user@workstation:~$ ssh installer@10.222.111.11
```

- A temporary random password for the installation was created and shared as well, which you should use without the leading and trailing double quotes:

```
"C7BZrW76s4mJzmpf4eUy"
```

- Hence the remote session for the installer can be opened by:

```
user@workstation:~$ ssh installer@10.222.111.11

The authenticity of host '10.222.111.11 (10.222.111.11)' can't be established.
ECDSA key fingerprint is SHA256:P+hBF3fj/pu6+0KaywUYii3Lyuc09Za9/a2elCD0gdE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.222.111.11' (ECDSA) to the list of known hosts.
installer@10.222.111.11's password: C7BZrW76s4mJzmpf4eUy
```

- One may swiftly see some login messages like the following ones:

```
Welcome to Ubuntu Focal Fossa (development branch) (GNU/Linux 5.4.0-42-generic s390x)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

System information as of Wed Jun  3 17:32:10 UTC 2020

System load:   0.0      Memory usage: 2%   Processes:    146
Usage of /home: unknown  Swap usage:   0%   Users logged in: 0

0 updates can be installed immediately.
0 of these updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

Eventually you'll reach the initial Subiquity installer screen:

```
=====
Willkommen! Bienvenue! Welcome! ????? ??????????! Welkom!
=====
Use UP, DOWN and ENTER keys to select your language.

[ English                > ]
[ Asturianu              > ]
[ Catal                 > ]
[ Hrvatski               > ]
[ Nederlands            > ]
[ Suomi                 > ]
[ Fran  ais             > ]
[ Deutsch               > ]
[ Magyar                > ]
[ Latvie  u             > ]
[ Norsk bokm  l         > ]
[ Polski                > ]
[ Espa  ol              > ]
```

From this point, you can follow the normal Subiquity installation. For more details, refer to the [Subiquity installer documentation](https://canonical-subiquity.readthedocs-hosted.com/en/latest/index.html)⁸³.

(I'm leaving some pretty standard screenshots here just to give an example for a basic installation ...)

```
=====
Keyboard configuration
=====
Please select your keyboard layout below, or select "Identify keyboard" to
detect your layout automatically.

Layout:  [ English (US)                v ]

Variant: [ English (US)                v ]

[ Identify keyboard ]

[ Done      ]
[ Back      ]
```

```
=====
Zdev setup
=====
ID                ONLINE  NAMES                ^
|
dasd-eckd
0.0.1600          >
0.0.1601          >
0.0.1602          >
0.0.1603          >
0.0.1604          >
0.0.1605          >
0.0.1606          >
0.0.1607          >
0.0.1608          >
0.0.1609          >
0.0.160a          >
0.0.160b          >
0.0.160c          >
0.0.160d          >          v

[ Continue      ]
[ Back          ]
```

⁸³ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/index.html>

```
=====
Zdev setup
=====
ID                ONLINE  NAMES                ^
|
dasd-eckd
0.0.1600          >
0.0.1601          > < (close)
0.0.1602          > Enable
0.0.1603          > Disable
0.0.1604          >
0.0.1605          >
0.0.1606          >
0.0.1607          >
0.0.1608          >
0.0.1609          >
0.0.160a          >
0.0.160b          >
0.0.160c          >
0.0.160d          > v

[ Continue ]
[ Back    ]
```

```
=====
Zdev setup
=====
ID                ONLINE  NAMES                ^
|
dasd-eckd
0.0.1600          >
0.0.1601          online  dasda  >
0.0.1602          >
0.0.1603          >
0.0.1604          >
0.0.1605          >
0.0.1606          >
0.0.1607          >
0.0.1608          >
0.0.1609          >
0.0.160a          >
0.0.160b          >
0.0.160c          >
0.0.160d          > v

[ Continue ]
[ Back    ]
```

- One may hit the End key here – that will automatically scroll down to the bottom of the Z devices list and screen:

```
=====
Zdev setup
=====
0.0.f1de:0.0.f1df  > ^
0.0.f1e0:0.0.f1e1  >

(continues on next page)
```

(continued from previous page)

```

0.0.f1e2:0.0.f1e3      >
0.0.f1e4:0.0.f1e5      >
0.0.f1e6:0.0.f1e7      >
0.0.f1e8:0.0.f1e9      >
0.0.f1ea:0.0.f1eb      >
0.0.f1ec:0.0.f1ed      >
0.0.f1ee:0.0.f1ef      >
0.0.f1f0:0.0.f1f1      >
0.0.f1f2:0.0.f1f3      >
0.0.f1f4:0.0.f1f5      >
0.0.f1f6:0.0.f1f7      >
0.0.f1f8:0.0.f1f9      >
0.0.f1fa:0.0.f1fb      >
0.0.f1fc:0.0.f1fd      > |
0.0.f1fe:0.0.f1ff      > v

[ Continue ]
[ Back     ]

```

```

=====
Network connections
=====
Configure at least one interface this server can use to talk to other
machines, and which preferably provides sufficient access for updates.

NAME      TYPE  NOTES
[ encc000  eth   -                > ]
  72:00:bb:00:aa:11 / Unknown Vendor / Unknown Model

[ encc000.1234  vlan -                > ]
  static        10.222.111.11/24
  VLAN 1234 on interface encc000

[ Create bond > ]

[ Continue ]
[ Back     ]

```

- Depending on the installer version you are using you may face a little bug here. In that case the button will be named 'Continue without network', but the network is there. If you see that, just ignore it and continue ... (If you wait long enough the label will be refreshed and corrected.)

```

=====
Configure proxy
=====
If this system requires a proxy to connect to the internet, enter its
details here.

Proxy address:

```

(continues on next page)

(continued from previous page)

If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of "http://[[user][:pass]@host[:port]]/".

[Done]
[Back]

=====

Configure Ubuntu archive mirror

=====

If you use an alternative mirror for Ubuntu, enter its details here.

```
Mirror address: http://ports.ubuntu.com/ubuntu-ports
You may provide an archive mirror that will be used instead
of the default.
```

[Done]

[Back]

=====

Guided storage configuration

=====

Configure a guided storage layout, **or** create a custom one:

(X) Use an entire disk

```
[ 0X1601      local disk 6.877G      v ]
```

- [] Set up this disk **as** an LVM group

- [] Encrypt the LVM group **with** LUKS

Passphrase:

(continues on next page)

(continued from previous page)

Confirm passphrase:

() Custom storage layout

[Done]
[Back]

Storage configuration

FILE SYSTEM SUMMARY

MOUNT POINT	SIZE	TYPE	DEVICE TYPE
[/	6.875G	new ext4	new partition of local disk >]

AVAILABLE DEVICES

No available devices

[Create software RAID (md) >]
[Create volume group (LVM) >]

USED DEVICES

[Done]
[Reset]
[Back]

Storage configuration

FILE SYSTEM SUMMARY

Confirm destructive action

Selecting Continue below will begin the installation process and result in the loss of data on the disks selected to be formatted.

You will not be able to return to this or a previous screen once the installation has started.

Are you sure you want to continue?

[No]
[Continue]

(continues on next page)

(continued from previous page)

```
[ Reset      ]
[ Back      ]
```

```
=====
Profile setup
=====
```

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name: Ed Example

Your server's name: s1lp11
The name it uses when it talks to other computers.

Pick a username: ubuntu

Choose a password: *****

Confirm your password: *****

```
[ Done      ]
```

```
=====
SSH Setup
=====
```

You can choose to install the OpenSSH server package to enable secure remote access to your server.

☐ Install OpenSSH server

Import SSH identity: ☐ No ☒ Yes
You can import your SSH keys from Github or Launchpad.

Import Username:

☒ Allow password authentication over SSH

```
[ Done      ]
[ Back      ]
```

It's a nice and convenient new feature to add the user's SSH keys during the installation to the system, since that makes the system login password-less for the initial login!

```
=====
SSH Setup
=====
```

You can choose to install the OpenSSH server package to enable secure remote access to your server.

☒ Install OpenSSH server

Import SSH identity: ☐ from Launchpad v ☐
You can import your SSH keys from Github or Launchpad.

Launchpad Username: user
Enter your Launchpad username.

☒ Allow password authentication over SSH

☐ Done ☐
☐ Back ☐

```
=====
SSH Setup
=====
```

You can choose to install the OpenSSH server package to enable secure remote access to your server.

```
----- Confirm SSH keys -----
```

Keys with the following fingerprints were fetched. Do you want to use them?

2048 SHA256:joGscmiamcaoincinaäonnväineorviZEdDWdR9Hpbc2KIw user@W520
(RSA)
521 SHA256:T3JzxvB6K1Gzidvoidhoidsaoicak0jhhgwbw01F7/fZ2c
ed.example@acme.com (ECDSA)

☐ Yes ☐
☐ No ☐

☐ Done ☐
☐ Back ☐

```
=====
Featured Server Snaps
=====
```

These are popular snaps in server environments. Select or deselect with SPACE, press ENTER to see more details of the package, publisher and versions available.

☐ kata-containers Lightweight virtual machines that seamlessly plug into >

(continues on next page)

(continued from previous page)

```
[ ] docker          Docker container runtime                >
[ ] mosquitto       Eclipse Mosquitto MQTT broker          >
[ ] etcd            Resilient key-value store by CoreOS    >
[ ] stress-ng       A tool to load, stress test and benchmark a computer s >
[ ] sabnzbd         SABnzbd                                >
[ ] wormhole        get things from one computer to another, safely >
[ ] slcli           Python based SoftLayer API Tool.              >
[ ] doctl           DigitalOcean command line tool          >
[ ] keepalived      High availability VRRP/BFD and load-balancing for Linu >
[ ] juju            Simple, secure and stable devops. Juju keeps complexit >
```

```
[ Done      ]
[ Back      ]
```

```
=====
Install complete!
=====
```

```

      configuring raid (mdadm) service                ^
      installing kernel
      setting up swap
      apply networking config
      writing etc/fstab
      configuring multipath
      updating packages on target system
      configuring pollinate user-agent on target
      updating initramfs configuration
finalizing installation
      running 'curtin hook'
      curtin command hook
      executing late commands
final system configuration
      configuring cloud-init                          ||
      installing openssh-server \                     v|
```

[View full log]

```
=====
Installation complete!
=====
```

```

      Finished install!
      apply networking config                          ^
      writing etc/fstab
      configuring multipath
      updating packages on target system
      configuring pollinate user-agent on target
      updating initramfs configuration
      finalizing installation
      running 'curtin hook'
      curtin command hook
      executing late commands
final system configuration
```

(continues on next page)

(continued from previous page)

```
| configuring cloud-init  
| installing openssh-server  
| restoring apt configuration  
| downloading and installing security updates v|
```

[View full log]
[Reboot]

Installation complete!

=====

```
----- Finished install! -----  
| apply networking config  
| writing etc/fstab  
| configuring multipath  
| updating packages on target system  
| configuring pollinate user-agent on target  
| updating initramfs configuration  
| finalizing installation  
| running 'curtin hook'  
|   curtin command hook  
| executing late commands  
| final system configuration  
| configuring cloud-init  
| installing openssh-server  
| restoring apt configuration  
| downloading and installing security updates v|
```

[Connection to 10.222.111.11 closed by remote host.

[Rebooting...]

Connection to 10.222.111.11 closed.

user@workstation:~\$

- Now type reset to clear the screen and reset it to its defaults.
- Before proceeding one needs to remove the old, temporary host key of the target system, since it was only for use during the installation:

```
user@workstation:~$ ssh-keygen -f "/home/user/.ssh/known_hosts" -R  
"s1lp11"  
  
# Host s1lp11 found: line 159  
/home/user/.ssh/known_hosts updated.  
Original contents retained as /home/user/.ssh/known_hosts.old
```

- And assuming the post-installation reboot is done, one can now login:

```
user@workstation:~$ ssh ubuntu@s1lp11
```

```
Warning: Permanently added the ECDSA host key for IP address
'10.222.111.11' to the list of known hosts.
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-42-generic s390x)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed 03 Jun 2020 05:50:05 PM UTC

System load: 0.08           Memory usage: 2%   Processes:      157
Usage of /: 18.7% of 6.70GB Swap usage:   0%   Users logged in: 0

0 updates can be installed immediately.
0 of these updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@s1lp11:~$ uptime
17:50:09 up 1 min,  1 user,  load average: 0.08, 0.11, 0.05
ubuntu@s1lp11:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 20.04.5 LTS
Release: 20.04
Codename: focal
ubuntu@s1lp11:~$ uname -a
Linux s1lp11 5.4.0-42-generic #30-Ubuntu SMP Wed Aug 05 16:57:22 UTC 2020
s390x s390x s390x GNU/Linux
ubuntu@s1lp11:~$ exit
logout
Connection to s1lp11 closed.
user@workstation:~$
```

Done !

Non-interactive IBM Z LPAR autoinstall (s390x)

This non-interactive installation uses `autoinstall`, which can be considered the successor to the Debian installer (`d-i`) and `preseed` on Ubuntu. This is a detailed step-by-step guide, including output and logs (which are partially a bit shortened, as indicated by ..., to limit the size of this document).

The example logical partition (LPAR) here uses zFCP storage and is connected to a VLAN network. For a *DASD* and a non-VLAN network example, please see the [non-interactive IBM z/VM \(s390x\) auto-installation](#) (page 116) guide.

Start with the preparation of the (FTP) install server (if it doesn't already exist).

```
user@local:~$ ssh admin@installserver.local
```

```
admin@installserver:~$ mkdir -p /srv/ftp/ubuntu-daily-live-server-20.04
```

```
admin@installserver:~$ wget http://cdimage.ubuntu.com/ubuntu-server/focal/
daily-live/current/focal-live-server-s390x.iso
--directory-prefix=/srv/ftp/ubuntu-daily-live-server-20.04

--2020-06-26 12:18:48-- http://cdimage.ubuntu.com/ubuntu-server/focal/daily-live/
current/focal-live-server-s390x.iso
Resolving cdimage.ubuntu.com (cdimage.ubuntu.com)... 2001:67c:1560:8001::1d,
2001:67c:1360:8001::28, 2001:67c:1360:8001::27,
...
Connecting to cdimage.ubuntu.com (cdimage.ubuntu.com)|2001:67c:1560:8001::1d|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 700952576 (668M) [application/x-iso9660-image]
Saving to: 'focal-live-server-s390x.iso'

focal-live-server-s 100%[=====] 668.48M  2.73MB/s   in 4m 54s

2020-06-26 12:23:42 (2.27 MB/s) - 'focal-live-server-s390x.iso' saved [700952576/
700952576]
```

The ISO image needs to be extracted now. Since files in its boot folder need to be modified, loopback mount is not an option here:

```
admin@installserver:~$ cd /srv/ftp/ubuntu-daily-live-server-20.04
```

```
admin@installserver:~/srv/ftp/ubuntu-daily-live-server-20.04$ mkdir iso
```

```
admin@installserver:~/srv/ftp/ubuntu-daily-live-server-20.04$ sudo mount -o
loop ./focal-live-server-s390x.iso ./iso

[sudo] password for admin:
mount: /home/user/iso-test/iso: WARNING: device write-protected, mounted read-only.
```

```
admin@installserver:~/srv/ftp/ubuntu-daily-live-server-20.04$ ls -l

total 684530
-rw-rw-r-- 1 user user 700952576 Jun 26 10:12 focal-live-server-s390x.iso
dr-xr-xr-x 10 root root      2048 Jun 26 10:12 iso
```

```
admin@installserver:~/srv/ftp/ubuntu-daily-live-server-20.04$ rsync -rtvz
./iso/ . && sync
```



```
sending incremental file list
skipping non-regular file "ubuntu"
skipping non-regular file "ubuntu-ports"
./
README.diskdefines
boot.catalog
md5sum.txt
ubuntu.ins
skipping non-regular file "dists/stable"
skipping non-regular file "dists/unstable"
.disk/
.disk/base_installable
.disk/casper-uuid-generic
.disk/cd_type
.disk/info
boot/
boot/README.boot
boot/initrd.off
boot/initrd.siz
boot/initrd.ubuntu
boot/kernel.ubuntu
boot/parmfile.ubuntu
boot/ubuntu.exec
boot/ubuntu.ikr
boot/ubuntu.ins
casper/
...
sent 681,509,758 bytes  received 1,857 bytes  22,344,643.11 bytes/sec
total size is 700,317,941  speedup is 1.03
```

```
admin@installserver:~/srv/ftp/ubuntu-daily-live-server-20.04$ ls -l

total 684578
dr-xr-xr-x  2 user user      4096 Jun 26 10:12 boot
-r--r--r--  1 user user      2048 Jun 26 10:12 boot.catalog
dr-xr-xr-x  3 user user      4096 Jun 26 10:12 casper
dr-xr-xr-x  3 user user      4096 Jun 26 10:11 dists
-rw-rw-r--  1 user user 700952576 Jun 26 10:12 focal-live-server-s390x.iso
dr-xr-xr-x  2 user user      4096 Jun 26 10:11 install
dr-xr-xr-x 10 root  root      2048 Jun 26 10:12 iso
-r--r--r--  1 user user      4944 Jun 26 10:12 md5sum.txt
dr-xr-xr-x  2 user user      4096 Jun 26 10:11 pics
dr-xr-xr-x  3 user user      4096 Jun 26 10:11 pool
dr-xr-xr-x  2 user user      4096 Jun 26 10:11 preseed
-r--r--r--  1 user user       236 Jun 26 10:11 README.diskdefines
-r--r--r--  1 user user       185 Jun 26 10:12 ubuntu.ins
```

Now create .ins and parmfiles dedicated to the LPAR that will be installed (here zlinlpar), based on the default .ins and parmfiles that are shipped with the ISO image:

```
admin@installserver:~/srv/ftp/ubuntu-daily-live-server-20.04$ chmod -R +rw
./boot > cp ./boot/ubuntu.ins ./boot/ubuntu_zlinlpar.ins > cp
./boot/parmfile.ubuntu ./boot/parmfile.zlinlpar
```

```
admin@installserver:~/srv/ftp/ubuntu-daily-live-server-20.04$ vi
./boot/ubuntu_zlinlpar.ins > cat ./boot/ubuntu_zlinlpar.ins

* Ubuntu for z Series (default kernel)
kernel.ubuntu 0x00000000
initrd.off 0x0001040c
initrd.siz 0x00010414
parmfile.zlinlpar 0x00010480
initrd.ubuntu 0x01000000
```

```
admin@installserver:~/srv/ftp/ubuntu-daily-live-server-20.04$ vi
./boot/parmfile.zlinlpar > cat ./boot/parmfile.zlinlpar

ip=10.11.12.42::10.11.12.1:255.255.255.0:zlinlpar:encc000.4711:none:10.11.12.1
vlan=encc000.4711:encc000 url=http://installserver.local:80/ubuntu-daily-live-
server-20.04/focal-live-server-s390x.iso autoinstall ds=nocloud-net;s=http://
installserver.local:80/autoinstall/zlinlpar/ --- quiet
```

Now make sure an FTP server is running in the installserver with /srv/ftp as ftp-server root (as used in this example).

Now prepare an *autoinstall* (HTTP) server, which hosts the configuration data for the non-interactive installation.

```
admin@installserver:~/srv/ftp/ubuntu-daily-live-server-20.04$ mkdir -p
/srv/www/autoinstall/zlinlpar > cd /srv/www/autoinstall/zlinlpar
```

```
admin@installserver:~/srv/www/autoinstall/zlinlpar$ echo "instance-id:
$(uuidgen || openssl rand -base64 8)" > meta-data > cat meta-data

instance-id: 2c2215fb-6a38-417f-b72f-376b1cc44f01
admin@installserver:~/srv/www/autoinstall/zlinlpar$
```

```
admin@installserver:~/srv/www/autoinstall/zlinlpar$ vi user-data > cat
user-data
```

```
#cloud-config
autoinstall:
  version: 1
  refresh-installer:
    update: yes
  reporting:
    builtin:
      type: print
  apt:
    preserve_sources_list: false
    primary:
      - arches: [amd64, i386]
        uri: http://archive.ubuntu.com/ubuntu
      - arches: [default]
        uri: http://ports.ubuntu.com/ubuntu-ports
  keyboard:
    layout: en
    variant: us
  locale: en_US
  identity:
    hostname: zlinlpar
    password:
"$6$ebJ1f8wxED22bTL4F46P0"
    username: ubuntu
  user-data:
    timezone: America/Boston
    users:
      - name: ubuntu
        password:
"$6$KwuxED22bTL4F46P0"
        lock_passwd: false
  early-commands:
    - touch /tmp/lets_activate_the_s390x_devices
    - chzdev zfcv -e e000
    - chzdev zfcv -e e100
    - chzdev zfcv-lun -e --online
    - touch /tmp/s390x_devices_activation_done
  network:
    ethernet:
      encc000: {}
    version: 2
    vlans:
      encc000.4711:
        addresses: [10.11.12.42/24]
        gateway4: 10.11.12.1
        id: 4711
        link: encc000
        nameservers:
          addresses: [10.11.12.1]
  ssh:
    install-server: true
    allow-pw: true
    authorized-keys: ['ssh-rsa meQwtZ user@workstation # ssh-import-id lp:user']
```

For s390x installations the early-commands section is the interesting part:

early-commands:

```
- touch /tmp/lets_activate_the_s390x_devices
- chzdev zfcplun -e e000
- chzdev zfcplun -e e100
- chzdev zfcplun -e --online
- touch /tmp/s390x_devices_activation_done
```

The first and last early-commands are optional; they only frame and indicate the real s390x command activation.

In this particular example, two zFCP hosts (host-bus-adapters) are enabled via their addresses *e000* (`chzdev zfcplun -e e000`) and *e100* (`chzdev zfcplun -e e100`). These have certain logical unit numbers (LUNs) assigned that are all automatically discovered and activated by `chzdev zfcplun -e --online`.

Activation of a direct-access storage device (DASD) would look like this: `chzdev dasd -e 1f00`, and a QETH device activation looks like: `chzdev qeth -e c000`.

See also

For more details about the autoinstall config options, please have a look at the [autoinstall reference](#)⁸⁴ and [autoinstall schema](#)⁸⁵ pages.

- Now make sure a HTTP server is running with `/srv/www` as web-server root (in this particular example).
- Next steps need to be done at the *hardware management console (HMC)*. First, connect to the HMC and proceed with the ‘Load From Removable Media and Server’ task.
- Then, start the ‘Load from Removable Media or Server’ task under ‘Recovery’ → ‘Load from Removable Media or Server’ on your specific LPAR that you are going to install, and fill out the following fields (the contents will be of course different on your system):

```
FTP Source
Host computer:      installserver.local
User ID:           ftpuser
Password:          *****
Account (optional):
File location (optional): ubuntu-daily-live-server-20.04/boot
```

- Now confirm the entered data and click ‘OK’.
- At the ‘Load from Removable Media or Server - Select Software to Install’ screen, choose the LPAR that is going to be installed, here:

```
ubuntu-daily-live-server-20.04/boot/ubuntu_zlinlpar.ins  Ubuntu for z Series
(default kernel)
```

- Confirm again with ‘OK’.
- And another confirmation about the ‘Load will cause jobs to be cancelled’.
- Then, another ‘Yes’ – understanding that it’s a disruptive task:

⁸⁴ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/reference/autoinstall-reference.html>

⁸⁵ <https://canonical-subiquity.readthedocs-hosted.com/en/latest/reference/autoinstall-schema.html>

Disruptive Task Confirmation : Load from Removable Media or Server

- Now monitor the 'Load from Removable media or Server Progress' screen and confirm it once again when the status changes from 'Please wait while the image is being loaded.' to 'Success'.
- Then navigate to 'Daily' -> 'Operating System Messages' to monitor the initial program load (IPL) of the install system ...

```
Message
chzdev: Unknown device type or device ID format: c000.4711
Use 'chzdev --help' for more information
QETH device 0.0.c000:0.0.c001:0.0.c002 configured
IP-Config: encc000.4711 hardware address 1a:3c:99:55:2a:ef mtu 1500
IP-Config: encc000.4711 guessed broadcast address 10.11.12.255
IP-Config: encc000.4711 complete:
address: 10.11.12.42    broadcast: 10.11.12.255    netmask: 255.255.255.0

gateway: 10.11.12.1    dns0      : 10.11.12.1    dns1     : 0.0.0.0

host   : zlinlpar
rootserver: 0.0.0.0 rootpath:
filename :
Connecting to installserver.local:80 (installserver.local:80)
focal-live-server-s3  5% |*                               | 39.9M  0:00:15 ETA
focal-live-server-s3 22% |*****                           | 147M   0:00:07 ETA
focal-live-server-s3 38% |*****                           | 254M   0:00:04 ETA
focal-live-server-s3 53% |*****                           | 355M   0:00:03 ETA
focal-live-server-s3 67% |*****                           | 453M   0:00:02 ETA
focal-live-server-s3 81% |*****                           | 545M   0:00:01 ETA
focal-live-server-s3 94% |*****                           | 633M   0:00:00 ETA
focal-live-server-s3 100% |*****                           | 668M   0:00:00 ETA
mount: mounting /cow on /root/cow failed: No such file or directory
Connecting to plymouth: Connection refused
passwd: password expiry information changed.
Using CD-ROM mount point /cdrom/
Identifying... [5d25356068b713167814807dd678c261-2]
Scanning disc for index files...
Found 2 package indexes, 0 source indexes, 0 translation indexes and 1 signature
Found label 'Ubuntu-Server 20.04 LTS _Focal Fossa_ - Release s390x (20200616)'
This disc is called:
'Ubuntu-Server 20.04 LTS _Focal Fossa_ - Release s390x (20200616)'
...
[ 61.190916] cloud-init[2076]: Cloud-init v. 20.1-10-g71af48df-0ubuntu5 running
'modules:final' at Fri, 26 Jun 2020 11:02:01 +0000. Up 61.09 seconds.
[ 61.191002] cloud-init[2076]: ci-info: no authorized SSH keys fingerprints fo
und for user installer.
[ 61.191071] cloud-init[2076]: Cloud-init v. 20.1-10-g71af48df-0ubuntu5 finished at
Fri, 26 Jun 2020 11:02:01 +0000.
Datasource DataSourceNoCloudNet [seed=cmdline,
/var/lib/cloud/seed/nocloud,
http://installserver.local:80/autoinstall/zlinlpar/]
[dsmode=net]. Up 61.18 seconds
[ 61.191136] cloud-init[2076]: Welcome to Ubuntu Server Installer!
[ 61.191202] cloud-init[2076]: Above you will find SSH host keys and a random
password set for the `installer` user. You can use these credentials to ssh-in
```

(continues on next page)

(continued from previous page)

and **complete** the installation. If you provided SSH keys **in** the cloud-init datasource, they were also provisioned to the installer user.

[61.191273] cloud-init[2076]: If you have access to the graphical console, like TTY1 or HMC ASCII terminal you can **complete** the installation there too.

It is possible to connect to the installer over the network, which might allow the use of a more capable terminal.

To connect, SSH to installer@10.11.12.42.

The password you should use is `'i7UFdP8fhiVVMme3qqH8''`.

The host key fingerprints are:

```
RSA      SHA256:rBXLeUke3D4gKdsruKEajHjocxc9hr3PI
ECDSA    SHA256:KZZYFswtKxFXWQPuQS9Qp0BUoS6RHswis
ED25519  SHA256:s+5tZfagx0zffc6gYRGW3t1KcBH6f+Vt0
```

```
Ubuntu 20.04 LTS ubuntu-server sclp_line0
```

- At short notice, you can even log in to the system with the user 'installer' and the temporary password that was given at the end of the boot-up process (see above) of the installation system:

```
user@workstation:~$ ssh installer@zlinlpar
```

```
The authenticity of host 'zlinlpar (10.11.12.42)' can't be established.  
ECDSA key fingerprint is SHA256:0/dU/D8jJAEGQcbqKGE9La24IRxUPLpzzs5li9F6Vvk.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'zlinlpar,10.11.12.42' (ECDSA) to the list of known  
hosts.  
installer@zlinlpar's password:  
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-37-generic s390x)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/pro  
  
System information as of Fri Jun 26 11:08:18 UTC 2020  
  
System load:      1.25      Memory usage: 4%    Processes:       192  
Usage of /home: unknown  Swap usage:   0%    Users logged in: 0  
  
0 updates can be installed immediately.  
0 of these updates are security updates.  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
the installer running on /dev/tty1 will perform the autoinstall  
  
press enter to start a shell
```

- Notice that it informs you about a currently-running autoinstall process:

```
the installer running on /dev/tty1 will perform the autoinstall
```

- Nevertheless, we can quickly check some things – though only until the autoinstall process has finished and the post-install reboot has been triggered:

```
root@ubuntu-server:/# ls -l /tmp/lets_activate_the_s390x_devices  
-rw-r--r-- 1 root root 0 Jun 26 11:08 /tmp/lets_activate_the_s390x_devices  
-rw-r--r-- 1 root root 0 Jun 26 11:09 /tmp/s390x_devices_activation_done  
  
root@ubuntu-server:/# lszdev | grep yes  
zfcplib 0.0.e000 yes yes  
zfcplib 0.0.e100 yes yes  
zfcplib 0.0.e000:0x50050763060b16b6:0x4026400200000000 yes yes sdb sg1  
zfcplib 0.0.e000:0x50050763061b16b6:0x4026400200000000 yes yes sda sg0  
zfcplib 0.0.e100:0x50050763060b16b6:0x4026400200000000 yes yes sdd sg3  
zfcplib 0.0.e100:0x50050763061b16b6:0x4026400200000000 yes yes sdc sg2  
qeth 0.0.c000:0.0.c001:0.0.c002 yes no encc000  
root@ubuntu-server:/#
```

- If you wait long enough you'll see the remote session get closed:

```
root@ubuntu-server:/# Connection to zlinlpar closed by remote host.
Connection to zlinlpar closed.
user@workstation:~$
```

- As well as at the console:

```
ubuntu-server login:

[[0;1;31mFAILED[0m] Failed unmounting [0;1;39m/cdrom[0m.
[ 169.161139] sd-umoun[15600]: Failed to unmount /oldroot: Device or resource busy
[ 169.161550] sd-umoun[15601]: Failed to unmount /oldroot/cdrom: Device or resource
busy
[ 169.168118] shutdown[1]: Failed to finalize file systems, loop devices, ignoring
Total: 282 Selected: 0

Command:
```

- ...and that the post-install reboot gets triggered:

```
Message
Mounting [0;1;39mKernel Configuration File System[0m...
Starting [0;1;39mApply Kernel Variables[0m...
[[0;32m OK [0m] Finished [0;1;39mRemount Root and Kernel File Systems[0m.
[[0;32m OK [0m] Finished [0;1;39mUncomplicated firewall[0m.
[[0;32m OK [0m] Mounted [0;1;39mFUSE Control File System[0m.
[[0;32m OK [0m] Mounted [0;1;39mKernel Configuration File System[0m.
...
[ 35.378928] cloud-init[2565]: Cloud-init v. 20.1-10-g71af48df-0ubuntu5 runnin
g 'modules:final' at Fri, 26 Jun 2020 11:10:44 +0000. Up 35.29 seconds.
[ 35.378978] cloud-init[2565]: Cloud-init v. 20.1-10-g71af48df-0ubuntu5 finish
ed at Fri, 26 Jun 2020 11:10:44 +0000. Datasource DataSourceNone. Up 35.37 seconds
[ 35.379008] cloud-init[2565]: 2020-06-26 11:10:44,359 - cc_final_message.py[W
ARNING]: Used fallback datasource
[[0;32m OK [0m] Finished [0;1;39mExecute cloud user/final scripts[0m.
[[0;32m OK [0m] Reached target [0;1;39mCloud-init target[0m.

zlinlpar login:
```

- With the completion of the reboot, the autoinstall is finished and the LPAR is ready to use:

```
user@workstation:~$ ssh-keygen -f "/home/user/.ssh/known_hosts" -R "zlinlpar"
# Host zlinlpar found: line 163
/home/user/.ssh/known_hosts updated.
Original contents retained as /home/user/.ssh/known_hosts.old
user@workstation:~$ ssh ubuntu@zlinlpar
The authenticity of host 'zlinlpar (10.11.12.42)' can't be established.
ECDSA key fingerprint is SHA256:iGtCAReg+ZnoZlgt0kvmyy0gPY8UEI+f7zoISO+f+m/0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'zlinlpar,10.11.12.42' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-39-generic s390x)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro
```

(continues on next page)

(continued from previous page)

```
System information as of Fri 26 Jun 2020 11:12:23 AM UTC

System load: 0.21          Memory usage: 3%   Processes:      189
Usage of /: 28.2% of 30.88GB  Swap usage:  0%   Users logged in: 0

10 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY,
to the extent permitted by applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@zlinlpar:~$ uptime
 11:12:35 up 2 min,  1 user,  load average: 0.18, 0.17, 0.08
ubuntu@zlinlpar:~$ lsb_release -a
No LSB modules are available.
Distributor ID:      Ubuntu
Description:         Ubuntu 20.04 LTS
Release:             20.04
Codename:            focal
ubuntu@zlinlpar:~$ uname -a
Linux zlinlpar 5.4.0-39-generic #43-Ubuntu SMP Fri Jun 19 10:27:17
UTC 2020 s390x s390x s390x
GNU/Linux
ubuntu@zlinlpar:~$ lsdev | grep yes
zfcplib 0.0.e000 yes yes
zfcplib 0.0.e100 yes yes
zfcplib 0.0.e000:0x50050763060b16b6:0x4026400200000000 yes yes
sdb sg1
zfcplib 0.0.e000:0x50050763061b16b6:0x4026400200000000 yes yes
sda sg0
zfcplib 0.0.e100:0x50050763060b16b6:0x4026400200000000 yes yes
sdc sg2
zfcplib 0.0.e100:0x50050763061b16b6:0x4026400200000000 yes yes
sdd sg3
qeth 0.0.c000:0.0.c001:0.0.c002 yes yes encc000
ubuntu@zlinlpar:~$ exit
logout
Connection to zlinlpar closed.
user@workstation:~$
```

Some closing notes

- It's always best to use the latest installer and autoinstall components. Be sure to update the installer to the most recent version, or just use a current daily live-server image.
- The ISO image specified with the kernel parameters needs to fit in the boot folder. Its kernel and initrd are specified in the 'Load from Removable Media and Server' task at the hardware management console (HMC).
- In addition to activating disk storage resources in `early-commands`, other devices like OSA/QETH can be added and activated there, too. This is not needed for the basic network device, as specified in the kernel parameters used for the installation (that one is automatically handled).
- If everything is properly set up – FTP server for the image, HTTP server for the autoinstall config files – the installation can be as quick as 2 to 3 minutes. Of course this depends on the complexity of the autoinstall YAML file.
- There is a simple way of generating a sample autoinstall YAML file; one can perform an interactive Subiquity installation, grab the file `/var/log/installer/autoinstall-user-data`, and use it as an example – but beware that the `early-commands` entries to activate the s390x-specific devices need to be added manually!

3.2.2. Security

[System security](#) (page 153) is a crucial topic for any Ubuntu user. In addition to general security topics such as setting up a firewall, AppArmor profiles and user/group management, you will also find how-to guides on:

- **Authentication** with Kerberos, network user authentication with SSSD and physical authentication with smart cards
- **Cryptography** with OpenSSH
- **Virtual Private Networks** OpenVPN and WireGuard VPN

Security

While a fresh Ubuntu installation is usually safe for immediate use, there are some additional steps you can take to introduce a layered approach to your system's security. If you are new to Ubuntu, you may want to refer to our [Introduction to security](#) (page 840) first for a general overview.

General configuration

- [Users and groups management](#) (page 154) for setting up user accounts, permissions and password policies
- [Firewalls](#) (page 163) are recommended for network security
- [AppArmor](#) (page 170) limits permissions and access for the software running on your system
- [Console security](#) (page 177) for an additional physical security barrier

- [TPM-backed LUKS decryption](#) (page 179) to automate decryption of disks

User management

User management is a critical part of maintaining a secure system. Ineffective user and privilege management often leads to a system being compromised. Therefore, it is important that you understand how to protect your server through simple and effective user account management techniques.

Where is root?

Ubuntu developers decided to disable the administrative root account by default in all Ubuntu installations. This does not mean that the root account has been deleted, or that it may not be accessed. Instead, it has been given a password hash that matches no possible value, and so may not log in directly by itself.

Instead, the `sudo` utility (“superuser do”) is used to carry out system administrative duties. `sudo` allows an authorized user to temporarily elevate their privileges using their own password instead of having to know the password belonging to the root account. This provides accountability for all user actions, and gives the administrator control over which actions a user can perform with said privileges.

`sudo-rs`

From Ubuntu 25.10 (Questing Quokka) onward, `sudo` is provided by the `sudo-rs` package (a rust implementation). While `sudo-rs` is not 100% compatible with `sudo.ws`, the majority of common use cases are supported and the change should be invisible to most users.

A list of key differences is [maintained as a reference](#) (page 839)

The original `sudo` utility (maintained by Todd C. Miller), `sudo.ws` remains supported in the 25.10 and the subsequent 26.04 LTS. You will find these utilities installed with `.ws` suffix (for example `sudo.ws`, `visudo.ws`, etc.). Switching the default back to `sudo.ws` is not recommended, but if you need to do so, follow these steps:

Interactive:

```
$ update-alternatives --config sudo
```

Non-interactive:

```
$ update-alternatives --set sudo /usr/bin/sudo.ws
```

You can always switch back to `sudo-rs` using:

```
$ update-alternatives --set sudo /usr/lib/cargo/bin/sudo
```

You can learn more about the motivation for this change in the blog post [Adopting sudo-rs By Default in Ubuntu 25.10](#)⁸⁶

⁸⁶ <https://discourse.ubuntu.com/t/adopting-sudo-rs-by-default-in-ubuntu-25-10/60583>

The rest of the article should work the same with both `sudo.ws` and `sudo-rs`.

Enabling the root account

If for some reason you wish to enable the root account, you will need to give it a password:

```
$ sudo passwd
```

`sudo` will prompt you for your password, and then ask you to supply a new password for root as shown below:

```
[sudo: authenticate] Password: (enter your own password)
New password: (enter a new password for root)
Retype new password: (repeat new password for root)
passwd: password updated successfully
```

Disabling the root account password

To disable the root account password, use the following `passwd` syntax:

```
$ sudo passwd -l root
```

You can learn more about `sudo` by reading the [sudo\(8\)](https://manpages.ubuntu.com/manpages/resolute/man8/sudo.8.html)⁸⁷ manual page: `man sudo`

By default, the initial user created by the Ubuntu installer is a member of the group `sudo` which is added to the file `/etc/sudoers` as an authorized `sudo` user. To give any other account full root access through `sudo`, add them to the `sudo` group.

Listing, adding, and deleting local users

Managing local users and groups differs very little from most other [GNU/Linux](#) operating systems. Ubuntu and other Debian-based distributions encourage the use of the `adduser` package for account management.

Local users and groups are those defined in the `/etc/passwd` and `/etc/group` files, respectively. Via the Name Service Switch ([NSS](#)) mechanism, however, user and group accounts can also come from other sources, like remote Active Directory or Samba servers, OpenLDAP directory services, and more. Managing accounts in those services is an entirely different matter, and not covered in this document.

⁸⁷ <https://manpages.ubuntu.com/manpages/resolute/man8/sudo.8.html>

Listing local users

The local users are defined one per line in the `/etc/passwd` text file. The `cat /etc/passwd` command can be used to list its contents (only a few lines shown below, for brevity):

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
...
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
```

Tip:

The structure of the `/etc/passwd` file is explained in the [passwd\(5\)](https://manpages.ubuntu.com/manpages/resolute/man5/passwd.5.html)⁸⁸ manual page.

⁸⁸ <https://manpages.ubuntu.com/manpages/resolute/man5/passwd.5.html>

There are basically three types of local users:

- system pre-installed users: These are system users essential to a working Linux system, like `root`, `man`, `lp`, and others. Their `UIDs` are in the range between 0 and 99.
- system users: These users are created dynamically, as services are installed. The `UID` range here is between 100 and 999.
- Regular users: Finally, the remaining range is usable for “real” users, generally representing persons using the system. This `UID` range starts at 1000 and, by convention, ends at 59999.

Tip:

For more details about these `UID` ranges, including groups and other configuration options for adding users, check the [adduser.conf.5](https://manpages.ubuntu.com/manpages/resolute/man5/adduser.conf.5.html)⁸⁹ manual page.

⁸⁹ <https://manpages.ubuntu.com/manpages/resolute/man5/adduser.conf.5.html>

To list all local usernames, we can list the contents of `/etc/passwd` with some string manipulation. For example:

```
$ cat /etc/passwd | cut -d : -f 1
```

This command will cut each line at the first column, using the “:” character as the delimiter, and result in an output showing all local users:

```
root
daemon
bin
sys
...
```

If you want to only show regular users, that is, users with an `UID` greater than 1000, we can use a regular expression for this filtering:

```
$ cat /etc/passwd | cut -d : -f 1,3 | grep -E ':[0-9]{4,}'
```

The regular expression `:[0-9]{4,}` means all digits, repeated 4 times or more. The output will be similar to this:

```
nobody:65534
ubuntu:1000
```

Tip:

By convention, the `nobody` users has an UID of 65534. That originated back when the user range was limited to 16 bits (so, 0-65535), and is kept for compatibility reasons.

A note about remote users

There is another command that, at first glance, looks like it lists all local users as well: `getent passwd`. What this tool is doing, however, is listing all entries from the `passwd` database, and this will include any NSS modules that might be installed and configured.

On a default installation of Ubuntu, with no extra configuration, that will indeed list only the local users, but not in the general case. If you have, for example, `libnss-ldap` installed and configured, the output of `getent passwd` will list all local users, **plus all the remote users** it finds in the LDAP server that it was configured with.

Performing such an exhaustive enumeration is frowned upon, because that can incur a large amount of network traffic, and place an unreasonable burden on the remote server. In fact, most NSS services block such enumeration actions by default.

For more information on the Name Service Switch (NSS) mechanism, please consult the [nss\(5\)](https://manpages.ubuntu.com/manpages/resolute/man5/nss.5.html)⁹⁰ and [nsswitch.conf\(5\)](https://manpages.ubuntu.com/manpages/resolute/man5/nsswitch.conf.5.html)⁹¹ manual pages.

Add a user

To add a user account, use the following syntax, and follow the prompts to give the account a password and identifiable characteristics, such as a full name, phone number, etc:

```
$ sudo adduser username
```

⁹⁰ <https://manpages.ubuntu.com/manpages/resolute/man5/nss.5.html>

⁹¹ <https://manpages.ubuntu.com/manpages/resolute/man5/nsswitch.conf.5.html>

Delete a user

To delete a user account and its primary group, use the following syntax:

```
$ sudo deluser username
```

Deleting an account does not remove their respective home folder. You must decide whether or not to delete the folder manually, or to keep it according to your desired retention policies.

Remember, any user added later with the same UID/GID as the previous owner will now have access to this folder if you have not taken the necessary precautions.

You may want to change the UID/GID values to something more appropriate, such as the root account, and perhaps even relocate the folder to avoid future conflicts:

```
$ sudo chown -R root:root /home/username/
```

```
$ sudo mkdir /home/archived_users/
```

```
$ sudo mv /home/username /home/archived_users/
```

Lock or unlock a password

To temporarily lock a user password, use the following syntax:

```
$ sudo passwd -l username
```

Similarly, to unlock a user password:

```
$ sudo passwd -u username
```

Add or delete a group

To add or delete a personalised group, use the following syntax, respectively:

```
$ sudo addgroup groupname
```

```
$ sudo delgroup groupname
```

Add a user to a group

To add a user to a group, use the following syntax:

```
$ sudo adduser username groupname
```

User profile security

When a new user is created, the `adduser` utility creates a brand new home directory named `/home/username`. The default profile is modelled after the contents found in the directory of `/etc/skel`, which includes all profile basics.

If your server will be home to multiple users, you should pay close attention to the user home directory permissions to ensure confidentiality.

Ubuntu 21.10 Impish and later

Home directories are created with private permissions (`0750` or `drwxr-x---`) by default. This ensures that only the user and members of their group can access the directory.

Ubuntu 21.04 Hirsute and earlier

Home directories were created with world read/execute permissions (`0755` or `drwxr-xr-x`) by default. This meant that all users on the system could browse and access the contents of other users home directories, which may not be suitable for your environment.

To verify your current user home directory permissions, use the following syntax:

```
$ ls -ld /home/username  
drwxr-xr-x  2 username username    4096 2007-10-02 20:03 username
```

The output shows that the directory `/home/username` has world-readable permissions. You can remove the world readable-permissions using the following command:

```
$ sudo chmod 0750 /home/username
```

Note:

Some people use the recursive option (`-R`) indiscriminately, which modifies all child folders and files. However, this is not necessary and may have undesirable/unintended consequences. Modifying only the parent directory is enough to prevent unauthorised access to anything below the parent.

A more efficient approach would be to modify the `adduser` global default permissions when creating user home folders. To do this, edit the `/etc/adduser.conf` file and modify the `DIR_MODE` variable to something appropriate, so that all new home directories will receive the correct permissions.


```
DIR_MODE=0750
```

After correcting the directory permissions using any of the previously mentioned techniques, verify the results as follows:

```
$ ls -ld /home/username
drwxr-x---  2 username username  4096 2007-10-02 20:03 username
```

Password policy

A strong password policy is one of the most important aspects of your security posture. Many successful security breaches involve simple **brute force** and **dictionary** attacks against weak passwords.

If you intend to offer any form of remote access involving your local password system, make sure you address minimum password complexity requirements, maximum password lifetimes, and frequent audits of your authentication systems.

Minimum password length

By default, Ubuntu requires a minimum password length of 6 characters, as well as some basic entropy checks. These values are controlled in the file `/etc/pam.d/common-password`, which is outlined below.

```
password      [success=1 default=ignore]      pam_unix.so obscure sha512
```

To adjust the minimum length to 8 characters, change the appropriate variable to `minlen=8`. The modification is outlined below:

```
password      [success=1 default=ignore]      pam_unix.so obscure sha512 minlen=8
```

Note:

Basic password entropy checks and minimum length rules do not apply to the administrator using `sudo`-level commands to setup a new user.

Password expiration

When creating user accounts, you should make it a policy to have a minimum and maximum password age, forcing users to change their passwords when they expire.

To view the current status of a user account:

```
$ sudo chage -l username
```

The output below shows interesting facts about the user account, namely that there are no policies applied:

```
Last password change           : Jan 20, 2015
Password expires               : never
Password inactive              : never
Account expires                : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

To set any of these values, use the `chage` ("change age") command, and follow the interactive prompts:

```
$ sudo chage username
```

The following is also an example of how you can manually change the explicit expiration date (-E) to 01/31/2015, minimum password age (-m) of 5 days, maximum password age (-M) of 90 days, inactivity period (-I) of 30 days after password expiration, and a warning time period (-W) of 14 days before password expiration:

```
$ sudo chage -E 01/31/2015 -m 5 -M 90 -I 30 -W 14 username
```

To verify changes, use the same syntax as mentioned previously:

```
$ sudo chage -l username
```

The output below shows the new policies that have been established for the account:

```
Last password change           : Jan 20, 2015
Password expires               : Apr 19, 2015
Password inactive              : May 19, 2015
Account expires                : Jan 31, 2015
Minimum number of days between password change : 5
Maximum number of days between password change : 90
Number of days of warning before password expires : 14
```

Other security considerations

Many applications use alternate authentication mechanisms that can be easily overlooked by even experienced system administrators. Therefore, it is important to understand and control how users authenticate and gain access to services and applications on your server.

SSH access by disabled passwords

Disabling or locking a user password will not prevent a user from logging into your server remotely if they have previously set up SSH public key authentication. They will still be able to gain shell access to the server, without the need for any password. Remember to check the user's home directory for files that will allow for this type of authenticated SSH access, e.g. `/home/username/.ssh/authorized_keys`.

Remove or rename the directory `.ssh/` in the user's home folder to prevent further SSH authentication access.

Be sure to check for any established SSH connections by the disabled account, as it is possible they may have existing inbound or outbound connections – then `kill` any that are found.

```
$ w | grep username (to get the pts/# terminal)
```

```
$ sudo kill -f pts/#
```

Restrict SSH access to only user accounts that should have it. For example, you may create a group called `sshlogin` and add the group name as the value associated with the `AllowGroups` variable located in the file `/etc/ssh/sshd_config`:

```
AllowGroups sshlogin
```

Then add your permitted SSH users to the group `sshlogin`, and restart the SSH service.

```
$ sudo adduser username sshlogin
```

```
$ sudo systemctl restart ssh.service
```

External user database authentication

Most enterprise networks require centralised authentication and access controls for all system resources. If you have configured your server to authenticate users against external databases, be sure to disable the user accounts both externally and locally. This way you ensure that local *fallbacks* authentication is not possible.

Firewall

The Linux kernel includes the **netfilter**⁹² subsystem, which is used to manipulate or decide the fate of network traffic headed into or through your server. All modern Linux firewall solutions use this system for packet filtering.

The kernel's packet filtering system would be of little use to administrators without a userspace interface to manage it. This is the purpose of the **iptables** utility: when a packet reaches your server, it will be handed off to the netfilter subsystem for acceptance, manipulation, or rejection based on the rules supplied to it from the userspace (via iptables). Thus, iptables is all you need to manage your firewall, if you're familiar with it, but many *frontends* are available to simplify the task. We'll take a look at the default frontend used in Ubuntu here.

ufw - Uncomplicated Firewall

The default firewall configuration tool for Ubuntu is ufw. Developed to ease iptables firewall configuration, ufw provides a user-friendly way to create an IPv4 or IPv6 host-based firewall.

ufw by default is initially disabled. From the *ufw(8)*⁹³ manual page:

ufw is not intended to provide complete firewall functionality via its command interface, but instead provides an easy way to add or remove simple rules. It is currently mainly used for host-based firewalls.

Enable or disable ufw

To enable ufw, run the following command from a terminal prompt:

```
$ sudo ufw enable
```

To disable it, you can use the following command:

```
$ sudo ufw disable
```

Open or close a port

To open a port (SSH in this case):

```
$ sudo ufw allow 22
```

Similarly, to close an opened port:

```
$ sudo ufw deny 22
```

⁹² <https://netfilter.org/>

⁹³ <https://manpages.ubuntu.com/manpages/resolute/man8/ufw.8.html>

Add or remove a rule

Rules can also be added using a **numbered** format:

```
$ sudo ufw insert 1 allow 80
```

To view the numbered format:

```
$ sudo ufw status numbered
```

To remove a rule, use delete followed by the rule:

```
$ sudo ufw delete deny 22
```

Allow access from specific hosts

It is possible to allow access from specific hosts or networks to a port. The following example allows SSH access from host 192.168.0.2 to any IP address on this host:

```
$ sudo ufw allow proto tcp from 192.168.0.2 to any port 22
```

Replace 192.168.0.2 with 192.168.0.0/24 to allow SSH access from the entire subnet.

The --dry-run option

Adding the --dry-run option to a ufw command will output the resulting rules, but not apply them. For example, the following is what would be applied if opening the HTTP port:

```
$ sudo ufw --dry-run allow http
```

```
*filter
:ufw-user-input - [0:0]
:ufw-user-output - [0:0]
:ufw-user-forward - [0:0]
:ufw-user-limit - [0:0]
:ufw-user-limit-accept - [0:0]
### RULES ###

### tuple ### allow tcp 80 0.0.0.0/0 any 0.0.0.0/0
-A ufw-user-input -p tcp --dport 80 -j ACCEPT

### END RULES ###
-A ufw-user-input -j RETURN
-A ufw-user-output -j RETURN
-A ufw-user-forward -j RETURN
-A ufw-user-limit -m limit --limit 3/minute -j LOG --log-prefix "[UFW LIMIT]: "
-A ufw-user-limit -j REJECT
-A ufw-user-limit-accept -j ACCEPT
COMMIT
Rules updated
```

Check the status

To see the firewall status, enter:

```
$ sudo ufw status
```

And for more verbose status information use:

```
$ sudo ufw status verbose
```

Note:

If the port you want to open or close is defined in `/etc/services`, you can use the port name instead of the number. In the above examples, replace 22 with `ssh`.

This is a quick introduction to using `ufw`. Please refer to the [ufw\(8\)](https://manpages.ubuntu.com/manpages/resolute/man8/ufw.8.html)⁹⁴ manual page for more information.

⁹⁴ <https://manpages.ubuntu.com/manpages/resolute/man8/ufw.8.html>

ufw application integration

Applications that open ports can include a ufw profile, which details the ports needed for the application to function properly. The profiles are kept in `/etc/ufw/applications.d`, and can be edited if the default ports have been changed.

To view which applications have installed a profile, run the following in a terminal:

```
$ sudo ufw app list
```

Similar to allowing traffic to a port, using an application profile is accomplished by entering:

```
$ sudo ufw allow Samba
```

An extended syntax is available as well:

```
$ ufw allow from 192.168.0.0/24 to any app Samba
```

Replace Samba and `192.168.0.0/24` with the application profile you are using and the IP range for your network.

Note:

There is no need to specify the **protocol** for the application, because that information is detailed in the profile. Also, note that the app name replaces the port number.

To view details about which ports and protocols, and so on, are defined for an application, enter:

```
$ sudo ufw app info Samba
```

Not all applications that require opening a network port come with ufw profiles, but if you have profiled an application and want the file to be included with the package, please file a bug against the package in Launchpad.

```
$ ubuntu-bug nameofpackage
```

IP masquerading

The purpose of IP masquerading is to allow machines with private, non-routable IP addresses on your network to access the Internet through the machine doing the masquerading. Traffic from your private network destined for the Internet must be manipulated for replies to be routable back to the machine that made the request.

To do this, the kernel must modify the **source** IP address of each packet so that replies will be routed back to it, rather than to the private IP address that made the request, which is impossible over the Internet. Linux uses **Connection Tracking** (*conntrack(8)*⁹⁵) to keep track of which connections belong to which machines and reroute each return packet accordingly.

⁹⁵ <https://manpages.ubuntu.com/manpages/resolute/man8/conntrack.8.html>

Traffic leaving your private network is thus “masqueraded” as having originated from your Ubuntu gateway machine. This process is referred to in Microsoft documentation as “Internet Connection Sharing”.

IP masquerading with ufw

IP masquerading can be achieved using custom ufw rules. This is possible because the current back-end for ufw is iptables - restore with the rules files located in `/etc/ufw/*.rules`. These files are a great place to add legacy iptables rules used without ufw, and rules that are more network gateway or bridge related.

The rules are split into two different files; rules that should be executed before ufw command line rules, and rules that are executed after ufw command line rules.

Enable packet forwarding

First, packet forwarding needs to be enabled in ufw. Two configuration files will need to be adjusted, so first, in `/etc/default/ufw` change the `DEFAULT_FORWARD_POLICY` to “ACCEPT”:

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Then, edit `/etc/ufw/sysctl.conf` and uncomment:

```
net/ipv4/ip_forward=1
```

Similarly, for IPv6 forwarding, uncomment:

```
net/ipv6/conf/default/forwarding=1
```

Add the configuration

Now add rules to the `/etc/ufw/before.rules` file. The default rules only configure the **filter** table, and to enable masquerading the **nat** table will need to be configured. Add the following to the top of the file, just after the header comments:

```
# nat Table rules
*nat
:POSTROUTING ACCEPT [0:0]

# Forward traffic from eth1 through eth0.
-A POSTROUTING -s 192.168.0.0/24 -o eth0 -j MASQUERADE

# don't delete the 'COMMIT' line or these nat table rules won't be processed
COMMIT
```

The comments are not strictly necessary, but it is considered good practice to document your configuration. Also, when modifying any of the rules files in `/etc/ufw`, make sure these lines are the last line for each table modified:

```
# don't delete the 'COMMIT' line or these rules won't be processed
COMMIT
```


For each **Table**, a corresponding COMMIT statement is required. In these examples only the nat and filter tables are shown, but you can also add rules for the raw and mangle tables.

Note:

In the above example, replace eth0, eth1, and 192.168.0.0/24 with the appropriate interfaces and IP range for your network.

Restart ufw

Finally, disable and re-enable ufw to apply the changes:

```
$ sudo ufw disable && sudo ufw enable
```

IP masquerading should now be enabled. You can also add any additional FORWARD rules to the /etc/ufw/before.rules. It is recommended that these additional rules be added to the ufw-before-forward chain.

IP masquerading with iptables

iptables can also be used to enable masquerading.

Similarly to ufw, the first step is to enable IPv4 packet forwarding by editing /etc/sysctl.conf and uncomment the following line:

```
net.ipv4.ip_forward=1
```

If you wish to enable IPv6 forwarding also uncomment:

```
net.ipv6.conf.default.forwarding=1
```

Next, run the sysctl command to enable the new settings in the configuration file:

```
$ sudo sysctl -p
```

IP masquerading can now be accomplished with a single iptables rule, which may differ slightly based on your network configuration:

```
$ sudo iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o ppp0 -j MASQUERADE
```

The above command assumes that your private address space is 192.168.0.0/16 and that your Internet-facing device is ppp0. The syntax is broken down as follows:

- -t nat – the rule is to go into the NAT table
- -A POSTROUTING – the rule is to be appended (-A) to the POSTROUTING chain
- -s 192.168.0.0/16 – the rule applies to traffic originating from the specified address space
- -o ppp0 – the rule applies to traffic scheduled to be routed through the specified network device

- -j MASQUERADE – traffic matching this rule is to “jump” (-j) to the MASQUERADE target to be manipulated as described above

Also, each chain in the filter table (the default table, and where most – or all – packet filtering occurs) has a default **policy** of ACCEPT, but if you are creating a firewall in addition to a gateway device, you may have set the policies to DROP or REJECT, in which case your masqueraded traffic needs to be allowed through the FORWARD chain for the above rule to work:

```
$ sudo iptables -A FORWARD -s 192.168.0.0/16 -o ppp0 -j ACCEPT
```

```
$ sudo iptables -A FORWARD -d 192.168.0.0/16 -m state \> --state  
ESTABLISHED,RELATED -i ppp0 -j ACCEPT
```

The above commands will allow all connections from your local network to the Internet and all traffic related to those connections to return to the machine that initiated them.

If you want masquerading to be enabled on reboot, which you probably do, edit `/etc/rc.local` and add any commands used above. For example add the first command with no filtering:

```
$ iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o ppp0 -j MASQUERADE
```

Logs

Firewall logs are essential for recognizing attacks, troubleshooting your firewall rules, and noticing unusual activity on your network. You must include logging rules in your firewall for them to be generated, though, and logging rules must come before any applicable terminating rule (a rule with a target that decides the fate of the packet, such as ACCEPT, DROP, or REJECT).

If you are using `ufw`, you can turn on logging by entering the following in a terminal:

```
$ sudo ufw logging on
```

To turn logging off in `ufw`, replace `on` with `off` in the above command:

If you are using `iptables` instead of `ufw`, run:

```
$ sudo iptables -A INPUT -m state --state NEW -p tcp --dport 80 \> -j LOG  
--log-prefix "NEW_HTTP_CONN: "
```

A request on port 80 from the local machine, then, would generate a log in *dmesg* that looks like this (single line split into 3 to fit this document):

```
[4304885.870000] NEW_HTTP_CONN: IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00  
SRC=127.0.0.1 DST=127.0.0.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=58288 DF PROTO=TCP  
SPT=53981 DPT=80 WINDOW=32767 RES=0x00 SYN URG=0
```

The above log will also appear in `/var/log/messages`, `/var/log/syslog`, and `/var/log/kern.log`. This behavior can be modified by editing `/etc/syslog.conf` appropriately or by installing and configuring `ulogd` and using the `ULOG` target instead of `LOG`. The `ulogd` daemon

is a userspace server that listens for logging instructions from the kernel – specifically for firewalls – and can log to any file you like, or even to a PostgreSQL or MySQL database. Making sense of your firewall logs can be simplified by using a log analysing tool such as logwatch, fwana1og, fwlogwatch, or lire.

Other tools

There are many tools available to help you construct a complete firewall without intimate knowledge of iptables. A command-line tool with plain-text configuration files, for example, is Shorewall⁹⁶; a powerful solution to help you configure an advanced firewall for any network.

Further reading

- The Ubuntu Firewall⁹⁷ wiki page contains information on the development of ufw
- Also, the `ufw(8)`⁹⁸ manual page contains some very useful information: `man ufw`
- See the packet-filtering how-to⁹⁹ for more information on using iptables
- The nat how-to¹⁰⁰ contains further details on masquerading
- The iptables how-to¹⁰¹ in the Ubuntu wiki is a great resource

AppArmor

AppArmor¹⁰² is an easy-to-use Linux Security Module implementation that restricts applications' capabilities and permissions with **profiles** that are set per-program. It provides mandatory access control (MAC) to supplement the more traditional UNIX model of discretionary access control (DAC).

In Ubuntu, AppArmor is installed and loaded by default – you can check this by running `aa-status`.

It uses **profiles** of an application to determine what files and permissions the application requires. Some packages will install their own profiles, and additional profiles can be found in the `apparmor-profiles` package.

⁹⁶ <https://shorewall.org/>

⁹⁷ <https://wiki.ubuntu.com/UncomplicatedFirewall>

⁹⁸ <https://manpages.ubuntu.com/manpages/resolute/man8/ufw.8.html>

⁹⁹ <https://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html>

¹⁰⁰ <https://www.netfilter.org/documentation/HOWTO/NAT-HOWTO.html>

¹⁰¹ <https://help.ubuntu.com/community/IptablesHowTo>

¹⁰² <https://apparmor.net/>

Install AppArmor Profiles

To install the `apparmor-profiles` package from a terminal prompt:

```
$ sudo apt install apparmor-profiles
```

AppArmor profiles have two modes of operation:

- **Complaining/Learning:** profile violations are permitted and logged. This is useful for testing and developing new profiles.
- **Enforced/Confined:** enforces profile policy in addition to logging the violation.

Using AppArmor

The optional `apparmor-utils` package contains command-line utilities you can use to change the AppArmor operation mode, find the status of a profile, create new profiles, etc.

AppArmor profiles are located in the `/etc/apparmor.d` directory. It also stores **abstractions** that can simplify profile authoring, such as `abstractions/base` that allows many shared libraries, writing logs to the journal, many pseudo-devices, receiving signals from unconfined processes, and many more things.

Common commands

- **apparmor_status** is used to view the current status of AppArmor profiles:

```
$ sudo apparmor_status
```

- **aa-complain** places a profile into complain mode:

```
$ sudo aa-complain /path/to/bin
```

- **aa-enforce** places a profile into enforce mode:

```
$ sudo aa-enforce /path/to/bin
```

- **apparmor_parser** is used to load a profile into the kernel. It can also be used to reload a currently-loaded profile using the `-r` option after modifying it to have the changes take effect. To reload a profile:

```
$ sudo apparmor_parser -r /etc/apparmor.d/profile.name
```

- **systemctl** can be used to reload all profiles:

```
$ sudo systemctl reload apparmor.service
```

Disabling or re-enabling a profile

The `/etc/apparmor.d/disable` directory can be used along with the `apparmor_parser -R` option to disable a profile:

```
$ sudo ln -s /etc/apparmor.d/profile.name /etc/apparmor.d/disable/
```

```
$ sudo apparmor_parser -R /etc/apparmor.d/profile.name
```

To re-enable a disabled profile, remove the symbolic link to the profile in `/etc/apparmor.d/disable/`, then load the profile using the `-a` option:

```
$ sudo rm /etc/apparmor.d/disable/profile.name
```

```
$ cat /etc/apparmor.d/profile.name | sudo apparmor_parser -a
```

Note:

Replace `profile.name` with the name of the profile you want to manipulate. Also, replace `/path/to/bin/` with the actual executable file path. For example, for the `ping` command use `/bin/ping`.

Profiles

AppArmor profiles are simple text files located in `/etc/apparmor.d/`. The files are named after the full path to the executable they profile, replacing the `/` with `.`.

For example `/etc/apparmor.d/bin.ping` is the AppArmor profile for the `/bin/ping` command.

There are two main type of rules used in profiles:

- **Path entries**, detailing which files an application can access in the file system.
- **Capability entries**, which determine what privileges a confined process is allowed to use.

As an example, take a look at `/etc/apparmor.d/bin.ping`:

```
#include <tunables/global>
/bin/ping flags=(complain) {
    #include <abstractions/base>
    #include <abstractions/consoles>
    #include <abstractions/nameservice>

    capability net_raw,
    capability setuid,
    network inet raw,

    /bin/ping mixr,
    /etc/modules.conf r,
}
```

Which can be broken down as follows:

- `#include <tunables/global>`: include statements from other files. This allows statements pertaining to multiple applications to be placed in a common file.
- `/bin/ping flags=(complain)`: path to the profiled program, also setting the mode to `complain`.
- `capability net_raw,:` allows the application access to the `CAP_NET_RAW` Posix.1e capability.
- `/bin/ping mixr,:` allows the application read and execute access to the file.

Note:

After editing a profile file the profile must be reloaded.

Create a Profile

- **Design a test plan:** Try to think about how the application should be exercised. The test plan should be divided into small test cases. Each test case should have a small description and list the steps to follow.

Some standard test cases are:

- Starting the program
- Stopping the program
- Reloading the program
- Testing all the commands supported by the init script

- **Generate the new profile:** Use `aa-genprof` to generate a new profile. From a terminal:

```
$ sudo aa-genprof executable
```

For example:

```
$ sudo aa-genprof slapd
```

- To get your new profile included in the `apparmor-profiles` package, [file a bug in Launchpad against the AppArmor package](#)¹⁰³:
 - Include your test plan and test cases
 - Attach your new profile to the bug

¹⁰³ <https://bugs.launchpad.net/ubuntu/+source/apparmor/+filebug>

Updating profiles

When the program is misbehaving, audit messages are sent to the log files. The program `aa-logprof` can be used to scan log files for AppArmor audit messages, review them and update the profiles. From a terminal:

```
$ sudo aa-logprof
```

Further pre-existing profiles

The packages `apparmor-profiles` and `apparmor-profiles-extra` ship some experimental profiles for AppArmor security policies. Do not expect these profiles to work out-of-the-box, but they can give you a head start when trying to create a new profile by starting off with a base that already exists.

These profiles are not considered mature enough to be shipped in `enforce` mode by default. Therefore, they are shipped in `complain` mode so that users can test them, choose which are desired, and help improve them upstream if needed.

Some even more experimental profiles carried by the package are placed in `/usr/share/doc/apparmor-profiles/extras/`

Checking and debugging denies

You will see in `dmesg` (and any log that collects kernel messages) if you have hit a **deny**. It is worth knowing that this will cover any access that was denied because it was not allowed, but explicit denies will put no message in your logs at all.

Examples might look like:

```
[1521056.552037] audit: type=1400 audit(1571868402.378:24425): apparmor="DENIED"
operation="open" profile="/usr/sbin/cups-browsed" name="/var/lib/libvirt/dnsmasq/"
pid=1128 comm="cups-browsed" requested_mask="r" denied_mask="r" fsuid=0 ouid=0
[1482106.651527] audit: type=1400 audit(1571829452.330:24323): apparmor="DENIED"
operation="sendmsg" profile="snap.lxd.lxc" pid=24115 comm="lxc" laddr=10.7.0.69
lport=48796 faddr=10.7.0.231 fport=445 family="inet" sock_type="stream" protocol=6
requested_mask="send" denied_mask="send"
```

That follows a generic structure starting with a timestamp, an audit tag and the category `apparmor="DENIED"`. From the following fields you can derive what was going on and why it was failing.

In the examples above that would be:

First example:

- operation: open (program tried to open a file)
- profile: `/usr/sbin/cups-browsed` (you'll find `/etc/apparmor.d/usr.bin.cups-browsed`)
- name: `/var/lib/libvirt/dnsmasq` (what it wanted to access)
- pid/comm: the program that triggered the access

- `requested_mask/denied_mask/fsuid/oid`: parameters of that open call

Second example:

- `operation`: `sendmsg` (program tried send via network)
- `profile`: `snap.lxd.lxc` (snaps are special, you'll find `/var/lib/snapd/apparmor/profiles/snap.lxd.lxc`)
- `pid/comm`: the program that triggered the access
- `laddr/lport/faddr/fport/family/sock_type/protocol`: parameters of the `sendmsg` call

That way you know in which profile and at what action you have to start if you consider either debugging or adapting the profiles.

Profile customization

Profiles are meant to provide security and so can't be too permissive. But often, a very special setup would work with a profile if it would *just allow this one extra access*. To handle that situation, there are three options:

- Modify the profile itself:
 - Always works, but has the drawback that profiles are in `/etc` and considered config files. So after modification on a related package update you might get a config file prompt. Worst case; depending on configuration, automatic updates might even override it and your custom rule is gone.
- Use tunables:
 - These provide variables that can be used in templates, for example if you want a custom `dir` considered as it would be a home directory. You could modify `/etc/apparmor.d/tunables/home`, which defines the base path rules used for home directories.
 - By design, these variables will only influence profiles that use them.
- Modify a local override:
 - To mitigate the drawbacks of above approaches, **local includes** were introduced, adding the ability to write arbitrary rules that not run into issues during upgrades that modify the packaged rule.
 - The files can be found in `/etc/apparmor.d/local/` and exist for the packages that are known to sometimes need slight tweaks for special setups.

Disabling or Re-enabling AppArmor

Starting with Ubuntu 24.04 and later, the AppArmor services are baked into the Ubuntu Kernel. In earlier versions of Ubuntu, you could disable AppArmor by not loading the service. However, it now requires setting a module parameter on the kernel command line to fully disable or re-enable AppArmor.

Disable AppArmor

Warning:

Disabling AppArmor reduces the security of your system! You should only disable AppArmor if you understand the security implications of disabling the service!

To disable AppArmor, you must do the following:

- Edit `/etc/default/grub` and add `apparmor=0` to `GRUB_CMDLINE_LINUX` in `/etc/default/grub`
- Run `sudo update-grub` to refresh the boot configuration
- Reboot the system.

Once rebooted, you can check the status of AppArmor with the following commands, and should see similar output to this:

```
$ sudo aa-status

apparmor module is loaded.
apparmor filesystem is not mounted.
```

```
$ systemctl status apparmor

❑ apparmor.service - Load AppArmor profiles
   Loaded: loaded (/usr/lib/systemd/system/apparmor.service; enabled; preset:
   enabled)
   Active: inactive (dead)
     Condition: start condition unmet at Tue 2025-01-14 08:45:04 UTC; 1min 20s ago
               └─ ConditionSecurity=apparmor was not met
        Docs: man:apparmor(7)
              https://gitlab.com/apparmor/apparmor/wikis/home/

Jan 14 08:45:04 n systemd[1]: apparmor.service - Load AppArmor profiles was skipped
because of an unmet condition check (ConditionSecurity=apparmor).
```

Re-enable AppArmor

- Remove the `apparmor=0` item from `GRUB_CMDLINE_LINUX` in `/etc/default/grub`.
- Run `sudo update-grub` to update your system boot configuration.
- Reboot your system.

You can check if AppArmor is then re-enabled with the following command, and you should see output similar to this:

```
$ sudo aa-status
```

```
apparmor module is loaded.  
119 profiles are loaded.  
24 profiles are in enforce mode.  
  /usr/bin/man  
...
```

Further reading

- See the [AppArmor Administration Guide](#)¹⁰⁴ for advanced configuration options.
- For details using AppArmor with other Ubuntu releases see the [AppArmor Community Wiki](#)¹⁰⁵ page.
- The [OpenSUSE AppArmor](#)¹⁰⁶ page is another introduction to AppArmor.
- (<https://wiki.debian.org/AppArmor>) is another introduction and basic how-to for AppArmor.
- A great place to get involved with the Ubuntu Server community and to ask for AppArmor assistance is the [#server channel on Matrix](#)¹⁰⁷. The [#security channel](#)¹⁰⁸ may also be of use.

Console security

It is difficult to defend against the damage that can be caused by someone with physical access to your environment, for example, due to theft of hard drives, power or service disruption, and so on.

Console security should be addressed as one component of your overall physical security strategy. A locked “screen door” may deter a casual criminal, or at the very least slow down a determined one, so it is still advisable to perform basic precautions with regard to console security.

Disable Ctrl+Alt+Delete

Anyone with physical access to the keyboard can use the Ctrl + Alt + Delete key combination to reboot the server without having to log on. While someone could also simply unplug the power source, you should still prevent the use of this key combination on a production server. This forces an attacker to take more drastic measures to reboot the server, and will prevent accidental reboots at the same time.

To disable the reboot action taken by pressing the Ctrl + Alt + Delete key combination, run the following two commands:

¹⁰⁴ http://www.novell.com/documentation/apparmor/apparmor201_sp10_admin/?page=/documentation/apparmor/apparmor201_sp10_admin/data/book_apparmor_admin.html

¹⁰⁵ <https://help.ubuntu.com/community/AppArmor>

¹⁰⁶ https://en.opensuse.org/SDB:AppArmor_geeks

¹⁰⁷ <https://matrix.to/#/#server:ubuntu.com>

¹⁰⁸ <https://matrix.to/#/#security:ubuntu.com>

```
$ sudo systemctl mask ctrl-alt-del.target
```

```
$ sudo systemctl daemon-reload
```

Note:

To re-enable this feature, use:

```
$ sudo systemctl unmask ctrl-alt-del.target
```

Disable Magic SysRq keys

The Linux kernel's [Magic SysRq](#)¹⁰⁹ feature allows keyboard shortcuts to send low-level commands directly to the kernel, bypassing any running applications and without requiring a login. By default, Ubuntu enables a subset of these functions, including the ability to immediately reboot (Alt + SysRq + b) or power off (Alt + SysRq + o) the machine. As with Ctrl + Alt + Delete, this is worth addressing as part of your physical access hardening.

Note:

On most keyboards, SysRq is labeled as PrtScn.

You can check which functions are currently enabled:

```
$ sysctl kernel.sysrq  
  
kernel.sysrq = 176
```

The default value of 176 enables filesystem sync (16), remounting read-only (32), and reboot/poweroff (128). To disable all SysRq functions, create an override file in `/etc/sysctl.d/` with a higher sort order than the default SysRq configuration file `10-magic-sysrq.conf`:

```
$ echo "kernel.sysrq = 0" | sudo tee /etc/sysctl.d/99-disable-sysrq.conf
```

```
$ sudo sysctl --system
```

The 99- prefix ensures this file is processed after `10-magic-sysrq.conf`, and the setting persists across reboots.

¹⁰⁹ <https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html>

Note:

A list of all of the available SysRq values and a brief description explaining what each value does can be found by examining `/etc/sysctl.d/10-magic-sysrq.conf`.

Beyond that, further explanation can be found by referencing the [SysRq section on the Linux Kernel documentation website](#)¹¹⁰.

¹¹⁰ <https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html>

Verify the change has taken effect:

```
$ sysctl kernel.sysrq  
  
kernel.sysrq = 0
```

Note:

To re-enable the default SysRq functions, remove the override file and reload:

```
$ sudo rm /etc/sysctl.d/99-disable-sysrq.conf
```

```
$ sudo sysctl --system
```

TPM-based LUKS decryption with Clevis

[Clevis](#)¹¹¹ is a pluggable framework for automated decryption. When combined with Trusted Platform Module ([TPM](#)) and Full Disk Encryption ([FDE](#)) via Linux Unified Key Setup ([LUKS](#)), Clevis can automatically unlock encrypted drives during boot without user intervention. The Clevis tooling is currently provided via the universe archive. While functional, it serves as a community-supported fallback for TPM-bound encryption until a fully integrated server FDE solution lands in a future LTS release.

This guide targets systems that use [dracut\(8\)](#)¹¹² as the initramfs generator.

Warning:

Keep at least one well-known LUKS passphrase for recovery. TPM measurements can change after firmware updates, Secure Boot state changes, or some [bootloader](#) changes, which can prevent automatic unlock until you rebind Clevis.

¹¹¹ <https://github.com/latchset/clevis>

¹¹² <https://manpages.ubuntu.com/manpages/resolute/man8/dracut.8.html>

Prerequisites

- An Ubuntu Server installation with LVM over LUKS, with Dracut as the initramfs generator
- A system with a TPM 2.0 module
- Root or sudo privileges

Install Clevis and Dracut integration

Install the necessary packages:

```
$ sudo apt update && sudo apt install clevis clevis-tpm2 clevis-dracut clevis-luks
```

Binding a LUKS volume

Before Dracut can auto-unlock a drive, the drive must be bound to the TPM. You can use Clevis to add a new key to the LUKS header, sealed against the TPM.

First, identify the encrypted partition before binding:

```
$ lsblk -f
```

Look for the partition with FSTYPE set to `crypto_LUKS`, then use that path in the following command as `<encrypted_partition>`.

Bind against Platform Configuration Register (PCR) 7, which tracks secure boot state. If your environment requires different trust guarantees, choose PCR values that match your threat model.

```
$ sudo clevis luks bind -d <encrypted_partition> tpm2 '{"pcr_ids": "7"}'
```

You will be prompted to enter the LUKS passphrase you created during installation. Clevis will generate a new cryptographic secret and store it in a new keyslot. It will also create a corresponding token linked to this new keyslot.

Updating the initial ramdisk

The `clevis-dracut` package provides the necessary Dracut modules to include Clevis decryption hooks in the early boot environment. After binding your root disk, you must regenerate the initial ramdisk.

To force a rebuild of the initial ramdisk for the current kernel, use:

```
$ sudo dracut -f
```

You can verify that the Clevis modules were successfully included with `lsinitrd(1)`¹¹³ by inspecting the generated image:

```
$ sudo lsinitrd | grep '^clevis'

clevis
clevis-pin-null
clevis-pin-sss
clevis-pin-tang
clevis-pin-tpm2
```

You can also verify that a Clevis token exists in the LUKS metadata – it should show a TPM pin bound to the device:

```
$ sudo clevis luks list -d <encrypted_partition>

2: tpm2 '{"hash":"sha256","key":"ecc"}'
```

For a lower-level check, inspect LUKS token metadata with `cryptsetup(8)`¹¹⁴, which should show the Clevis token and the keyslot it is bound to:

```
$ sudo cryptsetup luksDump <encrypted_partition> | grep -A2 -i Tokens

Tokens:
 0: clevis
    Keyslot: 2
```

After rebooting, confirm the root volume unlocks automatically and the system reaches the login prompt without asking for the LUKS passphrase.

Unlocking secondary disks

If your server has secondary encrypted data disks, you do not need to use dracut to unlock them, as they are unlocked later in the boot process by `systemd(1)`¹¹⁵.

To setup automatic unlocking, you must first bind the secondary disk to the TPM. Replace `<secondary_encrypted_partition>` with your secondary encrypted partition.

```
$ sudo clevis luks bind -d <secondary_encrypted_partition> tpm2 '{"pcr_ids":
"7"}'
```

Then, get the UUID of the locked LUKS partition:

```
$ blkid -s UUID -o value <secondary_encrypted_partition>
```

¹¹³ <https://manpages.ubuntu.com/manpages/resolute/man1/lsinitrd.1.html>

¹¹⁴ <https://manpages.ubuntu.com/manpages/resolute/man8/cryptsetup.8.html>

¹¹⁵ <https://manpages.ubuntu.com/manpages/resolute/man1/systemd.1.html>

Add the disk to `crypttab(5)`¹¹⁶, using `none` for the password file so `systemd(1)`¹¹⁷ automatically intercepts the prompt and recognizes the Clevis binding.

```
data_vol UUID=<UUID> none luks
```

The first field in the `/etc/crypttab` entry is the name used for the mapped device. In the example above the name is `data_vol`, so when the LUKS volume is unlocked the decrypted mapper device is exposed at `/dev/mapper/data_vol`. Add that device (`/dev/mapper/data_vol`) to `fstab(5)`¹¹⁸ as usual.

Removing a TPM binding

If you need to remove a Clevis binding (for example, if you are decommissioning a server or moving the drive), you can unbind it. First, list the active Clevis tokens to find the correct LUKS slot:

```
$ sudo clevis luks list -d <encrypted_partition>
```

Then, unbind the specific slot (replacing 1 with the slot number identified above).

```
$ sudo clevis luks unbind -d <encrypted_partition> -s 1
```

Authentication

These tools are particularly useful for more advanced or complex setups.

- [Kerberos](#) (page 182) is a network authentication protocol providing identity verification for distributed systems
- [Network user authentication with SSSD](#) (page 204) handles authentication, user/group information and authorisation from disparate network sources
- [Smart card authentication](#) (page 223) provides a physical authentication method

Kerberos

This section assumes you have some familiarity with the terms and concepts used in Kerberos.

You will need a properly configured DNS server set up before you begin. See the [Domain Name Service \(DNS\)](#) (page 299) page for instructions on how to set this up.

We recommend following this set of how-to guides in the order presented for best results.

¹¹⁶ <https://manpages.ubuntu.com/manpages/resolute/man5/crypttab.5.html>

¹¹⁷ <https://manpages.ubuntu.com/manpages/resolute/man1/systemd.1.html>

¹¹⁸ <https://manpages.ubuntu.com/manpages/resolute/man5/fstab.5.html>

How to install a Kerberos server

For this discussion, we will create an MIT Kerberos domain with the following features (edit them to fit your needs):

- **Realm:** EXAMPLE.COM
- **Primary KDC:** kdc01.example.com
- **Secondary KDC:** kdc02.example.com
- **User principal:** ubuntu
- **Admin principal:** ubuntu/admin

Prerequisites

Before installing the Kerberos server, a properly configured [DNS](#) server is needed for your domain. Since the Kerberos realm (by convention) matches the domain name, this section uses the EXAMPLE.COM domain configured in the primary server section of the [DNS documentation](#) (page 299).

Also, Kerberos is a time sensitive protocol. If the local system time between a client machine and the server differs by more than five minutes (by default), the workstation will not be able to authenticate. To correct the problem all hosts should have their time synchronized using the same Network Time Protocol (NTP) server. Check out the [NTP chapter](#) (page 937) for more details.

Install the Kerberos packages

The first step in creating a Kerberos realm is to install the krb5-kdc and krb5-admin-server packages. From a terminal enter:

```
$ sudo apt install krb5-kdc krb5-admin-server
```

You will be asked at the end of the install to supply the [hostname](#) for the Kerberos and Admin servers for the realm, which may or may not be the same server. Since we are going to create the realm, and thus these servers, type in the full hostname of this server.

Note:

By default the realm name will be domain name of the Key Distribution Center (KDC) server.

Next, create the new realm with the kdb5_newrealm utility:

```
$ sudo krb5_newrealm
```

It will ask you for a database password, which is used to encrypt the local database. Chose a secure password: its strength is not verified for you.

Configure the Kerberos server

The questions asked during installation are used to configure the `/etc/krb5.conf` and `/etc/krb5kdc/kdc.conf` files. The former is used by the Kerberos 5 libraries, and the latter configures the KDC. If you need to adjust the KDC settings, edit the file and restart the `krb5-kdc` daemon. If you need to reconfigure Kerberos from scratch, perhaps to change the realm name, you can do so by typing:

```
$ sudo dpkg-reconfigure krb5-kdc
```

Note:

The manual page for `krb5.conf` is in the `krb5-doc` package.

Let's create our first principal. Since there is no principal create yet, we need to use `kadmin.local`, which uses a local UNIX socket to talk to the KDC, and requires root privileges:

```
$ sudo kadmin.local

Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local: addprinc ubuntu
WARNING: no policy specified for ubuntu@EXAMPLE.COM; defaulting to no policy
Enter password for principal "ubuntu@EXAMPLE.COM":
Re-enter password for principal "ubuntu@EXAMPLE.COM":
Principal "ubuntu@EXAMPLE.COM" created.
kadmin.local: quit
```

To be able to use `kadmin` remotely, we should create an **admin principal**. Convention suggests it should be an **admin instance**, as that also makes creating a generic Access Control List (ACL) easier. Let's create an admin instance for the ubuntu principal:

```
$ sudo kadmin.local

Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local: addprinc ubuntu/admin
WARNING: no policy specified for ubuntu/admin@EXAMPLE.COM; defaulting to no policy
Enter password for principal "ubuntu/admin@EXAMPLE.COM":
Re-enter password for principal "ubuntu/admin@EXAMPLE.COM":
Principal "ubuntu/admin@EXAMPLE.COM" created.
kadmin.local: quit
```

Next, the new admin principal needs to have the appropriate ACL permissions. The permissions are configured in the `/etc/krb5kdc/kadm5.acl` file:

```
ubuntu/admin@EXAMPLE.COM
```

*

You can also use a more generic form for this ACL:

```
*/admin@EXAMPLE.COM
```

*

The above will grant all privileges to any admin instance of a principal. See the `kadm5`.

`acl(5)`¹¹⁹ manual page for details.

Now restart the `krb5-admin-server` for the new ACL to take effect:

```
$ sudo systemctl restart krb5-admin-server.service
```

The new user principal can be tested using the `kinit` utility:

```
$ kinit ubuntu/admin  
Password for ubuntu/admin@EXAMPLE.COM:
```

After entering the password, use the `klist` utility to view information about the Ticket Granting Ticket (TGT):

```
$ klist  
  
Ticket cache: FILE:/tmp/krb5cc_1000  
Default principal: ubuntu/admin@EXAMPLE.COM  
  
Valid starting    Expires          Service principal  
04/03/20 19:16:57 04/04/20 05:16:57 krbtgt/EXAMPLE.COM@EXAMPLE.COM  
        renew until 04/04/20 19:16:55
```

Where the cache filename `krb5cc_1000` is composed of the prefix `krb5cc_` and the user id (UID), which in this case is 1000.

`kinit` will inspect `/etc/krb5.conf` to find out which KDC to contact, and the corresponding address. The KDC can also be found via DNS lookups for special TXT and SRV records. You can add these records to your `example.com` DNS zone:

```
_kerberos._udp.EXAMPLE.COM.    IN SRV 1 0 88 kdc01.example.com.  
_kerberos._tcp.EXAMPLE.COM.    IN SRV 1 0 88 kdc01.example.com.  
_kerberos._udp.EXAMPLE.COM.    IN SRV 10 0 88 kdc02.example.com.  
_kerberos._tcp.EXAMPLE.COM.    IN SRV 10 0 88 kdc02.example.com.  
_kerberos-adm._tcp.EXAMPLE.COM. IN SRV 1 0 749 kdc01.example.com.  
_kpasswd._udp.EXAMPLE.COM.     IN SRV 1 0 464 kdc01.example.com.
```

See the [DNS chapter](#) (page 299) for detailed instructions on setting up DNS.

A very quick and useful way to troubleshoot what `kinit` is doing is to set the environment variable `KRB5_TRACE` to a file, or `stderr`, and it will show extra information. The output is quite verbose:

```
$ KRB5_TRACE=/dev/stderr kinit ubuntu/admin  
  
[2898] 1585941845.278578: Getting initial credentials for ubuntu/admin@EXAMPLE.COM  
[2898] 1585941845.278580: Sending unauthenticated request  
[2898] 1585941845.278581: Sending request (189 bytes) to EXAMPLE.COM  
[2898] 1585941845.278582: Resolving hostname kdc01.example.com  
(...)
```

Your new Kerberos realm is now ready to authenticate clients.

¹¹⁹ <https://manpages.ubuntu.com/manpages/resolute/man5/kadm5.acl.5.html>

How to configure Kerberos service principals

The specific steps to enable Kerberos for a service can vary, but in general both of the following are needed:

- A principal for the service – usually `service/host@REALM`
- A keytab accessible to the service wherever it's running – usually in `/etc/krb5.keytab`

For example, let's create a principal for an LDAP service running on the `ldap-server.example.com` host:

```
ubuntu@ldap-server:~$ sudo kadmin -p ubuntu/admin

Authenticating as principal ubuntu/admin with password.
Password for ubuntu/admin@EXAMPLE.COM:
kadmin: addprinc -randkey ldap/ldap-server.example.com
No policy specified for ldap/ldap-server.example.com@EXAMPLE.COM; defaulting to no
policy
Principal "ldap/ldap-server.example.com@EXAMPLE.COM" created.
```

Let's dig a bit into what is happening here:

- The `kadmin` command is being run on the `ldap-server` machine, not on the Key Distribution Center (KDC). We are using `kadmin` remotely.
- It's being run with `sudo`. The reason for this will become clear later.
- We are logged in on the server as `ubuntu`, but specifying an `ubuntu/admin` principal. Remember the `ubuntu` principal has no special privileges.
- The name of the principal we are creating follows the pattern `service/hostname`.
- In order to select a random secret, we pass the `-randkey` parameter. Otherwise we would be asked to type in a password.

With the principal created, we need to extract the key from the KDC and store it in the `ldap-server` host, so that the `ldap` service can use it to authenticate itself with the KDC. Still in the same `kadmin` session:

```
kadmin: ktadd ldap/ldap-server.example.com
Entry for principal ldap/ldap-server.example.com with kvno 2, encryption type aes256-cts-
hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.
Entry for principal ldap/ldap-server.example.com with kvno 2, encryption type aes128-cts-
hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.
```

This is why we needed to run `kadmin` with `sudo`: so that it can write to `/etc/krb5.keytab`. This is the system keytab file, which is the default file for all keys that might be needed for services on this host, and we can list them with `klist`. Back in the shell:

```
$ sudo klist -k
```

```
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
-----
```

```
2 ldap/ldap-server.example.com@EXAMPLE.COM
2 ldap/ldap-server.example.com@EXAMPLE.COM
```

If you don't have the `kadmin` utility on the target host, one alternative is to extract the keys on a different host and into a different file, and then transfer this file *securely* to the target server. For example:

```
kadmin: ktadd -k /home/ubuntu/ldap.keytab ldap/ldap-server.example.com
Entry for principal ldap/ldap-server.example.com with kvno 3, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/home/ubuntu/ldap.keytab.
Entry for principal ldap/ldap-server.example.com with kvno 3, encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/home/ubuntu/ldap.keytab.
```

Note:

Notice how the kvno changed from 2 to 3 in the example above, when using `ktadd` a second time? This is the key version, and it invalidated the previously extracted key with kvno 2. Every time a key is extracted with `ktadd`, its version is bumped and that invalidates the previous ones!

In this case, as long as the target location is writable, you don't even have to run `kadmin` with `sudo`.

Then use `scp` to transfer it to the target host:

```
$ scp /home/ubuntu/ldap.keytab ldap-server.example.com:
```

And over there copy it to `/etc/krb5.keytab`, making sure it's mode `0600` and owned by `root:root`.

Kerberos encryption types

Encryption is at the heart of Kerberos, and it supports multiple cryptographic algorithms. The default choices are good enough for most deployments, but specific situations might need to tweak these settings.

This document will explain the basic configuration parameters of Kerberos that control the selection of encryption algorithms used in a Kerberos deployment.

Server-side configuration

There are two main server-side configuration parameters that control the encryption types used on the server for its database and its collection of principals. Both exist in `/etc/krb5kdc/kdc.conf` inside the `[realms]` section and are as follows:

- `master_key_type` Specifies the key type of the key. This is used to encrypt the database, and the default is `aes256-cts-hmac-sha1-96`.
- `supported_etypes` Specifies the default key/salt combinations of principals for this realm. The default is `aes256-cts-hmac-sha1-96:normal aes128-cts-hmac-sha1-96:normal`, and the encryption types should be listed in order of preference.

Possible values for the encryption algorithms are listed in the [MIT documentation on encryption types](https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_files/kdc_conf.html#encryption-types)¹²⁰, and the salt types can be seen in the [MIT keysalt lists](https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_files/kdc_conf.html#keysalt-lists)¹²¹.

Here is an example showing the default values (other settings removed for brevity):

```
[realms]
EXAMPLE.INTERNAL = {
    (...)
    master_key_type = aes256-cts
    supported_etypes = aes256-cts-hmac-sha1-96:normal aes128-cts-hmac-sha1-96:normal
    (...)
}
```

The key is created once per realm, when the realm is bootstrapped. That is usually done with the `krb5_newrealm` tool (see [how to install a Kerberos server](#) (page 183) for details). You can check the key type with either of these commands on the KDC server:

```
$ sudo kadmin.local

kadmin.local: getprinc K/M
Principal: K/M@EXAMPLE.INTERNAL
(...)
Number of keys: 1
Key: vno 1, aes256-cts-hmac-sha1-96
(...)
```

```
$ sudo klist -ke /etc/krb5kdc/stash

Keytab name: FILE:/etc/krb5kdc/stash
KVNO Principal
-----
  1 K/M@EXAMPLE.INTERNAL (aes256-cts-hmac-sha1-96)
```

When a new Kerberos principal is created through the `kadmin` service (via the `kadmin` or `kadmin.local` utilities), the types of encryption keys it will get are controlled via the `supported_etypes` configuration parameter.

For example, let's create an `ubuntu` principal, and check the keys that were created for it (output abbreviated):

¹²⁰ https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_files/kdc_conf.html#encryption-types

¹²¹ https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_files/kdc_conf.html#keysalt-lists

```
$ sudo kadmin.local

Authenticating as principal root/admin@EXAMPLE.INTERNAL with password.
kadmin.local: addprinc ubuntu
No policy specified for ubuntu@EXAMPLE.INTERNAL; defaulting to no policy
Enter password for principal "ubuntu@EXAMPLE.INTERNAL":
Re-enter password for principal "ubuntu@EXAMPLE.INTERNAL":
Principal "ubuntu@EXAMPLE.INTERNAL" created.

kadmin.local: getprinc ubuntu
Principal: ubuntu@EXAMPLE.INTERNAL
(...)
Number of keys: 2
Key: vno 1, aes256-cts-hmac-sha1-96
Key: vno 1, aes128-cts-hmac-sha1-96
(...)
```

Two keys were created for the ubuntu principal, following the default setting of supported_ectypes in kdc.conf for this realm.

Note:

The server config supported_ectypes has the *default* list of key types that are created for a principal. This list applies to the moment when that principal is **created** by kadmin. Changing that setting after the fact won't affect the keys that the principal in question has after that event. In particular, principals can be created with specific key types regardless of the supported_ectypes setting. See the -e parameter for the `kadmin add_principal` command¹²².

¹²² <https://web.mit.edu/kerberos/krb5-latest/doc/admin/database.html>

If we had supported_ectypes set to aes256-sha2:normal aes128-sha2:normal camellia256-cts:normal in kdc.conf, then the ubuntu principal would get three key types:

```
kadmin.local: getprinc ubuntu
Principal: ubuntu@EXAMPLE.INTERNAL
(...)
Number of keys: 3
Key: vno 1, aes256-cts-hmac-sha384-192
Key: vno 1, aes128-cts-hmac-sha256-128
Key: vno 1, camellia256-cts-cmac
```

Note:

Bootstrapping a new Kerberos realm via the `krb5_newrealm` command also creates some system principals required by Kerberos, such as kadmin/admin, kadmin/changepw and others. They will all also get the same number of keys each: one per encryption type in supported_ectypes.

Client-side configuration

When we say “client-side”, we really mean “applications linked with the Kerberos libraries”. These live on the server too, so keep that in mind.

The encryption types supported by the Kerberos libraries are defined in the `/etc/krb5.conf` file, inside the `[libdefaults]` section, via the `permitted_enctypes` parameter.

Example:

```
[libdefaults]
(...)
permitted_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96
```

This parameter contains a space-separated list of encryption type names, in order of preference. Default value: `aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 aes256-cts-hmac-sha384-192 aes128-cts-hmac-sha256-128 camellia256-cts-cmac camellia128-cts-cmac`.

Possible values for the encryption algorithms are listed in [the MIT documentation](https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_files/krb5_conf.html#libdefaults)¹²³ (same ones as for the KDC).

Note:

There are more encryption-related parameters in `krb5.conf`, but most take their defaults from `permitted_enctypes`. See the [MIT libdefaults documentation](https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_files/krb5_conf.html#libdefaults)¹²⁴ for more information.

¹²⁴ https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_files/krb5_conf.html#libdefaults

Putting it all together

When a client performs Kerberos authentication and requests a ticket from the KDC, the encryption type used in that ticket is decided by picking the common set of:

- The encryption types supported by the server for that principal
- The encryption types supported by the client

If there is no common algorithm between what the client accepts, and what the server has to offer for that specific principal, then `kinit` will fail.

For example, if the principal on the server has:

```
kadmin.local: getprinc ubuntu
Principal: ubuntu@EXAMPLE.INTERNAL
(...)
Number of keys: 2
Key: vno 1, aes256-cts-hmac-sha384-192
Key: vno 1, aes128-cts-hmac-sha256-128
```

And the client's `krb5.conf` has:

¹²³ https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_files/kdc_conf.html#encryption-types

```
permitted_etypes = aes256-sha1 aes128-sha1
```

Then kinit will fail, because the client only supports sha1 variants, and the server only has sha2 to offer for that particular principal the client is requesting:

```
$ kinit ubuntu
kinit: Generic error (see e-text) while getting initial credentials
```

The server log (`journalctl -u krb5-admin-server.service`) will have more details about the error:

```
Apr 19 19:31:49 j-kdc krb5kdc[8597]: AS_REQ (2 etypes {aes256-cts-hmac-sha1-96(18),
aes128-cts-hmac-sha1-96(17)}) fd42:78f4:b1c4:3964:216:3eff:feda:118c: GET_LOCAL_TGT:
ubuntu@EXAMPLE.INTERNAL for krbtgt/EXAMPLE.INTERNAL@EXAMPLE.INTERNAL, No matching key in
entry having a permitted enctype
```

This log says that there was an AS-REQ request which accepted two encryption types, but there was no matching key type on the server database for that principal.

Changing encryption types

Changing encryption types of an existing Kerberos realm is no small task. Just changing the configuration settings won't recreate existing keys, nor add new ones. The modifications have to be done in incremental steps.

MIT Kerberos has a [guide on updating encryption types](#)¹²⁵ that covers many scenarios, including deployments with multiple replicating servers:

Further reading

- [Encryption types in MIT Kerberos](#)¹²⁶
- [krb5.conf encryption related configurations options](#)¹²⁷
- [Migrating away from older encryption types](#)¹²⁸
- [kdc.conf\(5\)](#)¹²⁹ manual page
- [krb5.conf\(5\)](#)¹³⁰ manual page
- [Kerberos V5 concepts](#)¹³¹

¹²⁵ <https://web.mit.edu/kerberos/krb5-latest/doc/admin/etypes.html#migrating-away-from-older-encryption-types>

¹²⁶ <https://web.mit.edu/kerberos/krb5-latest/doc/admin/etypes.html>

¹²⁷ <https://web.mit.edu/kerberos/krb5-latest/doc/admin/etypes.html#configuration-variables>

¹²⁸ <https://web.mit.edu/kerberos/krb5-latest/doc/admin/etypes.html#migrating-away-from-older-encryption-types>

¹²⁹ <https://manpages.ubuntu.com/manpages/resolute/man5/kdc.conf.5.html>

¹³⁰ <https://manpages.ubuntu.com/manpages/resolute/man5/krb5.conf.5.html>

¹³¹ <https://web.mit.edu/kerberos/krb5-latest/doc/basic/index.html>

How to set up secondary KDC

Once you have one Key Distribution Center (KDC) on your network, it is good practice to have a secondary KDC in case the primary becomes unavailable.

Also, if you have Kerberos clients that are on different networks (possibly separated by routers using NAT), it is wise to place a secondary KDC in each of those networks.

Note:

The native replication mechanism explained here relies on a cron job; it essentially dumps the DB on the primary and loads it back up on the secondary. You may want to take a look at using the `kldap` backend, which can use the OpenLDAP replication mechanism. This is explained in [How to set up Kerberos with OpenLDAP backend](#) (page 196).

Install the required packages

First, install the packages, and when asked for the Kerberos and Admin server names enter the name of the Primary KDC:

```
$ sudo apt install krb5-kdc krb5-admin-server
```

Once you have installed the packages, create the host principals for both KDCs. From a terminal prompt, enter:

```
$ kadmin -q "addprinc -randkey host/kdc01.example.com" > kadmin -q "addprinc  
-randkey host/kdc02.example.com"
```

Note:

The `kadmin` command defaults to using a principal like `username/admin@EXAMPLE.COM`, where `username` is your current shell user. If you need to override that, use `-p <principal-you-want>`.

Extract the **key file** for the `kdc02` principal, which is the server we are on:

```
$ sudo kadmin -p ubuntu/admin -q "ktadd host/kdc02.example.com"
```

Next, there needs to be a `kpropd.acl` file on each KDC that lists all KDCs for the realm. For example, on both the **primary and secondary KDC**, create `/etc/krb5kdc/kpropd.acl`:

```
host/kdc01.example.com@EXAMPLE.COM  
host/kdc02.example.com@EXAMPLE.COM
```

Note:

It's customary to allow both KDCs because one may want to switch their roles if one goes bad. For such an eventuality, both are already listed here.

Create an empty database on the **secondary KDC**:

```
$ sudo kdb5_util create -s
```

Now install kprop daemon, which listens for connections from the kprop utility from the **primary KDC**:

```
$ sudo apt install krb5-kpropd
```

The service will be running immediately after installation.

From a terminal on the **primary KDC**, create a dump file of the principal database:

```
$ sudo kdb5_util dump /var/lib/krb5kdc/dump
```

Still on the **Primary KDC**, extract its **key**:

```
$ sudo kadmin.local -q "ktadd host/kdc01.example.com"
```

On the **primary KDC**, run the kprop utility to push the database dump made before to the secondary KDC:

```
$ sudo kprop -r EXAMPLE.COM -f /var/lib/krb5kdc/dump kdc02.example.com  
Database propagation to kdc02.example.com: SUCCEEDED
```

Note the SUCCEEDED message, which signals that the propagation worked. If there is an error message, check /var/log/syslog on the secondary KDC for more information.

You may also want to create a cron job to periodically update the database on the **secondary KDC**. For example, the following will push the database every hour:

```
# m h dom mon dow   command  
0 * * * * root /usr/sbin/kdb5_util dump /var/lib/krb5kdc/dump && /usr/sbin/kprop -r  
EXAMPLE.COM -f /var/lib/krb5kdc/dump kdc02.example.com
```

Finally, start the krb5-kdc daemon on the **secondary KDC**:

```
$ sudo systemctl start krb5-kdc.service
```

Note:

The secondary KDC does not run an admin server, since it's a read-only copy.

From now on, you can specify both KDC servers in /etc/krb5.conf for the EXAMPLE.COM realm, in any host participating in this realm (including kdc01 and kdc02), but remember that there can only be one admin server and that's the one running on kdc01:

```
[realms]  
EXAMPLE.COM = {  
    kdc = kdc01.example.com
```

(continues on next page)

(continued from previous page)

```
}  
kdc = kdc02.example.com  
admin_server = kdc01.example.com
```

The **secondary KDC** should now be able to issue tickets for the realm. You can test this by stopping the `krb5-kdc` daemon on the primary KDC, then using `kinit` to request a ticket. If all goes well you should receive a ticket from the secondary KDC. Otherwise, check `/var/log/syslog` and `/var/log/auth.log` on the secondary KDC.

How to set up basic workstation authentication

In this section we'll look at configuring a Linux system as a Kerberos client. This will allow access to any "Kerberised" services once a user has successfully logged into the system.

Note that Kerberos alone is not enough for a user to exist in a Linux system. We cannot just point the system at a Kerberos server and expect all the Kerberos principals to be able to *log in* on the Linux system, simply because these users do not *exist* locally.

Kerberos only provides authentication: it doesn't know about user groups, Linux *UIDs* and *GIDs*, home directories, etc. Normally, another network source is used for this information, such as an LDAP or Windows server, and, in the old days, NIS was used for that as well.

Set up a Linux system as a Kerberos client

If you have local users matching the principals in a Kerberos realm, and just want to switch the authentication from local to remote using Kerberos, you can follow this section. This is not a very usual scenario, but serves to highlight the separation between user authentication and user information (full name, UID, GID, home directory, groups, etc). If you just want to be able to grab tickets and use them, it's enough to install `krb5-user` and run `kinit`.

We are going to use `sssd` with a trick so that it will fetch the user information from the local system files, instead of a remote source which is the more common case.

Install the required packages

To install the packages enter the following in a terminal prompt:

```
$ sudo apt install krb5-user sssd-krb5
```

You will be prompted for the addresses of your KDCs and admin servers. If you have followed our [how to install a Kerberos server](#) (page 183) and [how to set up a secondary KDC](#) (page 192) guides, the KDCs will be (space separated):

```
kdc01.example.com kdc02.example.com
```

And the admin server will be:

```
kdc01.example.com
```

Remember that kdc02 is a read-only copy of the primary KDC, so it doesn't run an admin server.

Note:

If you have added the appropriate SRV records to [DNS](#), none of those prompts will need answering.

Configure Kerberos

If you missed the questions earlier, you can reconfigure the package to fill them in again:

```
$ sudo dpkg-reconfigure krb5-config
```

You can test the Kerberos configuration by requesting a ticket using the kinit utility. For example:

```
$ kinit ubuntu
Password for ubuntu@EXAMPLE.COM:
```

Note that kinit doesn't need the principal to exist as a local user in the system. In fact, you can kinit any principal you want. If you don't specify one, then the tool will use the username of whoever is running kinit.

Configure sssd

The only remaining configuration now is for sssd. Create the file /etc/sss/sss.conf with the following content:

```
[sss]
config_file_version = 2
domains = example.com

[pam]

[domain/example.com]
id_provider = proxy
proxy_lib_name = files
auth_provider = krb5
krb5_server = kdc01.example.com,kdc02.example.com
krb5_kpasswd = kdc01.example.com
krb5_realm = EXAMPLE.COM
```

The above configuration will use Kerberos for **authentication** (auth_provider), but will use the local system users for user and group information (id_provider).

Adjust the permissions of the config file and start sssd:

```
$ sudo chown root:root /etc/sss/sss.conf
```

```
$ sudo chmod 0600 /etc/sss/sss.conf
```

```
$ sudo systemctl start sss
```

Just by having installed sss and its dependencies, PAM will already have been configured to use sss, with a *fallback* to local user authentication. To try it out, if this is a workstation, simply switch users (in the GUI), or open a login terminal (Ctrl-Alt-number), or spawn a login shell with `sudo login`, and try logging in using the name of a Kerberos principal. Remember that this user must already exist on the local system:

```
$ sudo login

focal-krb5-client login: ubuntu
Password:
Welcome to Ubuntu Focal Fossa (development branch) (GNU/Linux 5.4.0-21-generic x86_64)

(...)

Last login: Thu Apr  9 21:23:50 UTC 2020 from 10.20.20.1 on pts/0
$ klist
Ticket cache: FILE:/tmp/krb5cc_1000_NlfnSX
Default principal: ubuntu@EXAMPLE.COM

Valid starting    Expires          Service principal
04/09/20 21:36:12  04/10/20 07:36:12  krbtgt/EXAMPLE.COM@EXAMPLE.COM
    renew until 04/10/20 21:36:12
```

You will have a Kerberos ticket already right after login.

How to set up Kerberos with OpenLDAP backend

Kerberos supports a few different database backends. The default one (which we have used in our other Kerberos guides so far) is called db2. The [DB types documentation](#)¹³² shows all the options, one of which is LDAP.

Why use LDAP?

There are several reasons why one would want to have the Kerberos principals stored in LDAP as opposed to a local on-disk database. There are also cases when it is not a good idea. Each site has to evaluate the pros and cons. Here are a few:

- Pros:
 - If you already have OpenLDAP set up for other things, such as storing users and groups, adding the Kerberos attributes can be beneficial, providing an integrated story
- Cons:

¹³² <https://web.mit.edu/kerberos/krb5-latest/doc/admin/dbtypes.html>

- Setting up the LDAP backend isn't a trivial task and shouldn't be attempted by administrators without prior knowledge of OpenLDAP
- As highlighted in the [LDAP section of DB types](#)¹³³, since krb5kdc is single-threaded there may be higher latency in servicing requests when using the OpenLDAP backend

In this guide

In this section we'll configure a primary and secondary Kerberos server to use OpenLDAP for the principal database. Note that as of version 1.18, the Key Distribution Center (KDC) from MIT Kerberos [does not support](#)¹³⁴ a primary KDC using a read-only consumer (secondary) LDAP server. What we have to consider here is that a primary KDC is read-write, and it needs a read-write backend. The secondary KDCs can use both a read-write and read-only backend, because they are expected to be read-only. Therefore there are only some possible layouts we can use:

1. Simple case:

- Primary KDC connected to primary OpenLDAP
- Secondary KDC connected to both primary and secondary OpenLDAP

2. Extended simple case:

- Multiple primary KDCs connected to one primary OpenLDAP
- Multiple secondary KDCs connected to primary and secondary OpenLDAP

3. OpenLDAP with replication:

- Multiple primary KDCs connected to all primary OpenLDAP servers

We haven't covered OpenLDAP replication in this guide, so we will show the **simple case** only. The second scenario is an extension: just add another primary KDC to the mix, talking to the same primary OpenLDAP server.

Configure OpenLDAP

We are going to install the OpenLDAP server on the same host as the KDC, to simplify the communication between them. In such a setup, we can use the `ldapi:///` transport, which is via a UNIX socket, and we don't need to set up SSL certificates to secure the communication between the Kerberos services and OpenLDAP. Note, however, that SSL is still needed for the OpenLDAP replication. See [LDAP with TLS](#) (page 476) for details.

If you want to use an existing OpenLDAP server, that's also possible, but keep in mind that you should then use SSL for the communication between the KDC and this OpenLDAP server.

First, the necessary **schema** needs to be loaded on an OpenLDAP server that has network connectivity to both the **primary** and **secondary** KDCs. The rest of this section assumes that you also have LDAP replication configured between at least two servers. For information on setting up OpenLDAP see [OpenLDAP Server](#) (page 449).

¹³³ <https://web.mit.edu/kerberos/krb5-latest/doc/admin/dbtypes.html#ldap-module-ldap>

¹³⁴ <https://krbdev.mit.edu/rt/Ticket/Display.html?id=7754#>

Note:

cn=admin,dc=example,dc=com is a default admin user that is created during the installation of the slapd package (the OpenLDAP server). The domain component will change for your server, so adjust accordingly.

- Install the necessary packages (it's assumed that OpenLDAP is already installed):

```
$ sudo apt install krb5-kdc-ldap krb5-admin-server
```

- Next, extract the kerberos.schema.gz file:

```
$ sudo cp /usr/share/doc/krb5-kdc-ldap/kerberos.schema.gz  
/etc/ldap/schema/ > sudo gunzip /etc/ldap/schema/kerberos.schema.gz
```

- The **Kerberos schema** needs to be added to the cn=config tree. This schema file needs to be converted to LDIF format before it can be added. For that we will use a helper tool, called schema2ldif, provided by the package of the same name which is available in the Universe archive:

```
$ sudo apt install schema2ldif
```

- To import the Kerberos schema, run:

```
$ sudo ldap-schema-manager -i kerberos.schema  
  
SASL/EXTERNAL authentication started  
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth  
SASL SSF: 0  
executing 'ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/kerberos.ldif'  
SASL/EXTERNAL authentication started  
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth  
SASL SSF: 0  
adding new entry "cn=kerberos,cn=schema,cn=config"
```

- With the new schema loaded, let's index an attribute often used in searches:

```
$ sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF  
  
dn: olcDatabase={1}mdb,cn=config  
add: olcDbIndex  
olcDbIndex: krbPrincipalName eq,pres,sub  
EOF  
  
modifying entry "olcDatabase={1}mdb,cn=config"
```

- Let's create LDAP entries for the Kerberos administrative entities that will contact the OpenLDAP server to perform operations. There are two:
 - **ldap_kdc_dn**: needs to have read rights on the realm container, principal container and realm sub-trees. If **disable_last_success** and **disable_lockout** are not set,

however, then `ldap_kdc_dn` needs write access to the Kerberos container just like the admin `DN` below.

- `ldap_kadmin_dn`: needs to have read and write rights on the realm container, principal container and realm sub-trees

Here is the command to create these entities:

```
$ ldapadd -x -D cn=admin,dc=example,dc=com -W <<EOF

dn: uid=kdc-service,dc=example,dc=com
uid: kdc-service
objectClass: account
objectClass: simpleSecurityObject
userPassword: {CRYPT}x
description: Account used for the Kerberos KDC

dn: uid=kadmin-service,dc=example,dc=com
uid: kadmin-service
objectClass: account
objectClass: simpleSecurityObject
userPassword: {CRYPT}x
description: Account used for the Kerberos Admin server
EOF
Enter LDAP Password:
adding new entry "uid=kdc-service,dc=example,dc=com"

adding new entry "uid=kadmin-service,dc=example,dc=com"
```

Now let's set a password for them. Note that first the tool asks for the password you want for the specified user DN, and then for the password of the `cn=admin` DN:

```
$ ldappasswd -x -D cn=admin,dc=example,dc=com -W -S
uid=kdc-service,dc=example,dc=com

New password: <-- password you want for uid-kdc-service
Re-enter new password:
Enter LDAP Password: <-- password for the DN specified with the -D option
```

Repeat for the `uid=kadmin-service` DN. These passwords will be needed later.

You can test these with `ldapwhoami`:

```
$ ldapwhoami -x -D uid=kdc-service,dc=example,dc=com -W

Enter LDAP Password:
dn:uid=kdc-service,dc=example,dc=com
```

- Finally, update the Access Control Lists ([ACL](#)). These can be tricky, as it highly depends on what you have defined already. By default, the `slapd` package configures your database with the following ACLs:

```
olcAccess: {0}to attrs=userPassword by self write by anonymous auth by * none
olcAccess: {1}to attrs=shadowLastChange by self write by * read
olcAccess: {2}to * by * read
```


We need to insert new rules before the final to * by * read one, to control access to the Kerberos related entries and attributes:

```
$ sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF

dn: olcDatabase={1}mdb,cn=config
add: olcAccess
olcAccess: {2}to attrs=krbPrincipalKey
    by anonymous auth
    by dn.exact="uid=kdc-service,dc=example,dc=com" read
    by dn.exact="uid=kadmin-service,dc=example,dc=com" write
    by self write
    by * none
-
add: olcAccess
olcAccess: {3}to dn.subtree="cn=krbContainer,dc=example,dc=com"
    by dn.exact="uid=kdc-service,dc=example,dc=com" read
    by dn.exact="uid=kadmin-service,dc=example,dc=com" write
    by * none
EOF

modifying entry "olcDatabase={1}mdb,cn=config"
```

This will make the existing {2} rule become {4}. Check with `sudo slapcat -b cn=config` (the output below was reformatted a bit for clarity):

```
olcAccess: {0}to attrs=userPassword
    by self write
    by anonymous auth
    by * none
olcAccess: {1}to attrs=shadowLastChange
    by self write
    by * read
olcAccess: {2}to attrs=krbPrincipalKey by anonymous auth
    by dn.exact="uid=kdc-service,dc=example,dc=com" read
    by dn.exact="uid=kadmin-service,dc=example,dc=com" write
    by self write
    by * none
olcAccess: {3}to dn.subtree="cn=krbContainer,dc=example,dc=com"
    by dn.exact="uid=kdc-service,dc=example,dc=com" read
    by dn.exact="uid=kadmin-service,dc=example,dc=com" write
    by * none
olcAccess: {4}to * by * read
```

Your LDAP directory is now ready to serve as a Kerberos principal database.

Primary KDC configuration (LDAP)

With OpenLDAP configured it is time to configure the KDC. In this example we are doing it in the same OpenLDAP server to take advantage of local UNIX socket communication.

- Reconfigure the krb5-config package if needed to get a good starting point with /etc/krb5.conf:

```
$ sudo dpkg-reconfigure krb5-config
```

- Now edit /etc/krb5.conf adding the database_module option to the EXAMPLE.COM realm section:

```
[realms]
EXAMPLE.COM = {
    kdc = kdc01.example.com
    kdc = kdc02.example.com
    admin_server = kdc01.example.com
    default_domain = example.com
    database_module = openldap_ldapconf
}
```

Then also add these new sections:

```
[dbdefaults]
ldap_kerberos_container_dn = cn=krbContainer,dc=example,dc=com

[dbmodules]
openldap_ldapconf = {
    db_library = kldap

    # if either of these is false, then the ldap_kdc_dn needs to
    # have write access
    disable_last_success = true
    disable_lockout = true

    # this object needs to have read rights on
    # the realm container, principal container and realm sub-trees
    ldap_kdc_dn = "uid=kdc-service,dc=example,dc=com"

    # this object needs to have read and write rights on
    # the realm container, principal container and realm sub-trees
    ldap_kadmin_dn = "uid=kadmin-service,dc=example,dc=com"

    ldap_service_password_file = /etc/krb5kdc/service.keyfile
    ldap_servers = ldapi:///
    ldap_conns_per_server = 5
}
```

- Next, use the kdb5_ldap_util utility to create the realm:

```
$ sudo kdb5_ldap_util -D cn=admin,dc=example,dc=com create -subtrees
dc=example,dc=com -r EXAMPLE.COM -s -H ldapi:///
```

```
Password for "cn=admin,dc=example,dc=com":
Initializing database for realm 'EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
```

- Create a stash of the password used to bind to the LDAP server. Run it once for each `ldap_kdc_dn` and `ldap_kadmin_dn`:

```
$ sudo kdb5_ldap_util -D cn=admin,dc=example,dc=com stashsrvpw -f
/etc/krb5kdc/service.keyfile uid=kdc-service,dc=example,dc=com > sudo
kdb5_ldap_util -D cn=admin,dc=example,dc=com stashsrvpw -f
/etc/krb5kdc/service.keyfile uid=kadmin-service,dc=example,dc=com
```

Note:

The `/etc/krb5kdc/service.keyfile` file now contains clear text versions of the passwords used by the KDC to contact the LDAP server!

- Create a `/etc/krb5kdc/kadm5.acl` file for the admin server, if you haven't already:

```
*/admin@EXAMPLE.COM
```

```
*
```

- Start the Kerberos KDC and admin server:

```
$ sudo systemctl start krb5-kdc.service krb5-admin-server.service
```

You can now add Kerberos principals to the LDAP database, and they will be copied to any other LDAP servers configured for replication. To add a principal using the `kadmin.local` utility enter:

```
$ sudo kadmin.local

Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local: addprinc ubuntu
WARNING: no policy specified for ubuntu@EXAMPLE.COM; defaulting to no policy
Enter password for principal "ubuntu@EXAMPLE.COM":
Re-enter password for principal "ubuntu@EXAMPLE.COM":
Principal "ubuntu@EXAMPLE.COM" created.
kadmin.local:
```

The above will create an `ubuntu` principal with a DN of `krbPrincipalName=ubuntu@EXAMPLE.COM,cn=EXAMPLE.COM,cn=krbContainer,dc=example,dc=com`.

Let's say, however, that you already have a user in your directory, and it's in `uid=testuser1,ou=People,dc=example,dc=com`. How can you add the Kerberos attributes to it? You use the `-x` parameter to specify the location. For the `ldap_kadmin_dn` to be able to write to it, we first need to update the ACLs:

```
$ sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF

dn: olcDatabase={1}mdb,cn=config
add: olcAccess
olcAccess: {4}to dn.subtree="ou=People,dc=example,dc=com"
        by dn.exact="uid=kdc-service,dc=example,dc=com" read
        by dn.exact="uid=kadmin-service,dc=example,dc=com" write
        by * break
EOF
```

And now we can specify the new location:

```
$ sudo kadmin.local

Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local: addprinc -x dn=uid=testuser1,ou=People,dc=example,dc=com testuser1
WARNING: no policy specified for testuser1@EXAMPLE.COM; defaulting to no policy
Enter password for principal "testuser1@EXAMPLE.COM":
Re-enter password for principal "testuser1@EXAMPLE.COM":
Principal "testuser1@EXAMPLE.COM" created.
```

Since the specified DN already exists, `kadmin.local` will just add the required Kerberos attributes to this existing entry. If it didn't exist, it would be created from scratch, with only the Kerberos attributes, just like what happened with the `ubuntu` example above, but in the specified location.

Note:

If the `ldap_kadmin_dn` DN (`uid=kadmin-service` in our example) does not have write access to the location specified by the `-x` parameter, you will get an `Insufficient access` error.

Both places are visible for `kinit`, since, when the realm was created with `kdb5_ldap_util`, the default value for the search scope and base were taken: `subtree`, and `dc=example,dc=com`.

Secondary KDC configuration (LDAP)

The setup of the secondary KDC (and its OpenLDAP replica) is very similar. Once you have the OpenLDAP replication setup, repeat these steps on the secondary:

- Install `krb5-kdc-ldap`, `ldap-utils`. Do **not** install `krb5-admin-server`.
- Load the Kerberos schema using `schema2ldif`.
- Add the index for `krbPrincipalName`.
- Add the ACLs.
- Configure `krb5.conf` in the same way, initially. If you want to, and if you configured SSL properly, you can add `ldaps://kdc01.example.com` to the `ldap_servers` list after `ldapi://`, so that the secondary KDC can have two LDAP backends at its disposal.
- **DO NOT** run `kdb5_ldap_util`. There is no need to create the database since it's being replicated from the primary.

- Copy over the following files from the primary KDC and place them in the same location on the secondary:
 - `/etc/krb5kdc/stash`
 - `/etc/krb5kdc/service.keyfile`
- Start the KDC: `sudo systemctl start krb5-kdc.service`

Further reading

- [Configuring Kerberos with OpenLDAP back-end](#)¹³⁵
- [MIT Kerberos backend types](#)¹³⁶

See also

- Explanation: [Introduction to Kerberos](#) (page 846)

Network user authentication with SSSD

These guides will show you how to set up network user authentication with SSSD with...

How to set up SSSD with Active Directory

This section describes the use of SSSD to authenticate user logins against an Active Directory via using SSSD's "ad" provider. At the end, Active Directory users will be able to log in on the host using their AD credentials. Group membership will also be maintained.

Group Policies for Ubuntu

SSSD manages user authentication and sets initial security policies.

[ADSys](#) serves as a Group Policy client for Ubuntu, streamlining the configuration of Ubuntu systems within a Microsoft Active Directory environment. If you are interested in Group Policies support for Ubuntu, detailed information can be found in the [ADSys documentation](#)¹³⁷.

¹³⁵ https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_ldap.html#conf-ldap

¹³⁶ <https://web.mit.edu/kerberos/krb5-latest/doc/admin/dbtypes.html>

¹³⁷ <https://documentation.ubuntu.com/adsys/stable/>

Prerequisites and assumptions

This guide does not explain Active Directory, how it works, how to set one up, or how to maintain it. It assumes that a working Active Directory domain is already configured and you have access to the credentials to join a machine to that domain.

- The domain controller is:
 - Acting as an authoritative *DNS* server for the domain.
 - The primary DNS resolver (check with `resolvectl status`).
- System time is correct and in sync, maintained via a service like `chrony` or `ntp`.
- The domain used in this example is `ad1.example.com`.

Install necessary software

Install the following packages:

```
$ sudo apt install sssd-ad sssd-tools realmd adcli
```

Join the domain

We will use the `realm` command, from the `realmd` package, to join the domain and create the SSSD configuration.

Let's verify the domain is discoverable via DNS:

```
$ sudo realm -v discover ad1.example.com

* Resolving: _ldap._tcp.ad1.example.com
* Performing LDAP DSE lookup on: 10.51.0.5
* Successfully discovered: ad1.example.com
ad1.example.com
type: kerberos
realm-name: AD1.EXAMPLE.COM
domain-name: ad1.example.com
configured: no
server-software: active-directory
client-software: sssd
required-package: sssd-tools
required-package: sssd
required-package: libnss-sss
required-package: libpam-sss
required-package: adcli
required-package: samba-common-bin
```

This performs several checks and determines the best software stack to use with SSSD. SSSD can install the missing packages via `packagekit`, but we already installed them in the previous step.

Now let's join the domain:

```
$ sudo realm join ad1.example.com
```

```
Password for Administrator:
```

That was quite uneventful. If you want to see what it was doing, pass the `-v` option:

```
$ sudo realm join -v ad1.example.com
```

```
* Resolving: _ldap._tcp.ad1.example.com
* Performing LDAP DSE lookup on: 10.51.0.5
* Successfully discovered: ad1.example.com
Password for Administrator:
* Unconditionally checking packages
* Resolving required packages
* LANG=C /usr/sbin/adcli join --verbose --domain ad1.example.com --domain-realm
AD1.EXAMPLE.COM --domain-controller 10.51.0.5 --login-type user --login-user
Administrator --stdin-password
* Using domain name: ad1.example.com
* Calculated computer account name from fqdn: AD-CLIENT
* Using domain realm: ad1.example.com
* Sending NetLogon ping to domain controller: 10.51.0.5
* Received NetLogon info from: SERVER1.ad1.example.com
* Wrote out krb5.conf snippet to /var/cache/realmd/adcli-krb5-hUfTUg/krb5.d/adcli-
krb5-conf-hv2kzi
* Authenticated as user: Administrator@AD1.EXAMPLE.COM
* Looked up short domain name: AD1
* Looked up domain SID: S-1-5-21-2660147319-831819607-3409034899
* Using fully qualified name: ad-client.ad1.example.com
* Using domain name: ad1.example.com
* Using computer account name: AD-CLIENT
* Using domain realm: ad1.example.com
* Calculated computer account name from fqdn: AD-CLIENT
* Generated 120 character computer password
* Using keytab: FILE:/etc/krb5.keytab
* Found computer account for AD-CLIENT$ at: CN=AD-CLIENT,CN=Computers,DC=ad1,
DC=example,DC=com
* Sending NetLogon ping to domain controller: 10.51.0.5
* Received NetLogon info from: SERVER1.ad1.example.com
* Set computer password
* Retrieved kvno '3' for computer account in directory: CN=AD-CLIENT,CN=Computers,
DC=ad1,DC=example,DC=com
* Checking RestrictedKrbHost/ad-client.ad1.example.com
*   Added RestrictedKrbHost/ad-client.ad1.example.com
* Checking RestrictedKrbHost/AD-CLIENT
*   Added RestrictedKrbHost/AD-CLIENT
* Checking host/ad-client.ad1.example.com
*   Added host/ad-client.ad1.example.com
* Checking host/AD-CLIENT
*   Added host/AD-CLIENT
* Discovered which keytab salt to use
* Added the entries to the keytab: AD-CLIENT$@AD1.EXAMPLE.COM: FILE:/etc/krb5.
keytab
* Added the entries to the keytab: host/AD-CLIENT@AD1.EXAMPLE.COM: FILE:/etc/krb5.
keytab
* Added the entries to the keytab: host/ad-client.ad1.example.com@AD1.EXAMPLE.COM:
FILE:/etc/krb5.keytab
* Added the entries to the keytab: RestrictedKrbHost/AD-CLIENT@AD1.EXAMPLE.COM:
FILE:/etc/krb5.keytab
* Added the entries to the keytab: RestrictedKrbHost/ad-client.ad1.example.com@AD1.
EXAMPLE.COM: FILE:/etc/krb5.keytab
* /usr/sbin/update-rc.d sssd enable
* /usr/sbin/service sssd restart
* Successfully enrolled machine in realm
```


By default, `realm` will use the “Administrator” account of the domain to request the join. If you need to use another account, pass it to the tool with the `-U` option.

Another popular way of joining a domain is using a One Time Password (OTP) token. For that, use the `--one-time-password` option.

SSSD configuration

The `realm` tool already took care of creating an SSSD configuration, adding the PAM and NSS modules, and starting the necessary services.

Let’s take a look at `/etc/sss/sss.conf`:

```
[sss]
domains = ad1.example.com
config_file_version = 2
services = nss, pam

[domain/ad1.example.com]
default_shell = /bin/bash
krb5_store_password_if_offline = True
cache_credentials = True
krb5_realm = AD1.EXAMPLE.COM
realmd_tags = manages-system joined-with-adcli
id_provider = ad
fallback_homedir = /home/%u@%d
ad_domain = ad1.example.com
use_fully_qualified_names = True
ldap_id_mapping = True
access_provider = ad
```

Note:

Something very important to remember is that this file must have permissions `0600` and ownership `root:root`, or else SSSD won’t start!

Note:

```
[FAILED] Failed to listen on sssd-nss.socket...
[FAILED] Failed to listen on sssd-pam.socket...
[FAILED] Dependency failed for sssd-pam-priv.socket...
```

If you get those errors on startup, just remove the line `services = nss, pam`

Let’s highlight a few things from this config file:

- **cache_credentials:** This allows logins when the AD server is unreachable
- **fallback_homedir:** The home directory. By default, `/home/<user>@<domain>`. For example, the AD user *john* will have a home directory of `/home/john@ad1.example.com`.
- **use_fully_qualified_names:** Users will be of the form `user@domain`, not just `user`. This should only be changed if you are certain no other domains will ever join the AD forest, via one of the several possible trust relationships.

Automatic home directory creation

What the `realm` tool didn't do for us is setup `pam_mkhomedir`, so that network users can get a home directory when they login. This remaining step can be done by running the following command:

```
$ sudo pam-auth-update --enable mkhomedir
```

Testing our setup

You should now be able to fetch information about AD users. In this example, John Smith is an AD user:

```
$ getent passwd john@ad1.example.com

john@ad1.example.com:*:1725801106:1725800513:John Smith:/home/john@ad1.example.com:/bin/bash
```

Let's see his groups:

```
$ groups john@ad1.example.com

john@ad1.example.com : domain users@ad1.example.com engineering@ad1.example.com
```

Note:

If you just changed the group membership of a user, it may be a while before SSSD notices due to caching.

Finally, how about we try a login:

```
$ sudo login

ad-client login: john@ad1.example.com
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-24-generic x86_64)
...
Creating directory '/home/john@ad1.example.com'.

john@ad1.example.com@ad-client:~$
```

Notice how the home directory was automatically created.

You can also use SSH, but note that the command will look a bit funny because of the multiple @ signs:

```
$ ssh john@ad1.example.com@10.51.0.11
```

```
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-24-generic x86_64)
(...)
Last login: Thu Apr 16 21:22:55 2020

john@ad1.example.com@ad-client:~$
```

Note:

In the SSH example, public key authentication was used, so no password was required. Remember that SSH password authentication is disabled by default in `/etc/ssh/sshd_config`.

Kerberos tickets

If you install `krb5-user`, your AD users will also get a Kerberos ticket upon logging in:

```
john@ad1.example.com@ad-client:~$ klist

Ticket cache: FILE:/tmp/krb5cc_1725801106_9UxVIz
Default principal: john@AD1.EXAMPLE.COM

Valid starting    Expires          Service principal
04/16/20 21:32:12 04/17/20 07:32:12 krbtgt/AD1.EXAMPLE.COM@AD1.EXAMPLE.COM
                renew until 04/17/20 21:32:12
```

Note:

realm also configured `/etc/krb5.conf` for you, so there should be no further configuration prompts when installing `krb5-user`.

Let's test with `smbclient` using Kerberos authentication to list the shares of the domain controller:

```
john@ad1.example.com@ad-client:~$ smbclient -k -L server1.ad1.example.com

Sharename      Type      Comment
-----
ADMIN$         Disk      Remote Admin
C$             Disk      Default share
IPC$           IPC       Remote IPC
NETLOGON       Disk      Logon server share
SYSVOL         Disk      Logon server share
SMB1 disabled -- no workgroup available
```

Notice how we now have a ticket for the `cifs` service, which was used for the share list above:

```
john@ad1.example.com@ad-client:~$ klist
```

```
Ticket cache: FILE:/tmp/krb5cc_1725801106_9UxVIz
Default principal: john@AD1.EXAMPLE.COM

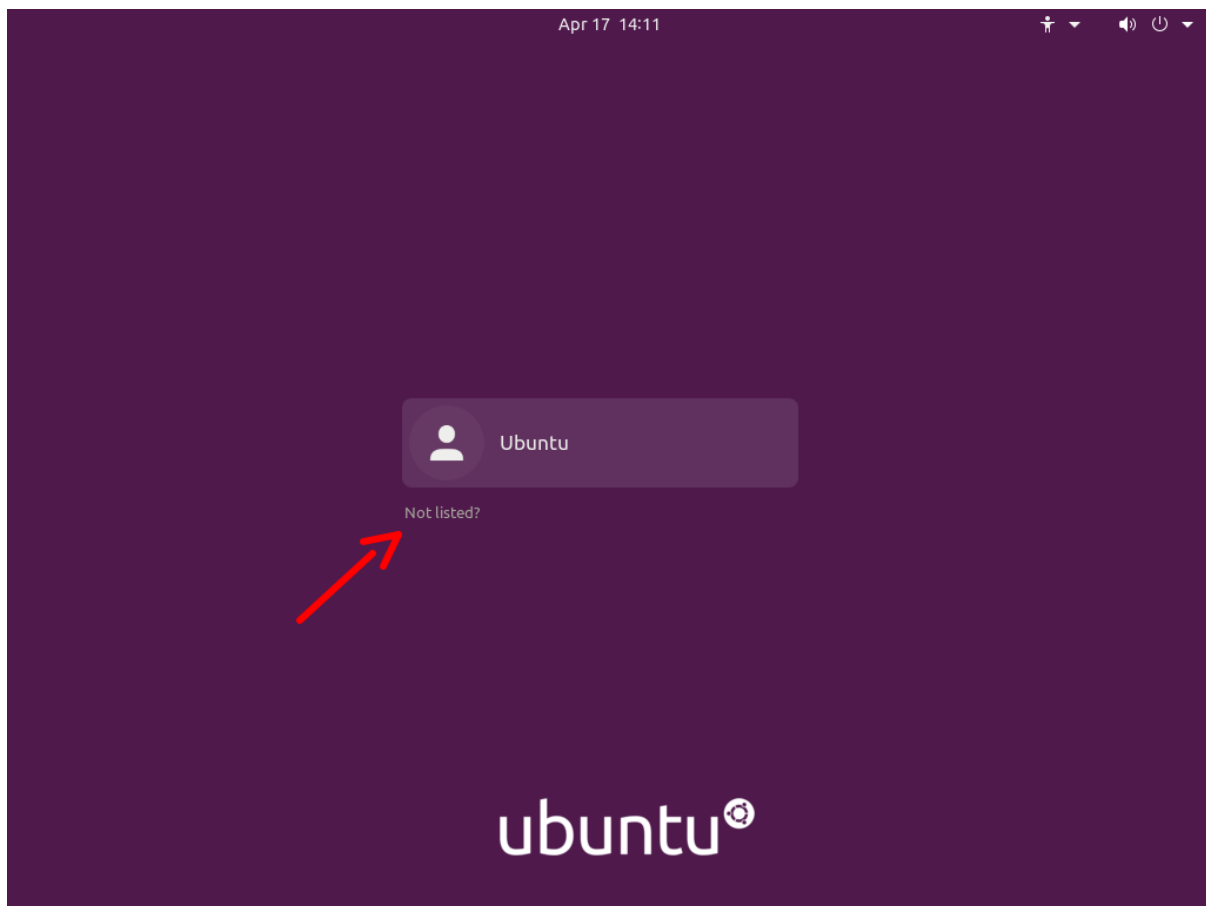
Valid starting    Expires          Service principal
04/16/20 21:32:12  04/17/20 07:32:12  krbtgt/AD1.EXAMPLE.COM@AD1.EXAMPLE.COM
        renew until 04/17/20 21:32:12
04/16/20 21:32:21  04/17/20 07:32:12  cifs/server1.ad1.example.com@AD1.EXAMPLE.COM
```

Ubuntu Desktop authentication

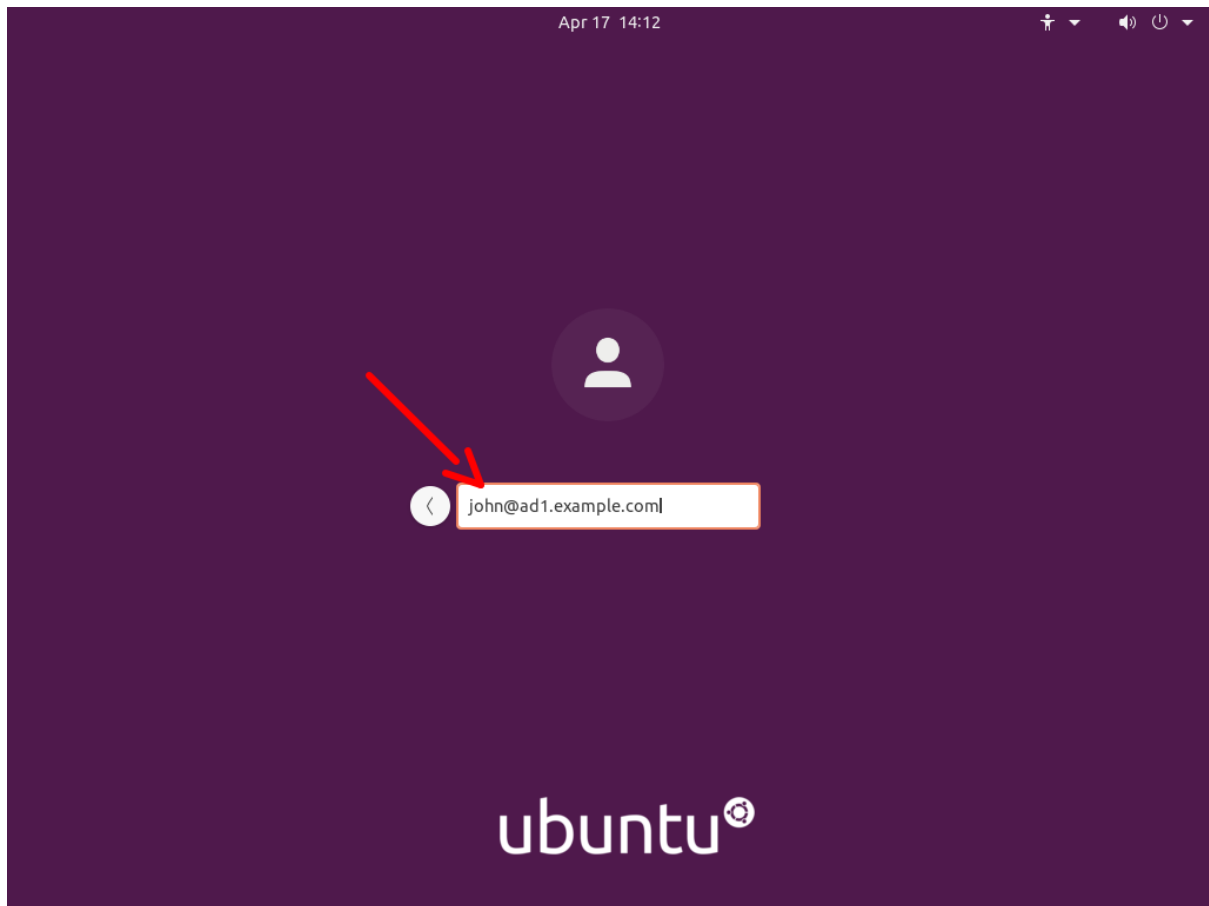
The desktop login only shows local users in the list to pick from, and that's on purpose.

To login with an Active Directory user for the first time, follow these steps:

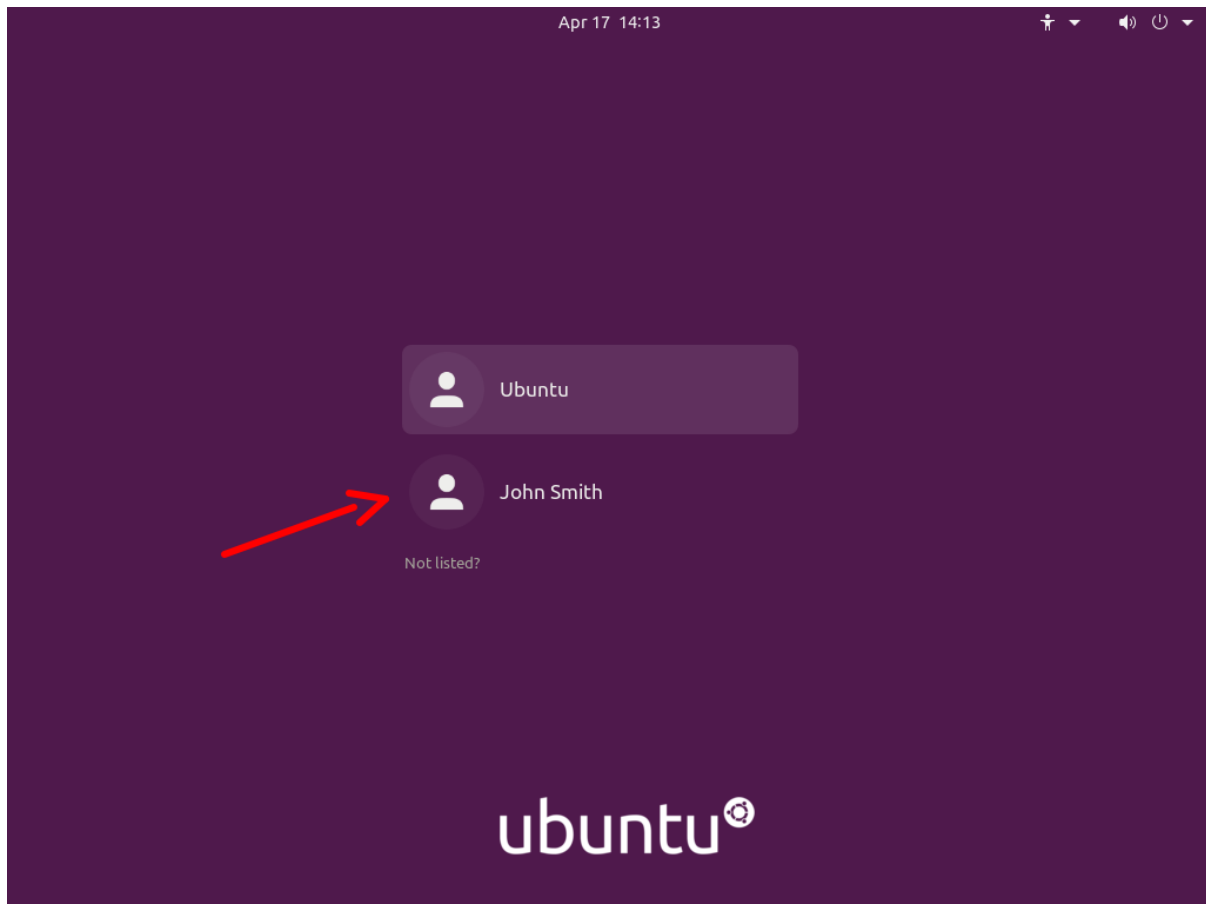
- Click on the “Not listed?” option:



- Type in the login name followed by the password:



- Next time you login, the AD user will be listed as if it was a local user:



Known issues

When logging in on a system joined with an Active Directory domain, sssd (the package responsible for this integration) will try to apply Group Policies by default. There are cases where if a specific policy is missing, the login will be denied.

This is being tracked in [LP: #1934997](https://bugs.launchpad.net/ubuntu/+bug/1934997)¹³⁸. Until the fix becomes available, please see comment number 5 in that bug report for existing workarounds.

Further reading

- [GitHub SSSD Project](#)¹³⁹
- [Active Directory DNS Zone Entries](#)¹⁴⁰

¹³⁸ <https://bugs.launchpad.net/ubuntu/+bug/1934997>

¹³⁹ <https://github.com/SSSD/sss>

¹⁴⁰ [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc759550\(v=ws.10\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc759550(v=ws.10)?redirectedfrom=MSDN)

How to set up SSSD with LDAP

SSSD can also use LDAP for authentication, authorisation, and user/group information. In this section we will configure a host to authenticate users from an OpenLDAP directory.

Prerequisites and assumptions

For this setup, we need:

- An existing OpenLDAP server with SSL enabled and using the RFC2307 schema for users and groups
- A client host where we will install the necessary tools and login as a user from the LDAP server

Install necessary software

Install the following packages:

```
$ sudo apt install sssd-ldap ldap-utils
```

Configure SSSD

Create the `/etc/sss/sss.conf` configuration file, with permissions `0600` and ownership `root:root`, and add the following content:

```
[sss]
config_file_version = 2
domains = example.com

[domain/example.com]
id_provider = ldap
auth_provider = ldap
ldap_uri = ldap://ldap01.example.com
cache_credentials = True
ldap_search_base = dc=example,dc=com
```

Make sure to start the `sss` service:

```
$ sudo systemctl start sss.service
```

Note:

`sss` will use `START_TLS` by default for authentication requests against the LDAP server (the **auth_provider**), but not for the **id_provider**. If you want to also enable `START_TLS` for the `id_provider`, specify `ldap_id_use_start_tls = true`.

Automatic home directory creation

To enable automatic home directory creation, run the following command:

```
$ sudo pam-auth-update --enable mkhomedir
```

Check SSL setup on the client

The client must be able to use START_TLS when connecting to the LDAP server, with full certificate checking. This means:

- The client host knows and trusts the CA that signed the LDAP server certificate,
- The server certificate was issued for the correct host (ldap01.example.com in this guide),
- The time is correct on all hosts performing the TLS connection, and
- That neither certificate (CA or server's) expired.

If using a custom CA, an easy way to have a host trust it is to place it in /usr/local/share/ca-certificates/ with a .crt extension and run sudo update-ca-certificates.

Alternatively, you can edit /etc/ldap/ldap.conf and point TLS_CACERT to the CA public key file.

Note:

You may have to restart sssd after these changes: sudo systemctl restart sssd

Once that is all done, check that you can connect to the LDAP server using verified SSL connections:

```
$ ldapwhoami -x -ZZ -H ldap://ldap01.example.com  
anonymous
```

and for ldaps (if enabled in /etc/default/slapd):

```
$ ldapwhoami -x -H ldaps://ldap01.example.com
```

The -ZZ parameter tells the tool to use START_TLS, and that it must not fail. If you have LDAP logging enabled on the server, it will show something like this:

```
slapd[779]: conn=1032 op=0 STARTTLS  
slapd[779]: conn=1032 op=0 RESULT oid= err=0 text=  
slapd[779]: conn=1032 fd=15 TLS established tls_ssf=256 ssf=256  
slapd[779]: conn=1032 op=1 BIND dn="" method=128  
slapd[779]: conn=1032 op=1 RESULT tag=97 err=0 text=  
slapd[779]: conn=1032 op=2 EXT oid=1.3.6.1.4.1.4203.1.11.3  
slapd[779]: conn=1032 op=2 WHOAMI  
slapd[779]: conn=1032 op=2 RESULT oid= err=0 text=
```


START_TLS with err=0 and TLS established is what we want to see there, and, of course, the WHOAMI extended operation.

Final verification

In this example, the LDAP server has the following user and group entry we are going to use for testing:

```
dn: uid=john,ou=People,dc=example,dc=com
uid: john
objectClass: inetOrgPerson
objectClass: posixAccount
cn: John Smith
sn: Smith
givenName: John
mail: john@example.com
userPassword: johnsecret
uidNumber: 10001
gidNumber: 10001
loginShell: /bin/bash
homeDirectory: /home/john

dn: cn=john,ou=Group,dc=example,dc=com
cn: john
objectClass: posixGroup
gidNumber: 10001
memberUid: john

dn: cn=Engineering,ou=Group,dc=example,dc=com
cn: Engineering
objectClass: posixGroup
gidNumber: 10100
memberUid: john
```

The user john should be known to the system:

```
ubuntu@ldap-client:~$ getent passwd john
john*:10001:10001:John Smith:/home/john:/bin/bash
```

```
ubuntu@ldap-client:~$ id john
uid=10001(john) gid=10001(john) groups=10001(john),10100(Engineering)
```

And we should be able to authenticate as john:

```
ubuntu@ldap-client:~$ sudo login
```

```
ldap-client login: john
Password:
Welcome to Ubuntu Focal Fossa (development branch) (GNU/Linux 5.4.0-24-generic x86_
64)
(...)
Creating directory '/home/john'.

john@ldap-client:~$
```

How to set up SSSD with LDAP and Kerberos

With SSSD we can create a setup that is very similar to Active Directory in terms of the technologies used: using LDAP for users and groups, and Kerberos for authentication.

Prerequisites and assumptions

For this setup, we will need:

- An existing [OpenLDAP server](#) (page 449) using the RFC2307 schema for users and groups. SSL support is recommended, but not strictly necessary because authentication in this setup is being done via Kerberos, and not LDAP.
- A [Kerberos server](#) (page 183). It doesn't have to be [using the OpenLDAP backend](#) (page 196).
- A client host where we will install and configure SSSD.

Install necessary software

On the client host, install the following packages:

```
$ sudo apt install sssd-ldap sssd-krb5 ldap-utils krb5-user
```

You may be asked about the default Kerberos realm. For this guide, we are using EXAMPLE.COM.

At this point, you should already be able to obtain tickets from your Kerberos server, assuming [DNS](#) records point at it:

```
$ kinit ubuntu

Password for ubuntu@EXAMPLE.COM:

ubuntu@ldap-krb-client:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: ubuntu@EXAMPLE.COM

Valid starting    Expires          Service principal
04/17/20 19:51:06 04/18/20 05:51:06 krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 04/18/20 19:51:05
```

But we want to be able to login as an LDAP user, authenticated via Kerberos. Let's continue with the configuration.

Configure SSSD

Create the `/etc/sss/sss.conf` configuration file, with permissions `0600` and ownership `root:root`, and add the following content:

```
[sss]
config_file_version = 2
domains = example.com

[domain/example.com]
id_provider = ldap
ldap_uri = ldap://ldap01.example.com
ldap_search_base = dc=example,dc=com
auth_provider = krb5
krb5_server = kdc01.example.com,kdc02.example.com
krb5_kpasswd = kdc01.example.com
krb5_realm = EXAMPLE.COM
cache_credentials = True
```

This example uses two KDCs, which made it necessary to also specify the `krb5_kpasswd` server because the second KDC is a replica and is not running the admin server.

Start the `sss` service:

```
$ sudo systemctl start sss.service
```

Automatic home directory creation

To enable automatic home directory creation, run the following command:

```
$ sudo pam-auth-update --enable mkhomedir
```

Final verification

In this example, the LDAP server has the following user and group entry we are going to use for testing:

```
dn: uid=john,ou=People,dc=example,dc=com
uid: john
objectClass: inetOrgPerson
objectClass: posixAccount
cn: John Smith
sn: Smith
givenName: John
mail: john@example.com
uidNumber: 10001
gidNumber: 10001
```

(continues on next page)

(continued from previous page)

```
loginShell: /bin/bash
homeDirectory: /home/john

dn: cn=john,ou=Group,dc=example,dc=com
cn: john
objectClass: posixGroup
gidNumber: 10001
memberUid: john

dn: cn=Engineering,ou=Group,dc=example,dc=com
cn: Engineering
objectClass: posixGroup
gidNumber: 10100
memberUid: john
```

Note how the user john has no userPassword attribute.

The user john should be known to the system:

```
ubuntu@ldap-client:~$ getent passwd john

john*:10001:10001:John Smith:/home/john:/bin/bash
```

```
ubuntu@ldap-client:~$ id john

uid=10001(john) gid=10001(john) groups=10001(john),10100(Engineering)
```

Let's try a login as this user:

```
ubuntu@ldap-krb-client:~$ sudo login

ldap-krb-client login: john
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-24-generic x86_64)
(...)
Creating directory '/home/john'.
```

```
john@ldap-krb-client:~$ klist

Ticket cache: FILE:/tmp/krb5cc_10001_B0rxWr
Default principal: john@EXAMPLE.COM

Valid starting    Expires          Service principal
04/17/20 20:29:50 04/18/20 06:29:50  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 04/18/20 20:29:50
```

We logged in using the Kerberos password, and user/group information from the LDAP server.

SSSD and KDC spoofing

When using SSSD to manage Kerberos logins on a Linux host, there is an attack scenario you should be aware of: KDC spoofing.

The objective of the attacker is to login on a workstation that is using Kerberos authentication. Let's say they know john is a valid user on that machine.

The attacker first deploys a rogue Key Distribution Center (KDC) server in the network, and creates the john principal there with a password of the attacker's choosing. What they must do now is have their rogue KDC respond to the login request from the workstation, before (or instead of) the real KDC. If the workstation isn't authenticating the KDC, it will accept the reply from the rogue server and let john in.

There is a configuration parameter that can be set to protect the workstation from this type of attack. It will have SSSD authenticate the KDC, and block the login if the KDC cannot be verified. This option is called `krb5_validate`, and it's `false` by default.

To enable it, edit `/etc/sss/sss.conf` and add this line to the domain section:

```
[sss]
config_file_version = 2
domains = example.com

[domain/example.com]
id_provider = ldap
...
krb5_validate = True
```

The second step is to create a host principal on the KDC for this workstation. This is how the KDC's authenticity is verified. It's like a "machine account", with a shared secret that the attacker cannot control and replicate in the rogue KDC. The host principal has the format `host/<fqdn>@REALM`.

After the host principal is created, its keytab needs to be stored on the workstation. This two step process can be easily done on the workstation itself via `kadmin` (not `kadmin.local`) to contact the KDC remotely:

```
$ sudo kadmin -p ubuntu/admin

kadmin: addprinc -randkey host/ldap-krb-client.example.com@EXAMPLE.COM
WARNING: no policy specified for host/ldap-krb-client.example.com@EXAMPLE.COM;
defaulting to no policy
Principal "host/ldap-krb-client.example.com@EXAMPLE.COM" created.

kadmin: ktadd -k /etc/krb5.keytab host/ldap-krb-client.example.com
Entry for principal host/ldap-krb-client.example.com with kvno 6, encryption type
aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/ldap-krb-client.example.com with kvno 6, encryption type
aes128-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
```

Then exit the tool and make sure the permissions on the keytab file are tight:

```
$ sudo chmod 0600 /etc/krb5.keytab
```

```
$ sudo chown root:root /etc/krb5.keytab
```

You can also do it on the KDC itself using `kadmin.local`, but you will have to store the keytab temporarily in another file and securely copy it over to the workstation.

Once these steps are complete, you can restart SSSD on the workstation and perform the login. If the rogue KDC notices the attempt and replies, it will fail the host verification. With debugging we can see this happening on the workstation:

```
==> /var/log/sss/krb5_child.log <==
(Mon Apr 20 19:43:58 2020) [[sss[krb5_child[2102]]] [validate_tgt] (0x0020): TGT
failed verification using key for [host/ldap-krb-client.example.com@EXAMPLE.COM].
(Mon Apr 20 19:43:58 2020) [[sss[krb5_child[2102]]] [get_and_save_tgt] (0x0020):
1741: [-1765328377][Server host/ldap-krb-client.example.com@EXAMPLE.COM not found in
Kerberos database]
```

And the login is denied. If the real KDC picks it up, however, the host verification succeeds:

```
==> /var/log/sss/krb5_child.log <==
(Mon Apr 20 19:46:22 2020) [[sss[krb5_child[2268]]] [validate_tgt] (0x0400): TGT
verified using key for [host/ldap-krb-client.example.com@EXAMPLE.COM].
```

And the login is accepted.

If you run into difficulties, refer to our troubleshooting guide.

Troubleshooting SSSD

Here are some tips to help troubleshoot SSSD.

debug_level

The debug level of SSSD can be changed on-the-fly via `sssctl`, from the `sss-tools` package:

```
$ sudo apt install sss-tools
```

```
$ sssctl debug-level <new-level>
```

Or add it to the config file and restart SSSD:

```
[sss]
config_file_version = 2
domains = example.com

[domain/example.com]
debug_level = 6
...
```

Either approach will yield more logs in `/var/log/sss/*.log` and can help identify what is happening. The `sssctl` approach has the clear advantage of not having to restart the service.

Caching

Caching is useful to speed things up, but it can get in the way big time when troubleshooting. It's useful to be able to remove the cache while chasing down a problem. This can also be done with the `sssctl` tool from the `sss-tools` package.

You can either remove the whole cache:

```
$ sssctl cache-remove

Creating backup of local data...
SSSD backup of local data already exists, override? (yes/no) [no] yes
Removing cache files...
SSSD= needs to be running. Start SSSD now? (yes/no) [yes] yes
```

Or just one element:

```
$ sssctl cache-expire -u john
```

Or expire everything:

```
$ sssctl cache-expire -E
```

DNS

Kerberos is quite sensitive to [DNS](#) issues. If you suspect something related to DNS, here are two suggestions:

FQDN hostname

Make sure `hostname -f` returns a *fully qualified domain name (FQDN)*. Set it in `/etc/hostname` if necessary, and use `sudo hostnamectl set-hostname <fqdn>` to set it at runtime.

Reverse name lookup

You can try disabling a default reverse name lookup, which the `krb5` libraries do, by editing (or creating) `/etc/krb5.conf` and setting `rdns = false` in the `[libdefaults]` section:

```
[libdefaults]
rdns = false
```

See also

- Explanation: *Introduction to network user authentication with SSSD* (page 847)

Smart card authentication

One of the most popular uses for smart cards is to control access to computer systems. The owner must physically *have* the smart card, and they must know the PIN to unlock it. This provides a higher degree of security than single-factor authentication (such as just using a password). In this page, we describe how to enable smart card authentication on Ubuntu.

Note:

This guide is meant for Ubuntu Server 20.04 and newer. If you want to configure a desktop installation refer to the [desktop guide](#)¹⁴¹.

¹⁴¹ <https://ubuntu.com/tutorials/how-to-use-smart-card-authentication-in-ubuntu-desktop#2-overview>

Software requirements

The following packages must be installed to obtain a smart card configuration on Ubuntu:

- **pcscd**: contains the drivers needed to communicate with the CCID smart card readers
- **opensc-pkcs11**: (optional, depending on your smart card hardware) contains the smart card drivers, such as Personal Identify Verification (PIV) or Common Access Card (CAC)
- **sssd**: the authentication daemon that manages smart card access and certificate verification

To install these packages, run the following command in your terminal:

```
$ sudo apt install opensc-pkcs11 pcscd sssd libpam-sss
```

Hardware requirements

Any PIV or CAC smart card with the corresponding reader should be sufficient. USB smart cards like Yubikey embed the reader, and work like regular PIV cards.

Each smart card is expected to contain an X.509 certificate and the corresponding private key to be used for authentication.

Smart card PKCS#11 modules

While `opensc-pkcs11` supports a wide number of smart cards, some of them may require specific PKCS#11 modules, and you must refer to your vendor to install the proper one. From Ubuntu 20.04 onward, all modules supported by `p11-kit`¹⁴² can be used.

If custom PKCS#11 modules are used, you need to ensure that `p11-kit` is properly configured¹⁴³.

In any case, `p11-kit` can be used to see all the configured modules that can be used for authentication:

```
$ p11-kit list-modules

p11-kit-trust: p11-kit-trust.so
  library-description: PKCS#11 Kit Trust Module
  library-manufacturer: PKCS#11 Kit
  library-version: 0.23
  token: System Trust
    manufacturer: PKCS#11 Kit
    model: p11-kit-trust
    serial-number: 1
    hardware-version: 0.23
    flags:
      write-protected
      token-initialized
opensc-pkcs11: opensc-pkcs11.so
  library-description: OpenSC smartcard framework
  library-manufacturer: OpenSC Project
  library-version: 0.20
  token: MARCO TREVISAN (PIN CNS0)
    manufacturer: IC: STMicroelectronics; mask:...
    model: PKCS#15 emulated
    serial-number: 6090010669298009
    flags:
      login-required
      user-pin-initialized
      token-initialized
      user-pin-locked
```

X.509 smart card certificates

The authentication is based on X.509 certificate validation and a smart card can provide one or more certificates that can be used for this purpose.

Before continuing, you may need to export or reference the certificate ID that must be used and associated to each user; such operations can be performed in one of the following three ways:

¹⁴² <https://p11-glue.github.io/p11-glue/p11-kit.html>

¹⁴³ <https://p11-glue.github.io/p11-glue/p11-kit/manual/config.html>

Using p11tool

This is a more generic implementation that just uses the PKCS#11 protocol so it should work with all modules:

```
$ sudo apt install gnutls-bin > p11tool --list-tokens
```

Alternatively, URLs can be listed via:

```
$ p11tool --list-token-urls

Token 1:
  URL: pkcs11:model=PKCS%2315%20emulated;manufacturer=IC%3A%20Infineon%3B%20mask%3A%20IDEMIA%20%280...;serial=6090033068507002;token=MARCO%20TREVISAN%20%28PIN%20CNS1%29
  Label: MARCO TREVISAN (PIN CNS1)
  Type: Hardware token
  Flags: Requires login
  Manufacturer: IC: Infineon; mask: IDEMIA (0...
  Model: PKCS#15 emulated
  Serial: 6090033068507002
  Module: opensc-pkcs11.so
```

The command above will show all the available smart cards in the system and their associated PKCS#11 URI. Copy the URI token of the selected card in the following command, which prints all certificates that can be used for authentication and their associated token URIs.

```
$ p11tool --list-all-certs 'pkcs11:token=[TOKEN-ID]'
```

So in the above example:

```
$ p11tool --list-all-certs
'pkcs11:token=MARCO%20TREVISAN%20%28PIN%20CNS1%29'

Object 0:
  URL: pkcs11:model=PKCS%2315%20emulated;manufacturer=IC%3A%20Infineon%3B%20mask%3A%20IDEMIA%20%280...;serial=6090033068507002;token=MARCO%20TREVISAN%20%28PIN%20CNS1%29;id=%02;object=CNS1;type=cert
  Type: X.509 Certificate (RSA-2048)
  Expires: ven 17 dic 2027, 00:00:00
  Label: CNS1
  ID: 02
```

Now, once the URI of the certificate that will be used for authentication is known, let's extract the **Common Name** from the certificate. In the example we are assuming that our certificate URI is `pkcs11:id=%02;type=cert`.

It can be exported as text Privacy Enhanced Mail (PEM) format using:

```
$ p11tool --export 'pkcs11:id=%02;type=cert'
```

Using opensc

```
$ sudo apt install opensc
```

Certificates can be via:

```
$ pkcs15-tool --list-certificates
```

And exported using

```
$ pkcs15-tool --read-certificate [CERTIFICATE_ID]
```

So, for example:

```
$ pkcs15-tool --list-certificates

Using reader with a card: Alcor Micro AU9560 00 00
X.509 Certificate [CNS1]
  Object Flags   : [0x00]
  Authority      : no
  Path           : 3f00140090012002
  ID             : 02
  Encoded serial : 02 10 0357B1EC0EB725BA67BD2D838DDF93D5
```

```
$ pkcs15-tool --read-certificate 2

Using reader with a card: Alcor Micro AU9560 00 00
-----BEGIN CERTIFICATE-----
MIIHXDCCBUSgAwIBAgIQA1ex7A6.....
```

Troubleshooting

The card certificate verification can be simulated using openssl:

```
$ sudo apt install openssl
```

Save the certificate, using one of the methods stated above:

```
$ pkcs15-tool --read-certificate 2 > card-cert.pem
```

```
$ p11tool --export 'pkcs11:id=%02;type=cert' > card-cert.pem
```

See the certificate contents with:

```
$ openssl x509 -text -noout -in card-cert.pem
```

Verify it is valid for the given CA, where 'Ca-Auth-CERT.pem' contains all the certificates chain:

```
$ openssl verify -verbose -CAfile CA-Auth-CERT.pem card-cert.pem
```

If only the parent CA Certificate is available, you can use `-partial_chain`:

```
$ openssl verify -verbose -partial_chain -CAfile intermediate_CA_cert.pem
```

PAM configuration

To enable smart card authentication we should rely on a module that allows PAM supported systems to use X.509 certificates to authenticate logins. The module relies on a PKCS#11 library, such as `opensc-pkcs11` to access the smart card for the credentials it will need.

When a PAM smart card module is enabled, the login process is as follows:

1. Enter login
2. Enter PIN
3. Validate the X.509 certificate
4. Map the certificate to a user
5. Verify the login and match

To enable that process we have to configure the PAM module, add the relevant certificate authorities, add the PAM module to PAM configuration and set the mapping of certificate names to logins.

Setup guide

This configuration uses SSSD as authentication mechanism, and the example shown here is showing a possible usage for local users, but more complex setups using external remote identity managers such as [FreeIPA](#), LDAP, Kerberos or others can be used.

Refer to [SSSD documentation](#)¹⁴⁴ to learn more about this.

Enable SSSD PAM service

Pam service must be enabled in SSSD configuration, it can be done by ensuring that `/etc/sss/sss.conf` contains:

```
[sssd]
services = pam

[pam]
pam_cert_auth = True
```

Further `[pam]` configuration options can be changed according to the [sssd.conf\(5\)](#)¹⁴⁵ manual page.

¹⁴⁴ <https://sssd.io/docs/introduction.html>

¹⁴⁵ <https://manpages.ubuntu.com/manpages/resolute/man5/sss.conf.5.html>

Configure SSSD Certificate Authorities database

The card certificate must be allowed by a Certificate Authority, these should be part of `/etc/sss/pki/sss_auth_ca_db.pem` (or any other location configured in `[pam]` config section of `sss.conf` as `pam_cert_db_path`).

As per SSSD using openssl, we need to add the whole certificates chain to the SSSD CA certificates path (if not changed via `sss.certificate_verification`), so adding the certificates to the `pam_cert_db_path` is enough:

```
$ sudo cat Ca-Auth-CERT*.pem >> /etc/sss/pki/sss_auth_ca_db.pem
```

Certification Revocation List can be also defined in `sss.conf`, providing a CRL file path in PEM format

```
[sss]
crl_file = /etc/sss/pki/sss_auth_crl.pem
soft_crl = /etc/sss/pki/sss_auth_soft_crl.pem
```

In case that a full certificate authority chain is not available, openssl won't verify the card certificate, and so SSSD should be instructed about.

This is not suggested, but it can be done changing `/etc/sss/sss.conf` so that it contains:

```
[sss]
certificate_verification = partial_chain
```

Troubleshooting

Card certificate verification can be simulated using SSSD tools directly, by using the command SSSD's `p11_child`:

22.04 LTS onward

```
$ sudo /usr/libexec/sss/p11_child --pre -d 10 --debug-fd=2
--ca_db=/etc/sss/pki/sss_auth_ca_db.pem
```

20.04 LTS and earlier

```
$ sudo /usr/libexec/sss/p11_child --pre -d 10 --debug-fd=2
--nssdb=/etc/sss/pki/sss_auth_ca_db.pem
```

If certificate verification succeeds, the tool should output the card certificate name, its ID and the certificate itself in base64 format (other than debug data):

```
(Mon Sep 11 16:33:32:129558 2023) [p11_child[1965]] [do_card] (0x4000): Found
certificate has key id [02].
MARCO TREVISAN (PIN CNS1)
/usr/lib/x86_64-linux-gnu/pkcs11/opensc-pkcs11.so
02
CNS1
MIIHXDCBUSgAwIBAgIQ1ex7....
```

For checking if the smart card works, without doing any verification check (and so for debugging purposes the option) `--verify=no_ocsp` can also be used, while `--verify=partial_chain` can be used to do partial CA verification.

Map certificates to user names

The sss PAM module allows certificates to be used for login, though our Linux system needs to know the username associated to a certificate. SSSD provides a variety of cert mappers to do this. Each cert mapper uses specific information from the certificate to map to a user on the system. The different cert mappers may even be stacked. In other words, if the first defined mapper fails to map to a user on the system, the next one will be tried, and so on until a user is found.

For the purposes of this guide, we will use a simple local user mapping as reference.

Mapping for more complex configurations can be done following the official [SSSD documentation](https://sssd.io/design-pages/matching_and_mapping_certificates.html)¹⁴⁶ depending on [providers](https://sssd.io/design-pages/certmaps_for_LDAP_AD_file.html)¹⁴⁷. For up-to-date information on certificate mapping, please also consult the [sss-certmap\(5\)](https://manpages.ubuntu.com/manpages/resolute/man5/sss-certmap.5.html)¹⁴⁸ manual page.

Local users mapping

When using only local users, SSSD can be easily configured to define an `implicit_domain` that maps all the local users.

Certificate mapping for local users can be easily done using the certificate Subject check, in our example:

```
$ openssl x509 -noout -subject -in card-cert.pem | sed "s/, /,/g;s/ = /=/g"
subject=C=IT,O=Actalis S.p.A.,OU=REGIONE TOSCANA,SN=TREVISAN,GN=MARCO,CN=TRVMRC[...
data-removed...]/6090033068507002.UyMnHxFF3gkAeBYHhxa6V1Edazs=
```

So we can use for the user foo:

```
[sssd]
enable_files_domain = True
services = pam

[certmap/implicit_files/foo]
```

(continues on next page)

¹⁴⁶ https://sssd.io/design-pages/matching_and_mapping_certificates.html

¹⁴⁷ https://sssd.io/design-pages/certmaps_for_LDAP_AD_file.html

¹⁴⁸ <https://manpages.ubuntu.com/manpages/resolute/man5/sss-certmap.5.html>

(continued from previous page)

```
matchrule = <SUBJECT>.*CN=TRVMRC[A-Z0-9]+/6090033068507002\ .UyMnHxFF3gkAeBYHhxa6V1Edazs=.*
```

```
[pam]
```

```
pam_cert_auth = True
```

Troubleshooting

User mapping can be tested working in versions newer than Ubuntu 20.04 with:

```
$ sudo dbus-send --system --print-reply \ >
--dest=org.freedesktop.sssd.infopipe \ > /org/freedesktop/sss/infopipe/Users
\ > org.freedesktop.sssd.infopipe.Users.ListByCertificate \ > string:"$(cat
card-cert.pem)" uint32:10
```

That should return the object path containing the expected user ID:

```
method return time=1605127192.698667 sender=:1.1628 -> destination=:1.1629 serial=6
reply_serial=2
  array [
    object path "/org/freedesktop/sss/infopipe/Users/implicit_5ffiles/1000"
  ]
```

Basic SSSD configuration

The SSSD configuration for accessing to the system is out of the scope of this document, however for smart card login it should contain at least such values:

```
[sssd]
```

```
# Comma separated list of domains
;domains = your-domain1, your-domain2
```

```
# comma-separated list of SSSD services
# pam might be implicitly loaded already, so the line is optional
services = pam
```

```
# You can enable debug of the SSSD daemon
# Logs will be in /var/log/sss/sss.log
;debug_level = 10
```

```
# A mapping between the SC certificate and users
;[certmap/your-domain1/<username>]
;matchrule = <SUBJECT>.*CN=<REGEX MATCHING YOUR CN>.*
```

```
[pam]
```

```
pam_cert_auth = True
```

```
# The Certificate DB to be used:
# - Needs to be an openssl CA certificates
;pam_cert_db_path = /etc/ssl/certs/ca-certificates.crt
```

(continues on next page)

(continued from previous page)

```
# You can enable debug infos for the PAM module
# Logs will be in /var/log/sss/sss_pam.log
# p11 child logs are in /var/log/sss/p11_child.log
# standard auth logs are in /var/log/auth.log
;pam_verbosity = 10
;debug_level = 10
```

In general what's in the configuration file will affect the way SSSD will call the `p11_child` tool (that is the one in charge for the actual authentication). Check `man sssd.conf` for details.

Remember that this file should be owned by root and have permission set to 600, otherwise won't be loaded and SSSD will not complain gracefully. On errors you can test running SSSD temporary with `sudo sssd -d9 -i`.

Every time the configuration is changed SSSD should be restarted (`systemctl restart sssd`).

Add `pam_sss` to PAM

The next step includes the `pam_sss` module into the PAM stack. There are various ways to do this depending on your local policy. The following example enables smart card support for general authentication.

Edit `/etc/pam.d/common-auth` to include the `pam_sss` module as follows:

For Ubuntu later than 23.10

```
$ sudo pam-auth-update
```

Then you can interactively enable SSSD profiles for smart-card-only or optional smart card access. You can also set this non-interactively.

To use smart-card only authentication:

```
$ $ sudo pam-auth-update --disable sss-smart-card-optional --enable
sss-smart-card-required
```

To use smart-card authentication with fallback:

```
$ sudo pam-auth-update --disable sss-smart-card-required --enable
sss-smart-card-optional
```


For Ubuntu 23.10 and earlier

```
$ require SSSD smart card login  
  
auth    [success=done default=die]    pam_sss.so allow_missing_name require_cert_  
auth
```

or only try to use it:

```
$ try SSSD smart card login  
  
auth    [success=ok default=ignore]    pam_sss.so allow_missing_name try_cert_auth
```

See [pam.conf\(5\)](#)¹⁴⁹, [pam_sss\(8\)](#)¹⁵⁰ for further details.

Warning:

A global configuration such as this requires a smart card for su and sudo authentication as well!

If you want to reduce the scope of this module, move it to the appropriate PAM configuration file in /etc/pam.d and ensure that it's referenced by pam_p11_allowed_services in sssd.conf.

The OS is now ready to do a smart card login for the user foo.

Troubleshooting

pamtester is your friend!

To get better debug logging, also increase the SSSD verbosity by changing /etc/sss/sss.conf so that it has:

```
[pam]  
pam_verbosity = 10  
debug_level = 10
```

You can use it to check your configuration without having to login/logout for real. First install it:

```
$ sudo apt install pamtester
```

Run the authentication service as standalone:

```
$ pamtester -v login $USER authenticate
```

Run the authentication service to get user from cert:

¹⁴⁹ <https://manpages.ubuntu.com/manpages/resolute/man5/pam.conf.5.html>

¹⁵⁰ https://manpages.ubuntu.com/manpages/resolute/man8/pam_sss.8.html

```
$ pamtester -v login "" authenticate
```

You can check what happened in the logs:

```
$ sudo less /var/log/auth.log
```

```
$ sudo less /var/log/sss/sss_pam.log
```

```
$ sudo less /var/log/sss/p11_child.log
```

Smart card authentication with SSH

One of the authentication methods supported by the SSH protocol is public key authentication. A public key is copied to the SSH server where it is stored and marked as authorized. The owner of the corresponding private key in the smart card can then SSH login to the server.

We will use `opensc-pkcs11` on the client to access the smart card drivers, and we will copy the public key from the smart card to the SSH server to make the authentication work.

The following instructions apply to Ubuntu 18.04 later.

Server configuration

The SSH server and client must be configured to permit smart card authentication.

Configure the SSH server

The SSH server needs to allow public key authentication set in its configuration file and it needs the user's public key.

Ensure the server has the `PubkeyAuthentication` option set to 'yes' in its `/etc/ssh/sshd_config` file. In a default `/etc/ssh/sshd_config` in Ubuntu, the `PubkeyAuthentication` option is commented out. However, the default is 'yes'. To ensure the setting, edit the `sshd_config` file and set accordingly.

```
PubkeyAuthentication yes
```

Restart the SSH server

```
$ sudo systemctl restart sshd
```

Set the public key on the server

Extract the user's public key from the smart card on the SSH client. Use `ssh-keygen` to read the public key from the smart card and into a format consumable for SSH.

```
$ ssh-keygen -D /usr/lib/x86_64-linux-gnu/opensc-pkcs11.so > smartcard.pub
```

Copy this key to the SSH server.

```
ssh-copy-id -f -i smartcard.pub ubuntu@server-2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "smartcard.pub"
ubuntu@server-2's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'ubuntu@server-2'"
and check to make sure that only the key(s) you wanted were added.
```

Client configuration

The SSH client needs to identify its PKCS#11 provider. To do that set the `PKCS11Provider` option in the `~/.ssh/config` file of each user desiring to use SSH smart card login.

```
PKCS11Provider /usr/lib/x86_64-linux-gnu/opensc-pkcs11.so
```

Use this method to enforce SSH smart card login on a per user basis.

After this step you can SSH into the server using the smart card for authentication.

SSH authentication

See [this page on SSH authentication with smart cards](#) (page 233).

Cryptography

The Secure Shell (SSH) cryptographic protocol that provides secure channels on an unsecured network. In Ubuntu, OpenSSH is the most commonly used implementation of SSH. It provides a suite of utilities for encrypting data transfers and can also be used for remote login and authentication.

- [Obtain TLS certificates](#) (page 245) covers both Let's Encrypt (for internet-facing servers) and running your own Certificate Authority (CA) for internal networks

OpenSSH server

OpenSSH is a powerful collection of tools for remotely controlling networked computers and transferring data between them. Here we'll describe some of the configuration settings possible with the OpenSSH server application and how to change them on your Ubuntu system.

OpenSSH is a freely available version of the Secure Shell (SSH) protocol family of tools. Traditional tools, such as `telnet` or `rcp`, are insecure and transmit the user's password in cleartext when used. OpenSSH provides a server daemon and client tools to facilitate secure, encrypted, remote control and file transfer operations, effectively replacing the legacy tools.

The OpenSSH server component, `sshd`, listens continuously for client connections from any of the client tools. When a connection request occurs, `sshd` sets up the correct connection depending on the type of client tool connecting. For example, if the remote computer is connecting with the SSH client application, the OpenSSH server sets up a remote control session after authentication. If a remote user connects to an OpenSSH server with `scp`, the OpenSSH server daemon initiates a secure copy of files between the server and client after authentication.

OpenSSH can use many authentication methods, including plain password, public key cryptography, and Kerberos tickets.

Install OpenSSH

To install the OpenSSH client applications on your Ubuntu system, use this command at a terminal prompt:

```
$ sudo apt install openssh-client
```

To install the OpenSSH server application, and related support files, use this command at a terminal prompt:

```
$ sudo apt install openssh-server
```

Configure OpenSSH

To configure the default behavior of the OpenSSH server application, `sshd`, you can modify the main configuration file at `/etc/ssh/sshd_config` or add modular configuration snippets to the `/etc/ssh/sshd_config.d/` directory.

Important:

By default, Ubuntu's configuration includes the line `Include /etc/ssh/sshd_config.d/*.conf` at the very top of `/etc/ssh/sshd_config`. Because OpenSSH uses the first value set for most directives, any settings defined in files within `sshd_config.d/` will override those in the main configuration file.

Using snippets in `sshd_config.d/` makes updates easier and keeps your custom changes separate from system defaults.

For information about the configuration directives, refer to the online [sshd_config\(5\)](https://manpages.ubuntu.com/manpages/resolute/man5/sshd_config.5.html)¹⁵¹ manual page or run `man sshd_config` at a terminal prompt.

Example configuration directive

Let's take a look at an example of a configuration directive change. To make your OpenSSH server display the contents of the `/etc/issue.net` file as a pre-login banner, you can add or modify this line in the `/etc/ssh/sshd_config` or in a snippet file in `/etc/ssh/sshd_config.d/`:

```
Banner /etc/issue.net
```

Since losing an SSH server might mean losing your way to reach a server, check the configuration after changing it and before restarting the server:

```
$ sudo sshd -t
```

After making changes to the configuration, save the file. Then, restart the `sshd` server application to effect the changes using the following command:

```
$ sudo systemctl restart ssh.service
```

Disable OS information disclosure through service banner

When a client establishes a TCP connection with the OpenSSH server, the server returns a banner providing information about the host OS and OpenSSH versions.

This banner is returned before the client authenticates to the service:

```
nc -nv 127.0.0.1 22
Connection to 127.0.0.1 22 port [tcp/*] succeeded!
SSH-2.0-OpenSSH_10.2p1 Ubuntu-2ubuntu1
```

The OS related information can be suppressed from that banner through the `DebianBanner` configuration directive in the `/etc/ssh/sshd_config` or in a snippet file in `/etc/ssh/sshd_config.d/`:

```
DebianBanner no
```

As shown in the previous example, restart the `sshd` server and verify that the banner no longer contains the snippet announcing the OS related data:

```
nc -nv 127.0.0.1 22
Connection to 127.0.0.1 22 port [tcp/*] succeeded!
SSH-2.0-OpenSSH_10.2p1
```

¹⁵¹ https://manpages.ubuntu.com/manpages/resolute/man5/sshd_config.5.html

Warning:

Many other configuration directives for `sshd` are available to change the server application's behavior to fit your needs. Be advised, however, if your only method of access to a server is SSH, and you make a mistake when configuring `sshd` you may find you are locked out of the server upon restarting it. Additionally, if an incorrect configuration directive is supplied, the `sshd` server may refuse to start, so be particularly careful when editing the configuration on a remote server.

Tip:

You can use [etckeeper](#) (page 557) to track changes in your `/etc/` with `git`. Alternatively, before editing the configuration file, you should make a copy of the original `/etc/ssh/sshd_config` file and protect it from writing so you will have the original settings as a reference and to reuse as necessary. You can do this with the following commands:

```
$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original
```

```
$ sudo chmod a-w /etc/ssh/sshd_config.original
```

SSH keys

SSH allows authentication between two hosts without the need of a password, using cryptographic keys instead. [SSH-key](#) authentication uses a **private key** and a **public key**.

We recommend the **ed25519 Elliptic Curve algorithm**¹⁵² (`-t ed25519`) due to shorter key size and lower computational requirements. Alternatively, you can create a key using the **RSA Algorithm** (`-t rsa -b 4096`, for 4096 bit key size) instead.

To generate a key pair, run the following command:

```
$ ssh-keygen -t ed25519
```

During creation, you will be prompted for a key passphrase, which you would enter to use that key (every time, or instead using an [ssh-agent\(1\)](#)¹⁵³.

By default, the public key is saved in the file `~/.ssh/id_<algorithm>.pub`, while `~/.ssh/id_<algorithm>` is the private key. To allow login for a user via key, append the content of `id_ed25519.pub` (`id_rsa.pub` for RSA) to `target_machine:~/.ssh/authorized_keys` by running:

```
$ ssh-copy-id username@target_machine
```

Finally, double check the permissions on the `authorized_keys` file – only the authenticated user must have write permissions. If the permissions are not correct then change them by:

¹⁵² <https://ed25519.cr.yp.to/>

¹⁵³ <https://manpages.ubuntu.com/manpages/resolute/man1/ssh-agent.1.html>

```
$ chmod go-w .ssh/authorized_keys
```

You should now be able to SSH to the `target_machine` without being prompted for a password.

To troubleshoot this, have a look at the `target_machine`'s live logs of `ssh.service`:

```
$ sudo journalctl -fu ssh.service
```

Since you can have multiple *SSH-keys*, you can configure which to use for which target machine in `~/.ssh/config` (see *ssh_config(5)*¹⁵⁴).

Connection multiplexing

To reduce connection setup times, multiple SSH sessions to the same `target_machine` can reuse the same *TCP* connection.

Configuration to activate multiplexing in `~/.ssh/config`:

```
ControlMaster auto
ControlPath %d/.ssh/ssh_mux_%u@%l_%r@%h:%p
```

Optionally, if you want the session to still be re-usable 1 second after disconnecting, set:

```
ControlPersist 1
```

Import keys from public key servers

These days many users have already SSH keys registered with services like Launchpad or GitHub. Those can be imported with:

```
$ ssh-import-id <username-on-remote-service>
```

The prefix `lp:` is implied and means fetching from Launchpad. The alternative `gh:` will make the tool fetch from GitHub instead.

Two factor authentication

You can add an extra layer of security to the default key-based authentication using two factor authentication. You can add two factor authentication *using U2F/FIDO hardware authentication devices* (page 243). Alternatively, in cases U2F/FIDO hardware authentication devices are unavailable or impractical for your use case you can add it *using HMAC/Time based One Time Passwords (HOTP/TOTP)* (page 239).

¹⁵⁴ https://manpages.ubuntu.com/manpages/resolute/man5/ssh_config.5.html

Handling unstable connections

When working on remote systems via SSH, unstable network connections or accidental disconnects can interrupt your work and terminate running processes. Terminal multiplexers solve this problem by allowing sessions to persist even after disconnection.

Using a *terminal multiplexer* (page 829) like `tmux` or `screen`, you can start a session on the remote machine that continues running independently of your SSH connection. If your connection drops, you can reconnect and reattach to your existing session, resuming your work exactly where you left off without losing any running processes or command output.

Further reading

- [Ubuntu Wiki SSH¹⁵⁵ page.](#)
- [OpenSSH Website¹⁵⁶](#)
- [Advanced OpenSSH Wiki Page¹⁵⁷](#)

Two factor authentication with TOTP/HOTP

For the best two factor authentication (2FA) security, we recommend using hardware authentication devices that support U2F/FIDO. See the guide on *two factor authentication with U2F/FIDO* (page 243) for details. However, if this is not possible or is impractical to implement in your case, TOTP/HOTP based 2FA is an improvement over no two factor at all. Smartphone apps to support this type of 2FA are common, such as Google Authenticator.

Background

The configuration presented here makes public key authentication the first factor, the TOTP/HOTP code the second factor, and makes password authentication unavailable. Apart from the usual setup steps required for public key authentication, all configuration and setup takes place on the server. No changes are required at the client end; the 2FA prompt appears in place of the password prompt.

The two supported methods are [HMAC-based One Time Password \(HOTP\)¹⁵⁸](#) and [Time-based One Time Password \(TOTP\)¹⁵⁹](#). Generally, TOTP is preferable if the 2FA device supports it.

HOTP is based on a sequence predictable only to those who share a secret. The user must take an action to cause the client to generate the next code in the sequence, and this response is sent to the server. The server also generates the next code, and if it matches the one supplied by the user, then the user has proven to the server that they share the secret. A downside of this approach is that if the user generates codes without the server following along, such as in the case of a typo, then the sequence generators can fall “out of sync”. Servers compensate by allowing a gap in the sequence and considering a few subsequent codes to also be valid; if this mechanism is used, then the server “skips ahead” to sync back

¹⁵⁵ <https://help.ubuntu.com/community/SSH>

¹⁵⁶ <http://www.openssh.org/>

¹⁵⁷ <https://wiki.ubuntu.com/AdvancedOpenSSH>

¹⁵⁸ https://en.wikipedia.org/wiki/HMAC-based_one-time_password

¹⁵⁹ https://en.wikipedia.org/wiki/Time-based_one-time_password

up. But to remain secure, this can only go so far before the server must refuse. When HOTP falls out of sync like this, it must be reset using some out-of-band method, such as authenticating using a second backup key in order to reset the secret for the first one.

TOTP avoids this downside of HOTP by using the current timezone-independent date and time to determine the appropriate position in the sequence. However, this results in additional requirements and a different failure mode. Both devices must have the ability to tell the time, which is not practical for a USB 2FA token with no battery, for example. And both the server and client must agree on the correct time. If their clocks are skewed, then they will disagree on their current position in the sequence. Servers compensate for clock skew by allowing a few codes either side to also be valid. But like HOTP, they can only go so far before the server must refuse. One advantage of TOTP over HOTP is that correcting for this condition involves ensuring the clocks are correct at both ends; an out-of-band authentication to reset unfortunate users' secrets is not required. When using a modern smartphone app, for example, the requirement to keep the clock correct isn't usually a problem since this is typically done automatically at both ends by default.

Note:

It is not recommended to configure U2F/FIDO at the same time as TOTP/HOTP. This combination has not been tested, and using the configuration presented here, TOTP/HOTP would become mandatory for everyone, whether or not they are also using U2F/FIDO.

Install required software

From a terminal prompt, install the google-authenticator PAM module:

```
$ sudo apt update && sudo apt install libpam-google-authenticator
```

Note:

The libpam-google-authenticator package is in Ubuntu's universe archive component, which receives best-effort community support only.

Configure users

Since public key authentication with TOTP/HOTP 2FA is about to be configured to be mandatory for all users, each user who wishes to continue using SSH must first set up public key authentication and then configure their 2FA keys by running the user setup tool. If this isn't done first, users will not be able to do it later over SSH, since at that point they won't have public key authentication and/or 2FA configured to authenticate with.

Configure users' key-based authentication

To set up key-based authentication, see “SSH Keys” above. Once this is done, it can be tested independently of subsequent 2FA configuration. At this stage, user authentication should work with keys only, requiring the supply of the private key passphrase only if it was configured. If configured correctly, the user should not be prompted for their password.

Configure users' TOTP/HOTP 2FA secrets

Each user needs to run the setup tool to configure 2FA. This will ask some questions, generate a key, and display a QR code for the user to import the secret into their smartphone app, such as the Google Authenticator app on Android. The tool creates the file `~/.google-authenticator`, which contains a shared secret, emergency passcodes and per-user configuration.

As a user who needs 2FA configured, from a terminal prompt run the following command:

```
$ google-authenticator
```

Follow the prompts, scanning the QR code into your 2FA app as directed.

It's important to plan for the eventuality that the 2FA device gets lost or damaged. Will this lock the user out of their account? In mitigation, it's worthwhile for each user to consider doing one or more of the following:

- Use the 2FA device's backup or cloud sync facility if it has one.
- Write down the backup codes printed by the setup tool.
- Take a photo of the QR code.
- (TOTP only) Scan the QR code on multiple 2FA devices. This only works for TOTP, since multiple HOTP 2FA devices will not be able to stay in sync.
- Ensure that the user has a different authentication path to be able to rerun the setup tool if required.

Of course, any of these backup steps also negate any benefit of 2FA should someone else get access to the backup, so the steps taken to protect any backup should be considered carefully.

Configure the SSH server

Once all users are configured, configure `sshd` itself by creating a new configuration file in `/etc/ssh/sshd_config.d/` with the following directives:

```
KbdInteractiveAuthentication yes
PasswordAuthentication no
AuthenticationMethods publickey,keyboard-interactive
```

Depending on your installation, some of these settings may be configured already, but not necessarily with the values required for this configuration. Check for and adjust existing occurrences of these configuration directives, or add new ones, as required.

Note:

On Ubuntu 20.04 “Focal Fossa” and earlier, use `ChallengeResponseAuthentication yes` instead of `KbdInteractiveAuthentication yes`.

Restart the ssh service to pick up configuration changes:

```
$ sudo systemctl try-reload-or-restart ssh
```

Edit `/etc/pam.d/sshd` and replace the line:

```
@include common-auth
```

with:

```
auth required pam_google_authenticator.so
```

Changes to PAM configuration have immediate effect, and no separate reloading command is required.

Log in using 2FA

Now when you log in using SSH, in addition to the normal public key authentication, you will be prompted for your TOTP or HOTP code:

```
$ ssh ubuntu.server

Enter passphrase for key 'id_rsa':
(ubuntu@ubuntu.server) Verification code:
Welcome to Ubuntu (...)
ubuntu@ubuntu.server:~$
```

Special cases

On Ubuntu, the following settings are default in `/etc/ssh/sshd_config`, but if you have overridden them, note that they are required for this configuration to work correctly and must be restored as follows:

```
UsePAM yes
PubkeyAuthentication yes
```

Remember to run `sudo systemctl try-reload-or-restart ssh` for any changes made to `sshd` configuration to take effect.

Further reading

- [Wikipedia on TOTP](#)¹⁶⁰
- [Wikipedia on HOTP](#)¹⁶¹

Two factor authentication with U2F/FIDO

OpenSSH 8.2 has added [support for U2F/FIDO hardware authentication devices](#)¹⁶². These devices are used to provide an extra layer of security on top of the existing key-based authentication, as the hardware token needs to be present to finish the authentication.

It's very simple to use and setup. The only extra step is to generate a new keypair that can be used with the hardware device. For that, there are two key types that can be used: `ecdsa-sk` and `ed25519-sk`. The former has broader hardware support, while the latter might need a more recent device.

Once the keypair is generated, it can be used as you would normally use any other type of key in OpenSSH. The only requirement is that in order to use the private key, the U2F device has to be present on the host.

Example with U2F

For example, plug the U2F device in and generate a keypair to use with it:

```
$ ssh-keygen -t ecdsa-sk

Generating public/private ecdsa-sk key pair.
You may need to touch your authenticator to authorize key generation. <-- touch device
Enter file in which to save the key (/home/ubuntu/.ssh/id_ecdsa_sk):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ecdsa_sk
Your public key has been saved in /home/ubuntu/.ssh/id_ecdsa_sk.pub
The key fingerprint is:
SHA256:V9PQ1MqaU8FODXdHqDiH9Mxb8XK3o5aVYDQLVl9IFRo ubuntu@focal
```

Now transfer the public part to the server to `~/.ssh/authorized_keys` and you are ready to go:

```
$ ssh -i .ssh/id_ecdsa_sk ubuntu@focal.server

Confirm user presence for key ECDSA-SK
SHA256:V9PQ1MqaU8FODXdHqDiH9Mxb8XK3o5aVYDQLVl9IFRo <-- touch device
Welcome to Ubuntu Focal Fossa (GNU/Linux 5.4.0-21-generic x86_64)
(...)
ubuntu@focal.server:~$
```

¹⁶⁰ https://en.wikipedia.org/wiki/Time-based_one-time_password

¹⁶¹ https://en.wikipedia.org/wiki/HMAC-based_one-time_password

¹⁶² <https://www.openssh.org/txt/release-8.2>

FIDO2 resident keys

FIDO2 private keys consist of two parts: a **key handle** part, stored in the private key file on disk, and a **per-device key**, which is unique to each FIDO2 token and cannot be exported from the token hardware. These are combined by the hardware at authentication time to derive the real key, which is used to sign authentication challenges.

For tokens that are required to move between computers, it can be cumbersome to have to move the private key file first. To avoid this, tokens implementing the newer FIDO2 standard support **resident keys**, where it is possible to retrieve the key handle part of the key from the hardware.

Using resident keys increases the likelihood of an attacker being able to use a stolen token device. For this reason, tokens normally enforce PIN authentication before allowing the download of keys, and users should set a PIN on their tokens before creating any resident keys. This is done via the hardware token management software.

OpenSSH allows resident keys to be generated using the `ssh-keygen` flag `-O resident` at key generation time:

```
$ ssh-keygen -t ecdsa-sk -O resident -O application=ssh:mykeyname

Generating public/private ecdsa-sk key pair.
You may need to touch your authenticator to authorize key generation.
Enter PIN for authenticator:
Enter file in which to save the key (/home/ubuntu/.ssh/id_ecdsa_sk): mytoken
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in mytoken
(...)
```

This will produce a public/private key pair as usual, but it will be possible to retrieve the private key part (the key handle) from the token later. This is done by running:

```
$ ssh-keygen -K

Enter PIN for authenticator:
You may need to touch your authenticator to authorize key download.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Saved ECDSA-SK key ssh:mytoken to id_ecdsa_sk_rk_mytoken
```

It will use the part after `ssh:` from the `application` parameter from before as part of the key filenames:

```
$ ls id_ecdsa_sk_rk_mytoken*

-rw----- 1 ubuntu ubuntu 598 out  4 18:49 id_ecdsa_sk_rk_mytoken
-rw-r--r-- 1 ubuntu ubuntu 228 out  4 18:49 id_ecdsa_sk_rk_mytoken.pub
```

If you set a passphrase when extracting the keys from the hardware token, and later use these keys, you will be prompted for both the key passphrase *and* the hardware key PIN. You will also have to touch the token:

```
$ ssh -i ./id_ecdsa_sk_rk_mytoken ubuntu@focal.server

Enter passphrase for key './id_ecdsa_sk_rk_mytoken':
Confirm user presence for key ECDSA-SK
SHA256:t+l26IgTXeURY6e36wtrq7wVYJtDVZr0+iuobs1CvVQ
User presence confirmed
(...)
```

It is also possible to download and add resident keys directly to ssh-agent by running

```
$ ssh-add -K
```

In this case, no file is written and the public key can be printed by running `ssh-add -L`.

Note:

If you used the `-o verify-required` option when generating the keys, or if that option is set on the SSH server via the `/etc/ssh/sshd_config` setting `PubkeyAuthOptions verify-required`, then using the agent won't work (in Ubuntu 22.04 LTS).

Further reading

- [OpenSSH 8.2 release notes](#)¹⁶³
- [Yubikey documentation for OpenSSH FIDO/FIDO2 usage](#)¹⁶⁴

Obtain TLS certificates

Transport Layer Security (TLS) certificates authenticate your server's identity and enable encrypted connections. There are two main approaches:

- **Let's Encrypt** (recommended for internet-facing servers): a free, automated Certificate Authority (CA) whose certificates are trusted by all major browsers and clients.
- **Manual CA** (for internal networks or air-gapped environments): run your own CA and sign certificates yourself; requires distributing the CA certificate to all clients.

Use Let's Encrypt

Let's Encrypt¹⁶⁵ is operated by the Internet Security Research Group (ISRG). It issues certificates via the Automated Certificate Management Environment (ACME) protocol. Certificates are valid for 90 days and can be renewed automatically.

¹⁶³ <https://www.openssh.org/txt/release-8.2>

¹⁶⁴ https://developers.yubico.com/SSH/Securing_SSH_with_FIDO2.html

¹⁶⁵ <https://letsencrypt.org/>

Prerequisites

- A registered domain name with DNS records pointing to the server.
- Port 80 open in your firewall. The default [HTTP-01 challenge](#)¹⁶⁶ verifies domain ownership by making an HTTP request to your server on port 80, so it must be reachable from the internet during certificate issuance and renewal.

Install Certbot

[Certbot](#)¹⁶⁷ is the recommended ACME client for Let's Encrypt. Install it from the snap store:

```
$ sudo snap install --classic certbot
```

Obtain a certificate

Certbot supports several modes depending on whether a web server is already running.

With an nginx server running:

Certbot's [nginx plugin](#)¹⁶⁸ finds the server block matching the specified domains in your nginx configuration (typically in `/etc/nginx/sites-enabled/`) and adds the necessary TLS directives in place, then reloads nginx:

```
$ sudo certbot --nginx -d example.com -d www.example.com
```

With an Apache server running:

Certbot's [Apache plugin](#)¹⁶⁹ finds the VirtualHost block matching the specified domains in your Apache configuration (typically in `/etc/apache2/sites-enabled/`) and adds the necessary TLS directives in place, then reloads Apache:

```
$ sudo certbot --apache -d example.com -d www.example.com
```

Without a web server (standalone mode):

Certbot starts a temporary web server on port 80 to complete the ACME challenge. Stop any service already listening on port 80 before running this.

```
$ sudo certbot certonly --standalone -d example.com -d www.example.com
```

With a web server managing the document root (webroot mode):

If you want to obtain a certificate without modifying your web server configuration, specify the directory the web server serves:

¹⁶⁶ <https://letsencrypt.org/docs/challenge-types/#http-01-challenge>

¹⁶⁷ <https://certbot.eff.org/>

¹⁶⁸ <https://certbot.eff.org/docs/using.html#nginx>

¹⁶⁹ <https://certbot.eff.org/docs/using.html#apache>

```
$ sudo certbot certonly --webroot -w /var/www/html -d example.com
```

Multiple domains:

Each `-d` flag adds a domain as a Subject Alternative Name (SAN), so a single certificate can cover multiple domains and subdomains. If a certificate for any of the listed domains already exists, Certbot expands it to include all specified domains. Pass the complete desired domain list — both existing and new — in a single command:

```
$ sudo certbot --nginx -d example.com -d www.example.com -d mail.example.com
```

Other challenge methods

The modes above all use the [HTTP-01 challenge](#)¹⁷⁰ and require port 80. For servers not reachable on port 80, or when you need wildcard certificates (e.g., `*.example.com`), you can use the [DNS-01 challenge](#)¹⁷¹ instead: it verifies domain ownership via a DNS TXT record and has no inbound port requirements. Certbot supports DNS-01 via [DNS plugins](#)¹⁷² for common providers (Cloudflare, Route 53, etc.).

For environments where running an ACME client on each server is not practical — such as air-gapped networks or large fleets — an [ACME proxy server](#)¹⁷³ (such as `step-ca`) can act as an internal CA that forwards requests to Let's Encrypt, or a centralised ACME client such as [acme.sh](#)¹⁷⁴ or [lego](#)¹⁷⁵ can manage certificate issuance and distribution centrally.

Locate the issued certificate files

After a successful run, Certbot writes the certificate files to `/etc/letsencrypt/live/<domain>/:`

File	Contents
<code>fullchain.pem</code>	Server certificate plus any intermediate certificates (use this in most service configurations)
<code>privkey.pem</code>	Private key
<code>cert.pem</code>	Server certificate only
<code>chain.pem</code>	Intermediate certificates only

Point your service configuration at the files in this directory. For example in nginx:

¹⁷⁰ <https://letsencrypt.org/docs/challenge-types/#http-01-challenge>

¹⁷¹ <https://letsencrypt.org/docs/challenge-types/#dns-01-challenge>

¹⁷² <https://certbot.eff.org/docs/using.html#dns-plugins>

¹⁷³ <https://smallstep.com/docs/step-ca/>

¹⁷⁴ <https://acme.sh>

¹⁷⁵ <https://go-acme.github.io/lego/>


```
ssl_certificate      /etc/letsencrypt/live/example.com/fullchain.pem;  
ssl_certificate_key  /etc/letsencrypt/live/example.com/privkey.pem;
```

Automatic renewal

Certbot installs a systemd timer that attempts renewal twice a day. Check that it is active:

```
$ sudo systemctl status snap.certbot.renew.timer
```

Test the renewal process without making any changes:

```
$ sudo certbot renew --dry-run
```

Services configured using the Certbot nginx or Apache plugins are reloaded automatically after a successful renewal. For other services, add a renewal hook. Create `/etc/letsencrypt/renewal-hooks/deploy/reload-service.sh`:

```
#!/bin/sh  
systemctl reload <your-service>
```

Then make it executable:

```
$ sudo chmod +x /etc/letsencrypt/renewal-hooks/deploy/reload-service.sh
```

Replace `<your-service>` with the name of the service to reload after renewal (e.g., postfix, dovecot, slapd).

Set up a manual Certificate Authority

If the services on your network require more than a few self-signed certificates it may be worth the additional effort to set up your own internal Certification Authority (CA). Using certificates signed by your own CA allows the various services using the certificates to easily trust other services using certificates issued from the same CA.

Note:

Clients that connect to your services must trust your CA. You will need to install your CA certificate in the trust store of every client, or distribute it to users. See [Install a root CA certificate in the trust store](#) (page 252) for how to add a CA certificate to Ubuntu's trust store.

Create the CA directory structure

First, create the directories to hold the CA certificate and related files:

```
$ sudo mkdir /etc/ssl/CA
```

```
$ sudo mkdir /etc/ssl/newcerts
```

The CA needs a few additional files to operate: one to keep track of the last serial number used by the CA (each certificate must have a unique serial number), and another file to record which certificates have been issued:

```
$ sudo sh -c "echo '01' > /etc/ssl/CA/serial"
```

```
$ sudo touch /etc/ssl/CA/index.txt
```

Configure OpenSSL

The third file needed is a CA configuration file. Though not strictly necessary, it is convenient when issuing multiple certificates. Edit `/etc/ssl/openssl.cnf`, and in the `[ CA_default ]` section, change:

```
dir           = /etc/ssl           # Where everything is kept
database      = $dir/CA/index.txt  # database index file.
certificate    = $dir/certs/cacert.pem # The CA certificate
serial        = $dir/CA/serial     # The current serial number
private_key   = $dir/private/cakey.pem # The private key
```

Create the self-signed root certificate

```
$ openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem
-days 3650
```

You will then be asked to enter the details about the certificate. Next, install the root certificate and key:

```
$ sudo mv cakey.pem /etc/ssl/private/
```

```
$ sudo mv cacert.pem /etc/ssl/certs/
```

Generate a Certificate Signing Request (CSR)

You are now ready to start signing certificates. The first item needed is a Certificate Signing Request (CSR). If the certificate will be used by service daemons, such as Apache, Postfix, Dovecot, etc., a key without a passphrase is often appropriate. Not having a passphrase allows the services to start without manual intervention, usually the preferred way to start a daemon.

Warning:

Running your secure service without a passphrase is convenient but insecure – a compromise of the key means a compromise of the server as well.

To generate the keys for the Certificate Signing Request (CSR) run the following command from a terminal prompt:

```
$ openssl genrsa -des3 -out server.key 2048

Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
```

You can now enter your passphrase. For best security, it should contain **at least** eight characters. The minimum length when specifying `-des3` is four characters. As a best practice it should also include numbers and/or punctuation and not be a word in a dictionary. Also remember that your passphrase is case-sensitive.

Re-type the passphrase to verify. Once you have re-typed it correctly, the server key is generated and stored in the `server.key` file.

Now create the insecure key, the one without a passphrase, and shuffle the key names:

```
$ openssl rsa -in server.key -out server.key.insecure > mv server.key
server.key.secure > mv server.key.insecure server.key
```

The insecure key is now named `server.key`, and you can use this file to generate the CSR without a passphrase.

To create the CSR, run the following command at a terminal prompt:

```
$ openssl req -new -key server.key -out server.csr
```

It will prompt you to enter the passphrase. If you enter the correct passphrase, it will prompt you to enter 'Company Name', 'Site Name', 'Email ID', etc. Once you enter all these details, your CSR will be created and it will be stored in the `server.csr` file.

Sign the certificate

Enter the following to generate a certificate signed by the CA:

```
$ sudo openssl ca -in server.csr -config /etc/ssl/openssl.cnf
```

After entering the password for the CA key, you will be prompted to sign the certificate, and again to commit the new certificate. You should then see a somewhat large amount of output related to the certificate creation.

There should now be a new file, `/etc/ssl/newcerts/01.pem`, containing the same output. Copy and paste everything beginning with the `-----BEGIN CERTIFICATE-----` line and continuing through to the `-----END CERTIFICATE-----` lines to a file named after the *hostname* of the server where the certificate will be installed. For example `mail.example.com.crt` is a descriptive name.

Subsequent certificates will be named `02.pem`, `03.pem`, etc.

Note:

Replace `mail.example.com.crt` with your own descriptive name.

Install the certificate

Copy the new certificate to the host that needs it, and configure the appropriate applications to use it. The default location to install certificates is `/etc/ssl/certs`. This enables multiple services to use the same certificate without overly complicated file permissions.

```
$ sudo cp server.key /etc/ssl/private/
```

```
$ sudo cp mail.example.com.crt /etc/ssl/certs/
```

For applications that can be configured to use a CA certificate, you should also copy the `/etc/ssl/certs/cacert.pem` file to the `/etc/ssl/certs/` directory on each server that needs to trust certificates issued by your CA.

Further reading

- [Let's Encrypt documentation](https://letsencrypt.org/docs/)¹⁷⁶
- [Certbot documentation](https://certbot.eff.org/docs/)¹⁷⁷
- The Wikipedia article on [Transport Layer Security](https://en.wikipedia.org/wiki/Transport_Layer_Security)¹⁷⁸ provides background on TLS and its history.
- [About certificates](#) (page 901) explains the concepts behind public-key cryptography and certificate types.

¹⁷⁶ <https://letsencrypt.org/docs/>

¹⁷⁷ <https://certbot.eff.org/docs/>

¹⁷⁸ https://en.wikipedia.org/wiki/Transport_Layer_Security

Install a root CA certificate in the trust store

Enterprise environments sometimes have a local Certificate Authority (CA) that issues certificates for use within the organization. For an Ubuntu server to be functional, and to trust the hosts in this environment, this CA must be installed in Ubuntu's trust store.

Certificate formats

There are two encoding formats for certificates:

- **Privacy Enhanced Mail (PEM):** These are human-readable and in Base64-encoded **ASCII** format.
- **Distinguished Encoding Rules (DER):** These are encoded in a more compact **binary** format, and not human readable.

To install a certificate in the trust store it must be in PEM format. A PEM certificate starts with the line `-----BEGIN CERTIFICATE-----`. If you see this, you're ready to install. If not, it is probably a DER certificate and needs to be converted before you can install it in the trust store.

Install a PEM-format certificate

Assuming your PEM-formatted root CA certificate is in `local-ca.crt`, run the following commands to install it:

- Generate a random certificate

Answer the questions asked after executing the command. When asked to input a commonName (CN), it should match the hostname of the server.

```
$ openssl req -x509 -new -nodes -keyout local-ca.key -out local-ca.crt
```

- Install the CA certificate package

```
$ sudo apt-get install -y ca-certificates
```

- Copy your certificate to the local CA certificates directory

```
$ sudo cp local-ca.crt /usr/local/share/ca-certificates
```

- Add the certificate to your trust store

```
$ sudo update-ca-certificates
```

- Verify that your certificate is in PEM format

```
$ sudo ls /etc/ssl/certs/ | grep local-ca  
local-ca.pem
```

- You can also verify that your certificate is available in the trust store by selecting a few texts from your certificate and comparing them with the certificate at the end of the trust store file to see if it matches yours.

```
$ sudo cat local-ca.crt
```

```
...  
L4z0d3b41xJtYldofPve  
-----END CERTIFICATE-----
```

```
$ sudo cat /etc/ssl/certs/ca-certificates.crt | grep L4z0d3b41xJtYldofPve  
  
L4z0d3b41xJtYldofPve
```

Note:

It is important that the certificate file has the .crt extension, otherwise it will not be processed.

After this point, you can use tools like `curl` and `wget` to connect to local sites.

Note:

This procedure updates the system trust store and affects applications that rely on the host system's OpenSSL configuration, such as `curl` and `wget`. Snap applications, including snap-packaged browsers, are unlikely to automatically trust certificates installed in the system trust store due to snap confinement.

Uninstall a PEM-format certificate

- Verify that the certificate you'd like to uninstall exists.

```
$ sudo ls /usr/local/share/ca-certificates/local-ca.crt  
  
/usr/local/share/ca-certificates/local-ca.crt
```

- Delete the certificate

```
$ sudo rm /usr/local/share/ca-certificates/local-ca.crt
```

- The certificate still exists in the trust store.

```
$ sudo cat /etc/ssl/certs/ca-certificates.crt | grep  
L4z0d3b41xJtYldofPve  
  
L4z0d3b41xJtYldofPve
```

- Update the trust store

```
$ sudo update-ca-certificates --fresh
```

- Verify that the certificate has been uninstalled

```
$ sudo cat /etc/ssl/certs/ca-certificates.crt | grep  
L4z0d3b41xJtYldofPve
```

Convert from DER to PEM format

You can convert a DER-formatted certificate called `local-ca.der` to PEM form like this:

```
$ sudo openssl x509 -inform der -outform pem -in local-ca.der -out  
local-ca.crt`
```

The CA trust store location

The CA trust store (as generated by `update-ca-certificates`) is available at the following locations:

- As a single file (PEM bundle) in `/etc/ssl/certs/ca-certificates.crt`
- As an OpenSSL-compatible certificate directory in `/etc/ssl/certs`

Virtual Private Network (VPN)

VPNs are commonly used to provide encrypted, secure access to a network. Two of the most popular choices in Ubuntu are OpenVPN and WireGuard VPN.

- [OpenVPN](#) (page 254) is a well-established option that supports many platforms besides Linux
- [WireGuard VPN](#) (page 267) is a modern and performant option that removes a lot of the complexity from configuring a VPN

How to install and use OpenVPN

OpenVPN is a flexible, reliable and secure Virtual Private Networking (VPN) solution. It belongs to the family of SSL/TLS VPN stacks (different from IPsec VPNs). This chapter will show how to install and configure OpenVPN to create a VPN.

Install the server

To install OpenVPN, run the following command in your terminal:

```
$ sudo apt install openvpn easy-rsa
```

Set up the Public Key Infrastructure (PKI)

If you want more than just pre-shared keys, OpenVPN makes it easy to set up a Public Key Infrastructure (PKI) to use SSL/TLS certificates for authentication and key exchange between the VPN server and clients.

OpenVPN can be used in a routed or bridged VPN mode and can be configured to use either UDP or TCP. The port number can be configured as well, but port 1194 is the official one; this single port is used for all communication. [VPN client implementations](#) (page 910) are available for almost anything including all Linux distributions, macOS, Windows and OpenWRT-based WLAN routers.

The first step in building an OpenVPN configuration is to establish a PKI, which consists of:

- A separate certificate (also known as a public key) and private key for the server and each client
- A primary Certificate Authority (CA) certificate and key, used to sign the server and client certificates

OpenVPN supports bi-directional authentication based on certificates, meaning that the client must authenticate the server certificate and the server must authenticate the client certificate before mutual trust is established.

Both the server and the client will authenticate each other by first verifying that the presented certificate was signed by the primary Certificate Authority (CA), and then by testing information in the now-authenticated certificate header, such as the certificate common name or certificate type (client or server).

Set up the Certificate Authority

OpenVPN uses its own internal Public Key Infrastructure (PKI) where every client and server holds a certificate signed by the same CA. This is different from obtaining a public TLS certificate for a web service; the CA here is private and only used to authenticate VPN participants.

To set up your own CA, and generate certificates and keys for an OpenVPN server with multiple clients, first copy the `easy-rsa` directory to `/etc/openvpn`. This will ensure that any changes to the scripts will not be lost when the package is updated. From a terminal, run:

```
$ sudo make-cadir /etc/openvpn/easy-rsa
```

Note:

You can alternatively edit `/etc/openvpn/easy-rsa/vars` directly, adjusting it to your needs.

As a root user, change to the newly created directory `/etc/openvpn/easy-rsa` and run:

```
$ ./easyrsa init-pki && ./easyrsa build-ca
```

The PEM passphrase set when creating the CA will be asked for every time you need to encrypt the output of a command (such as a private key). The encryption here is important, to avoid printing any private key in plain text.

Create server keys and certificates

Next, we will generate a key pair for the server:

```
$ ./easyrsa gen-req myservername nopass
```

parameters must be generated for the OpenVPN server. The following command will place them in `pki/dh.pem`:

```
$ ./easyrsa gen-dh
```

And finally, create a certificate for the server:

```
$ ./easyrsa sign-req server myservername
```

All certificates and keys have been generated in subdirectories. Common practice is to copy them to `/etc/openvpn/`:

```
$ cp pki/dh.pem pki/ca.crt pki/issued/myservername.crt  
pki/private/myservername.key /etc/openvpn/
```

Simple server configuration

Included with your OpenVPN installation are these (and many more) sample configuration files:

```
/usr/share/doc/openvpn/examples/sample-config-files/client.conf  
/usr/share/doc/openvpn/examples/sample-config-files/server.conf
```

If these files under `/usr/share/doc/*` are not available (for example, the system was provisioned as a minimal install), you can fetch them directly from the Ubuntu package repositories:

- [client.conf](https://git.launchpad.net/ubuntu/+source/openvpn/tree/sample/sample-config-files/client.conf)¹⁷⁹
- [server.conf](https://git.launchpad.net/ubuntu/+source/openvpn/tree/sample/sample-config-files/server.conf)¹⁸⁰

Start by copying the example server configuration to `/etc/openvpn/myserver.conf`:

¹⁷⁹ <https://git.launchpad.net/ubuntu/+source/openvpn/tree/sample/sample-config-files/client.conf>

¹⁸⁰ <https://git.launchpad.net/ubuntu/+source/openvpn/tree/sample/sample-config-files/server.conf>

```
$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf
/etc/openvpn/myserver.conf
```

Note:

In Ubuntu 20.04 or older, the file is compressed, so do this instead:

```
$ sudo cp
/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
/etc/openvpn/myserver.conf.gz
```

```
$ sudo gzip -d /etc/openvpn/myserver.conf.gz
```

Edit `/etc/openvpn/myserver.conf` to make sure the following lines are pointing to the certificates and keys you created in the section above.

```
ca ca.crt
cert myservername.crt
key myservername.key
dh dh.pem
tls-crypt ta.key
```

Note:

The `tls-crypt` option was introduced in OpenVPN 2.4.0. If you are running an Ubuntu Server release older than 18.04 LTS (Bionic Beaver), please use `tls-auth ta.key 0` instead.

Complete this set with a TLS Authentication (TA) key in `/etc/openvpn` for `tls-crypt` like this:

```
$ sudo openvpn --genkey secret ta.key
```

Add a config file to `/etc/sysctl.conf.d` to enable IP forwarding:

```
$ echo "net.ipv4.ip_forward = 1" | sudo tee
/etc/sysctl.d/50-enable-ipv4-forwarding.conf
```

Then apply this file through `sysctl`:

```
$ sudo sysctl -p /etc/sysctl.d/50-enable-ipv4-forwarding.conf
```

This is the minimum you need to configure to get a working OpenVPN server. You can consider studying and tweaking all the default settings we got from the sample `server.conf` file. Now you can start the server.

```
$ sudo systemctl start openvpn@myserver
```

Note:

Be aware that the `systemctl start openvpn` is **not** starting the openvpn you just defined. OpenVPN uses templated systemd jobs, `openvpn@CONFIGFILENAME`. So if, for example, your configuration file is `myserver.conf` your service is called `openvpn@myserver`. You can run all kinds of service and `systemctl` commands like `start/stop/enable/disable/preset` against a templated service like `openvpn@server`.

You will find logging and error messages in the journal. For example, if you started a [templated service](#)¹⁸¹ `openvpn@server` you can filter for this particular message source with:

```
$ sudo journalctl -u openvpn@myserver -xe
```

The same templated approach works for all of `systemctl`, which shows the current server status:

```
$ sudo systemctl status openvpn@myserver

Loaded: loaded (/usr/lib/systemd/system/openvpn@.service; disabled; preset:
enabled)
Active: active (running) since Wed 2026-01-21 10:38:01 UTC; 37s ago
Invocation: b2a3d68935e7423e8f822bc0c88db764
Docs: man:openvpn(8)
      https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
      https://community.openvpn.net/openvpn/wiki/HOWTO
Main PID: 10556 (openvpn)
Status: "Initialization Sequence Completed"
Tasks: 1 (limit: 10)
Memory: 1.7M (peak: 1.9M)
CPU: 17ms
CGroup: /system.slice/system-openvpn.slice/openvpn@myserver.service
        └─10556 /usr/sbin/openvpn --daemon ovpn-myserver --status /run/openvpn/
myserver.status 10 --cd /etc/openvpn --script-security 2 --config /etc/openvpn/
myserver.conf --writepid /r>

Jan 21 10:38:01 server ovpn-myserver[10556]: net_iface_up: set tun0 up
Jan 21 10:38:01 server ovpn-myserver[10556]: net_addr_v4_add: 10.8.0.1/24 dev tun0
Jan 21 10:38:01 server ovpn-myserver[10556]: Could not determine IPv4/IPv6 protocol.
Using AF_INET
Jan 21 10:38:01 server ovpn-myserver[10556]: Socket Buffers: R=[212992->212992]
S=[212992->212992]
Jan 21 10:38:01 server ovpn-myserver[10556]: UDPv4 link local (bound): [AF_
INET][undef]:1194
Jan 21 10:38:01 server ovpn-myserver[10556]: UDPv4 link remote: [AF_UNSPEC]
Jan 21 10:38:01 server ovpn-myserver[10556]: MULTI: multi_init called, r=256 v=256
Jan 21 10:38:01 server ovpn-myserver[10556]: IFCONFIG POOL IPv4: base=10.8.0.2
size=253
Jan 21 10:38:01 server ovpn-myserver[10556]: IFCONFIG POOL LIST
Jan 21 10:38:01 server ovpn-myserver[10556]: Initialization Sequence Completed
```

You can enable/disable various OpenVPN services on one system, but you could also let Ubuntu do it for you. There is a config for `AUTOSTART` in `/etc/default/openvpn`. Allowed values are "all", "none" or a space-separated list of names of the VPNs. If empty, "all" is assumed.

¹⁸¹ <https://www.freedesktop.org/software/systemd/man/latest/systemd.unit.html>

The VPN name refers to the VPN configuration file name, i.e., home would be `/etc/openvpn/home.conf`.

If you're running `systemd`, changing this variable requires running `systemctl daemon-reload` followed by a restart of the `openvpn` service (if you removed entries you may have to stop those manually).

After `systemctl daemon-reload`, a restart of the "generic" OpenVPN will restart all dependent services that the generator in `/lib/systemd/system-generators/openvpn-generator` created for your conf files when you called `daemon-reload`.

Now, check if OpenVPN created a `tun0` interface:

```
$ ip addr show dev tun0
```

Due to restricting it to `dev tun0` this will report just that:

```
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UNKNOWN group default qlen 100
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::b5ac:7829:f31e:32c5/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
```

Create client certificates

The VPN client will also need a certificate to authenticate itself to the server. Usually you create a different certificate for each client.

This can be done either on the server (as with the keys and certificates above) and then securely distributed to the client, or the client can generate and submit a request that is sent and signed by the server.

To create the certificate, enter the following in a terminal as a root user:

```
$ ./easyrsa gen-req myclient1 nopass && ./easyrsa sign-req client myclient1
```

If the first command above was done on a remote system, then copy the `.req` file to the CA server. From there, you can import it via `easyrsa import-req /incoming/myclient1.req myclient1`. Then you can go on with the second `sign-req` command.

After this is done, in both cases you will need to copy the following files to the client using a secure method:

- `pki/ca.crt`
- `pki/issued/myclient1.crt`
- `pki/private/myclient1.key`
- `ta.key`

Since the client certificates and keys are only required on the client machine, you can remove them from the server.

Simple client configuration

There are various different OpenVPN client implementations – both with and without *GUIs*. You can read more about clients in [our page on OpenVPN Clients](#) (page 910). For now, we use the command-line/service-based OpenVPN client for Ubuntu, which is part of the same package as the server. So you must install the `openvpn` package again on the client machine:

```
$ sudo apt install openvpn
```

This time, copy the `client.conf` sample config file to `/etc/openvpn/`:

```
$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf  
/etc/openvpn/
```

Copy the following client keys and certificate files you created in the section above to e.g. `/etc/openvpn/` and edit `/etc/openvpn/client.conf` to make sure the following lines are pointing to those files. If you have the files in `/etc/openvpn/` you can omit the path:

```
ca ca.crt  
cert myclient1.crt  
key myclient1.key  
tls-crypt ta.key
```

Note:

The `tls-crypt` option was introduced in OpenVPN 2.4.0. If you are running an Ubuntu Server release older than 18.04 LTS (Bionic Beaver), please use `tls-auth ta.key 1` instead.

And you have to specify the OpenVPN server name or address. Make sure the keyword **client** is in the config file, since that's what enables client mode.

```
client  
remote vpnserver.example.com 1194
```

Now start the OpenVPN client with the same templated mechanism:

```
$ sudo systemctl start openvpn@client
```

You can check the status as you did on the server:

```
$ sudo systemctl status openvpn@client
```

```

❏ openvpn@client.service - OpenVPN connection to client
   Loaded: loaded (/usr/lib/systemd/system/openvpn@.service; disabled; preset:
   enabled)
   Active: active (running) since Wed 2026-01-21 10:54:27 UTC; 55s ago
   Invocation: f65eae29fc8e4403bd06236309a1db2c
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 11361 (openvpn)
     Status: "Initialization Sequence Completed"
    Tasks: 1 (limit: 10)
   Memory: 1.9M (peak: 2.1M)
      CPU: 22ms
   CGroup: /system.slice/system-openvpn.slice/openvpn@client.service
           └─11361 /usr/sbin/openvpn --daemon ovpn-client --status /run/openvpn/
client.status 10 --cd /etc/openvpn --script-security 2 --config /etc/openvpn/client.
conf --writepid /run/openvpn/client.pid

```

```

Jan 21 10:54:27 client systemd[1]: Starting openvpn@client.service - OpenVPN
connection to client...
Jan 21 10:54:27 client ovpn-client[11361]: Note: --cipher is not set. OpenVPN
versions before 2.5 defaulted to BF-CBC as fallback when cipher negotiation failed
in this case. If you need this fallback please add '--data-ciphers-fallback BF-CBC'
to your configuration and/or add BF-CBC to --data-ciphers.
Jan 21 10:54:27 client ovpn-client[11361]: Note: Kernel support for ovpn-dco
missing, disabling data channel offload.
Jan 21 10:54:27 client ovpn-client[11361]: OpenVPN 2.6.15 x86_64-pc-linux-gnu [SSL
(OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
Jan 21 10:54:27 client ovpn-client[11361]: library versions: OpenSSL 3.5.3 16 Sep
2025, LZO 2.10
Jan 21 10:54:27 client systemd[1]: Started openvpn@client.service - OpenVPN
connection to client.
Jan 21 10:54:27 client ovpn-client[11361]: DCO version: N/A
Jan 21 10:54:27 client ovpn-client[11361]: TCP/UDP: Preserving recently used remote
address: [AF_INET]10.185.198.84:1194
Jan 21 10:54:27 client ovpn-client[11361]: Socket Buffers: R=[212992->212992]
S=[212992->212992]
Jan 21 10:54:27 client ovpn-client[11361]: UDPv4 link local: (not bound)
Jan 21 10:54:27 client ovpn-client[11361]: UDPv4 link remote: [AF_INET]10.185.198.
84:1194
Jan 21 10:54:27 client ovpn-client[11361]: TLS: Initial packet from [AF_INET]10.185.
198.84:1194, sid=46d7b1e7 da534d2d
Jan 21 10:54:27 client ovpn-client[11361]: VERIFY OK: depth=1, CN=server
Jan 21 10:54:27 client ovpn-client[11361]: VERIFY KU OK
Jan 21 10:54:27 client ovpn-client[11361]: Validating certificate extended key usage
Jan 21 10:54:27 client ovpn-client[11361]: ++ Certificate has EKU (str) TLS Web
Server Authentication, expects TLS Web Server Authentication
Jan 21 10:54:27 client ovpn-client[11361]: VERIFY EKU OK
Jan 21 10:54:27 client ovpn-client[11361]: VERIFY OK: depth=0, CN=server
Jan 21 10:54:27 client ovpn-client[11361]: Control Channel: TLSv1.3, cipher TLSv1.3
TLS_AES_256_GCM_SHA384, peer certificate: 2048 bits RSA, signature: RSA-SHA256
Jan 21 10:54:27 client ovpn-client[11361]: [server] Peer Connection Initiated with
[AF_INET]10.185.198.84:1194
Jan 21 10:54:27 client ovpn-client[11361]: TLS: move_session: dest=TM_ACTIVE src=TM_
INITIAL reinit_src=1
Jan 21 10:54:27 client ovpn-client[11361]: TLS: tls_multi_process: initial untrusted
session promoted to trusted
Jan 21 10:54:27 client ovpn-client[11361]: PUSH: Received control message: 'PUSH_
REPLY,route-gateway 10.8.0.1,topology subnet,ping 10,ping-restart 120,ifconfig 10.8.
0.2 255.255.255.0,peer-id 1,cipher AES-256-GCM,protocol-flags cc-exit tls-ekm dyn-
tls-crypt,tun-mtu 1500'
Jan 21 10:54:27 client ovpn-client[11361]: OPTIONS IMPORT: --ifconfig/up options
modified
Jan 21 10:54:27 client ovpn-client[11361]: OPTIONS IMPORT: extra related options

```

```
ovpn-myserver[10556]: read UDPv4 [ECONNREFUSED]: Connection refused (fd=7,code=111)
ovpn-myserver[10556]: 10.185.198.41:40732 VERIFY OK: depth=1, CN=server
ovpn-myserver[10556]: 10.185.198.41:40732 VERIFY OK: depth=0, CN=client
ovpn-myserver[10556]: 10.185.198.41:40732 peer info: IV_VER=2.6.15
ovpn-myserver[10556]: 10.185.198.41:40732 peer info: IV_PLAT=linux
ovpn-myserver[10556]: 10.185.198.41:40732 peer info: IV_TCPNL=1
ovpn-myserver[10556]: 10.185.198.41:40732 peer info: IV_MTU=1600
ovpn-myserver[10556]: 10.185.198.41:40732 peer info: IV_NCP=2
ovpn-myserver[10556]: 10.185.198.41:40732 peer info: IV_CIPHERS=AES-256-GCM:AES-128-
GCM:CHACHA20-POLY1305
ovpn-myserver[10556]: 10.185.198.41:40732 peer info: IV_PROTO=990
ovpn-myserver[10556]: 10.185.198.41:40732 peer info: IV_LZO_STUB=1
ovpn-myserver[10556]: 10.185.198.41:40732 peer info: IV_COMP_STUB=1
ovpn-myserver[10556]: 10.185.198.41:40732 peer info: IV_COMP_STUBv2=1
ovpn-myserver[10556]: 10.185.198.41:40732 TLS: move_session: dest=TM_ACTIVE src=TM_INITIAL
reinit_src=1
ovpn-myserver[10556]: 10.185.198.41:40732 TLS: tls_multi_process: initial untrusted
session promoted to trusted
ovpn-myserver[10556]: 10.185.198.41:40732 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_
AES_256_GCM_SHA384, peer certificate: 2048 bits RSA, signature: RSA-SHA256, peer temporary
key: 768 bits X25519MLKEM768
ovpn-myserver[10556]: 10.185.198.41:40732 [client] Peer Connection Initiated with [AF_
INET]10.185.198.41:40732
ovpn-myserver[10556]: MULTI: new connection by client 'client' will cause previous active
sessions by this client to be dropped. Remember to use the --duplicate-cn option if you
want multiple clients using the same certificate or username to concurrently connect.
ovpn-myserver[10556]: MULTI_sva: pool returned IPv4=10.8.0.2, IPv6=(Not enabled)
ovpn-myserver[10556]: MULTI: Learn: 10.8.0.2 -> client/10.185.198.41:40732
ovpn-myserver[10556]: MULTI: primary virtual IP for client/10.185.198.41:40732: 10.8.0.2
ovpn-myserver[10556]: SENT CONTROL [client]: 'PUSH_REPLY,route-gateway 10.8.0.1,topology
subnet,ping 10,ping-restart 120,ifconfig 10.8.0.2 255.255.255.0,peer-id 0,cipher AES-256-
GCM,protocol-flags cc-exit tls-ekm dyn-tls-crypt,tun-mtu 1500' (status=1)
ovpn-myserver[10556]: client/10.185.198.41:40732 Data Channel: cipher 'AES-256-GCM', peer-
id: 0
ovpn-myserver[10556]: client/10.185.198.41:40732 Timers: ping 10, ping-restart 240
ovpn-myserver[10556]: client/10.185.198.41:40732 Protocol options: explicit-exit-notify 1,
protocol-flags cc-exit tls-ekm dyn-tls-crypt
```

```
$ ip addr show dev tun0
```

```
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.2/24 brd 10.8.0.255 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::3c08:9432:f898:d1c0/64 scope link stable-privacy proto kernel_ll
        valid_lft forever preferred_lft forever
```

```
$ ping 10.8.0.1 -c 1
```

```
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.  
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=0.335 ms
```

Note:

The OpenVPN server always uses the first usable IP address in the client network and only that IP is pingable. E.g., if you configured a /24 for the client network mask, the .1 address will be used. The P-t-P address you see in the `ip addr` output above does not usually answer ping requests.

```
$ ip route show dev tun0
```

```
10.8.0.0/24 proto kernel scope link src 10.8.0.2
```

First troubleshooting

-
-
-
-
-
-

Advanced configuration

Advanced routed VPN configuration on server

Note:

Read the OpenVPN [hardening security guide](https://openvpn.net/community-docs/security.html)¹⁸² for further security advice.

¹⁸² <https://openvpn.net/community-docs/security.html>

Advanced bridged VPN configuration on server

DHCP

Prepare interface config for bridging on server

```
network:
  version: 2
  renderer: networkd
  ethernet:
    enp0s31f6:
      dhcp4: no
  bridge:
    br0:
      interfaces: [enp0s31f6]
      dhcp4: no
      addresses: [10.0.1.100/24]
      routes:
        - to: default
          via: 10.0.1.1
      nameservers:
        addresses: [10.0.1.1]
```

```
$ sudo apt update && sudo apt install networkd-dispatcher
```

```
#!/bin/sh
set -e
if [ "$IFACE" = br0 ]; then
    # no networkd-dispatcher event for 'carrier' on the physical interface
    ip link set enp0s31f6 up promisc on
fi
```

```
$ sudo chmod +x /etc/networkd-dispatcher/dormant.d/promisc_bridge
```

Prepare server config for bridging

```
;dev tun
dev tap
;server 10.8.0.0 255.255.255.0
server-bridge 10.0.0.4 255.255.255.0 10.0.0.128 10.0.0.254
```

```
$ sudo systemctl restart openvpn@myserver
```

Prepare client config for bridging

```
dev tap
;dev tun
```

```
$ sudo systemctl restart openvpn@client
```

Using OpenSSL providers

Legacy provider

```
providers legacy default
```

```
$ openvpn --providers legacy default ...
```

Note:

On Ubuntu 22.04 LTS and earlier, legacy algorithms are included in OpenVPN by default.

TPM 2.0

```
providers legacy default tpm2
```

```
$ openvpn --providers legacy default tpm2 ...
```

Note:

The provider order matters here as tpm2 currently requires the legacy provider to load first.

Further reading

- [EasyRSA](#)¹⁸³
- [OpenVPN quick start guide](#)¹⁸⁴
- [easy-openvpn](#)¹⁸⁵
- [OpenVPN Guide](#)¹⁸⁶

WireGuard VPN

WireGuard VPN peer-to-site

another system in the home network

on the router

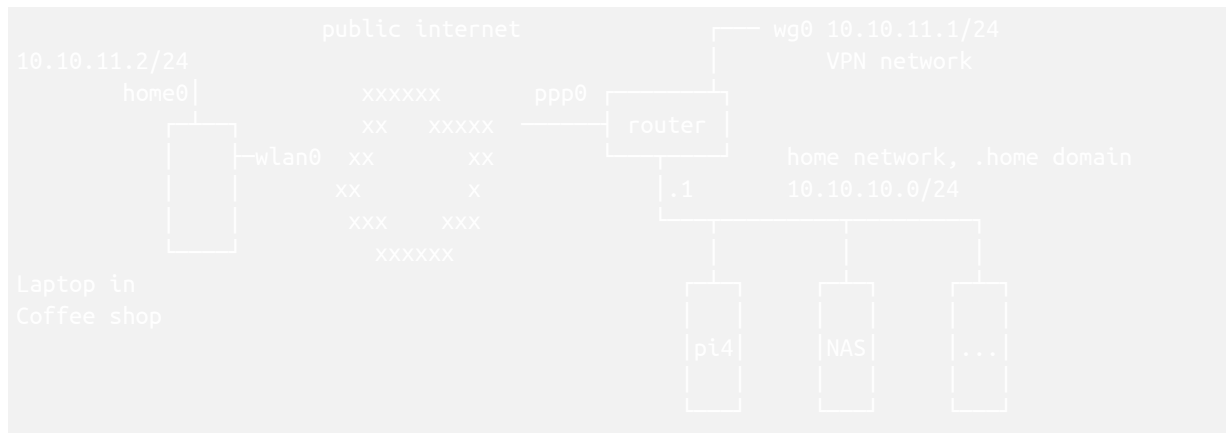
¹⁸³ <https://github.com/OpenVPN/easy-rsa/blob/master/README.quickstart.md>

¹⁸⁴ <https://openvpn.net/as-docs/getting-started.html>

¹⁸⁵ <https://snapcraft.io/easy-openvpn-server>

¹⁸⁶ <https://wiki.debian.org/OpenVPN>

WireGuard VPN peer-to-site (on router)



-
-
-
-
-
-

Configuration

```
$ umask 077
```

```
$ wg genkey > laptop-private.key
```

```
$ wg pubkey < laptop-private.key > laptop-public.key
```

```
$ wg genkey > router-private.key
```

```
$ wg pubkey < router-private.key > router-public.key
```

[Interface]

```
PrivateKey = <contents-of-router-private.key>  
ListenPort = 51000  
Address = 10.10.11.1/24
```

[Peer]

```
PublicKey = <contents-of-laptop-public.key>  
AllowedIPs = 10.10.11.2
```

Important:

This configuration file contains a secret: **PrivateKey**. Make sure to adjust its permissions accordingly, as follows:

```
$ sudo chmod 0600 /etc/wireguard/wg0.conf
```

```
$ sudo chown root: /etc/wireguard/wg0.conf
```

-
-

[Interface]

```
PrivateKey = <contents-of-laptop-private.key>
ListenPort = 51000
Address = 10.10.11.2/24
```

[Peer]

```
PublicKey = <contents-of-router-public.key>
Endpoint = <home-ppp0-IP-or-hostname>:51000
AllowedIPs = 10.10.11.0/24,10.10.10.0/24
```

Important:

As before, this configuration file contains a secret: **PrivateKey**. You need to adjust its permissions accordingly, as follows:

```
$ sudo chmod 0600 /etc/wireguard/home0.conf
```

```
$ sudo chown root: /etc/wireguard/home0.conf
```

Note:

You may have noticed by now that address allocation is manual, and not via something like *DHCP*. Keep tabs on it!

-
-

DNS

-

Default Gateway

Testing

```
$ sudo wg-quick up wg0

[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.10.11.1/24 dev wg0
[#] ip link set mtu 1378 up dev wg0
```

```
$ ip a show dev wg0

9: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1378 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/none
    inet 10.10.11.1/24 scope global wg0
        valid_lft forever preferred_lft forever
```

```
$ ip route | grep wg0

10.10.11.0/24 dev wg0 proto kernel scope link src 10.10.11.1
```

```
$ sudo wg show

interface: wg0
  public key: <router public key>
  private key: (hidden)
  listening port: 51000

peer: <laptop public key>
  allowed ips: 10.10.11.2/32
```



```
$ sudo watch wg show
```

```
$ sudo wg-quick up home0
```

```
[#] ip link add home0 type wireguard  
[#] wg setconf home0 /dev/fd/63  
[#] ip -4 address add 10.10.11.2/24 dev home0  
[#] ip link set mtu 1420 up dev home0  
[#] ip -4 route add 10.10.10.0/24 dev home0
```

```
$ ip a show dev home0
```

```
24: home0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN  
group default qlen 1000  
    link/none  
    inet 10.10.11.2/24 scope global home0  
        valid_lft forever preferred_lft forever
```

```
$ ip route | grep home0
```

```
10.10.10.0/24 dev home0 scope link  
10.10.11.0/24 dev home0 proto kernel scope link src 10.10.11.2
```

```
$ ping -c 3 10.10.10.1
```

```
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.  
64 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=603 ms  
64 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=300 ms  
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=304 ms
```

```
$ sudo wg show

interface: wg0
  public key: <router public key>
  private key: (hidden)
  listening port: 51000

peer: <laptop public key>
  endpoint: <laptop public IP>:51000
  allowed ips: 10.10.11.2/32
  latest handshake: 1 minute, 8 seconds ago
  transfer: 564 B received, 476 B sent
```

WireGuard on an internal system (peer-to-site)

on the home router itself

DHCP

Router changes

Network	10.10.10.0/24
Usable addresses	10.10.10.2 – 10.10.10.254 (.1 is the router)
DHCP range	10.10.10.50 – 10.10.10.254
VPN range	10.10.10.10 – 10.10.10.49

Configure the internal WireGuard endpoint

```
$ sudo apt install wireguard
```

```
$ umask 077
```

```
$ wg genkey > internal-private.key
```

```
$ wg pubkey < internal-private.key > internal-public.key
```

[Interface]

```
Address = 10.10.10.10/32
```

```
ListenPort = 51000
```

```
PrivateKey = <contents of internal-private.key>
```

(continues on next page)

(continued from previous page)

```
[Peer]
# laptop
PublicKey = <contents of laptop-public.key>
AllowedIPs = 10.10.10.11/32 # any available IP in the VPN range
```

Note:

Just like in the *peer-to-site* (page 268) scenario with WireGuard on the router, there is no Endpoint configuration here for the laptop peer, because we don't know where it will be connecting from beforehand.

-
-

```
net.ipv4.ip_forward = 1
net.ipv4.conf.all.proxy_arp = 1
```

```
$ sudo sysctl -p /etc/sysctl.d/70-wireguard-routing.conf -w
```

```
$ sudo wg-quick up wg0
```

Configuring the peer

```
[Interface]
ListenPort = 51000
Address = 10.10.10.11/24
PrivateKey = <contents of the private key for this system>
```

```
[Peer]
PublicKey = <contents of internal-public.key>
Endpoint = <home-ppp0-IP-or-hostname>:51000
AllowedIPs = 10.10.10.0/24
```

```
$ sudo wg-quick up home0
```

Note:

There is no need to add an index number to the end of the interface name. That is a convention, but not strictly a requirement.

Testing

WireGuard VPN site-to-site

-
-
-
-
-

```
ubuntu@b1:~$ mtr -n -r 10.10.10.230
```

Start: 2022-09-02T18:56:51+0000

HOST: b1	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. -- 10.10.11.1	0.0%	10	0.1	0.1	0.1	0.2	0.0
2. -- 10.10.9.0	0.0%	10	299.6	299.3	298.3	300.0	0.6
3. -- 10.10.10.230	0.0%	10	299.1	299.1	298.0	300.2	0.6

Note:

Technically, a /31 Classless Inter-Domain Routing (CIDR) network has no usable IP addresses, since the first one is the network address, and the second (and last) one is the broadcast address. RFC 3021¹⁸⁷ allows for it, but if you encounter routing or other networking issues, switch to a /30 CIDR and its two valid host IPs.

¹⁸⁷ <https://www.ietf.org/rfc/rfc3021.txt>

Configure WireGuard

```
$ sudo apt install wireguard
```

```
$ wg genkey | sudo tee /etc/wireguard/wgA.key
```

```
$ sudo cat /etc/wireguard/wgA.key | wg pubkey | sudo tee  
/etc/wireguard/wgA.pub
```

[Interface]

```
PostUp = wg set %i private-key /etc/wireguard/%i.key  
Address = 10.10.9.0/31  
ListenPort = 51000
```

[Peer]

```
# beta site  
PublicKey = <contents of /etc/wireguard/wgB.pub>  
AllowedIPs = 10.10.11.0/24,10.10.9.0/31  
Endpoint = <beta-gw-ip>:51000
```

```
$ sudo apt install wireguard
```

```
$ wg genkey | sudo tee /etc/wireguard/wgB.key
```

```
$ sudo cat /etc/wireguard/wgB.key | wg pubkey | sudo tee  
/etc/wireguard/wgB.pub
```

[Interface]

```
Address = 10.10.9.1/31  
PostUp = wg set %i private-key /etc/wireguard/%i.key
```

(continues on next page)

(continued from previous page)

```
ListenPort = 51000
```

[Peer]

```
# alpha site
```

```
PublicKey = <contents of /etc/wireguard/wgA.pub>
```

```
AllowedIPs = 10.10.10.0/24,10.10.9.0/31
```

```
Endpoint = <alpha-gw-ip>:51000
```

Important:

WireGuard is being set up on the gateways for these two networks. As such, there are no changes needed on individual hosts of each network, but keep in mind that the WireGuard tunneling and encryption is only happening between the *alpha* and *beta* gateways, and **NOT** between the hosts of each network.

Bring the interfaces up

```
$ sudo systemctl enable --now wg-quick@wgA
```

```
$ sudo systemctl enable --now wg-quick@wgB
```

Firewall and routing

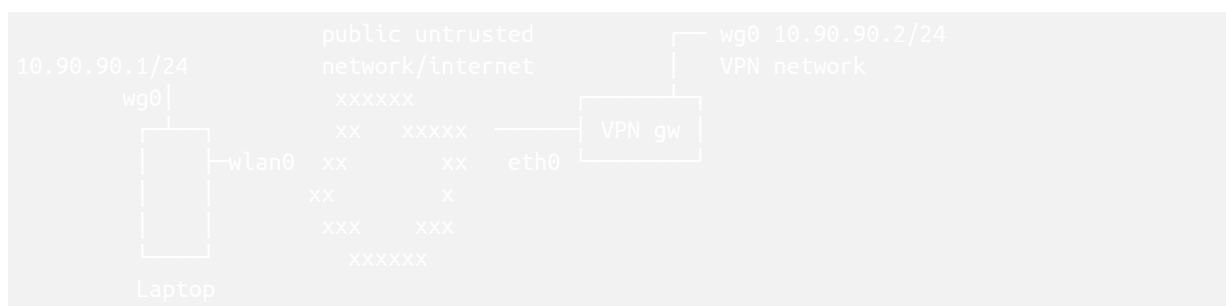
•

-
-
-

```
$ sudo iptables -A FORWARD -i wgA -p tcp --dport 22 -j REJECT
```

```
$ sudo iptables -A FORWARD -i wgB -p tcp --dport 22 -j REJECT
```

Using the VPN as the default gateway



Install and configure WireGuard

```
$ sudo apt install wireguard
```

```
$ umask 077
```

```
$ wg genkey > wg0.key
```

```
$ wg pubkey < wg0.key > wg0.pub
```

```
$ sudo mv wg0.key wg0.pub /etc/wireguard
```

```
$ sudo apt install wireguard
```

```
$ umask 077
```

```
$ wg genkey > gateway0.key
```

```
$ wg pubkey < gateway0.key > gateway0.pub
```

```
$ sudo mv gateway0.key gateway0.pub /etc/wireguard
```

[Interface]

```
PostUp = wg set %i private-key /etc/wireguard/wg0.key
```

```
ListenPort = 51000
```

```
Address = 10.90.90.1/24
```

[Peer]

```
PublicKey = <contents of gateway0.pub>
```

```
Endpoint = <public IP of gateway server>
```

```
AllowedIPs = 0.0.0.0/0
```

-

-

-

```
[Interface]
PostUp = wg set %i private-key /etc/wireguard/%i.key
Address = 10.90.90.2/24
ListenPort = 51000

[Peer]
PublicKey = <contents of wg0.pub>
AllowedIPs = 10.90.90.1/32
```

Routing and masquerading

-

-

```
net.ipv4.ip_forward = 1
```

```
$ sudo sysctl -p /etc/sysctl.d/70-wireguard-routing.conf -w
```

```
$ sudo iptables -t nat -A POSTROUTING -s 10.90.90.0/24 -o eth0 -j MASQUERADE
```

```
#!/bin/sh
iptables -t nat -A POSTROUTING -s 10.90.90.0/24 -o eth0 -j MASQUERADE
```

Testing

```
$ sudo wg-quick up gateway0

[#] ip link add gateway0 type wireguard
[#] wg setconf gateway0 /dev/fd/63
[#] ip -4 address add 10.90.90.2/24 dev gateway0
[#] ip link set mtu 1378 up dev gateway0
[#] wg set gateway0 private-key /etc/wireguard/gateway0.key
```

```
$ sudo wg-quick up wg0

[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.90.90.1/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
[#] wg set wg0 fwmark 51820
[#] ip -4 route add 0.0.0.0/0 dev wg0 table 51820
[#] ip -4 rule add not fwmark 51820 table 51820
[#] ip -4 rule add table main suppress_prefixlength 0
[#] sysctl -q net.ipv4.conf.all.src_valid_mark=1
[#] nft -f /dev/fd/63
[#] wg set wg0 private-key /etc/wireguard/wg0.key
```

```
$ mtr -r 1.1.1.1

Start: 2022-09-01T12:42:59+0000
HOST: laptop.lan
      Loss%  Snt  Last  Avg  Best  Wrst StDev
  1. |-- 10.90.90.2          0.0%   10 184.9 185.5 184.9 186.9   0.6
  2. |-- 10.48.128.1        0.0%   10 185.6 185.8 185.2 188.3   0.9
  (...)
  7. |-- one.one.one.one    0.0%   10 186.2 186.3 185.9 186.6   0.2
```

```
$ ip route

default via 192.168.122.1 dev enp1s0 proto dhcp src 192.168.122.160 metric 100
10.90.90.0/24 dev wg0 proto kernel scope link src 10.90.90.1
192.168.122.0/24 dev enp1s0 proto kernel scope link src 192.168.122.160 metric 100
192.168.122.1 dev enp1s0 proto dhcp scope link src 192.168.122.160 metric 100
```

DNS leaks

DNS

//dnsleaktest.com

https:

Selecting a DNS server

```
$ resolvectl status
```

```
Global
    Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
    resolv.conf mode: stub

Link 2 (ens2)
    Current Scopes: DNS
        Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
    Current DNS Server: 10.48.0.5
        DNS Servers: 10.48.0.5
        DNS Domain: openstacklocal

Link 5 (gateway0)
    Current Scopes: none
        Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
```

[Interface]

```
PostUp = wg set %i private-key /etc/wireguard/wg0.key
PostUp = resolvectl dns %i 10.48.0.5; resolvectl domain %i \~.
ListenPort = 51000
Address = 10.90.90.1/24
```

[Peer]

```
PublicKey = <contents of gateway0.pub>
Endpoint = <public IP of gateway server>
AllowedIPs = 0.0.0.0/0
```

```
$ sudo resolvectl dns wg0 10.48.0.5; sudo resolvectl domain wg0 \~.
```

```
$ sudo wg-quick down wg0; sudo wg-quick up wg0
```

Installing your own DNS server

```
$ sudo apt install bind9
```

[Interface]

```
PostUp = wg set %i private-key /etc/wireguard/wg0.key  
PostUp = resolvectl dns %i 10.90.90.2; resolvectl domain %i |~.  
ListenPort = 51000  
Address = 10.90.90.1/24
```

[Peer]

```
PublicKey = <contents of gateway0.pub>  
Endpoint = <public IP of gateway server>  
AllowedIPs = 0.0.0.0/0
```

Common tasks in WireGuard VPN

Controlling the WireGuard interface with systemd

```
$ sudo systemctl enable wg-quick@wg0
```

```
$ sudo systemctl reload wg-quick@wg0
```

DNS resolving

DNS

-
-
-

[Interface]

...

```
PostUp = resolvectl dns %i 10.10.10.1; resolvectl domain %i \~home
```

*wg-quick(8)*¹⁸⁸

-
-

```
$ sudo wg-quick up home0
```

```
[#] ip link add home0 type wireguard
[#] wg setconf home0 /dev/fd/63
[#] ip -4 address add 10.10.11.2/24 dev home0
[#] ip link set mtu 1420 up dev home0
[#] ip -4 route add 10.10.10.0/24 dev home0
[#] resolvectl dns home0 10.10.10.1; resolvectl domain home0 \~home
```

hostname

```
$ resolvectl status home0
```

¹⁸⁸ <https://manpages.ubuntu.com/manpages/resolute/man8/wg-quick.8.html>

```
Link 26 (home0)
  Current Scopes: DNS
    Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
Current DNS Server: 10.10.10.1
  DNS Servers: 10.10.10.1
  DNS Domain: ~home
```

Note:

The *wg-quick(8)*¹⁸⁹ manual page documents the DNS setting of the WireGuard interface which has the same purpose, but only works if you have `resolveconf` installed. Ubuntu systems by default don't, and rely on `systemd-resolved` instead.

¹⁸⁹ <https://manpages.ubuntu.com/manpages/resolute/man8/wg-quick.8.html>

Adding another peer

- 1.
- 2.
- 3.

```
$ umask 077
```

```
$ wg genkey > ontheroad-private.key
```

```
$ wg pubkey < ontheroad-private.key > ontheroad-public.key
```

```
$ ls -la ontheroad.*
```

```
-rw----- 1 ubuntu ubuntu 45 Aug 22 20:12 ontheroad-private.key
-rw----- 1 ubuntu ubuntu 45 Aug 22 20:13 ontheroad-public.key
```

[Interface]

```
PrivateKey = <contents-of-ontheroad-private.key>
ListenPort = 51000
Address = 10.10.11.3/24
```

(continues on next page)

(continued from previous page)

[Peer]

```
PublicKey = <contents-of-router-public.key>
Endpoint = <home-ppp0-IP-or-hostname>:51000
AllowedIPs = 10.10.11.0/24,10.10.10.0/24
```

[Interface]

```
PrivateKey = <contents-of-router-private.key>
ListenPort = 51000
Address = 10.10.11.1/24
```

[Peer]

```
# laptop
PublicKey = <contents-of-laptop-public.key>
AllowedIPs = 10.10.11.2
```

[Peer]

```
# ontheroad
PublicKey = <contents-of-ontheroad-public.key>
AllowedIPs = 10.10.11.3
```

```
$ systemctl reload wg-quick@wg0
```

Note:

For this case of a “server” or “VPN gateway”, where we are just adding another peer to an existing config, the `systemctl reload` action will work well enough to insert the new peer into the WireGuard configuration. However, it won’t create new routes, or do any of the other steps that `wg-quick` does. Depending on your setup, you might need a full restart so that `wg-quick` can fully do its job.

Adding a smartphone peer

[upstream installation page](#)¹⁹⁰

```
$ sudo apt install qrencode
```

¹⁹⁰ <https://www.wireguard.com/install/>

```
$ cat phone.conf | qrencode -t ansiutf8
```

Important:

Treat this QR code as a secret, as it contains the private key for the WireGuard interface!

Security tips for WireGuard VPN

Traffic goes both ways

Using PreSharedKey

```
$ wg genpsk  
vxlX6eMMin8uhxbKEhe/i0xi8ru+q1qWzCdjESXoFZY=
```

```
[Peer]  
PublicKey = ....  
Endpoint = ....  
AllowedIPs = ....  
PresharedKey = vxlX6eMMin8uhxbKEhe/i0xi8ru+q1qWzCdjESXoFZY=
```

Note:

Both sides need to have the same PresharedKey in their respective [Peer] sections.

Preventing accidental leakage of private keys

```
[Interface]
ListenPort = 51000
Address = 10.10.11.3/24
PostUp = wg set %i private-key /etc/wireguard/%i.key

[Peer]
PublicKey = <contents-of-router-public.key>
Endpoint = 10.48.132.39:51000
AllowedIPs = 10.10.11.0/24,10.10.10.0/24
```

-
-
-

Troubleshooting WireGuard VPN

-
-
-
-
-
-

The watch wg command

```
Every 2.0s: wg                               j-wg: Fri Aug 26 17:44:37 2022

interface: wg0
  public key: +T3T3HTMeyrEDvim8FBxbYjbz+/P0eOtG3Rl9kjmM=
  private key: (hidden)
  listening port: 51000

peer: 2cJdFcNzXv4YUGyDTaht0frbsrFsCByatPnNzKts0Qo=
  endpoint: 10.172.196.106:51000
  allowed ips: 10.10.11.2/32
  latest handshake: 3 hours, 27 minutes, 35 seconds ago
  transfer: 3.06 KiB received, 2.80 KiB sent

peer: ZliZ1hlarZqvfxPMyme2ECtXdk611NB7uzLAD4McpGI=
  allowed ips: 10.10.11.3/32
```

Kernel debug messages

Note:

To be able to use the commands outlined below `kernel_lockdown(7)`¹⁹¹ needs to be disabled. That is because the debug interfaces used are considered a way to tamper with the integrity of the kernel.

Lockdown is enabled by default in secure boot mode. Check the status via:

```
$ cat /sys/kernel/security/lockdown
none [integrity] confidentiality
```

Or that when disabled:

```
$ cat /sys/kernel/security/lockdown
[none] integrity confidentiality
```

To disable it either disable secure boot in *BIOS* or your virtualization platform.

If that is impossible, consider not using the dynamic module debug control shown below, but instead using a kernel boot parameter `wireguard.dyndbg=+p` which has the drawback that it needs a reboot to enable/disable it.

Before 20.04 [Kernel Lockdown](#)¹⁹² was not yet present, therefore such systems can skip this step.

¹⁹¹ https://manpages.ubuntu.com/manpages/resolute/man7/kernel_lockdown.7.html

¹⁹² <https://documentation.ubuntu.com/security/security-features/kernel-protections/#kernel-lockdown>

```
$ echo "module wireguard +p" | sudo tee
/sys/kernel/debug/dynamic_debug/control
```

```
$ sudo dmesg -wT
```

```
$ echo "module wireguard -p" | sudo tee
/sys/kernel/debug/dynamic_debug/control
```

Destination address required

```
$ ping 10.10.11.2
```

```
PING 10.10.11.2 (10.10.11.2) 56(84) bytes of data.  
From 10.10.11.1 icmp_seq=1 Destination Host Unreachable  
ping: sendmsg: Destination address required
```

Required key not available

```
$ ping 10.10.11.1
```

```
PING 10.10.11.1 (10.10.11.1) 56(84) bytes of data.  
From 10.10.11.2 icmp_seq=1 Destination Host Unreachable  
ping: sendmsg: Required key not available
```

```
wireguard: home0: No peer has allowed IPs matching 10.10.11.1
```

See also

- [Introduction to WireGuard VPN](#)

See also

- [Introduction to security](#)
- [Security topics](#)

Our networking section

-
-

Networking

[networking](#)

[introduction to](#)

Configuration

[Configuring networks](#)

[the Netplan documentation](#)¹⁹³

Network tools

¹⁹³ <https://netplan.readthedocs.io/en/stable/>

Set up an FTP server

tation

OpenSSH documen-

-
-

vsftpd – FTP server installation

```
$ sudo apt install vsftpd
```

Anonymous FTP configuration

```
anonymous_enable=YES
```

```
$ sudo systemctl restart vsftpd
```


FTP files location

```
$ sudo mkdir -p /srv/files/ftp
```

```
$ sudo usermod -d /srv/files/ftp ftp
```

```
$ sudo systemctl restart vsftpd.service
```

User authenticated FTP configuration

```
write_enable=YES
```

```
$ sudo systemctl restart vsftpd.service
```

```
anon_upload_enable=YES
```

Warning:

Enabling anonymous FTP upload can be an extreme security risk. It is best to not enable anonymous upload on servers accessed directly from the Internet.

Securing FTP

```
chroot_local_user=YES
```

```
chroot_list_enable=YES  
chroot_list_file=/etc/vsftpd.chroot_list
```

Warning:

If `chroot_local_user` is set to YES (setting all users to be limited to just their home directory), `/etc/vsftpd.chroot_list` becomes a list of users which are NOT limited to just their home directory

```
$ sudo systemctl restart vsftpd.service
```

```
ssl_enable=YES
```

```
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem  
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

Security - Certificates

```
$ sudo systemctl restart vsftpd.service
```

```
# /etc/shells: valid login shells
/bin/csh
/bin/sh
  /usr/bin/es
  /usr/bin/ksh
  /bin/ksh
  /usr/bin/rc
  /usr/bin/tcsh
  /bin/tcsh
  /usr/bin/esh
  /bin/dash
  /bin/bash
/bin/rbash
/usr/bin/screen
/usr/sbin/nologin
```

```
auth    required    pam_shells.so
```

Further reading

- [vsftpd website](http://vsftpd.beasts.org/vsftpd_conf.html)¹⁹⁴

¹⁹⁴ http://vsftpd.beasts.org/vsftpd_conf.html

Domain Name Service (DNS)

domain names (FQDN)

fully qualified

Install DNS

```
$ sudo apt install bind9
```

```
$ sudo apt install dnsutils
```

DNS configuration overview

-
-
-

Configuration files

-
-
-

Set up a caching nameserver

```
options {  
    forwarders {  
        1.2.3.4;  
        5.6.7.8;  
    };  
};
```

Note:

Replace 1.2.3.4 and 5.6.7.8 with the IP addresses of actual nameservers.

```
$ sudo systemctl restart bind9.service
```

See the dig section

Set up a primary server

Forward zone file

```
zone "example.com" {  
    type master;  
    file "/etc/bind/db.example.com",  
};
```

Note:

If BIND will be receiving automatic updates to the file as with [DDNS](#), then use `/var/lib/bind/db.example.com` rather than `/etc/bind/db.example.com` both here and in the copy command below.

FQDN

```
;
; BIND data file for example.com
;
$TTL      604800
@         IN      SOA      example.com. root.example.com. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       ns.example.com.
@         IN      A        192.168.1.10
@         IN      AAAA     2001:db8::10
ns        IN      A        192.168.1.10
```

Common record types

Note:

Many admins like to use the “last edited” date as the Serial of a zone, such as **2020012100** which is **yyyymmddss** (where **ss** is the Serial Number)

```
$ sudo systemctl restart bind9.service
```

Reverse zone file

```
zone "1.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.192";  
};
```

Note:

Replace 1.168.192 with the first three octets of whatever network you are using. Also, name the zone file `/etc/bind/db.192` appropriately. It should match the first octet of your network.

```
;   
; BIND reverse data file for local 192.168.1.XXX net   
;   
$TTL      604800   
@         IN      SOA      ns.example.com. root.example.com. (   
                                2          ; Serial   
                                604800     ; Refresh   
                                86400      ; Retry   
                                2419200    ; Expire   
                                604800 )   ; Negative Cache TTL   
;   
@         IN      NS       ns.   
10        IN      PTR      ns.example.com.
```

```
$ sudo systemctl restart bind9.service
```

Set up a secondary server

```
zone "example.com" {
    type master;
    file "/etc/bind/db.example.com";
    allow-transfer { 192.168.1.11; };
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
    allow-transfer { 192.168.1.11; };
};
```

Note:

Replace 192.168.1.11 with the IP address of your secondary nameserver.

```
$ sudo systemctl restart bind9.service
```

```
zone "example.com" {
    type secondary;
    file "db.example.com";
    masters { 192.168.1.10; };
};

zone "1.168.192.in-addr.arpa" {
    type secondary;
    file "db.192";
    masters { 192.168.1.10; };
};
```

```
$ sudo systemctl restart bind9.service
```

```
client 192.168.1.10#39448: received notify for zone '1.168.192.in-addr.arpa'
zone 1.168.192.in-addr.arpa/IN: Transfer started.
transfer of '100.18.172.in-addr.arpa/IN' from 192.168.1.10#53:
  connected using 192.168.1.11#37531
zone 1.168.192.in-addr.arpa/IN: transferred serial 5
transfer of '100.18.172.in-addr.arpa/IN' from 192.168.1.10#53:
  Transfer completed: 1 messages,
  6 records, 212 bytes, 0.002 secs (106000 bytes/sec)
zone 1.168.192.in-addr.arpa/IN: sending notifies (serial 5)
```

(continues on next page)

(continued from previous page)

```
client 192.168.1.10#20329: received notify for zone 'example.com'
zone example.com/IN: Transfer started.
transfer of 'example.com/IN' from 192.168.1.10#53: connected using 192.168.1.11#38577
zone example.com/IN: transferred serial 5
transfer of 'example.com/IN' from 192.168.1.10#53: Transfer completed: 1 messages,
8 records, 225 bytes, 0.002 secs (112500 bytes/sec)
```

Note:

A zone is only transferred if the Serial Number on the primary is larger than the one on the secondary. If you want to have your primary DNS notify other secondary DNS servers of zone changes, you can add `also-notify { ipaddress; };` to `/etc/bind/named.conf.local` as shown in the example below:

```
zone "example.com" {
    type master;
    file "/etc/bind/db.example.com";
    allow-transfer { 192.168.1.11; };
    also-notify { 192.168.1.11; };
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
    allow-transfer { 192.168.1.11; };
    also-notify { 192.168.1.11; };
};
```

Note:

The default directory for non-authoritative zone files is `/var/cache/bind/`. This directory is also configured in AppArmor to allow the named daemon to write to it. See this page for *more information on AppArmor* (page 170).

Testing your setup

resolv.conf

DNS client configuration

```
nameserver 127.0.0.53
search example.com
```

```
$ resolvectl status
```

Note:

You should also add the IP address of the secondary nameserver to your client configuration in case the primary becomes unavailable.

dig

```
$ dig -x 127.0.0.1  
;; Query time: 1 msec  
;; SERVER: 192.168.1.10#53(192.168.1.10)
```

```
$ dig ubuntu.com  
;; Query time: 49 msec
```

```
;; Query time: 1 msec
```

ping

```
$ ping example.com
```

```
PING ns.example.com (192.168.1.10) 56(84) bytes of data.  
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.800 ms  
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.813 ms
```

named-checkzone

```
$ named-checkzone example.com /etc/bind/db.example.com  
  
zone example.com/IN: loaded serial 6  
OK
```

```
$ named-checkzone 1.168.192.in-addr.arpa /etc/bind/db.192  
  
zone 1.168.192.in-addr.arpa/IN: loaded serial 3  
OK
```

Note:

The Serial Number of your zone file will probably be different.

Quick temporary query logging

```
$ sudo rndc querylog on
```

```
$ sudo rndc querylog off
```

```
Jan 20 19 40:50 new-n1 named[816]: received control channel command 'querylog on'  
Jan 20 19 40:50 new-n1 named[816]: query logging is now on  
Jan 20 19 40:57 new-n1 named[816]: client @0x7f48ec101480 192.168.1.10#36139 (ubuntu.com):  
query: ubuntu.com IN A +E(0)K (192.168.1.10)
```

Note:

The amount of logs generated by enabling querylog could be huge!

Logging

```
logging {  
    category default { default_syslog; default_debug; };  
    category unmatched { null; };  
};
```

```
logging {  
    channel query.log {  
        file "/var/log/named/query.log";  
        severity debug 3;  
    };  
    category queries { query.log; };  
};
```

Note:

The debug option can be set from 1 to 3. If a level isn't specified, level 1 is the default.

```
$ sudo mkdir /var/log/named
```

```
$ sudo chown bind:bind /var/log/named
```

```
$ sudo systemctl restart bind9.service
```

Common record types

- *hostname*

www	IN	A	192.168.1.12

-

web	IN	CNAME	www

-

@	IN	MX	1	mail.example.com.
mail	IN	A	192.168.1.13	

-

@	IN	NS	ns.example.com.
@	IN	NS	ns2.example.com.
ns	IN	A	192.168.1.10
ns2	IN	A	192.168.1.11

Further reading

- *Installing DNS Security Extensions (DNSSEC)*
- Upstream BIND9 Documentation¹⁹⁵
- DNS and BIND¹⁹⁶
BIND on IPv6¹⁹⁷

DNS and

Installing DNS Security Extensions (DNSSEC)

DNS

¹⁹⁵ <https://bind9.readthedocs.io/en/latest/>

¹⁹⁶ <https://www.oreilly.com/library/view/dns-and-bind/0596100574/>

¹⁹⁷ <https://www.oreilly.com/library/view/dns-and-bind/9781449308025/>

Starting point

DNS How-To

```
$ sudo apt install bind9 -y
```

```
zone "example.internal" {  
    type master;  
    file "/var/lib/bind/db.example.internal";  
};
```

```
$TTL 86400      ; 1 day  
example.internal. IN SOA  example.internal. root.example.internal. (  
                           1          ; serial  
                           43200      ; refresh (12 hours)  
                           900        ; retry (15 minutes)  
                           1814400    ; expire (3 weeks)  
                           7200       ; minimum (2 hours)  
                           )  
example.internal. IN NS   ns.example.internal.  
ns                 IN  A   192.168.1.10  
noble              IN  A   192.168.1.11
```

```
$ sudo systemctl restart named
```

```
$ dig @127.0.0.1 +short noble.example.internal  
192.168.1.11
```

Enabling DNSSEC

```
zone "example.internal" {  
    type master;  
    file "/var/lib/bind/db.example.internal";  
    dnssec-policy default;  
    inline-signing yes;  
};
```

-
-

```
$ sudo rndc reconfig
```

```
$ sudo journalctl -u named.service -f
```

```
named[3246]: zone example.internal/IN (unsigned): loaded serial 1  
named[3246]: zone example.internal/IN (signed): loaded serial 1  
named[3246]: zone example.internal/IN (signed): receive_secure_serial: unchanged  
named[3246]: zone example.internal/IN (signed): sending notifies (serial 1)  
named[3246]: zone example.internal/IN (signed): reconfiguring zone keys  
named[3246]: keymgr: DNSKEY example.internal/ECDSAP256SHA256/44911 (CSK) created for  
policy default  
named[3246]: Fetching example.internal/ECDSAP256SHA256/44911 (CSK) from key repository.  
named[3246]: DNSKEY example.internal/ECDSAP256SHA256/44911 (CSK) is now published  
named[3246]: DNSKEY example.internal/ECDSAP256SHA256/44911 (CSK) is now active  
named[3246]: zone example.internal/IN (signed): next key event: 23-Oct-2024 22:47:12.544  
named[3246]: any newly configured zones are now loaded  
named[3246]: running  
named[3246]: resolver priming query complete: success  
named[3246]: managed-keys-zone: Key 20326 for zone . is now trusted (acceptance timer
```

(continues on next page)

(continued from previous page)

```
complete)
named[3246]: zone example.internal/IN (signed): sending notifies (serial 3)
```

-
-
-
-

```
-rw-r--r-- 1 bind bind 413 Oct 23 20:50 Kexample.internal.+013+48112.key
-rw----- 1 bind bind 215 Oct 23 20:50 Kexample.internal.+013+48112.private
-rw-r--r-- 1 bind bind 647 Oct 23 20:50 Kexample.internal.+013+48112.state
```

Verification

```
$ cp /var/cache/bind/Kexample.internal.+013+48112.key /tmp/example.key
```

```
example.internal. 3600 IN DNSKEY 257 3 13 jkmSShfyY3nSww...
```


-
-
-
-
-

```
trust-anchors {  
    example.internal. static-key 257 3 13 "jkmS5hfyY3nSww....";  
};
```

```
$ delv @127.0.0.1 -a /tmp/example.key +root=example.internal  
noble.example.internal. +multiline  
  
; fully validated  
noble.example.internal. 86400 IN A 192.168.1.11  
noble.example.internal. 86400 IN RRSIG A 13 3 86400 (  
                                20241106131533 20241023195023 48112 example.  
internal.  
                                5fL4apIwCD9kt4XbzzLLxMXy3mj8Li1WZu3qzlcBpERp  
                                lXPgLODbRrWyp7L81xEFnfhecKtEYv+6Y0Xa5iVRug== )
```

-
-

Connecting the dots: the parent zone

DS record format

```
$ dnssec-dsfromkey /var/cache/bind/Kexample.internal.+013+48112.key
example.internal. IN DS 48112 13 2
1212DE7DA534556F1E11898F2C7A66736D5107962A19A0BFE1C2A67D6841962A
```

DNSKEY record format

```
$ cat /var/cache/bind/Kexample.internal.+013+48112.key
; This is a key-signing key, keyid 48112, for example.internal.
; Created: 20241023205023 (Wed Oct 23 20:50:23 2024)
; Publish: 20241023205023 (Wed Oct 23 20:50:23 2024)
; Activate: 20241023205023 (Wed Oct 23 20:50:23 2024)
; SyncPublish: 20241024215523 (Thu Oct 24 21:55:23 2024)
example.internal. 3600 IN DNSKEY 257 3 13 jkmS5hfyY3nSww4tD9Fy5d+GGc3A/
zR1CFUxmN8T2TKtkgWp8dusllM 7TrIZTEg6wZxmMs754/ftoTA6jmM1g==
```

```
example.internal. 3600 IN DNSKEY 257 3 13 (jkmS5hfyY3nSww4tD9Fy5d+GGc3A/
zR1CFUxmN8T2TKtkgWp8dusllM 7TrIZTEg6wZxmMs754/ftoTA6jmM1g==);
```

Further reading

- The DNSSEC Guide from bind9¹⁹⁸
- Easy-Start Guide for Signing Authoritative Zones¹⁹⁹
- Creating a Custom DNSSEC Policy²⁰⁰
- Detailed DNSSEC chapter from the bind9 documentation²⁰¹
- *delv(1)*²⁰²
- Working with the parent zone²⁰³

¹⁹⁸ <https://bind9.readthedocs.io/en/stable/dnssec-guide.html>

¹⁹⁹ <https://bind9.readthedocs.io/en/stable/dnssec-guide.html#signing>

²⁰⁰ <https://bind9.readthedocs.io/en/stable/dnssec-guide.html#signing-custom-policy>

²⁰¹ <https://bind9.readthedocs.io/en/stable/chapter5.html>

²⁰² <https://manpages.ubuntu.com/manpages/resolute/man1/delv.1.html>

²⁰³ <https://bind9.readthedocs.io/en/latest/dnssec-guide.html#working-with-the-parent-zone>

Basic DNSSEC troubleshooting

DNSSEC

DNS

Handy “bad” and “good” DNSSEC domains

- - *dnssec-failed.org*²⁰⁴
 -
- - *isc.org*²⁰⁵
 -

Logs

BIND9 server

```
$ dig @127.0.0.1 -t A www.dnssec-failed.org

; <<>> DiG 9.20.11-Ubuntu <<>> @127.0.0.1 -t A www.dnssec-failed.org
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 26260
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags;; udp: 1232
; COOKIE: 6339d7228b8587f401000000671bc2eb2fe25bdf099ef1af (good)
;; QUESTION SECTION:
;www.dnssec-failed.org.      IN      A

;; Query time: 460 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Fri Oct 25 16:10:19 UTC 2024
;; MSG SIZE rcvd: 78
```

²⁰⁴ <http://dnssec-failed.org>

²⁰⁵ <http://isc.org>

```
$ journalctl -u named.service -f
```

```
(...)
```

```
named[286]: validating dnssec-failed.org/DNSKEY: no valid signature found (DS)
named[286]: no valid RRSIG resolving 'dnssec-failed.org/DNSKEY/IN': 68.87.85.132#53
named[286]: validating dnssec-failed.org/DNSKEY: no valid signature found (DS)
named[286]: no valid RRSIG resolving 'dnssec-failed.org/DNSKEY/IN': 68.87.68.244#53
named[286]: validating dnssec-failed.org/DNSKEY: no valid signature found (DS)
named[286]: no valid RRSIG resolving 'dnssec-failed.org/DNSKEY/IN': 68.87.76.228#53
named[286]: validating dnssec-failed.org/DNSKEY: no valid signature found (DS)
named[286]: no valid RRSIG resolving 'dnssec-failed.org/DNSKEY/IN': 68.87.72.244#53
named[286]: validating dnssec-failed.org/DNSKEY: no valid signature found (DS)
named[286]: no valid RRSIG resolving 'dnssec-failed.org/DNSKEY/IN': 69.252.250.103
#53
named[286]: broken trust chain resolving 'www.dnssec-failed.org/A/IN': 68.87.72.244
#53
```

Client-side tooling: dig

-
-
-
-

*isc.org*²⁰⁶

Note:

We need to (temporarily) enable DNSSEC validation and potentially set DNSSEC capable upstream servers (e.g. Cloudflare's 1.1.1.1, in case your upstream DNS server cannot properly handle it) in systemd-resolved, as shown below. After our experiments, we might want to restore the default settings, using: `resolvectl revert eth0`

²⁰⁶ <http://isc.org>

```
$ resolvectl dnssec eth0 true
```

```
$ resolvectl dns eth0 1.1.1.1
```

```
$ dig @127.0.0.53 -t A +dnssec +multiline isc.org

; <<>> DiG 9.20.11-Ubuntu <<>> @127.0.0.53 -t A +dnssec +multiline isc.org
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 23376
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 65494
;; QUESTION SECTION:
;isc.org.                IN A

;; ANSWER SECTION:
isc.org.                299 IN A 151.101.2.217
isc.org.                299 IN A 151.101.66.217
isc.org.                299 IN A 151.101.130.217
isc.org.                299 IN A 151.101.194.217
isc.org.                299 IN RRSIG A 13 2 300 (
                        20260130043603 20260116034316 27566 isc.org.
                        kp+MQSLXw7x0qU072l4g3/Fq815uc0v5rN77kxizdxnL
                        EMwYbgtcvI+AgDIwhm9qgLhc2G06pUUF+puJCokxNA== )

;; Query time: 26 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Thu Jan 22 13:56:58 UTC 2026
;; MSG SIZE rcvd: 203
```

-
-
-

```
$ dig @127.0.0.53 -t A +dnssec +multiline dnssec-failed.org
```

```
; <<>> DiG 9.20.11-Ubuntu <<>> @127.0.0.53 -t A +dnssec +multiline dnssec-failed.org
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: SERVFAIL, id: 2028
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 65494
; EDE: 9 (DNSKEY Missing): (no SEP matching the DS found for dnssec-failed.org.)
;; QUESTION SECTION:
;dnssec-failed.org.          IN A

;; Query time: 219 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Thu Jan 22 13:57:33 UTC 2026
;; MSG SIZE rcvd: 103
```

-
-
-

```
$ journalctl -u systemd-resolved.service -f

(...)
systemd-resolved[9203]: [ ] DNSSEC validation failed for question dnssec-failed.org
IN A: no-signature
```

```
$ dig @127.0.0.53 -t A +dnssec +cd +multiline dnssec-failed.org
```

```
; <<>> DiG 9.20.11-Ubuntu <<>> @127.0.0.53 -t A +dnssec +cd +multiline dnssec-
failed.org
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32064
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 65494
; EDE: 9 (DNSKEY Missing): (no SEP matching the DS found for dnssec-failed.org.)
;; QUESTION SECTION:
;dnssec-failed.org.          IN A

;; ANSWER SECTION:
dnssec-failed.org.          299 IN A 96.99.227.255
dnssec-failed.org.          299 IN RRSIG A 5 2 300 (
    20260204145120 20260118144620 44973 dnssec-failed.org.
    U8NkuyOJBdX6vA/f+o4DcLA0x5YrCV0pMCGyCnXqaTTv
    8+QhDuwRGsAhXYtAZTx4k6q7B2NbqPFsavHKbbsEY46J
    SLpA1MIYydHTnUPHnu+foH4Dq/QctoXbUejMySdyZwPp
    MU5hm5IkwN8LgKjoE0YL89mr4ueu01DtESG6P+I= )

;; Query time: 119 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Thu Jan 22 13:58:49 UTC 2026
;; MSG SIZE rcvd: 296
```

-
-
-

Digging a bit deeper: delv

```
$ delv @127.0.0.53 -t A +dnssec +multiline dnssec-failed.org
;; resolution failed: SERVFAIL
```

```
$ delv @127.0.0.53 -t A +dnssec +cd +multiline dnssec-failed.org
;; validating dnssec-failed.org/DNSKEY: no valid signature found (DS)
;; no valid RRSIG resolving 'dnssec-failed.org/DNSKEY/IN': 127.0.0.1#53
;; broken trust chain resolving 'dnssec-failed.org/A/IN': 127.0.0.1#53
;; resolution failed: broken trust chain
```

```
$ delv @127.0.0.53 -i -t A +dnssec +cd +multiline dnssec-failed.org
; answer not validated
dnssec-failed.org.      100 IN A 96.99.227.255
```

```
$ delv @127.0.0.53 -t A +multiline +cd isc.org
; fully validated
isc.org.      299 IN A 151.101.2.217
isc.org.      299 IN A 151.101.66.217
isc.org.      299 IN A 151.101.130.217
isc.org.      299 IN A 151.101.194.217
isc.org.      299 IN RRSIG A 13 2 300 (
20250908112301 20250825104017 27566 isc.org.
9ocLTno0Ub2NmUEXdyLv0zqwPBbbUVmT3RX4aP4BQQ+h
4g839JXuCKHufXSPkWh/GJe/MveP83dDvJkrMmEIzg== )
```

Client-side tooling: resolvectl

Note:

We need to (temporarily) enable DNSSEC validation and potentially set DNSSEC capable upstream servers (e.g. Cloudflare's 1.1.1.1, in case your upstream DNS server cannot properly handle it) in systemd-resolved, as shown below. After our experiments, we might want to restore the default settings, using: `resolvectl revert eth0`

```
$ sudo resolvectl flush-caches
```

```
$ sudo resolvectl reset-server-features
```

```
$ sudo resolvectl dnssec eth0 yes
```

```
$ sudo resolvectl dns eth0 1.1.1.1
```

```
$ resolvectl dnssec
```

```
Global: no  
Link 44 (eth0): yes
```

```
$ resolvectl query --type=MX isc.org
```

```
isc.org IN MX 5 mx.pao1.isc.org      -- link: eth0  
isc.org IN MX 10 mx.ams1.isc.org     -- link: eth0
```

```
-- Information acquired via protocol DNS in 3.0ms.  
-- Data is authenticated: yes; Data was acquired via local or encrypted transport:  
no  
-- Data from: network
```

Incorrect time

```
noble.example.internal. 86400 IN RRSIG A 13 3 86400 (  
20241106131533 20241023195023 48112 example.internal.  
5fL4apIwCD9kt4XbzzLLxMXY3mj8Li1WZu3qzlcBpERp  
LXPgLODbRrWyp7L81xEFnfhcKtEYv+6Y0Xa5iVRug== )
```

-
-

```
$ date
```

```
Tue Oct 10 10:10:19 UTC 2000
```

```
$ delv @10.10.17.229 -a example.internal.key +root=example.internal  
+multiline noble.example.internal
```

```
;; validating example.internal/DNSKEY: verify failed due to bad signature  
(keyid=48112): RRSIG validity period has not begun  
;; validating example.internal/DNSKEY: no valid signature found (DS)  
;; no valid RRSIG resolving 'example.internal/DNSKEY/IN': 10.10.17.229#53  
;; broken trust chain resolving 'noble.example.internal/A/IN': 10.10.17.229#53  
;; resolution failed: broken trust chain
```

```
named[3593]: managed-keys-zone: DNSKEY set for zone '.' could not be verified with current  
keys  
named[3593]: validating ./DNSKEY: verify failed due to bad signature (keyid=20326):  
RRSIG validity period has not begun  
named[3593]: validating ./DNSKEY: no valid signature found (DS)  
named[3593]: broken trust chain resolving './NS/IN': 199.7.83.42#53  
named[3593]: resolver priming query complete: broken trust chain
```

Third-party web-based diagnostics

- <https://dnsviz.net>
- <https://dnssec-debugger.verisignlabs.com>

Further reading

- [bind9's guide to DNSSEC troubleshooting](#)²⁰⁷
- [How to validate DNSSEC using the command line](#)²⁰⁸
- [delv\(1\)](#)²⁰⁹
- [dig\(1\)](#)²¹⁰

How to use Open vSwitch with DPDK

DPDK is just a library

```
$ sudo apt-get install openvswitch-switch-dpdk
```

```
$ sudo update-alternatives --set ovs-vswitchd  
/usr/lib/openvswitch-switch-dpdk/ovs-vswitchd-dpdk
```

```
$ ovs-vsctl set Open_vSwitch . "other_config:dpdk-init=true"
```

•

```
$ ovs-vsctl set Open_vSwitch . "other_config:dpdk-lcore-mask=0x1"
```

•

```
$ ovs-vsctl set Open_vSwitch . "other_config:dpdk-alloc-mem=2048"
```

•

```
$ ovs-vsctl set Open_vSwitch .  
"other_config:dpdk-extra=-pci-whitelist=0000:04:00.0"
```

```
$ sudo service openvswitch-switch restart
```

²⁰⁷ <https://bind9.readthedocs.io/en/latest/dnssec-guide.html#basic-dnssec-troubleshooting>

²⁰⁸ <https://www.cyberciti.biz/faq/unix-linux-test-and-validate-dnssec-using-dig-command-line/>

²⁰⁹ <https://manpages.ubuntu.com/manpages/resolute/man1/delv.1.html>

²¹⁰ <https://manpages.ubuntu.com/manpages/resolute/man1/dig.1.html>

Note:

You need to assign devices to DPDK-compatible drivers before restarting – see the DPDK section on *unassigning the default kernel drivers* (page 930).

Open vSwitch setup²¹¹

Attach DPDK ports to Open vSwitch

```
$ ovs-vsctl add-br ovsdpkbr0 -- set bridge ovsdpkbr0 datapath_type=netdev
```

```
$ ovs-vsctl add-port ovsdpkbr0 dpdk0 -- set Interface dpdk0 type=dpdk  
"options:dpdk-devargs=${OVSDEV_PCIID}"
```

```
$ ovs-vsctl set Interface dpdk0 "options:n_rxq=2"
```

Open vSwitch DPDK to KVM guests

```
$ ovs-vsctl add-port ovsdpkbr0 vhost-user-1 -- set Interface vhost-user-1  
type=dpdkvhostuserclient  
"options:vhost-server-path=/var/run/vhostuserclient/vhost-user-client-1"
```

²¹¹ <https://docs.openvswitch.org/en/latest/intro/install/dpdk/#setup-ovs>

```
<interface type='vhostuser'>
<source type='unix'
path='/var/run/vhostuserclient/vhost-user-client-1'
mode='server' />
<model type='virtio' />
</interface>
```

Tuning Open vSwitch-DPDK

```
$ ovs-vsctl set Open_vSwitch . other_config:n-dpdk-rxqs=2
```

```
$ ovs-vsctl set Open_vSwitch . other_config:pmd-cpu-mask=0x6
```

See also

Check the links to [EAL Command-line Options](#) and “Open vSwitch DPDK installation” at the end of this document for more information.

Support and troubleshooting

- [DPDK Mailing Lists](#)²¹²
- [OpenStack Mailing Lists](#)²¹³
- [DPDK Launchpad Area](#)²¹⁴

²¹² <https://www.dpdk.org/contribute/>

²¹³ <http://www.openvswitch.org/mlists/>

²¹⁴ <https://launchpad.net/ubuntu/+source/dpdk>

```
ovs-ctl[3560]: EAL: PCI device 0000:04:00.1 on NUMA socket 0
ovs-ctl[3560]: EAL:   probe driver: 8086:1528 rte_ixgbe_pmd
ovs-ctl[3560]: EAL:   PCI memory mapped at 0x7f2140000000
ovs-ctl[3560]: EAL:   PCI memory mapped at 0x7f2140200000
```

#Note: This log was taken with dpdk 2.2 and openvswitch 2.5 but still looks quite similar (a bit extended) these days

Captions:

CMD: that you enter

SYSLOG: (Including EAL and OVS Messages)

OVS-LOG: (Openvswitch messages)

#PREPARATION

Bind an interface to DPDK UIO drivers, make Hugepages available, enable DPDK on OVS

CMD: sudo service openvswitch-switch restart

SYSLOG:

2016-01-22T08:58:31.372Z|00003|daemon_unix(monitor)|INFO|pid 3329 died, killed (Terminated), exiting

2016-01-22T08:58:33.377Z|00002|vlog|INFO|opened log file /var/log/openvswitch/ovs-vswitchd.log

2016-01-22T08:58:33.381Z|00003|ovs_numa|INFO|Discovered 12 CPU cores on NUMA node 0

2016-01-22T08:58:33.381Z|00004|ovs_numa|INFO|Discovered 1 NUMA nodes and 12 CPU cores

2016-01-22T08:58:33.381Z|00005|reconnect|INFO|unix:/var/run/openvswitch/db.sock: connecting...

2016-01-22T08:58:33.383Z|00006|reconnect|INFO|unix:/var/run/openvswitch/db.sock: connected

2016-01-22T08:58:33.386Z|00007|bridge|INFO|ovs-vswitchd (Open vSwitch) 2.5.0

OVS-LOG:

systemd[1]: Stopping Open vSwitch...

systemd[1]: Stopped Open vSwitch.

systemd[1]: Stopping Open vSwitch Internal Unit...

ovs-ctl[3541]: * Killing ovs-vswitchd (3329)

ovs-ctl[3541]: * Killing ovssdb-server (3318)

systemd[1]: Stopped Open vSwitch Internal Unit.

systemd[1]: Starting Open vSwitch Internal Unit...

ovs-ctl[3560]: * Starting ovssdb-server

ovs-vsctl: ovs|00001|vsctl|INFO|Called as ovs-vsctl --no-wait -- init -- set Open_vSwitch . db-version=7.12.1

ovs-vsctl: ovs|00001|vsctl|INFO|Called as ovs-vsctl --no-wait set Open_vSwitch . ovs-version=2.5.0 "external-ids:system-id=\"e7c5ba80-bb14-45c1-b8eb-628f3ad03903\"" "system-type=\"Ubuntu\"" "system-version=\"16.04-xenial\""

ovs-ctl[3560]: * Configuring Open vSwitch system IDs

ovs-ctl[3560]: 2016-01-22T08:58:31Z|00001|dpdk|INFO|No -vhost_sock_dir provided - defaulting to /var/run/openvswitch

ovs-vswitchd: ovs|00001|dpdk|INFO|No -vhost_sock_dir provided - defaulting to /var/run/openvswitch

ovs-ctl[3560]: EAL: Detected lcore 0 as core 0 on socket 0

(continues on next page)

(continued from previous page)

```
ovs-ctl[3560]: EAL: Detected lcore 1 as core 1 on socket 0
ovs-ctl[3560]: EAL: Detected lcore 2 as core 2 on socket 0
ovs-ctl[3560]: EAL: Detected lcore 3 as core 3 on socket 0
ovs-ctl[3560]: EAL: Detected lcore 4 as core 4 on socket 0
ovs-ctl[3560]: EAL: Detected lcore 5 as core 5 on socket 0
ovs-ctl[3560]: EAL: Detected lcore 6 as core 0 on socket 0
ovs-ctl[3560]: EAL: Detected lcore 7 as core 1 on socket 0
ovs-ctl[3560]: EAL: Detected lcore 8 as core 2 on socket 0
ovs-ctl[3560]: EAL: Detected lcore 9 as core 3 on socket 0
ovs-ctl[3560]: EAL: Detected lcore 10 as core 4 on socket 0
ovs-ctl[3560]: EAL: Detected lcore 11 as core 5 on socket 0
ovs-ctl[3560]: EAL: Support maximum 128 logical core(s) by configuration.
ovs-ctl[3560]: EAL: Detected 12 lcore(s)
ovs-ctl[3560]: EAL: VFIO modules not all loaded, skip VFIO support...
ovs-ctl[3560]: EAL: Setting up physically contiguous memory...
ovs-ctl[3560]: EAL: Ask a virtual area of 0x100000000 bytes
ovs-ctl[3560]: EAL: Virtual area found at 0x7f2040000000 (size = 0x100000000)
ovs-ctl[3560]: EAL: Requesting 4 pages of size 1024MB from socket 0
ovs-ctl[3560]: EAL: TSC frequency is ~2397202 KHz
ovs-vswitchd[3592]: EAL: TSC frequency is ~2397202 KHz
ovs-vswitchd[3592]: EAL: Master lcore 0 is ready (tid=fc6cbb00;cpuset=[0])
ovs-vswitchd[3592]: EAL: PCI device 0000:04:00.0 on NUMA socket 0
ovs-vswitchd[3592]: EAL:   probe driver: 8086:1528 rte_ixgbe_pmd
ovs-vswitchd[3592]: EAL:   Not managed by a supported kernel driver, skipped
ovs-vswitchd[3592]: EAL: PCI device 0000:04:00.1 on NUMA socket 0
ovs-vswitchd[3592]: EAL:   probe driver: 8086:1528 rte_ixgbe_pmd
ovs-vswitchd[3592]: EAL:   PCI memory mapped at 0x7f2140000000
ovs-vswitchd[3592]: EAL:   PCI memory mapped at 0x7f2140200000
ovs-ctl[3560]: EAL: Master lcore 0 is ready (tid=fc6cbb00;cpuset=[0])
ovs-ctl[3560]: EAL: PCI device 0000:04:00.0 on NUMA socket 0
ovs-ctl[3560]: EAL:   probe driver: 8086:1528 rte_ixgbe_pmd
ovs-ctl[3560]: EAL:   Not managed by a supported kernel driver, skipped
ovs-ctl[3560]: EAL: PCI device 0000:04:00.1 on NUMA socket 0
ovs-ctl[3560]: EAL:   probe driver: 8086:1528 rte_ixgbe_pmd
ovs-ctl[3560]: EAL:   PCI memory mapped at 0x7f2140000000
ovs-ctl[3560]: EAL:   PCI memory mapped at 0x7f2140200000
ovs-vswitchd[3592]: PMD: eth_ixgbe_dev_init(): MAC: 4, PHY: 3
ovs-vswitchd[3592]: PMD: eth_ixgbe_dev_init(): port 0 vendorID=0x8086 deviceID=0x1528
ovs-ctl[3560]: PMD: eth_ixgbe_dev_init(): MAC: 4, PHY: 3
ovs-ctl[3560]: PMD: eth_ixgbe_dev_init(): port 0 vendorID=0x8086 deviceID=0x1528
ovs-ctl[3560]: Zone 0: name:<RG_MP_log_history>, phys:0x83ffffdec0, len:0x2080,
virt:0x7f213ffffdec0, socket_id:0, flags:0
ovs-ctl[3560]: Zone 1: name:<MP_log_history>, phys:0x83fd73d40, len:0x28a0c0,
virt:0x7f213fd73d40, socket_id:0, flags:0
ovs-ctl[3560]: Zone 2: name:<rte_eth_dev_data>, phys:0x83fd43380, len:0x2f700,
virt:0x7f213fd43380, socket_id:0, flags:0
ovs-ctl[3560]: * Starting ovs-vswitchd
ovs-ctl[3560]: * Enabling remote OVSDB managers
systemd[1]: Started Open vSwitch Internal Unit.
systemd[1]: Starting Open vSwitch...
systemd[1]: Started Open vSwitch.

CMD: sudo ovs-vsctl add-br ovsdpdkbr0 -- set bridge ovsdpdkbr0 datapath_type=netdev

SYSLOG:
```

(continues on next page)

(continued from previous page)

```
2016-01-22T08:58:56.344Z|00008|memory|INFO|37256 kB peak resident set size after 24.5
seconds
2016-01-22T08:58:56.346Z|00009|ofproto_dpif|INFO|netdev@ovs-netdev: Datapath supports
recirculation
2016-01-22T08:58:56.346Z|00010|ofproto_dpif|INFO|netdev@ovs-netdev: MPLS label stack
length probed as 3
2016-01-22T08:58:56.346Z|00011|ofproto_dpif|INFO|netdev@ovs-netdev: Datapath supports
unique flow ids
2016-01-22T08:58:56.346Z|00012|ofproto_dpif|INFO|netdev@ovs-netdev: Datapath does not
support ct_state
2016-01-22T08:58:56.346Z|00013|ofproto_dpif|INFO|netdev@ovs-netdev: Datapath does not
support ct_zone
2016-01-22T08:58:56.346Z|00014|ofproto_dpif|INFO|netdev@ovs-netdev: Datapath does not
support ct_mark
2016-01-22T08:58:56.346Z|00015|ofproto_dpif|INFO|netdev@ovs-netdev: Datapath does not
support ct_label
2016-01-22T08:58:56.360Z|00016|bridge|INFO|bridge ovspdkbr0: added interface ovspdkbr0
on port 65534
2016-01-22T08:58:56.361Z|00017|bridge|INFO|bridge ovspdkbr0: using datapath ID
00005a4a1ed0a14d
2016-01-22T08:58:56.361Z|00018|connmgr|INFO|ovspdkbr0: added service controller "punix:/
var/run/openvswitch/ovspdkbr0.mgmt"

OVS-LOG:
ovs-vsctl: ovs|00001|vsctl|INFO|Called as ovs-vsctl add-br ovspdkbr0 -- set bridge
ovspdkbr0 datapath_type=netdev
systemd-udevd[3607]: Could not generate persistent MAC address for ovs-netdev: No such
file or directory
kernel: [50165.886554] device ovs-netdev entered promiscuous mode
kernel: [50165.901261] device ovspdkbr0 entered promiscuous mode

CMD: sudo ovs-vsctl add-port ovspdkbr0 dpdk0 -- set Interface dpdk0 type=dpdk

SYSLOG:
2016-01-22T08:59:06.369Z|00019|memory|INFO|peak resident set size grew 155% in last 10.0
seconds, from 37256 kB to 95008 kB
2016-01-22T08:59:06.369Z|00020|memory|INFO|handlers:4 ports:1 revalidators:2 rules:5
2016-01-22T08:59:30.989Z|00021|dpdk|INFO|Port 0: 8c:dc:d4:b3:6d:e9
2016-01-22T08:59:31.520Z|00022|dpdk|INFO|Port 0: 8c:dc:d4:b3:6d:e9
2016-01-22T08:59:31.521Z|00023|dpif_netdev|INFO|Created 1 pmd threads on numa node 0
2016-01-22T08:59:31.522Z|00001|dpif_netdev(pmd16)|INFO|Core 0 processing port 'dpdk0'
2016-01-22T08:59:31.522Z|00024|bridge|INFO|bridge ovspdkbr0: added interface dpdk0 on
port 1
2016-01-22T08:59:31.522Z|00025|bridge|INFO|bridge ovspdkbr0: using datapath ID
00008cdcd4b36de9
2016-01-22T08:59:31.523Z|00002|dpif_netdev(pmd16)|INFO|Core 0 processing port 'dpdk0'

OVS-LOG:
ovs-vsctl: ovs|00001|vsctl|INFO|Called as ovs-vsctl add-port ovspdkbr0 dpdk0 -- set
Interface dpdk0 type=dpdk
ovs-vswitchd[3595]: PMD: ixgbe_dev_tx_queue_setup(): sw_ring=0x7f211a79ebc0 hw_
ring=0x7f211a7a6c00 dma_addr=0x81a7a6c00
ovs-vswitchd[3595]: PMD: ixgbe_set_tx_function(): Using simple tx code path
ovs-vswitchd[3595]: PMD: ixgbe_set_tx_function(): Vector tx enabled.
ovs-vswitchd[3595]: PMD: ixgbe_dev_rx_queue_setup(): sw_ring=0x7f211a78a6c0 sw_sc_
```

(continues on next page)

(continued from previous page)

```
ring=0x7f211a786580 hw_ring=0x7f211a78e800 dma_addr=0x81a78e800
ovs-vswitchd[3595]: PMD: ixgbe_set_rx_function(): Vector rx enabled, please make sure RX
burst size no less than 4 (port=0).
ovs-vswitchd[3595]: PMD: ixgbe_dev_tx_queue_setup(): sw_ring=0x7f211a79ebc0 hw_
ring=0x7f211a7a6c00 dma_addr=0x81a7a6c00
...

CMD: sudo ovs-vsctl add-port ovspdkbr0 vhost-user-1 -- set Interface vhost-user-1
type=dpmkvhostuser

OVS-LOG:
2016-01-22T09:00:35.145Z|00026|dpdk|INFO|Socket /var/run/openvswitch/vhost-user-1 created
for vhost-user port vhost-user-1
2016-01-22T09:00:35.145Z|00003|dpif_netdev(pmd16)|INFO|Core 0 processing port 'dpdk0'
2016-01-22T09:00:35.145Z|00004|dpif_netdev(pmd16)|INFO|Core 0 processing port 'vhost-user-
1'
2016-01-22T09:00:35.145Z|00027|bridge|INFO|bridge ovspdkbr0: added interface vhost-user-1
on port 2

SYSLOG:
ovs-vsctl: ovs|00001|vsctl|INFO|Called as ovs-vsctl add-port ovspdkbr0 vhost-user-1 --
set Interface vhost-user-1 type=dpmkvhostuser
ovs-vswitchd[3595]: VHOST_CONFIG: socket created, fd:46
ovs-vswitchd[3595]: VHOST_CONFIG: bind to /var/run/openvswitch/vhost-user-1

Eventually we can see the poll thread in top
  PID USER      PR  NI   VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 3595 root        10  -10 4975344 103936   9916 S 100.0   0.3   33:13.56 ovs-vswitchd
```

Further reading

- [DPDK documentation](#)²¹⁵
- [Linux DPDK user getting started](#)²¹⁶
- [EAL command-line options](#)²¹⁷
- [DPDK API documentation](#)²¹⁸
- [Open Vswitch DPDK installation](#)²¹⁹
- [Wikipedia's definition of DPDK](#)²²⁰

²¹⁵ <http://dpdk.org/doc>

²¹⁶ http://dpdk.org/doc/guides/linux_gsg/index.html

²¹⁷ http://dpdk.org/doc/guides/testpmd_app_ug/run_app.html

²¹⁸ <http://dpdk.org/doc/api/>

²¹⁹ <https://github.com/openvswitch/ovs/blob/branch-2.5/INSTALL.DPDK.md>

²²⁰ https://en.wikipedia.org/wiki/Data_Plane_Development_Kit

Install and maintain DOCA-OFED

Note:

DOCA-OFED²²¹ is NVIDIA's distribution of OpenFabrics Enterprise Distribution (OFED) drivers and userspace libraries for high-performance networking. DOCA-OFED is available for Ubuntu 26.04 LTS (Resolute Raccoon).

²²¹ <https://developer.nvidia.com/networking/doca>

Available metapackages

Metapackage	Kernel flavor
doca-ofed-aws	aws
doca-ofed-aws-64k	aws-64k
doca-ofed-azure	azure
doca-ofed-azure-fde	azure-fde
doca-ofed-gcp	gcp
doca-ofed-gcp-64k	gcp-64k
doca-ofed-generic	generic
doca-ofed-generic-64k	generic-64k
doca-ofed-generic-64k-hwe-26.04	generic-64k-hwe-26.04
doca-ofed-generic-64k-hwe-26.04-edge	generic-64k-hwe-26.04-edge
doca-ofed-generic-hwe-26.04	generic-hwe-26.04
doca-ofed-generic-hwe-26.04-edge	generic-hwe-26.04-edge
doca-ofed-oracle	oracle
doca-ofed-oracle-64k	oracle-64k

Install DOCA-OFED

*uname(1)*²²²

```
$ uname -r
```

*add-apt-repository(1)*²²³

```
$ sudo add-apt-repository -y ppa:canonical-nvidia/doca-stable
```

*apt(8)*²²⁴

```
$ sudo apt install -y doca-ofed-<your_kernel_flavor>
```

```
$ sudo apt install -y doca-ofed-generic
```

Maintain and upgrade DOCA-OFED

```
$ apt list --upgradable
```

```
$ sudo apt update && sudo apt full-upgrade
```

Warning:

Use `sudo apt full-upgrade` for DOCA-OFED updates. Do not use only `sudo apt upgrade` for this workflow. (DOCA-OFED updates from our PPA require uninstalling the old version's package names and installing the new version's package names (ex: `doca-ofed-userspace-3.3` is uninstalled, and `doca-ofed-userspace-3.4` is installed on update). `sudo apt upgrade` will hold such updates back.)

²²² <https://manpages.ubuntu.com/manpages/resolute/man1/uname.1.html>

²²³ <https://manpages.ubuntu.com/manpages/resolute/man1/add-apt-repository.1.html>

²²⁴ <https://manpages.ubuntu.com/manpages/resolute/man8/apt.8.html>

Collect a sosreport for support

```
$ apt list --installed | grep doca-ofed-userspace
```

```
$ sudo apt install doca-sosreport-<doca_version>
```

DHCP

How to install and configure isc-kea

Kea²²⁵

DHCP

refer to this guide instead

Install isc-kea

```
$ sudo apt install kea
```

-
-
-
-

DNS

LP: #2007312²²⁶

²²⁵ <https://www.isc.org/kea/>

²²⁶ <https://bugs.launchpad.net/bugs/2007312>

-
-
-

Configure kea-dhcp4

```
{
  "Dhcp4": {
    "interfaces-config": {
      "interfaces": ["eth4"]
    },
    "control-socket": {
      "socket-type": "unix",
      "socket-name": "/run/kea/kea4-ctrl-socket"
    },
    "lease-database": {
      "type": "memfile",
      "lfc-interval": 3600
    },
    "valid-lifetime": 600
    "max-valid-lifetime": 7200
    "subnet4": [{
      "id": 1,
      "subnet": "192.168.1.0/24",
      "pools": [{
        "pool": "192.168.1.150 - 192.168.1.200"
      }],
      "option-data": [{
        "name": "routers",
        "data": "192.168.1.254"
      },
      {
        "name": "domain-name-servers",
        "data": "192.168.1.1, 192.168.1.2"
      },
      {
        "name": "domain-name",
        "data": "mydomain.example"
      }
    ]
  }
}]
```

(continues on next page)

(continued from previous page)

```
}  
}
```

```
$ kea-shell --host 127.0.0.1 --port 8000 --auth-user kea-api --auth-password  
$(cat /etc/kea/kea-api-password) --service dhcp4 config-reload
```

```
[ { "result": 0, "text": "Configuration successful." } ]
```

```
DHCP4_DYNAMIC_RECONFIGURATION_SUCCESS dynamic server reconfiguration succeeded with file:  
/etc/kea/kea-dhcp4.conf
```

```
$ journalctl -u kea-dhcp4-server
```

```
$ systemctl restart kea-dhcp4-server
```

Further reading

- [ISC Kea Documentation](#)²²⁷

²²⁷ <https://kb.isc.org/docs/kea-administrator-reference-manual>

How to install and configure isc-dhcp-server

Warning:

The `isc-dhcp-server` package is deprecated and unsupported since Ubuntu 24.04 LTS. It is no longer supported by its vendor. Please use [Kea](https://www.isc.org/kea/)²²⁸ or [dnsmasq](https://thekelleys.org.uk/dnsmasq/doc.html)²²⁹ instead.

²²⁸ <https://www.isc.org/kea/>

²²⁹ <https://thekelleys.org.uk/dnsmasq/doc.html>

guide instead

DHCPD

refer to this

Install isc-dhcp-server

```
$ sudo apt install isc-dhcp-server
```

Note:

You can find diagnostic messages from `dhcpcd` in `syslog`.

Configure isc-dhcp-server

```
# minimal sample /etc/dhcp/dhcpd.conf
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.150 192.168.1.200;
    option routers 192.168.1.254;
    option domain-name-servers 192.168.1.1, 192.168.1.2;
    option domain-name "nydomain.example";
}
```

DNS

```
INTERFACESv4="eth4"
```

```
$ sudo systemctl restart isc-dhcp-server.service
```

Further reading

- *dhcpcd.conf(5)*²³⁰
- ISC dhcp-server²³¹

Time synchronization

Synchronize time using chrony

synchronize time using systemd

Check status of chrony client

```
Local time: Mo 2025-06-16 15:21:46 CEST
Universal time: Mo 2025-06-16 13:21:46 UTC
RTC time: Mo 2025-06-16 13:21:46
Time zone: Europe/Berlin (CEST, +0200)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

²³⁰ <https://manpages.ubuntu.com/manpages/resolute/man5/dhcpd.conf.5.html>

²³¹ <https://www.isc.org/dhcp/>


```
Reference ID      : B97DBE7B (2.ntp.ubuntu.com)
Stratum          : 3
Ref time (UTC)   : Mon Jun 16 13:06:04 2025
System time      : 0.000000004 seconds slow of NTP time
Last offset      : +0.001758954 seconds
RMS offset       : 0.017604901 seconds
Frequency        : 3.889 ppm slow
Residual freq    : +0.202 ppm
Skew             : 1.458 ppm
Root delay       : 0.022837413 seconds
Root dispersion  : 0.003050051 seconds
Update interval  : 1032.5 seconds
Leap status      : Normal
```

Network Time Security (NTS)

```
pool 1.ntp.ubuntu.com iburst maxsources 1 nts prefer
pool 2.ntp.ubuntu.com iburst maxsources 1 nts prefer
pool 3.ntp.ubuntu.com iburst maxsources 1 nts prefer
pool 4.ntp.ubuntu.com iburst maxsources 1 nts prefer
pool ntp-bootstrap.ubuntu.com iburst maxsources 1 nts certset 1
```

```
pool ntp-bootstrap.ubuntu.com iburst maxsources 1 nts certset 1
```

```
$ openssl s_client -connect 1.ntp.ubuntu.com:4460 -servername
1.ntp.ubuntu.com

Server certificate
...
subject=CN=ntp.ubuntu.com
issuer=C=US, O=Let's Encrypt, CN=R13
---
```

```
Server certificate
...
subject=CN=ntp-bootstrap.ubuntu.com
issuer=CN=ubuntu
```

```
ntstrustedcerts 1 /etc/chrony/nts-bootstrap-ubuntu.crt
```

```
$ sudo chronyc -N authdata
```

Name/IP address	Mode	KeyID	Type	KLen	Last	Atmp	NAK	Cook	CLen
1.ntp.ubuntu.com	NTS	6	30	128	14d	0	0	8	64
2.ntp.ubuntu.com	NTS	6	30	128	14d	0	0	8	64
3.ntp.ubuntu.com	NTS	1	30	128	27d	0	0	8	64
4.ntp.ubuntu.com	NTS	2	30	128	20d	0	0	5	64
ntp-bootstrap.ubuntu.com	NTS	3	30	128	7d	0	0	8	64

NTS related constraints

Warning:

If the network does not allow access to the Ubuntu NTS servers and required ports, with the default configuration is in place, chrony will not be able to adjust the system's clock. To revert to NTP, edit the configuration file in `/etc/chrony/sources.d/ubuntu-ntp-pools.sources` and revert to using the listed NTP servers in favor of the NTS ones.

Note:

The certificate for `ntp-bootstrap.ubuntu.com` is installed in `/etc/chrony/nts-bootstrap-ubuntu.crt` and is used specifically for the Ubuntu NTS bootstrap server, needed for when the clock is too far off. This is added to certificate set ID "1", and defined via `/etc/chrony/conf.d/ubuntu-nts.conf`.

Configure chrony

```
# Use NTS by default
# NTS uses an additional port to negotiate security: 4460/tcp
# The normal NTP port remains in use: 123/udp
pool 1.ntp.ubuntu.com iburst maxsources 1 nts prefer
pool 2.ntp.ubuntu.com iburst maxsources 1 nts prefer
pool 3.ntp.ubuntu.com iburst maxsources 1 nts prefer
pool 4.ntp.ubuntu.com iburst maxsources 1 nts prefer
# The bootstrap server is needed by systems without a hardware clock, or a very
# large initial clock offset. The specified certificate set is defined in
# /etc/chrony/conf.d/ubuntu-nts.conf.
pool ntp-bootstrap.ubuntu.com iburst maxsources 1 nts certset 1
```

Time sources provided by DHCP (option 42)

```
# Use time sources from DHCP.
# sourcedir /run/chrony-dhcp
```

Chrony time-daemon

```
$ systemctl status chrony.service

❏ chrony.service - chrony, an NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chrony.service; enabled; preset:
   enabled)
   Active: active (running) since Mon 2025-06-02 11:27:09 CEST; 2 weeks 0 days ago
     Docs: man:chronyd(8)
           man:chronyc(1)
           man:chrony.conf(5)
  Main PID: 36027 (chronyd)
    Tasks: 2 (limit: 28388)
   Memory: 5.8M (peak: 6.8M swap: 604.0K swap peak: 4.5M)
      CPU: 5.038s
   CGroup: /system.slice/chrony.service
           └─36027 /usr/sbin/chronyd -F 1
             └─36028 /usr/sbin/chronyd -F 1

Jun 02 11:27:09 questing chronyd[36027]: Using right/UTC timezone to obtain leap
second data
Jun 02 11:27:09 questing chronyd[36027]: Loaded seccomp filter (level 1)
Jun 02 11:27:09 questing chronyd[36027]: Added pool 1.ntp.ubuntu.com
Jun 02 11:27:09 questing chronyd[36027]: Added pool 2.ntp.ubuntu.com
Jun 02 11:27:09 questing chronyd[36027]: Added pool 3.ntp.ubuntu.com
Jun 02 11:27:09 questing chronyd[36027]: Added pool 4.ntp.ubuntu.com
Jun 02 11:27:09 questing chronyd[36027]: Added pool ntp-bootstrap.ubuntu.com
Jun 02 11:27:09 questing systemd[1]: Started chrony.service - chrony, an NTP
client/server.
```

*chrony.conf(5)*²³²

```
$ sudo systemctl restart chrony.service
```

Next steps

[how to configure your chrony setup](#)

²³² <https://manpages.ubuntu.com/manpages/resolute/man5/chrony.conf.5.html>

Further reading

- *chronyc(1)*²³³
- *chronyd(8)*²³⁴
- *chrony.conf(5)*²³⁵

Synchronize time using `timedatectl` and `timesyncd`

Network Time Security (NTS)

client *serve the Network Time Protocol*

```
$ sudo apt-mark auto chrony && apt install systemd-timesyncd
```

Note:

If `chrony` is installed, `timedatectl` steps back to let `chrony` handle timekeeping. This ensures that no two time-syncing services will be in conflict.

Check status of `timedatectl`

```
Local time: Wed 2023-06-14 12:05:11 BST
Universal time: Wed 2023-06-14 11:05:11 UTC
RTC time: Wed 2023-06-14 11:05:11
Time zone: Europe/Isle_of_Man (BST, +0100)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

```
[...]
systemd-timesyncd.service active: no
```

²³³ <https://manpages.ubuntu.com/manpages/resolute/man1/chronyc.1.html>

²³⁴ <https://manpages.ubuntu.com/manpages/resolute/man8/chronyd.8.html>

²³⁵ <https://manpages.ubuntu.com/manpages/resolute/man5/chrony.conf.5.html>

Configure timedatectl

Check status of timesyncd

```
$ systemctl status systemd-timesyncd

systemd-timesyncd.service - Network Time Synchronization
   Loaded: loaded (/lib/systemd/system/systemd-timesyncd.service; enabled;
   vendor preset: enabled)
   Active: active (running) since Fri 2018-02-23 08:55:46 UTC; 10s ago
     Docs: man:systemd-timesyncd.service(8)
  Main PID: 3744 (systemd-timesyn)
    Status: "Synchronized to time server 91.189.89.198:123 (ntp.ubuntu.com)."
```

Tasks: 2 (limit: 4915)
CGroup: /system.slice/systemd-timesyncd.service
|-3744 /lib/systemd/systemd-timesyncd

```
Feb 23 08:55:46 test-host systemd[1]: Starting Network Time Synchronization...
Feb 23 08:55:46 test-host systemd[1]: Started Network Time Synchronization.
Feb 23 08:55:46 test-host systemd-timesyncd[3744]: Synchronized to time server
91.189.89.198:123 (ntp.ubuntu.com).
```

Configure timesyncd

Next steps

how to install and configure your setup

Further reading

- `timedatectl(1)`²³⁶
- `systemd-timesyncd.service(8)`²³⁷
- [Ubuntu Time wiki page](https://help.ubuntu.com/community/UbuntuTime)²³⁸

How to serve the Network Time Protocol with Chrony

functioning as a client

Install chronyd

```
$ sudo apt install chrony
```

-
-

²³⁶ <https://manpages.ubuntu.com/manpages/resolute/man1/timedatectl.1.html>

²³⁷ <https://manpages.ubuntu.com/manpages/resolute/man8/systemd-timesyncd.service.8.html>

²³⁸ <https://help.ubuntu.com/community/UbuntuTime>

Enable serving the Network Time Protocol

```
allow 1.2.3.4
```

*chrony.conf(5)*²³⁹

View chrony status

```
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^+ gamma.rueckgr.at          2   8   377   135  -1048us[-1048us] +/-  29ms
^- 2b.ncomputers.org         2   8   377   204  -1141us[-1124us] +/-  50ms
^+ www.kashra.com            2   8   377   139  +3483us[+3483us] +/-  18ms
^+ stratum2-4.NTP.TechFak.U> 2   8   377   143  -2090us[-2073us] +/-  19ms
^- zepto.mcl.gg              2   7   377    9   -774us[ -774us] +/-  29ms
^- mirrorhost.pw            2   7   377   78   -660us[ -660us] +/-  53ms
^- atto.mcl.gg               2   7   377    8   -823us[ -823us] +/-  50ms
^- static.140.107.46.78.cli> 2   8   377    9  -1503us[-1503us] +/-  45ms
^- 4.53.160.75               2   8   377   137   -11ms[  -11ms] +/- 117ms
^- 37.44.185.42              3   7   377   10  -3274us[-3274us] +/-   70ms
^- bagnikita.com             2   7   377   74  +3131us[+3131us] +/-   71ms
^- europa.ellipse.net        2   8   377   204   -790us[ -773us] +/-   97ms
^- tethys.hot-chilli.net     2   8   377   141   -797us[ -797us] +/-   59ms
^- 66-232-97-8.static.hvvc.> 2   7   377   206  +1669us[+1686us] +/- 133ms
^+ 85.199.214.102            1   8   377   205  +175us[ +192us] +/-   12ms
^* 46-243-26-34.tangos.nl    1   8   377   141   -123us[ -106us] +/-   10ms
^- pugot.canonical.com       2   8   377   21    -95us[  -95us] +/-   57ms
^- alphyn.canonical.com      2   6   377   23  -1569us[-1569us] +/-   79ms
^- golem.canonical.com       2   7   377   92  -1018us[-1018us] +/-   31ms
^- chilipepper.canonical.com 2   8   377   21  -1106us[-1106us] +/-   27ms
```

²³⁹ <https://manpages.ubuntu.com/manpages/resolute/man5/chrony.conf.5.html>

210 Number of sources = 20

Name/IP Address	NP	NR	Span	Frequency	Freq Skew	Offset	Std Dev
gamma.rueckgr.at	25	15	32m	-0.007	0.142	-878us	106us
2b.ncomputers.org	26	16	35m	-0.132	0.283	-1169us	256us
www.kashra.com	25	15	32m	-0.092	0.259	+3426us	195us
stratum2-4.NTP.TechFak.U>	25	14	32m	-0.018	0.130	-2056us	96us
zepto.mcl.gg	13	11	21m	+0.148	0.196	-683us	66us
mirrorhost.pw	6	5	645	+0.117	0.445	-591us	19us
atto.mcl.gg	21	13	25m	-0.069	0.199	-904us	103us
static.140.107.46.78.cli>	25	18	34m	-0.005	0.094	-1526us	78us
4.53.160.75	25	10	32m	+0.412	0.110	-11ms	84us
37.44.185.42	24	12	30m	-0.983	0.173	-3718us	122us
bagnikita.com	17	7	31m	-0.132	0.217	+3527us	139us
europa.ellipse.net	26	15	35m	+0.038	0.553	-473us	424us
tethys.hot-chilli.net	25	11	32m	-0.094	0.110	-864us	88us
66-232-97-8.static.hvvc.>	20	11	35m	-0.116	0.165	+1561us	109us
85.199.214.102	26	11	35m	-0.054	0.390	+129us	343us
46-243-26-34.tangos.nl	25	16	32m	+0.129	0.297	-307us	198us
pugot.canonical.com	25	14	34m	-0.271	0.176	-143us	135us
alphyn.canonical.com	17	11	1100	-0.087	0.360	-1749us	114us
golem.canonical.com	23	12	30m	+0.057	0.370	-988us	229us
chilipepper.canonical.com	25	18	34m	-0.084	0.224	-1116us	169us

Pulse-Per-Second (PPS) support

GPSD

jitter

Example configuration for GPSD to feed chrony

```
$ sudo apt install gpsd chrony
```

```
$ sudo apt install pps-tools gpsd-clients
```

GPSD hardware²⁴⁰*dmesg*

```
[ 52.442199] usb 1-1.1: new full-speed USB device number 3 using xhci_hcd
[ 52.546639] usb 1-1.1: New USB device found, idVendor=067b, idProduct=2303,
bcdDevice= 4.00
[ 52.546654] usb 1-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 52.546665] usb 1-1.1: Product: USB-Serial Controller D
[ 52.546675] usb 1-1.1: Manufacturer: Prolific Technology Inc.
[ 52.602103] usbcore: registered new interface driver usbserial_generic
[ 52.602244] usbserial: USB Serial support registered for generic
[ 52.609471] usbcore: registered new interface driver pl2303
[ 52.609503] usbserial: USB Serial support registered for pl2303
[ 52.609564] pl2303 1-1.1:1.0: pl2303 converter detected
[ 52.618366] usb 1-1.1: pl2303 converter now attached to ttyUSB0
```

```
refclock SHM 0 refid GPS precision 1e-1 offset 0.9999 delay 0.2
refclock SOCK /var/run/chrony.ttyUSB0.sock refid PPS
```

```
$ sudo systemctl restart chrony
```

```
DEVICES="/dev/ttyUSB0"
```

•

—

•

—

²⁴⁰ <https://gpsd.gitlab.io/gpsd/hardware.html>

```
pps_ldisc: PPS line discipline registered
pps pps0: new PPS source usbserial0
pps pps0: source "/dev/ttyUSB0" added
```

```
pps0: name=usbserial0 path=/dev/ttyUSB0
```

```
...
| Status:          3D FIX (7 secs) ...
```

```
trying PPS source "/dev/pps0"
found PPS source "/dev/pps0"
ok, found 1 source(s), now start fetching data...
source 0 - assert 1588140739.099526246, sequence: 69 - clear 1588140739.999663721,
sequence: 70
source 0 - assert 1588140740.099661485, sequence: 70 - clear 1588140739.999663721,
sequence: 70
source 0 - assert 1588140740.099661485, sequence: 70 - clear 1588140740.999786664,
sequence: 71
source 0 - assert 1588140741.099792447, sequence: 71 - clear 1588140740.999786664,
sequence: 71
```

```
chronyc> sources
```

```
210 Number of sources = 10
```

```
MS Name/IP address          Stratum Poll Reach LastRx Last sample
```

```
=====
#? GPS                      0    4    0    -    +0ns[  +0ns] +/-    0ns
#? PPS                      0    4    0    -    +0ns[  +0ns] +/-    0ns
```

```
chronyc> sources
```

```
210 Number of sources = 10
```

```
MS Name/IP address      Stratum Poll Reach LastRx Last sample
```

```
=====
#x GPS                  0   4   177   24 -876ms[ -876ms] +/- 200ms
#- PPS                  0   4   177   21 +916us[ +916us] +/- 63us
^- chilipepper.canonical.com 2   6   37   53 +33us[ +33us] +/- 33ms
```

```
chronyc> sources
```

```
210 Number of sources = 10
```

```
MS Name/IP address      Stratum Poll Reach LastRx Last sample
```

```
=====
#x GPS                  0   4   377   20 -884ms[ -884ms] +/- 200ms
#* PPS                  0   4   377   18 +6677ns[ +52us] +/- 58us
^- alphyn.canonical.com 2   6   377   20 -1303us[-1258us] +/- 114ms
```

```
chronyc> sources
```

```
210 Number of sources = 11
```

```
MS Name/IP address      Stratum Poll Reach LastRx Last sample
```

```
=====
#? GPS                  0   4     0    -  +0ns[ +0ns] +/- 0ns
#+ PPS                  0   4   377   22 +154us[ +154us] +/- 8561us
^* chilipepper.canonical.com 2   6   377   50 -353us[ -300us] +/- 44ms
```

```
chronyc> sourcestats
```

```
210 Number of sources = 10
```

```
Name/IP Address        NP  NR  Span  Frequency  Freq Skew  Offset  Std Dev
```

```
=====
GPS                    20   9  302   +1.993    11.501    -868ms  1208us
PPS                     6   3   78    +0.324     5.009    +3365ns  41us
golem.canonical.com    15  10  783    +0.859     0.509    -750us  108us
```

```
$ sudo ntpshmmon -o
```

```
ntpshmmon: version 3.20
```

#	Name	Offset	Clock	Real	L	Prc
sample	NTP1	0.000223854	1588265805.000223854	1588265805.000000000	0	-10
sample	NTP0	0.125691783	1588265805.125999851	1588265805.000308068	0	-20
sample	NTP1	0.000349341	1588265806.000349341	1588265806.000000000	0	-10
sample	NTP0	0.130326636	1588265806.130634945	1588265806.000308309	0	-20
sample	NTP1	0.000485216	1588265807.000485216	1588265807.000000000	0	-10

NTS Support

curity “NTS”²⁴¹

Network Time Secu-

NTS server

```
ntsservercert /etc/chrony/fullchain.pem  
ntsserverkey /etc/chrony/privkey.pem
```

```
$ sudo chown _chrony:_chrony /etc/chrony/*.pem
```

```
$ sudo chronyc -N serverstats
```

²⁴¹ <https://web.archive.org/web/20241130024605/http://networktimesecurity.org/>

```
NTP packets received      : 213
NTP packets dropped       : 0
Command packets received  : 117
Command packets dropped   : 0
Client log records dropped : 0
NTS-KE connections accepted: 2
NTS-KE connections dropped : 0
Authenticated NTP packets : 197
```

```
$ sudo chronyc -N clients -k
```

Hostname	NTP	Drop	Int	IntL	Last	NTS-KE	Drop	Int	Last
=====	=====	=====	=====	=====	=====	=====	=====	=====	=====
10.172.196.173	197	0	10	-	595	2	0	5	48h
...									

*chrony.conf(5)*²⁴²

About certificate placement

chrony, by default, is isolated via AppArmor and uses a number of protect* features of systemd. Due to that, there are not many paths chrony can access for the certificates. But /etc/chrony/* is allowed as read-only and that is enough. Check /etc/apparmor.d/usr.sbin.chrond if you want other paths or allow custom paths in /etc/apparmor.d/local/usr.sbin.chrond.

Further reading

- [chrony FAQ](#)²⁴³
- [ntp.org: home of the Network Time Protocol project](#)²⁴⁴
- [pool.ntp.org: project of virtual cluster of timeservers](#)²⁴⁵
- [Freedesktop.org info on timedatectl](#)²⁴⁶
- [Freedesktop.org info on systemd-timesyncd service](#)²⁴⁷
- [Feeding chrony from GPSD](#)²⁴⁸
- [Ubuntu Time](#)²⁴⁹

²⁴² <https://manpages.ubuntu.com/manpages/resolute/man5/chrony.conf.5.html>

²⁴³ <https://chrony-project.org/faq.html>

²⁴⁴ <http://www.ntp.org/>

²⁴⁵ <https://www.ntppool.org/en/>

²⁴⁶ <https://www.freedesktop.org/software/systemd/man/latest/timedatectl.html>

²⁴⁷ <https://www.freedesktop.org/software/systemd/man/latest/systemd-timesyncd.service.html>

²⁴⁸ https://gpsd.gitlab.io/gpsd/gpsd-time-service-howto.html#_feeding_chrony_from_gpsd

²⁴⁹ <https://help.ubuntu.com/community/UbuntuTime>

Network shares

Network File System (NFS)

-
-
-

Warning:

If you use **NFS** and **authd** at the same time, you must add a Kerberos configuration on both the client and the server. Otherwise, you will encounter permission issues due to mismatched user and group identifiers. For details, see [Using authd with NFS](https://documentation.ubuntu.com/authd/stable-docs/howto/use-with-nfs/)²⁵⁰.

²⁵⁰ <https://documentation.ubuntu.com/authd/stable-docs/howto/use-with-nfs/>

Installation

```
$ sudo apt install nfs-kernel-server
```

```
$ sudo systemctl start nfs-kernel-server.service
```

Configuration

```
/srv      *(ro,sync,subtree_check)
/home     *.hostname.com(rw,sync,no_subtree_check)
/scratch  *(rw,async,no_subtree_check,no_root_squash)
```

```
$ sudo mkdir /scratch
```

```
$ sudo exportfs -a
```

hostname

filesystem

*exports(5)*²⁵¹

²⁵¹ <https://manpages.ubuntu.com/manpages/resolute/man5/exports.5.html>

NFS Client Configuration

```
$ sudo apt install nfs-common
```

```
$ sudo mkdir /opt/example
```

```
$ sudo mount example.hostname.com:/srv /opt/example
```

Warning:

The mount point directory `/opt/example` must exist. There should be no files or subdirectories in the `/opt/example` directory, else they will become inaccessible until the `nfs` filesystem is unmounted.

```
example.hostname.com:/srv /opt/example nfs
```

Note:

Details about all the NFS related options that can go into `/etc/fstab` can be found in the [*nfs\(5\)*](#)²⁵² manual page.

²⁵² <https://manpages.ubuntu.com/manpages/resolute/man5/nfs.5.html>

```
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
```

```
$ sudo systemctl daemon-reload
```

Advanced configuration

*nfs.conf(5)*²⁵³

*nfsconf(8)*²⁵⁴

Restarting NFS services

Service	nfs-utils. service	nfs-server. service	config file
nfs-blkmap	PartOf		nfs.conf
nfs-mountd		BindsTo	nfs.conf
nfsdclld			
nfs-idmapd		BindsTo	nfs.conf, idmapd.conf
rpc-gssd	PartOf		nfs.conf
rpc-statd	PartOf		nfs.conf
rpc-svcgssd	PartOf	BindsTo	nfs.conf

²⁵³ <https://manpages.ubuntu.com/manpages/resolute/man5/nfs.conf.5.html>

²⁵⁴ <https://manpages.ubuntu.com/manpages/resolute/man8/nfsconf.8.html>

*nfs.systemd(7)*²⁵⁵

NFS with Kerberos

NFS server (using Kerberos)

- *DNS*
-
-
-
-

```
$ sudo apt install krb5-user
```

```
$ kadmin -p ubuntu/admin -q "addprinc -randkey nfs/nfs-server.vms"
```

```
$ sudo kadmin -p ubuntu/admin -q "ktadd nfs/nfs-server.vms"
```

Authenticating as principal ubuntu/admin with password.

Password for ubuntu/admin@VMS:

Entry for principal nfs/nfs-server.vms with kvno 2, encryption type aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.

Entry for principal nfs/nfs-server.vms with kvno 2, encryption type aes128-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.

²⁵⁵ <https://manpages.ubuntu.com/manpages/resolute/man7/nfs.systemd.7.html>

```
$ sudo klist -k
```

```
Keytab name: FILE:/etc/krb5.keytab  
KVNO Principal
```

```
-----  
  2 nfs/nfs-server.vms@VMS  
  2 nfs/nfs-server.vms@VMS
```

```
$ sudo apt install nfs-kernel-server
```

```
$ sudo systemctl restart nfs-server
```

```
/storage *(rw,sync,no_subtree_check,sec=krb5p)
```

```
$ sudo exportfs -rav
```

```
exporting */:/storage
```

exports(5)²⁵⁶

-
-
-

²⁵⁶ <https://manpages.ubuntu.com/manpages/resolute/man5/exports.5.html>

NFS client (using Kerberos)

```
$ sudo apt install krb5-user
```

```
$ kadmin -p ubuntu/admin -q "addprinc -randkey host/nfs-client.vms"
```

```
$ sudo kadmin -p ubuntu/admin -q "ktadd host/nfs-client.vms"
```

```
$ sudo apt install nfs-common
```

```
$ sudo systemctl restart nfs-client.target
```

```
$ sudo mount nfs-server:/storage /mnt
```

```
# mount nfs-server:/storage /mnt
# ls -l /mnt/*
-rw-r--r-- 1 root root 0 Apr  5 14:50 /mnt/hello-from-nfs-server.txt
# klist
klist: No credentials cache found (filename: /tmp/krb5cc_0)
```

```
# sudo -u ubuntu -i
$ ls -l /mnt/*
ls: cannot access '/mnt/*': Permission denied
```

```
$ kinit
```

```
Password for ubuntu@VMS:
```

```
$ ls -l /mnt/*
```

```
-rw-r--r-- 1 root root 0 Apr  5 14:50 /mnt/hello-from-nfs-server.txt
```

```
$ klist
```

```
Ticket cache: FILE:/tmp/krb5cc_1000
```

```
Default principal: ubuntu@VMS
```

Valid starting	Expires	Service principal
04/05/22 17:48:50	04/06/22 03:48:50	krbtgt/VMS@VMS
	renew until 04/06/22 17:48:48	
04/05/22 17:48:52	04/06/22 03:48:50	nfs/nfs-server.vms@
	renew until 04/06/22 17:48:48	
Ticket server: nfs/nfs-server.vms@VMS		

rpc.gssd(8)²⁵⁷

•

Warning:

Note that this prevents automatic NFS mounts via `/etc/fstab`, unless a Kerberos ticket is obtained before.

[gssd]

`use-machine-creds=0`

²⁵⁷ <https://manpages.ubuntu.com/manpages/resolute/man8/rpc.gssd.8.html>

Upgrading from Ubuntu 20.04 (“Focal”) and earlier

Warning:

The `NEED_*` parameters have no effect on systemd-based installations, like Ubuntu 20.04 LTS (“Focal”) and Ubuntu 18.04 LTS (“Bionic”). In those systems, to control whether a service should be running or not, use `systemctl enable` or `systemctl disable`, respectively.

-
-

[file a bug for the nfs-utils package](#)²⁵⁸

²⁵⁸ <https://bugs.launchpad.net/ubuntu/+source/nfs-utils/+filebug>

Samba

Provisioning a Samba Active Directory Domain Controller

See also

For more information on why the Samba AD/DC server should not be used to provide file and print services, please refer to this [list of reasons and caveats in the Samba Wiki](#)²⁵⁹.

Installation

```
$ sudo apt install samba-ad-dc krb5-user bind9-dnsutils
```

```
$ sudo systemctl disable --now smbd nmbd winbind
```

```
$ sudo systemctl mask smbd nmbd winbind
```

```
$ sudo systemctl unmask samba-ad-dc
```

```
$ sudo systemctl enable samba-ad-dc
```

²⁵⁹ [https://wiki.samba.org/index.php/Setting_up_Samba_as_an_Active_Directory_Domain_Controller#Using_the_Domain_Controller_as_a_File_Server_\(Optional\)](https://wiki.samba.org/index.php/Setting_up_Samba_as_an_Active_Directory_Domain_Controller#Using_the_Domain_Controller_as_a_File_Server_(Optional))

Note:

A Samba AD/DC deployment represents a collection of services connected to each other, and needs its own specific systemd service unit.

```
$ sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.orig
```

Provisioning

-
-
-

```
$ sudo samba-tool domain provision \ > --domain EXAMPLE \ >  
--realm=EXAMPLE.INTERNAL \ > --adminpass=Passw0rd \ > --server-role=dc \ >  
--use-rfc2307 \ > --dns-backend=SAMBA_INTERNAL
```

Warning:

Providing passwords in the command line is generally unsafe. Other users on the system who can see the process listing can spot the password, and it will also be saved in the shell history, unless the command starts with a blank space.

```
(...)  
INFO ... #498: Server Role:      active directory domain controller  
INFO ... #499: Hostname:        ad  
INFO ... #500: NetBIOS Domain:  EXAMPLE  
INFO ... #501: DNS Domain:      example.internal  
INFO ... #502: DOMAIN SID:      S-1-5-21-2373640847-2123283686-338028823
```

```
INFO ... #497: Admin password: sbruR-Py>];k=KdN1H58PB#
```

Post-installation steps

DNS

```
[global]  
dns forwarder = 127.0.0.53
```

```
Global  
    Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported  
    resolv.conf mode: stub  
  
Link 2 (enp5s0)  
    Current Scopes: DNS  
        Protocols: +DefaultRoute -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported  
Current DNS Server: 10.10.17.1  
    DNS Servers: 10.10.17.1  
    DNS Domain: lxd
```

```
[global]  
dns forwarder = 10.10.17.1
```

```
$ sudo unlink /etc/resolv.conf
```

Note:

This will make sudo issue complaints about DNS from this point on, until the Samba DNS service is up and running.

```
nameserver 127.0.0.1  
search example.internal
```

```
$ sudo systemctl disable --now systemd-resolved
```

```
$ sudo cp -f /var/lib/samba/private/krb5.conf /etc/krb5.conf
```

```
$ sudo systemctl start samba-ad-dc
```

Verification

Kerberos authentication

```
$ kinit Administrator
```

```
Password for Administrator@EXAMPLE.INTERNAL:
```

See also

If you are not familiar with Kerberos, please see our [Introduction to Kerberos](#) (page 846).

```
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: Administrator@EXAMPLE.INTERNAL

Valid starting    Expires          Service principal
07/24/24 13:52:33 07/24/24 23:52:33 krbtgt/EXAMPLE.INTERNAL@EXAMPLE.INTERNAL
    renew until 07/25/24 13:52:02
```

DNS tests

name

host-

```
$ hostname
```

```
$ hostname -f
```

Server Information

```
$ samba-tool dns serverinfo $(hostname -f)

dwVersion           : 0xece0205
fBootMethod         : DNS_BOOT_METHOD_DIRECTORY
fAdminConfigured    : FALSE
fAllowUpdate        : TRUE
fDsAvailable        : TRUE
pszServerName       : AD.example.internal
pszDsContainer      : CN=MicrosoftDNS,DC=DomainDnsZones,DC=example,DC=internal
(...)
```

```
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: Administrator@EXAMPLE.INTERNAL

Valid starting    Expires          Service principal
07/24/24 14:29:55 07/25/24 00:29:55 krbtgt/EXAMPLE.INTERNAL@EXAMPLE.INTERNAL
                renew until 07/25/24 14:29:53
07/24/24 14:29:59 07/25/24 00:29:55 host/ad.example.internal@EXAMPLE.INTERNAL
                renew until 07/25/24 14:29:53
```

DNS records

```
for srv in _ldap._tcp._kerberos._tcp._kerberos._udp._kpasswd._udp; do
    echo -n "${srv}.example.internal: "
    dig @localhost +short -t SRV ${srv}.example.internal
done
```

```
_ldap._tcp.example.internal: 0 100 389 ad.example.internal.
_kerberos._tcp.example.internal: 0 100 88 ad.example.internal.
_kerberos._udp.example.internal: 0 100 88 ad.example.internal.
_kpasswd._udp.example.internal: 0 100 464 ad.example.internal.
```

```
$ dig @localhost +short -t A $(hostname -f)
```

User creation test

```
$ samba-tool user add noble
```

Note:

samba-tool creates **Samba** users, not local Linux users! These Samba users only exist for domain joined computers and other SMB connections/shares.

```
$ sudo samba-tool domain passwordsettings show

Password information for domain 'DC=example,DC=internal'

Password complexity: on
Store plaintext passwords: off
Password history length: 24
Minimum password length: 7
Minimum password age (days): 1
Maximum password age (days): 42
Account lockout duration (mins): 30
Account lockout threshold (attempts): 0
Reset account lockout after (mins): 30
```

```
$ sudo samba-tool domain passwordsettings set --min-pwd-length=12
```

```
$ samba-tool domain passwordsettings set --help
```

Next steps

Further reading

- [Active Directory Domain Services Overview](#)²⁶⁰
- *samba-tool(8)*²⁶¹
- - *Choosing an integration method*

²⁶⁰ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>

²⁶¹ <https://manpages.ubuntu.com/manpages/resolute/man8/samba-tool.8.html>

- *Joining a Member Server*
- *Joining Workstations (without Samba services)*

Member server in an Active Directory domain

SSSD

Network User Authentication with

See also

Samba itself has the necessary tooling to join an Active Directory domain. It requires a sequence of manual steps and configuration file editing, which is **thoroughly documented on the Samba wiki**²⁶². It's useful to read that documentation to get an idea of the steps necessary, and the decisions you will need to make.

Use realmd to join the Active Directory domain

Install realmd

```
$ sudo apt install realmd samba
```

DNS

*FQDN
hostname*

```
$ sudo hostnamectl hostname <yourfqdn>
```

²⁶² https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Domain_Member

Verify the AD server

```
$ sudo realm discover internal.example.fake

internal.example.fake
  type: kerberos
  realm-name: INTERNAL.EXAMPLE.FAKE
  domain-name: internal.example.fake
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
```

Join the AD domain

```
$ sudo realm join -v --membership-software=samba --client-software=winbind
internal.example.fake

* Resolving: _ldap._tcp.internal.example.fake
* Performing LDAP DSE lookup on: 10.0.16.5
* Successfully discovered: internal.example.fake
Password for Administrator:
* Unconditionally checking packages
* Resolving required packages
* Installing necessary packages: libnss-winbind samba-common-bin libpam-winbind
winbind
* LANG=C LOGNAME=root /usr/bin/net --configfile /var/cache/realmd/realmd-smb-conf.
A53N01 -U Administrator --use-kerberos=required ads join internal.example.fake
Password for [INTEAMPLE\Administrator]:
Using short domain name -- INTEAMPLE
Joined 'J1' to dns domain 'internal.example.fake'
* LANG=C LOGNAME=root /usr/bin/net --configfile /var/cache/realmd/realmd-smb-conf.
A53N01 -U Administrator ads keytab create
Password for [INTEAMPLE\Administrator]:
* /usr/sbin/update-rc.d winbind enable
* /usr/sbin/service winbind restart
* Successfully enrolled machine in realm
```


Note:

This command also installed the `libpam-winbind` package, **which allows AD users to authenticate to other services on this system via PAM, like SSH or console logins**. For example, if your SSH server allows password authentication (`PasswordAuthentication yes` in `/etc/ssh/sshd_config`), then the domain users will be allowed to login remotely on this system via SSH. If you don't expect or need AD users to log into this system (unless it's via Samba or Windows), then it's safe and probably best to remove the `libpam-winbind` package.

LP: #1980246²⁶³

-

```
passwd:      files systemd winbind
group:       files systemd winbind
```

```
$ getent passwd INTEXAMPLE\Administrator
```

```
INTEXAMPLE administrator:*:2000500:2000513::/home/administrator@INTEXAMPLE:/
bin/bash
```

Common installation options

*smb.conf(5)*²⁶⁴

-
-
-
-

²⁶³ <https://bugs.launchpad.net/bugs/1980246>

²⁶⁴ <https://manpages.ubuntu.com/manpages/resolute/man5/smb.conf.5.html>

```
$ sudo pam-auth-update --enable mkhomedir
```

Export shares

```
[storage]
path = /storage
comment = Storage share
writable = yes
guest ok = no
```

```
$ sudo mkdir -m 1777 /storage
```

```
$ sudo smbcontrol smbd reload-config
```

```
$ smbclient //j1.internal.example.fake/storage -U INTEXAMPLE\\ubuntu
Enter INTEXAMPLE\\ubuntu's password:
Try "help" to get a list of possible commands.
smb: \>
```

```
$ sudo smbstatus
```

```
Samba version 4.15.5-Ubuntu
PID      Username      Group      Machine
Version  Encryption    Signing
-----
3631     INTEXAMPLE\ubuntu INTEXAMPLE\domain users 10.0.16.1 (ipv4:10.0.16.1:39534)
SMB3_11  -              partial(AES-128-CMAC)

Service  pid    Machine    Connected at      Encryption
Signing
-----
storage  3631   10.0.16.1  Wed Jun 29 17:42:54 2022 UTC  -          -

No locked files
```

```
[storage]
path = /storage
comment = Storage share
writable = yes
guest ok = no
valid users = "@INTEXAMPLE\ LTS Releases"
```

Choose an idmap backend

- *idmap_ad(8)*²⁶⁵
- *idmap_autorid(8)*²⁶⁶
- *idmap_rid(8)*²⁶⁷

Choosing an idmap backend²⁶⁸

²⁶⁵ https://manpages.ubuntu.com/manpages/resolute/man8/idmap_ad.8.html

²⁶⁶ https://manpages.ubuntu.com/manpages/resolute/man8/idmap_autorid.8.html

²⁶⁷ https://manpages.ubuntu.com/manpages/resolute/man8/idmap_rid.8.html

²⁶⁸ https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Domain_Member#Choosing_an_idmap_backend

```
idmap config * : range = 10000-999999
idmap config intexample : backend = rid
idmap config intexample : range = 2000000-2999999
idmap config * : backend = tdb
```

```
$ id INTEXAMPLE\Administrator

uid=2000500(INTEXAMPLE administrator)
gid=2000513(INTEXAMPLE domain users)
groups=2000513(INTEXAMPLE domain users),
        2000500(INTEXAMPLE administrator),
        2000572(INTEXAMPLE denied rod password replication group),
        2000519(INTEXAMPLE enterprise admins),
        2000518(INTEXAMPLE schema admins),
        2000520(INTEXAMPLE group policy creator owners),
        2000512(INTEXAMPLE domain admins),
        10001(BUILTIN users),
        10000(BUILTIN administrators)
```

Further reading

- [The Samba Wiki](#)²⁶⁹

Set up sharing services

²⁶⁹ <https://wiki.samba.org>

Set up Samba as a file server

Share Access Control

Warning:

If you use **Samba and authd** at the same time, you must specify user and group mapping. Otherwise, you will encounter permission issues due to mismatched user and group identifiers.

If you *are* using Samba with authd, you should follow the instructions in the [steps for the server](#)²⁷⁰ guide in the authd documentation instead.

²⁷⁰ <https://documentation.ubuntu.com/authd/edge-docs/howto/use-with-samba/#steps-for-the-server>

Install Samba

```
$ sudo apt install samba
```

Configure Samba as a file server

Note:

Not all available options are included in the default configuration file. See the [smb.conf man page](#)²⁷¹ or the [Samba HOWTO Collection](#)²⁷² for more details.

²⁷¹ <https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html>

²⁷² <https://www.samba.org/samba/docs/old/Samba3-HOWTO/>

```
workgroup = EXAMPLE
```

[share]

```
comment = Ubuntu File Server Share
path = /srv/samba/share
browsable = yes
guest ok = yes
read only = no
create mask = 0755
```

-
-

Note:

This example uses `/srv/samba/sharename` because, according to the *Filesystem Hierarchy Standard (FHS)*, `/srv273` is where site-specific data should be served. Technically, Samba shares can be placed anywhere on the *filesystem* as long as the permissions are correct, but adhering to standards is recommended.

²⁷³ <https://www.pathname.com/fhs/pub/fhs-2.3.html#SRVDATAFORSERVICESPROVIDEDBYSYSTEM>

-
-
-
-

Create the directory

```
$ sudo mkdir -p /srv/samba/share
```

```
$ sudo chown nobody:nogroup /srv/samba/share/
```

Enable the new configuration

```
$ sudo systemctl restart smbd.service nmbd.service
```

Warning:

Once again, the above configuration gives full access to any client on the local network. For a more secure configuration see [Share Access Control](#) (page 376).

Further reading

- [Samba how-to collection](#)²⁷⁴
- [Samba in printed format](#)²⁷⁵
- [Using Samba](#)²⁷⁶
- [Ubuntu Wiki Samba](#)²⁷⁷

Set up Samba as a print server

using Samba as a file server

Share access controls

²⁷⁴ <https://www.samba.org/samba/docs/old/Samba3-HOWTO/>

²⁷⁵ <https://www.amazon.com/exec/obidos/tg/detail/-/0131882228>

²⁷⁶ <https://www.oreilly.com/library/view/using-samba-3rd/0596007698/>

²⁷⁷ <https://help.ubuntu.com/community/Samba>

Install and configure CUPS

our guide on CUPS

Install Samba

```
$ sudo apt install samba
```

Configure Samba

```
workgroup = EXAMPLE
```

```
browsable = yes  
guest ok = yes
```

```
$ sudo systemctl restart smbd.service nmbd.service
```

Further reading

- [Samba HOWTO Collection](#)²⁷⁸
- [Samba HOWTO Collection](#) in [printed format](#)²⁷⁹
- [Using Samba](#)²⁸⁰
- [CUPS Website](#)²⁸¹
- [Ubuntu Wiki Samba](#)²⁸²

²⁷⁸ <https://www.samba.org/samba/docs/old/Samba3-HOWTO/>

²⁷⁹ <http://www.amazon.com/exec/obidos/tg/detail/-/0131882228>

²⁸⁰ <http://www.oreilly.com/catalog/9780596007690/>

²⁸¹ <http://www.cups.org/>

²⁸² <https://help.ubuntu.com/community/Samba>

Access controls

Share access controls

Authenticating Samba users

See also

How to create and manage Linux users is covered in *Users and groups management* (page 154).

```
$ sudo smbpasswd -a melissa
```

Note:

As this is a separate user database, the password selected for the Samba user does not need to be the same as the Linux password for that user. In fact, most Samba servers setup this way will have the Linux users setup without a valid password: these users only exist so that the corresponding Samba users can be created.

```
$ sudo smbpasswd melissa
```

Groups

Read and write permissions

```
read list = @qa  
write list = @sysadmin, vincent
```

```
admin users = melissa
```

Note:

Remember that the users listed in `smb.conf` for these access controls need to exist both as Linux users, and Samba users.

```
$ sudo smbcontrol smbd reload-config
```

Filesystem permissions

filesystem

ACL

```
UUID=66bcdd2e-8861-4fb0-b7e4-e61c569fe17d /srv ext3 noatime,relatime,acl 0 1
```

```
$ sudo mount -v -o remount /srv
```

Note:

This example assumes `/srv` is on a separate partition. If `/srv`, or wherever you have configured your share path, is part of the `/` partition then a reboot may be required.

```
$ sudo chown -R melissa /srv/samba/share/
```

```
$ sudo chgrp -R sysadmin /srv/samba/share/
```

```
$ sudo setfacl -R -m g:qa:rx /srv/samba/share/
```

Note:

The `setfacl` command above gives *execute* permissions to all files in the `/srv/samba/share` directory, which you may or may not want.

*acl(5)*²⁸³ *setfacl(1)*²⁸⁴

²⁸³ <https://manpages.ubuntu.com/manpages/resolute/man5/acl.5.html>

²⁸⁴ <https://manpages.ubuntu.com/manpages/resolute/man1/setfacl.1.html>

Further reading

- [Samba HOWTO Collection](#)²⁸⁵
- [Samba HOWTO Collection](#) printed format²⁸⁶
- [Using Samba](#)²⁸⁷
- [Chapter 18](#)²⁸⁸
- [Samba ACLs page](#)²⁸⁹
- [Ubuntu Wiki Samba](#)²⁹⁰

Create a Samba AppArmor profile

in this guide

Install apparmor-profiles

```
$ sudo apt install apparmor-profiles apparmor-utils
```

Note:

This package contains profiles for several other binaries.

AppArmor profile modes

²⁸⁵ <https://www.samba.org/samba/docs/old/Samba3-HOWTO/>

²⁸⁶ <http://www.amazon.com/exec/obidos/tg/detail/-/0131882228>

²⁸⁷ <http://www.oreilly.com/catalog/9780596007690/>

²⁸⁸ <https://www.samba.org/samba/docs/old/Samba3-HOWTO/securing-samba.html>

²⁸⁹ <https://www.samba.org/samba/docs/old/Samba3-HOWTO/AccessControls.html>

²⁹⁰ <https://help.ubuntu.com/community/Samba>

```
Jun 30 14:41:09 ubuntu kernel: [ 621.478989] audit:
type=1400 audit(1656600069.123:418):
apparmor="ALLOWED" operation="exec" profile="smbd"
name="/usr/lib/x86_64-linux-gnu/samba/samba-bgqd" pid=4122 comm="smbd"
requested_mask="x" denied_mask="x" fsuid=0 ouid=0
target="smbd//null-/usr/lib/x86_64-linux-gnu/samba/samba-bgqd"
```

```
$ sudo aa-enforce /usr/sbin/smbd
```

```
$ sudo apparmor_parser -r -W -T /etc/apparmor.d/usr.sbin.smbd
```

Further reading

- [the Debian AppArmor guide²⁹¹](#)

How to mount CIFS shares permanently

Introduction to Samba

Prerequisites

-
-
-

²⁹¹ <https://wiki.debian.org/AppArmor/HowToUse>

-
-
-

Install CIFS

```
$ sudo apt-get install cifs-utils
```

Mount unprotected (guest) network folders

```
$ sudo mkdir /media/windowsshare
```

```
//servername/sharename /media/windowsshare cifs guest,uid=1000 0 0
```

- *hostname*
-
-
-

```
$ sudo mount /media/windowsshare
```

Mount password-protected network folders

```
//servername/sharename /media/windowsshare cifs username=msusername,password=mspassword 0 0
```

Create a credentials file

```
$ gedit ~/.smbcredentials
```

```
username=msusername  
password=mppassword
```

```
$ chmod 600 ~/.smbcredentials
```

```
//servername/sharename /media/windowsshare cifs credentials=/home/ubuntuusername/.  
smbcredentials 0 0
```

```
$ sudo mount /media/windowsshare
```

Changing the share ownership

GID

```
//servername/sharename /media/windowsshare cifs uid=ubuntuusername,credentials=/home/  
ubuntuusername/.smbcredentials 0 0
```

Mount network folders with authd

steps for the client²⁹²

Mount password-protected shares using libpam-mount

Install libpam-mount

```
$ sudo apt-get install libpam-mount
```

```
$ sudo gedit /etc/security/pam_mount.conf.xml
```

```
$ gedit ~/.pam_mount.conf.xml
```

²⁹² <https://documentation.ubuntu.com/authd/edge-docs/howto/use-with-samba/#steps-for-the-client>


```
<?xml version="1.0" encoding="utf-8" ?>

<pan_mount>

<volume options="uid=%(USER),gid=100,dir_mode=0700,credentials=/home/ubuntuusername/.
smbcredentials,nosuid,nodev" user="*" mountpoint="/media/windowsshare" path="sharename"
server="servername" fstype="cifs" />

</pan_mount>
```

Troubleshooting

Login errors

-
-
-
-

*mount.cifs(8)*²⁹³

Mount after login instead of boot

```
//servername/sharename /media/windowsshare cifs noauto,credentials=/home/ubuntuusername/.
smbpasswd 0 0
```

²⁹³ <https://manpages.ubuntu.com/manpages/resolute/man8/mount.cifs.8.html>

Legacy options

NT4 domain controller (legacy)

Note:

This section is flagged as *legacy* because nowadays, Samba can be deployed in full Active Directory domain controller mode, and the old-style NT4 Primary Domain Controller is deprecated.

Primary domain controller

Install Samba

```
$ sudo apt install samba libpam-winbind
```

Configure Samba

```
workgroup = EXAMPLE  
...  
security = user
```

```
domain logons = yes  
logon path = \\%N%\%U\profile  
logon drive = H:  
logon home = \\%N%\%U  
logon script = logon.cmd  
add machine script = sudo /usr/sbin/useradd -N -g machines -c Machine -d  
/var/lib/samba -s /bin/false %u
```

Note:

If you wish to not use *Roaming Profiles* leave the `logon home` and `logon path` options commented out.

-
-
-
-
-
-

Security - Users: Adding and Deleting Users

Mapping shares

[homes]

```
comment = Home Directories
browseable = no
read only = no
create mask = 0700
directory mask = 0700
valid users = %S
```

[netlogon]

```
comment = Network Logon Service
path = /srv/samba/netlogon
guest ok = yes
read only = yes
share modes = no
```

Note:

The original `netlogon` share path is `/home/samba/netlogon`, but according to the *Filesystem Hierarchy Standard (FHS)*, `/srv` is the correct location²⁹⁴ for site-specific data provided by the system.

²⁹⁴ <https://www.pathname.com/fhs/pub/fhs-2.3.html#SRVDATAFORSERVICESPROVIDEDBYSYSTEM>

```
$ sudo mkdir -p /srv/samba/netlogon
```

```
$ sudo touch /srv/samba/netlogon/logon.cmd
```

```
$ sudo systemctl restart smbd.service nmbd.service
```

Final setup tasks

```
$ sudo net groupmap add ntgroup="Domain Admins" unixgroup=sysadmin rid=512  
type=d
```

```
$ sudo smbpasswd -a sysadmin
```

```
$ net rpc rights grant -U sysadmin "EXAMPLE\Domain Admins"  
SeMachineAccountPrivilege \ > SePrintOperatorPrivilege SeAddUsersPrivilege  
SeDiskOperatorPrivilege \ > SeRemoteShutdownPrivilege
```

Backup domain controller

BDC

DAP Backend

Samba - OpenL-

```
$ sudo apt install samba libpam-winbind
```

```
workgroup = EXAMPLE  
...  
security = user
```

```
domain logons = yes  
domain master = no
```

```
$ sudo chgrp -R admin /var/lib/samba
```

```
$ sudo scp -r username@pdc:/var/lib/samba /var/lib
```

hostname

```
$ sudo systemctl restart smb.service nmbd.service
```

Further reading

- [Samba HOWTO Collection](#)²⁹⁵
- [in printed format](#)²⁹⁶
- [Using Samba](#)²⁹⁷
- [Chapter 4](#)²⁹⁸
- [Chapter 5](#)²⁹⁹
- [Ubuntu Wiki Samba](#)³⁰⁰

OpenLDAP backend (legacy)

Note:

This section is flagged as *legacy* because nowadays, Samba 4 is best integrated with its own LDAP server in Active Directory mode. Integrating Samba with LDAP as described here covers the NT4 mode, which has been deprecated for many years.

Install LDAP

LDAP with Transport Layer Security

²⁹⁵ <https://www.samba.org/samba/docs/old/Samba3-HOWTO/>

²⁹⁶ <https://www.amazon.com/exec/obidos/tg/detail/-/0131882228>

²⁹⁷ <https://www.oreilly.com/library/view/using-samba-3rd/0596007698/>

²⁹⁸ <https://www.samba.org/samba/docs/old/Samba3-HOWTO/samba-pdc.html>

²⁹⁹ <https://www.samba.org/samba/docs/old/Samba3-HOWTO/samba-bdc.html>

³⁰⁰ <https://help.ubuntu.com/community/Samba>

Install the software

```
$ sudo apt install samba smbldap-tools
```

Configure LDAP

-
-
-

Samba schema

```
$ sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f  
/usr/share/doc/samba/examples/LDAP/samba.ldif
```

```
$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=schema,cn=config  
'cn=*samba*'
```

Samba indices

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
replace: olcDbIndex
olcDbIndex: objectClass eq
olcDbIndex: uidNumber,gidNumber eq
olcDbIndex: loginShell eq
olcDbIndex: uid,cn eq,sub
olcDbIndex: memberUid eq,sub
olcDbIndex: member,uniqueMember eq
olcDbIndex: sambaSID eq
olcDbIndex: sambaPrimaryGroupSID eq
olcDbIndex: sambaGroupType eq
olcDbIndex: sambaSIDList eq
olcDbIndex: sambaDomainName eq
olcDbIndex: default sub,eq
```

```
$ sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f samba_indices.ldif
```

```
$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H \> ldapi:/// -b cn=config
olcDatabase={1}mdb olcDbIndex
```

Adding Samba LDAP objects

- *hostname*
-

```
$ sudo smbldap-config
```


-
-
-
-

DN

```
$ sudo smbldap-populate -g 10000 -u 10000 -r 10000
```

gid

Samba configuration

```
# passwd backend = tdbsam
workgroup = EXAMPLE

# LDAP Settings
passdb backend = ldapsam:ldap://ldap01.example.com
ldap suffix = dc=example,dc=com
ldap user suffix = ou=People
ldap group suffix = ou=Groups
ldap machine suffix = ou=Computers
ldap idmap suffix = ou=Idmap
ldap admin dn = cn=admin,dc=example,dc=com
ldap ssl = start tls
ldap passwd sync = yes
```

Note:

The `smb.conf` as shipped by the package is quite long and has many configuration examples. An easy way to visualize it without any comments is to run `testparm -s`.

```
$ sudo smbpasswd -W
```

Network User Authentication with SSSD

```
$ sudo apt install sssd-ldap
```

```
[sssds]  
config_file_version = 2  
domains = example.com  
  
[domain/example.com]  
id_provider = ldap  
auth_provider = ldap  
ldap_uri = ldap://ldap01.example.com  
cache_credentials = True  
ldap_search_base = dc=example,dc=com
```

```
$ sudo chmod 0600 /etc/sssds/sssds.conf
```

```
$ sudo chown root:root /etc/sssds/sssds.conf
```

```
$ sudo systemctl start sssds
```

```
$ sudo systemctl restart smbds.service nmbds.service
```

```
$ getent group Replicators  
Replicators:*:552:
```

Note:

The names are case sensitive!

```
$ sudo smbpasswd -a username
```

•

```
$ sudo smbldap-useradd -a -P -m username
```

```
$ getent passwd username
```

•

```
$ sudo smbldap-userdel username
```

•

```
$ sudo smbldap-groupadd -a groupname
```

•

```
$ sudo smbldap-groupmod -m username groupname
```

•

```
$ sudo smbldap-groupmod -x username groupname
```

•

```
$ sudo smbldap-useradd -t 0 -w username
```

Further reading

- [Upstream Samba documentation collection](#)³⁰¹
- [Upstream Samba wiki](#)³⁰²

See also

- [Introduction to Samba](#)

Active Directory integration

AD integration explanations

Join a domain with winbind: preparation

```
$ sudo apt install winbind libnss-winbind libpam-winbind
```

DNS

DHCP

```
network:  
  version: 2  
  ethernet:  
    eth0:  
      dhcp4: true
```

```
network:  
  version: 2  
  ethernet:  
    eth0:  
      dhcp4: true
```

(continues on next page)

³⁰¹ <https://www.samba.org/samba/docs/>

³⁰² https://wiki.samba.org/index.php/Main_Page

(continued from previous page)

```
nameservers
addresses  [10.10.4.5]
search:    [example.internal]
```

```
$ sudo netplan generate
```

```
$ sudo netplan apply
```

Note:

Be careful whenever changing network parameters over an SSH connection. If there are any mistakes, you might lose remote access!

```
$ resolvectl status

Global
    Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
    resolv.conf mode: stub

Link 281 (eth0)
    Current Scopes: DNS
        Protocols: +DefaultRoute -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
    DNS Servers: 10.10.4.5 10.10.4.1
    DNS Domain: example.internal
```

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:      files systemd winbind
group:       files systemd winbind
(...)
```

```
$ sudo pam-auth-update --enable mkhomedir
```

Join a simple domain with the rid backend

```
[global]
security = ads
realm = EXAMPLE.INTERNAL
workgroup = EXAMPLE

idmap config * : backend      = tdb
idmap config * : range       = 100000 - 199999
idmap config EXAMPLE : backend = rid
idmap config EXAMPLE : range  = 1000000 - 1999999

# allow logins when the DC is unreachable
winbind offline logon = yes
# this can be yes if there is absolute certainty that there is only a
# single domain involved
winbind use default domain = no
# setting these enumeration options to yes has a high performance impact
# and can cause instabilities
winbind enum groups = no
winbind enum users = no
winbind refresh tickets = yes
# if domain users should be allowed to login, they will need a login shell
template shell = /bin/bash
# the home directory template for domain users
template homedir = /home/%D/%U
kerberos method = secrets and keytab
```

```
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed by GnuTLS (e.g. NTLM as a compatibility fallback)

Server role: ROLE_DOMAIN_MEMBER

Press enter to see a dump of your service definitions
(...)
```

hostname

DNS

FQDN

```
$ sudo hostnamectl hostname n1.example.internal
```

```
$ sudo net ads join -U Administrator

Password for [EXAMPLE\Administrator]:
Using short domain name -- EXAMPLE
Joined 'N1' to dns domain 'example.internal'
```

```
$ sudo hostnamectl hostname n1
```

```
$ sudo systemctl restart winbind.service
```

Verifying the join

```
$ sudo wbinfo -t

checking the trust secret for domain EXAMPLE via RPC calls succeeded
```

```
$ id jammy@example.internal

uid=1001103(EXAMPLE\jammy) gid=1000513(EXAMPLE\domain users) groups=1000513(EXAMPLE\domain users),1001103(EXAMPLE\jammy)
```

```
$ id EXAMPLE\\jammy

uid=1001103(EXAMPLE\jammy) gid=1000513(EXAMPLE\domain users) groups=1000513(EXAMPLE\domain users),1001103(EXAMPLE\jammy)
```

```
$ n1 login: jammy@example.internal

Password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.5.0-26-generic x86_64)
(...)
Creating directory '/home/EXAMPLE/jammy'.

EXAMPLE\jammy@n1:~$
```

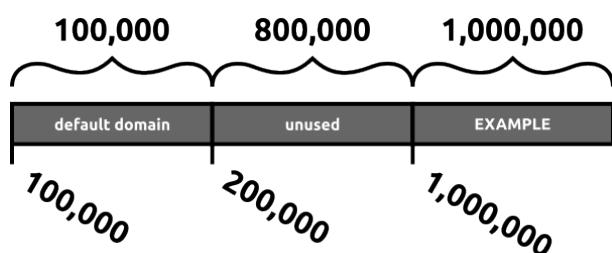
Note:

The actual login name used can have multiple formats: DOMAIN\user at the terminal login prompt, DOMAIN\\user when referred to in shell scripts (note the escaping of the '\' character), and user@domain is also accepted.

Join a forest with the rid backend

```
[global]
security = ads
realm = EXAMPLE.INTERNAL
workgroup = EXAMPLE

idmap config * : backend = tdb
# 100,000 - 199,999
idmap config * : range = 100000 - 199999
idmap config EXAMPLE : backend = rid
# 1,000,000 - 1,999,999
idmap config EXAMPLE : range = 1000000 - 1999999
```




```
$ testparm
```

```
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed by GnuTLS (e.g. NTLM as a compatibility fallback)
```

```
Server role: ROLE_DOMAIN_MEMBER
```

hostname

```
$ sudo hostnamectl hostname n3.example.internal
```

```
$ hostname
```

```
n3.example.internal
```

```
$ hostname -f
```

```
n3.example.internal
```

```
$ sudo net ads join -U Administrator
```

```
Password for [EXAMPLE\Administrator]:
Using short domain name -- EXAMPLE
Joined 'N3' to dns domain 'example.internal'
```

```
$ sudo hostnamectl hostname n3
```

```
$ sudo systemctl restart winbind.service
```

```
$ id jammy@example.internal
```

```
uid=1001103(EXAMPLE\jammy) gid=1000513(EXAMPLE\domain users) groups=1000513(EXAMPLE\
domain users),1001103(EXAMPLE\jammy)
```

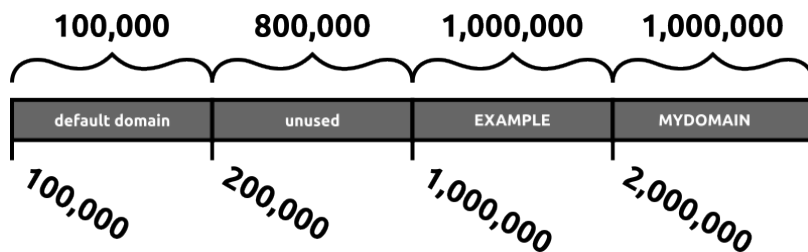
```
$ id noble@mydomain.internal
```

```
uid=1000000(MYDOMAIN\noble) gid=1000000(MYDOMAIN\domain users) groups=1000000(MYDOMAIN\
domain users)
```

```
[global]
security = ads
realm = EXAMPLE.INTERNAL
workgroup = EXAMPLE

idmap config * : backend          = tdb
# 100,000 - 199,999
idmap config * : range           = 100000 - 199999
idmap config EXAMPLE : backend = rid
# 1,000,000 - 1,999,999
idmap config EXAMPLE : range     = 1000000 - 1999999

# MYDOMAIN.INTERNAL idmap configuration
idmap config MYDOMAIN : backend = rid
# 2,000,000 - 2,999,999
idmap config MYDOMAIN : range   = 2000000 - 2999999
```



```
$ id jammy@example.internal

uid=1001103(EXAMPLE\jammy) gid=1000513(EXAMPLE\domain users) groups=1000513(EXAMPLE\
domain users),1001103(EXAMPLE\jammy)
```

```
$ id noble@mydomain.internal

uid=2001104(MYDOMAIN\noble) gid=2000513(MYDOMAIN\domain users)
groups=2000513(MYDOMAIN\domain users),2001104(MYDOMAIN\noble)
```

-
-
-
-

Join a forest with the autorid backend

[global]

```
security = ads
realm = EXAMPLE.INTERNAL
workgroup = EXAMPLE

idmap config * : backend = autorid
# 1,000,000 - 19,999,999
idmap config * : range   = 1000000 - 19999999
# 1,000,000
idmap config * : rangesize = 1000000

# allow logins when the DC is unreachable
winbind offline logon = yes
# this *can* be yes if there is absolute certainty that there is only a
# single domain involved
winbind use default domain = no
# setting these enumeration options to yes has a high performance impact
# and can cause instabilities
winbind enum groups = no
winbind enum users = no
winbind refresh tickets = yes
# if domain users should be allowed to login, they will need a login shell
template shell = /bin/bash
# the home directory template for domain users
template homedir = /home/%D/%U
kerberos method = secrets and keytab
```

-
-
-

```
$ testparm

Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed by GnuTLS (e.g. NTLM as a compatibility fallback)

Server role: ROLE_DOMAIN_MEMBER

Press enter to see a dump of your service definitions
```

DNS

hostname

```
$ sudo hostnamectl hostname n2.example.internal
```

```
$ sudo net ads join -U Administrator

Password for [EXAMPLE\Administrator]:
Using short domain name -- EXAMPLE
Joined 'N2' to dns domain 'example.internal'
```

```
$ sudo hostnamectl hostname n2
```

```
$ hostname

n2
```

```
$ hostname -f  
n2.example.internal
```

```
$ sudo systemctl restart winbind.service
```

Verifying the join

```
$ sudo wbinfo -t  
checking the trust secret for domain EXAMPLE via RPC calls succeeded
```

```
$ id jammy@example.internal  
uid=2001103(EXAMPLE\jammy) gid=2000513(EXAMPLE\domain users) groups=2000513(EXAMPLE\  
domain users),2001103(EXAMPLE\jammy)
```

```
$ id EXAMPLE\jammy  
uid=2001103(EXAMPLE\jammy) gid=2000513(EXAMPLE\domain users) groups=2000513(EXAMPLE\  
domain users),2001103(EXAMPLE\jammy)
```

```
$ n2 login: jammy@example.internal  
Password:  
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.5.0-26-generic x86_64)  
(...)  
Creating directory '/home/EXAMPLE/jammy'.  
EXAMPLE\jammy@n1:~$
```

```
$ id noble@mydomain.internal

uid=3001104(MYDOMAIN\noble) gid=3000513(MYDOMAIN\domain users)
groups=3000513(MYDOMAIN\domain users),3001104(MYDOMAIN\noble)
```

```
$ id MYDOMAIN\\noble

uid=3001104(MYDOMAIN\noble) gid=3000513(MYDOMAIN\domain users)
groups=3000513(MYDOMAIN\domain users),3001104(MYDOMAIN\noble)
```

```
$ n2 login: noble@mydomain.internal

Password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-31-generic x86_64)
(...)
Creating directory '/home/MYDOMAIN/noble'.

MYDOMAIN\noble@n2:~$
```

Note:

The actual login name used can have multiple formats: DOMAIN\user at the terminal login prompt, DOMAIN\\user when referred to in shell scripts (note the escaping of the '\' character), and user@domain is also accepted.

See also

- [Introduction to Active Directory integration](#)
- [Samba](#)
- [Active Directory integration](#)

Traffic management

Traffic shaping with tc and CAKE

What is bufferbloat?

Prerequisites

- [WAN](#)
-
-

Enable Explicit Congestion Notification (ECN)

```
# Enable explicit congestion notification for TCP (covers IPv4 and IPv6)
net.ipv4.tcp_ecn = 1
```

```
$ sudo sysctl --system
```

Create the IFB interface

```
[NetDev]
Kind=ifb
Name=wanifb
MTUBytes=1492
```

Note:

Adjust MTUBytes to match your WAN connection. For *PPPoE* connections, 1492 is typical. For standard Ethernet, use 1500. You can measure the *MTU* to be sure.

```
[Match]
Name=wanifb

[Link]
ActivationPolicy=always-up
```

```
$ sudo networkctl reload
```

```
$ ip link show wanifb
```

Create the traffic shaping script

```
#!/usr/bin/env python3
"""
Traffic shaping using CAKE qdisc.

Shapes both upload (egress) and download (ingress via IFB redirect) traffic
to reduce bufferbloat and improve latency for interactive applications.

Test results at: https://www.waveform.com/tools/bufferbloat
"""

import subprocess
import argparse
import shlex
```

(continues on next page)


```
# Configuration - adjust these values for your connection
CONFIG = {
    "wan_iface": "ext0",          # Physical WAN interface
    "ifb_iface": "wanifb",       # IFB interface for download shaping
    # set to ~90-95% of what your ISP effectively delivers (not just their promise)
    # please measure your bandwidth, and remember it can have time-of-day dependent
    # variations
    "upload_speed": "30mbit",    # you -> WAN
    "download_speed": "80mbit",  # WAN -> you
}

def run(cmd, ignore_errors=False):
    """Run a shell command."""
    print f" {cmd}"
    subprocess.run(shlex.split(cmd), check=(not ignore_errors))

def clear_all():
    """Remove existing traffic shaping rules."""
    run(f"tc qdisc del dev {CONFIG['wan_iface']} root", ignore_errors=True)
    run(f"tc qdisc del dev {CONFIG['wan_iface']} ingress", ignore_errors=True)
    run(f"tc qdisc del dev {CONFIG['ifb_iface']} root", ignore_errors=True)

def start():
    """Apply traffic shaping rules."""
    print("Clearing existing rules...")
    clear_all()

    # Ensure IFB interface is up
    run(f"ip link set dev {CONFIG['ifb_iface']} up", ignore_errors=True)

    # Shape upload (egress on WAN interface)
    print f"Setting upload limit to {CONFIG['upload_speed']}..."
    run(f"tc qdisc add dev {CONFIG['wan_iface']} root cake "
        f"bandwidth {CONFIG['upload_speed']} nat")

    # Redirect ingress traffic to IFB interface
    print "Redirecting ingress traffic to IFB interface..."
    run(f"tc qdisc add dev {CONFIG['wan_iface']} handle ffff: ingress")
    run(f"tc filter add dev {CONFIG['wan_iface']} parent ffff: protocol all "
        f"u32 match u32 0 0 action mirrored egress redirect dev {CONFIG['ifb_iface']}")

    # Shape download (egress on IFB interface)
    print f"Setting download limit to {CONFIG['download_speed']}..."
    run(f"tc qdisc add dev {CONFIG['ifb_iface']} root cake "
        f"bandwidth {CONFIG['download_speed']} wash")

    print("Traffic shaping enabled.")

def stop():
    """Remove traffic shaping rules."""
    print("Removing traffic shaping rules...")
```

(continues on next page)

(continued from previous page)

```
clear_all()
print("Traffic shaping disabled.")

def status():
    """Show current traffic shaping statistics."""
    print f"\n=== Upload ({CONFIG['wan_iface']}) ==="
    run(f"tc -s qdisc show dev {CONFIG['wan_iface']}")
    print f"\n=== Download ({CONFIG['ifb_iface']}) ==="
    run(f"tc -s qdisc show dev {CONFIG['ifb_iface']}")

def main():
    parser = argparse.ArgumentParser(
        description="Manage traffic shaping with CAKE qdisc"
    )
    parser.add_argument(
        "action",
        choices=["start", "stop", "status"],
        help="Action to perform"
    )
    args = parser.parse_args()

    actions = {"start": start, "stop": stop, "status": status}
    actions[args.action]()

if __name__ == "__main__":
    main()
```

```
$ sudo chmod +x /usr/local/bin/traffic-shaping
```

- `systemd.link(5)`³⁰³
-
-
-

Note:

Setting speeds slightly below your actual bandwidth leaves headroom for the shaper to work effectively. If you set the limit too high, bufferbloat can still occur in upstream devices.

³⁰³ <https://manpages.ubuntu.com/manpages/resolute/man5/systemd.link.5.html>

Create a systemd service

```
[Unit]
Description=Traffic shaping with CAKE
After=network-online.target
Wants=network-online.target
# Bind to the WAN and IFB interfaces
BindsTo=sys-subsystem-net-devices-ext0.device
After=sys-subsystem-net-devices-ext0.device
BindsTo=sys-subsystem-net-devices-wanifb.device
After=sys-subsystem-net-devices-wanifb.device

[Service]
Type=oneshot
ExecStart=/usr/local/bin/traffic-shaping start
ExecStop=/usr/local/bin/traffic-shaping stop
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Note:

Replace `ext0` in the `BindsTo` and `After` lines with your actual WAN interface name - or rename your external interface to `ext0`. The interface name becomes part of the systemd device unit path.

```
$ sudo systemctl daemon-reload > sudo systemctl enable --now
traffic-shaping.service
```

Verify the configuration

```
$ sudo systemctl status traffic-shaping.service
```

```
$ sudo /usr/local/bin/traffic-shaping status
```

Test for bufferbloat

1. Waveform Bufferbloat Test³⁰⁴
- 2.
- 3.

CAKE options explained

- - *NAT*
 - *DSCP*
 - *DSCP*
 -
 -
 -
- tc-cake(8)*³⁰⁵

Troubleshooting

- 1.

```
$ ip link show ext0 > ip link show wanifb
```

- 2.

```
$ sudo journalctl -eu traffic-shaping.service
```

- 3.

```
$ lsmod | grep cake
```

³⁰⁴ <https://www.waveform.com/tools/bufferbloat>

³⁰⁵ <https://manpages.ubuntu.com/manpages/resolute/man8/tc-cake.8.html>

4.

```
$ sudo /usr/local/bin/traffic-shaping stop > sudo  
/usr/local/bin/traffic-shaping start
```

Further reading

- [Bufferbloat project](#)³⁰⁶
- [tc\(8\)](#)³⁰⁷
- [tc-cake\(8\)](#)³⁰⁸

Printing

Install and configure a CUPS print server

Common UNIX Printing System, or CUPS³⁰⁹

Install CUPS

```
$ sudo apt install cups
```

³⁰⁶ <https://www.bufferbloat.net/>

³⁰⁷ <https://manpages.ubuntu.com/manpages/resolute/man8/tc.8.html>

³⁰⁸ <https://manpages.ubuntu.com/manpages/resolute/man8/tc-cake.8.html>

³⁰⁹ <https://openprinting.github.io/cups/doc/overview.html>

Configure the CUPS server

Make a copy of the configuration file

```
$ sudo cp /etc/cups/cupsd.conf /etc/cups/cupsd.conf.original
```

```
$ sudo chmod a-w /etc/cups/cupsd.conf.original
```

Configure Server administrator

```
ServerAdmin bjoy@somebigco.com
```

Configure Listen

hostname

```
Listen 127.0.0.1:631      # existing loopback Listen
Listen /var/run/cups/cups.sock # existing socket Listen
Listen 192.168.10.250:631  # Listen on the LAN interface, Port 631 (IPP)
```

```
Listen socrates:631 # Listen on all interfaces for the hostname 'socrates'
```

```
Port 631 # Listen on port 631 on all interfaces
```

```
$ man cupsd.conf
```

Configure network access

```
# Allow access to the web interface from the local network
<Location />
  Order allow,deny
  Allow localhost
  Allow 192.168.10.*
</Location>

# Allow access to the administration pages from the local network
<Location /admin>
  Order allow,deny
  Allow localhost
  Allow 192.168.10.*
</Location>

# Allow access to the admin configuration from the local network
<Location /admin/conf>
  AuthType Default
  Require user @SYSTEM
  Order allow,deny
  Allow localhost
  Allow 192.168.10.*
</Location>
```

firewall

```
$ sudo ufw allow 631/tcp
```

Post-configuration restart

```
$ sudo systemctl restart cups.service
```

Web Interface

```
$ sudo usermod -aG lpadmin username
```

Error logs

Further reading

- [CUPS Website](http://www.cups.org/)³¹⁰
- [Debian Printing Wiki](https://wiki.debian.org/Printing)³¹¹

³¹⁰ <http://www.cups.org/>

³¹¹ <https://wiki.debian.org/Printing>

See also

- [Networking section](#)

Managing software

-
-
-

Managing software

Install and manage packages

Update the package index

```
$ sudo apt update
```

Note:

Ubuntu releases prior to 24.04 LTS (Noble), which don't use the deb822 standard by default, have repository information defined in the `/etc/apt/sources.list` file.

Install a package

```
$ sudo apt install nmap
```

Tip:

You can install or remove multiple packages at once by separating them with spaces.

Remove a package

```
$ sudo apt remove nmap
```

Note:

While apt is a command-line tool, it is intended to be used interactively, and not to be called from non-interactive scripts. The apt-get command should be used in scripts (perhaps with the -quiet flag). For basic commands the syntax of the two tools is identical.

Upgrading packages

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

guide on upgrading releases
APT User's

Guide³¹²

Aptitude

frontend

³¹² <https://www.debian.org/doc/user-manuals#apt-guide>

```
$ sudo apt install aptitude
```

```
$ sudo aptitude
```

Installing packages

Remove Packages

Updating the package index

Upgrade packages

-
-
-
-
-
-
-
-

Command-line Aptitude

```
$ sudo aptitude install nmap
```

```
$ sudo aptitude remove nmap
```

*Aptitude manual pages*³¹³

³¹³ <https://manpages.ubuntu.com/manpages/resolute/man8/aptitude-curses.8.html>

dpkg

List packages

```
$ dpkg -l
```

```
$ dpkg -l | grep apache2
```

List files

```
$ dpkg -L ufw
```

```
$ dpkg -S /etc/host.conf  
base-files: /etc/host.conf
```

Note:

Many files are automatically generated during the package install process, and even though they are on the *filesystem*, `dpkg -S` may not know which package they belong to.

Installing a deb file

```
$ sudo dpkg -i zip_3.0-4_amd64.deb
```

Uninstalling packages

```
$ sudo dpkg -r zip
```

Caution:

Uninstalling packages using dpkg, is **NOT** recommended in most cases. It is better to use a package manager that handles dependencies to ensure that the system is left in a consistent state. For example, using `dpkg -r zip` will remove the zip package, but any packages that depend on it will still be installed and may no longer function correctly as a result.

*dpkg(1)*³¹⁴

APT configuration

Automatic updates

Automatic updates

Extra repositories

using third-party repositories

³¹⁴ <https://manpages.ubuntu.com/manpages/resolute/man1/dpkg.1.html>

Warning:

Be advised that packages in Universe and Multiverse are not officially supported and do not receive security patches, except through Ubuntu Pro's [Expanded Security Maintenance](#)³¹⁵. A subscription to [Ubuntu Pro](#)³¹⁶ is free for personal use on up to five machines. Packages in the *multiverse* repository often have licensing issues that prevent them from being distributed with a free operating system, and they may be illegal in your locality.

³¹⁵ <https://ubuntu.com/security/esm>

³¹⁶ <https://ubuntu.com/pro>

```
Types: deb
URIs: http://archive.ubuntu.com/ubuntu
Suites: DISTRO-SHORT-CODENAME DISTRO-SHORT-CODENAME-updates DISTRO-SHORT-CODENAME-backports
Components: main universe restricted multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg
```

Logging

Further reading

- [Installing Software](#)³¹⁷
- [APT User's Guide](#)³¹⁸
- [systemd\(1\)](#)³¹⁹ [systemd.timer\(5\)](#)³²⁰ [sys -](#)
- [Aptitude user's manual](#)³²¹

³¹⁷ <https://help.ubuntu.com/community/InstallingSoftware>

³¹⁸ <https://www.debian.org/doc/user-manuals#apt-guide>

³¹⁹ <https://manpages.ubuntu.com/manpages/resolute/man1/systemd.1.html>

³²⁰ <https://manpages.ubuntu.com/manpages/resolute/man5/systemd.timer.5.html>

³²¹ <https://www.debian.org/doc/user-manuals#aptitude>

- [Adding Repositories HOWTO \(Ubuntu Wiki\)](#)³²²
- *Package management*

Updates

Automatic updates

-
-
-

Important:

Just adding another package repository to an Ubuntu system WILL NOT make unattended-upgrades consider it for updates! This is explained in *where to pick updates from* (page 425) later in this document.

Configuration layout

```
$ sudo apt install unattended-upgrades
```

-
-
-

Expanded Security Maintenance (ESM)³²³

³²² <https://help.ubuntu.com/community/Repositories/Ubuntu>

³²³ <https://ubuntu.com/security/esm>

Enabling and disabling unattended upgrades

Upgrading

packages

```
APT::Periodic::Update-Package-Lists "1";  
APT::Periodic::Unattended-Upgrade "1";
```

```
APT::Periodic::Update-Package-Lists "0";  
APT::Periodic::Unattended-Upgrade "0";
```

```
[Timer]  
Persistent=false
```

*systemd.timer(5)*³²⁴

³²⁴ <https://manpages.ubuntu.com/manpages/resolute/man5/systemd.timer.5.html>

Where to pick updates from

Ubuntu Project documenta-

tion³²⁵

```
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
    // Extended Security Maintenance; doesn't necessarily exist for
    // every release and this system may not have it installed, but if
    // available, the policy for updates is such that unattended-upgrades
    // should also install from here by default.
    "${distro_id}ESMA:${distro_codename}-apps-security";
    "${distro_id}ESM:${distro_codename}-infra-security";
    // "${distro_id}:${distro_codename}-updates";
    // "${distro_id}:${distro_codename}-proposed";
    // "${distro_id}:${distro_codename}-backports";
};
```

Note:

The double `“//”` indicates a comment, so whatever follows `“//”` will not be evaluated.

```
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
    // Extended Security Maintenance; doesn't necessarily exist for
    // every release and this system may not have it installed, but if
    // available, the policy for updates is such that unattended-upgrades
    // should also install from here by default.
    "${distro_id}ESMA:${distro_codename}-apps-security";
    "${distro_id}ESM:${distro_codename}-infra-security";
    "${distro_id}:${distro_codename}-updates";
    // "${distro_id}:${distro_codename}-proposed";
    // "${distro_id}:${distro_codename}-backports";
};
```

³²⁵ <https://documentation.ubuntu.com/project/how-ubuntu-is-made/concepts/package-archive/#pockets>

Automatic upgrades from a PPA

Launchpad PPA³²⁶

<https://launchpad.net/>

~canonical-server/+archive/ubuntu/server-backports

```
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
    // Extended Security Maintenance; doesn't necessarily exist for
    // every release and this system may not have it installed, but if
    // available, the policy for updates is such that unattended-upgrades
    // should also install from here by default.
    "${distro_id}ESMAppls:${distro_codename}-apps-security";
    "${distro_id}ESM:${distro_codename}-infra-security";
    "${distro_id}:${distro_codename}-updates";
    // "${distro_id}:${distro_codename}-proposed";
    // "${distro_id}:${distro_codename}-backports";
    "LP-PPA-canonical-server-server-backports:${distro_codename}";
};
```

```
2025-03-13 22:44:29,802 INFO Starting unattended upgrades script
2025-03-13 22:44:29,803 INFO Allowed origins are: o=Ubuntu,a=noble, o=Ubuntu,a=noble-
security, o=UbuntuESMAppls,a=noble-apps-security, o=UbuntuESM,a=noble-infra-security, o=LP-
PPA-canonical-server-server-backports,a=noble
2025-03-13 22:44:29,803 INFO Initial blacklist:
2025-03-13 22:44:29,803 INFO Initial whitelist (not strict):
2025-03-13 22:44:33,029 INFO Option --dry-run given, *not* performing real actions
2025-03-13 22:44:33,029 INFO Packages that will be upgraded: ibverbs-providers libibverbs1
rdma-core
2025-03-13 22:44:33,029 INFO Writing dpkg log to /var/log/unattended-upgrades/unattended-
upgrades-dpkg.log
2025-03-13 22:44:34,421 INFO All upgrades installed
2025-03-13 22:44:34,855 INFO The list of kept packages can't be calculated in dry-run
mode.
```

³²⁶ <https://documentation.ubuntu.com/launchpad/user/Packaging/PPA/>

```
/var/lib/apt/lists/ppa.launchpadcontent.net_canonical-server_server-backports_ubuntu_
dists_noble_InRelease
```

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Origin: LP-PPA-canonical-server-server-backports
Label: Server Team Backports
Suite: noble
Version: 24.04
Codename: noble
Date: Tue, 03 Dec 2024 6:00:43 UTC
Architectures: amd64 arm64 armhf i386 ppc64el riscv64 s390x
Components: main
Description: Ubuntu Noble 24.04
(...)
```

How to block certain packages

Python Regular Expressions³²⁷

Note:

Keep in mind that blocking a package might prevent other updates from being installed if they depend on the blocked package!

```
Unattended-Upgrade::Package-Blacklist {
    "linux-";
}
```

```
Unattended-Upgrade::Package-Blacklist {
    "libc6$";
    "libc-bin$";
}
```

³²⁷ <https://docs.python.org/3/howto/regex.html>

Note:

The regular expressions used here behave as if the “^” character is present at the start, i.e., the `libc6$` expression will match `libc6`, but will NOT match `glibc6` for example.

```
Unattended-Upgrade::Package-Blacklist {  
    "libc(6|-bin)$";  
}
```

Notifications

-

-

-

-

-

Note:

Sending out emails like this requires the separate configuration of a package like `ssmtp(8)`³²⁸ or another minimalistic mail client that is capable of sending messages to a mail server.

³²⁸ <https://manpages.ubuntu.com/manpages/resolute/man8/ssmtp.8.html>

Notification examples

No changes applied, no errors

Subject: unattended-upgrades result for <hostname>: SUCCESS

Unattended upgrade result: No packages found that can be upgraded unattended and no pending auto-removals

Unattended-upgrades log:

Starting unattended upgrades script

Allowed origins are: o=Ubuntu,a=noble, o=Ubuntu,a=noble-security, o=UbuntuESMApps,a=noble-apps-security, o=UbuntuESM,a=noble-infra-security, o=Ubuntu,a=noble, o=Ubuntu,a=noble-security, o=UbuntuESMApps,a=noble-apps-security, o=UbuntuESM,a=noble-infra-security

Initial blacklist:

Initial whitelist (not strict):

No packages found that can be upgraded unattended and no pending auto-removals

Upgrades applied, no errors

Subject: unattended-upgrades result for nuc1: SUCCESS

Unattended upgrade result: All upgrades installed

Packages that were upgraded:

linux-firmware

Package installation log:

Log started: 2025-03-13 06:19:10

Preparing to unpack

.../linux-firmware_20240318.git3b128b60-0ubuntu2.10_amd64.deb ...

Unpacking linux-firmware (20240318.git3b128b60-0ubuntu2.10) over (20240318.git3b128b60-0ubuntu2.9) ...

Setting up linux-firmware (20240318.git3b128b60-0ubuntu2.10) ...

Processing triggers for initramfs-tools (0.142ubuntu25.5) ...

update-initramfs: Generating /boot/initrd.img-6.8.0-55-generic

Running kernel seems to be up-to-date.

The processor microcode seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

Log ended: 2025-03-13 06:19:26

(continues on next page)

(continued from previous page)

```
Unattended-upgrades log:
Starting unattended upgrades script
Allowed origins are: o=Ubuntu,a=noble, o=Ubuntu,a=noble-security,
o=UbuntuESMApps,a=noble-apps-security,
o=UbuntuESM,a=noble-infra-security, o=Ubuntu,a=noble,
o=Ubuntu,a=noble-security, o=UbuntuESMApps,a=noble-apps-security,
o=UbuntuESM,a=noble-infra-security
Initial blacklist:
Initial whitelist (not strict):
Packages that will be upgraded: linux-firmware
Writing dpkg log to /var/log/unattended-upgrades/unattended-upgrades-dpkg.log
All upgrades installed
```

Reboots

-
-
-

*shutdown(8)*³²⁹

Note:

For more information about this time specification for the reboot, and other options like cancelling a scheduled reboot, see the *shutdown(8)*³³⁰ manual page.

³³⁰ <https://manpages.ubuntu.com/manpages/resolute/man8/shutdown.8.html>

```
2025-03-13 20:43:25,923 INFO Starting unattended upgrades script
2025-03-13 20:43:25,924 INFO Allowed origins are: o=Ubuntu,a=noble, o=Ubuntu,a=noble-
```

(continues on next page)

³²⁹ <https://manpages.ubuntu.com/manpages/resolute/man8/shutdown.8.html>

(continued from previous page)

```
security, o=UbuntuESMApps,a=noble-apps-security, o=UbuntuESM,a=noble-infra-security
2025-03-13 20:43:25,924 INFO Initial blacklist:
2025-03-13 20:43:25,924 INFO Initial whitelist (not strict):
2025-03-13 20:43:29,082 INFO Packages that will be upgraded: libc6 python3-jinja2
2025-03-13 20:43:29,082 INFO Writing dpkg log to /var/log/unattended-upgrades/unattended-
upgrades-dpkg.log
2025-03-13 20:43:39,532 INFO All upgrades installed
2025-03-13 20:43:40,201 WARNING Found /var/run/reboot-required, rebooting
2025-03-13 20:43:40,207 WARNING Shutdown msg: b"Reboot scheduled for Thu 2025-03-13
20:45:00 UTC, use 'shutdown -c' to cancel."
```

When to consider disabling automatic updates

Systems which just get recreated

Manual steps required

Too much of a risk

Postponable updates

Fleet management

-
-
-

Landscape³³¹

Postponable updates

³³¹ <https://ubuntu.com/landscape>

```
Unattended-Upgrade::Postpone-For-Days "3";
```

Prompt duration

Who can postpone

[polkit documentation](#)³³²

Notifications in different environments

Testing and troubleshooting

*manual page*³³³

-
-

³³² <https://www.freedesktop.org/software/polkit/docs/latest/polkit.8.html>

³³³ <https://manpages.ubuntu.com/manpages/resolute/man8/unattended-upgrade.8.html>

```
$ sudo unattended-upgrade -v --dry-run
```

```
Starting unattended upgrades script
Allowed origins are: o=Ubuntu,a=noble, o=Ubuntu,a=noble-security, o=UbuntuESMApps,
a=noble-apps-security, o=UbuntuESM,a=noble-infra-security, o=LP-PPA-canonical-
server-server-backports,a=noble
Initial blacklist:
Initial whitelist (not strict):
Option --dry-run given, *not* performing real actions
Packages that will be upgraded: rdma-core
Writing dpkg log to /var/log/unattended-upgrades/unattended-upgrades-dpkg.log
/usr/bin/unattended-upgrade:567: DeprecationWarning: This process (pid=1213) is
multi-threaded, use of fork() may lead to deadlocks in the child.
  pid = os.fork()
/usr/bin/dpkg --status-fd 10 --no-triggers --unpack --auto-deconfigure /var/cache/
apt/archives/rdma-core_52.0-2ubuntu1~backport24.04.202410192216~ubuntu24.04.1_amd64.
deb
/usr/bin/dpkg --status-fd 10 --configure --pending
All upgrades installed
The list of kept packages can't be calculated in dry-run mode.
```

-
-

```
$ apt-cache policy rdma-core
```

```
rdma-core:
Installed: 50.0-2build2
Candidate: 52.0-2ubuntu1~backport24.04.202410192216~ubuntu24.04.1
Version table:
   52.0-2ubuntu1~backport24.04.202410192216~ubuntu24.04.1 500
      500 https://ppa.launchpadcontent.net/canonical-server/server-backports/
*** 50.0-2build2 500
      500 http://br.archive.ubuntu.com/ubuntu noble/main amd64 Packages
      100 /var/lib/dpkg/status
```

How to upgrade your Ubuntu release

Ubuntu Pro subscription³³⁴

release lifecycle and support period³³⁵

Understanding Upgrade Paths

Pre-upgrade checklist

-
-

Ubuntu Wiki Releases Page³³⁶

explanation on phased updates

1.

```
$ sudo apt update
```

```
$ sudo apt dist-upgrade -o
```

```
$ APT::Get::Always-Include-Phased-Updates=true
```

³³⁴ <https://ubuntu.com/pro>

³³⁵ <https://ubuntu.com/about/release-cycle>

³³⁶ <https://wiki.ubuntu.com/Releases>

2.

3.

-

-

-

-

Upgrade the system

```
$ sudo do-release-upgrade
```

Note:

Upgrading to a development release of Ubuntu is available using the `-d` flag. However, using the development release (or the `-d` flag) is **not recommended** for production environments.

Pre-upgrade summary

Do you want to start the upgrade?

5 installed packages are no longer supported by Canonical. You can still get support from the community.

4 packages are going to be removed. 117 new packages are going to be installed. 424 packages are going to be upgraded.

You have to download a total of 262 M. This download will take about 33 minutes with a 1Mbit DSL connection and about 10 hours with a 56k modem.

Fetching and installing the upgrade can take several hours. Once the download has finished, the process cannot be canceled.

Continue [yN] Details [d]

Configuration changes

```
Configuration file '/etc/ssh/ssh_config'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ? Your options are:
  Y or I : install the package maintainer's version
  N or O : keep your currently-installed version
  D      : show the differences between the versions
  Z      : start a shell to examine the situation
The default action is to keep your current version.
*** ssh_config (Y/I/N/O/D/Z) [default=N] ?
```

Removing Obsolete Packages

```
Remove obsolete packages?  
  
30 packages are going to be removed.  
  
Continue [yN] Details [d]
```

Reboot

```
System upgrade is complete.  
  
Restart required  
  
To finish the upgrade, a restart is required.  
If you select 'y' the system will be restarted.  
  
Continue [yN]
```

Further reading

- [List of releases](#)³³⁷

How to use the Ubuntu snapshot service

-
-
-

³³⁷ <https://documentation.ubuntu.com/project/release-team/list-of-releases/>

Prerequisites

```
deb http://archive.ubuntu.com/ubuntu jammy main universe restricted multiverse
deb http://archive.ubuntu.com/ubuntu jammy-updates main universe restricted multiverse
deb http://archive.ubuntu.com/ubuntu jammy-backports main restricted universe multiverse
deb http://security.ubuntu.com/ubuntu jammy-security main restricted universe multiverse
```

all components in all pockets³³⁸

```
deb [snapshot=yes] http://archive.ubuntu.com/ubuntu jammy main universe restricted
multiverse
deb [snapshot=yes] http://archive.ubuntu.com/ubuntu jammy-updates main universe restricted
multiverse
deb [snapshot=yes] http://archive.ubuntu.com/ubuntu jammy-backports main restricted
universe multiverse
deb [snapshot=yes] http://security.ubuntu.com/ubuntu jammy-security main restricted
universe multiverse
```

Ubuntu Release	Supports Snapshot Service?	Need to configure sources?
Next Ubuntu releases	Yes	No
Ubuntu 24.04 LTS	Yes	No
Ubuntu 22.04 LTS	Yes (with apt >= 2.4.11)	Yes
Ubuntu 20.04 LTS	Yes (with apt >= 2.0.10)	Yes

³³⁸ <https://documentation.ubuntu.com/project/how-ubuntu-is-made/concepts/package-archive/#pockets>

Quick start

```
$ sudo apt install hello --update --snapshot 20240301T030400Z
```

Using the snapshot service

The --snapshot CLI option

```
$ sudo apt update --snapshot $SNAPSHOT_ID
```

```
$ sudo apt install $PACKAGE_NAME --snapshot $SNAPSHOT_ID
```

```
$ sudo apt install $PACKAGE_NAME --update --snapshot $SNAPSHOT_ID
```

```
$ sudo apt install $PACKAGE_NAME --update -S $SNAPSHOT_ID
```

```
$ sudo apt update --snapshot 20240501T120000Z
```

```
$ apt policy docker.io --snapshot 20240501T120000Z

docker.io:
  Installed: (none)
  Candidate: 24.0.7-0ubuntu4
  Version table:
    24.0.7-0ubuntu4 500
    500 https://snapshot.ubuntu.com/ubuntu/20240501T120000Z noble/universe amd64
Packages
```

```
$ apt policy docker.io

docker.io:
  Installed: (none)
  Candidate: 28.2.2-0ubuntu1~24.04.1
  Version table:
    28.2.2-0ubuntu1~24.04.1 500
    500 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages
    27.5.1-0ubuntu3~24.04.2 500
    500 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages
    24.0.7-0ubuntu4 500
    500 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages
```

```
$ sudo apt install docker.io --snapshot 20240501T120000Z

...
```

```
$ docker --version

Docker version 24.0.7, build 24.0.7-0ubuntu4
```

Docker package publishing history in Launchpad³³⁹

```
$ sudo apt install docker.io --update --snapshot 20241126T230000Z

...
```

³³⁹ <https://launchpad.net/ubuntu/+source/docker.io-app/+publishinghistory>

```
$ docker --version
```

```
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
```

Note:

Note that, at the time of writing, the version of docker.io³⁴⁰ in the noble-security pocket is 27.5.1-0ubuntu3~24.04.2, i.e., the version used in the example above may be affected by known vulnerabilities. When using the snapshot service, do make sure to check the latest version available in your Ubuntu release's -security pocket to make sure you are not installing a vulnerable package.

³⁴⁰ <http://docker.io>

Configuring the snapshot service for specific repositories

```
Snapshot: $SNAPSHOT_ID
```

```
deb [snapshot=$SNAPSHOT_ID] http://archive.ubuntu.com/ubuntu ...
```

```
Types: deb
URIs: http://archive.ubuntu.com/ubuntu
Suites: noble noble-updates noble-backports
Components: main universe restricted multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg
Snapshot: 20250530T223000Z
```

```
Types: deb
URIs: http://security.ubuntu.com/ubuntu
Suites: noble-security
Components: main universe restricted multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg
Snapshot: 20250530T223000Z
```

```
$ sudo apt install --update docker.io
```

```
$ docker --version
```

```
Docker version 27.5.1, build 27.5.1-0ubuntu3~24.04.1
```

```
$ sudo apt install --update docker.io --snapshot 20251013T2300Z
```

```
...
```

```
Hit:5 https://snapshot.ubuntu.com/ubuntu/20250530T223000Z noble InRelease
```

```
Hit:6 https://snapshot.ubuntu.com/ubuntu/20250530T223000Z noble-updates InRelease
```

```
Hit:7 https://snapshot.ubuntu.com/ubuntu/20250530T223000Z noble-backports InRelease
```

```
Hit:8 https://snapshot.ubuntu.com/ubuntu/20250530T223000Z noble-security InRelease
```

```
...
```

```
docker.io is already the newest version (27.5.1-0ubuntu3~24.04.1).
```

```
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

Configuring the snapshot service globally

Warning:

If you are following these examples until here, make sure to revert the changes to the sources configuration file made in the previous section.

```
$ echo 'APT::Snapshot "20250801T111111Z";' | sudo tee  
/etc/apt/apt.conf.d/50snapshot
```

```
$ sudo apt install docker.io --update
```

```
...
```

```
$ docker --version
```

```
Docker version 27.5.1, build 27.5.1-0ubuntu3~24.04.2
```

```
$ sudo apt install docker.io --update --snapshot 20251013T120000Z
```

```
$ docker --version
```

```
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
```

```
$ sudo rm /etc/apt/apt.conf.d/50snapshot
```

Further reading

- [The Ubuntu snapshot service official page and documentation](#)³⁴¹
- [Securing multiple Ubuntu instances while maximising uptime](#)³⁴²
- [Integrating the Ubuntu Snapshot Service into systems management and update tools](#)³⁴³
- [Automatic updates](#)
- [Upgrade your release](#)
- [Snapshot service](#)

Troubleshooting

- [Reporting bugs](#)

How to report a bug in Ubuntu Server

uses [Launchpad](#)³⁴⁴

³⁴¹ <https://snapshot.ubuntu.com/>

³⁴² <https://ubuntu.com/blog/securing-multiple-ubuntu-instances-while-maximising-uptime>

³⁴³ <https://ubuntu.com/blog/integrating-the-ubuntu-snapshot-service-into-systems-management-and-update-tools>

³⁴⁴ <https://launchpad.net/>

Report bugs with apport-cli

Note:

The commands `apport-cli` and `ubuntu-bug` should give the same results on a command-line interface (CLI) server. The latter is actually a symlink to `apport-bug`, which is intelligent enough to know whether a desktop environment is in use, and will choose `apport-cli` if not. Since server systems tend to be CLI-only, `apport-cli` was chosen from the outset in this guide.

```
$ apport-cli PACKAGENAME
```

```
*** Collecting problem information
```

```
The collected information can be sent to the developers to improve the
application. This might take a few minutes.
```

```
...
```

```
*** Send problem report to the developers?
```

```
After the problem report has been sent, please fill out the form in the
automatically opened web browser.
```

```
What would you like to do? Your options are:
```

```
S: Send report (2.8 KB)
```

```
V: View report
```

```
K: Keep report file for sending later or copying to somewhere else
```

```
I: Cancel and ignore future crashes of this program version
```

```
C: Cancel
```

```
Please choose (S/V/K/I/C):
```

S: Send report

```
*** Uploading problem information

The collected information is being sent to the bug tracking system.
This might take a few minutes.
94%

*** To continue, you must visit the following URL:

https://bugs.launchpad.net/ubuntu/+source/vim/+filebug/09b2495a-e2ab-11e3-879b-
68b5996a96c8?

You can launch a browser now, or copy this URL into a browser on another computer.

Choices:
  1: Launch a browser now
  C: Cancel
Please choose (1/C): 1
```

*Debian alternatives system*³⁴⁵

V: View

K: Keep

```
What would you like to do? Your options are:
S: Send report (2.8 KB)
V: View report
K: Keep report file for sending later or copying to somewhere else
I: Cancel and ignore future crashes of this program version
C: Cancel
Please choose (S/V/K/I/C): k
Problem report file: /tmp/apport.vim.1pg92p02.apport
```

³⁴⁵ <https://manpages.ubuntu.com/manpages/resolute/man1/update-alternatives.1.html>

```
$ apport-cli apport.vim.1pg92p02.apport
```

```
$ apport-cli vim --save apport.vim.test.apport
```

Note:

If this Internet-enabled system is non-Ubuntu/Debian, `apport-cli` is not available so the bug will need to be created manually. An apport report is also not to be included as an attachment to a bug either so it is completely useless in this scenario.

Reporting application crashes

```
-rw-r----- 1 peter   whoopsie 150K Jul 24 16:17 _usr_lib_x86_64-linux-gnu_libmenu-  
cache2_libexec_menu-cached.1000.crash
```

```
$ apport-cli
```

```
*** Send problem report to the developers?
```

After the problem report has been sent, please fill out the form in the automatically opened web browser.

What would you like to do? Your options are:

S: Send report (153.0 KB)

V: View report

K: Keep report file for sending later or copying to somewhere else

I: Cancel and ignore future crashes of this program version

C: Cancel

Please choose (S/V/K/I/C): s


```
-rw-r----- 1 peter    whoopsie 150K Jul 24 16:17 _usr_lib_x86_64-linux-gnu_libmenu-  
cache2_libexec_menu-cached.1000.crash  
-rw-rw-r-- 1 peter    whoopsie   0 Jul 24 16:37 _usr_lib_x86_64-linux-gnu_libmenu-  
cache2_libexec_menu-cached.1000.upload  
-rw----- 1 whoopsie whoopsie   0 Jul 24 16:37 _usr_lib_x86_64-linux-gnu_libmenu-  
cache2_libexec_menu-cached.1000.uploaded
```

Further reading

- [Reporting Bugs](#)³⁴⁶
- [the Appport page](#)³⁴⁷
GUI

See also

- [Managing software](#)

The data and storage section

-
-

Data and storage

Data management

- [OpenLDAP](#)
- [Databases](#)

³⁴⁶ <https://help.ubuntu.com/community/ReportingBugs>

³⁴⁷ <https://wiki.ubuntu.com/Appport>

OpenLDAP

Install and configure LDAP

Lightweight Directory Access Protocol³⁴⁸

Introduction to OpenLDAP

slapd (the Stand-alone LDAP Daemon)³⁴⁹

Note:

This guide uses a database suffix of **dc=example,dc=com**. Change this to match your particular setup.

Install slapd

```
$ sudo apt install slapd ldap-utils
```

³⁴⁸ <https://ldap.com/>

³⁴⁹ <https://www.openldap.org/software/man.cgi?query=slapd>

Change the instance suffix (optional)

```
$ sudo dpkg-reconfigure slapd
```

- 1.
- 2.
- 3.
- 4.

```
BASE dc=example,dc=com  
URI ldap://ldap01.example.com
```

Default tree contents

Note:

Although the configuration is accessible via the LDAP protocol under the `cn=config` suffix, it is also stored on disk under `/etc/ldap/slapd.d/`. This allows the configuration to be replicated to other servers and backed up alongside the data.

•

-

DN

SASL

Default configuration tree

```
$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config dn
```

-

-

-

-

-

-

```
dn: cn=config
dn: cn=module{0},cn=config
dn: cn=schema,cn=config
dn: cn={0}core,cn=schema,cn=config
dn: cn={1}cosine,cn=schema,cn=config
dn: cn={2}nis,cn=schema,cn=config
dn: cn={3}inetorgperson,cn=schema,cn=config
dn: olcDatabase={-1}frontend,cn=config
dn: olcDatabase={0}config,cn=config
dn: olcDatabase={1}mdb,cn=config
```

-

-

-

-

-

-

-
-
-
-

Frontend

```
$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config
```

```
dn: cn=config
objectClass: olcGlobal
cn: config
olcArgsFile: /var/run/slapd/slapd.args
olcLogLevel: none
...
```

Default “data” tree

```
$ ldapsearch -x -LLL -H ldap:/// -b dc=example,dc=com dn
```

-
-

bindDN

```
dn: dc=example,dc=com
```

Who am I?

```
$ ldapwhoami -x
```

```
anonymous
```

```
$ ldapwhoami -x -D cn=admin,dc=example,dc=com -W
```

```
Enter LDAP Password:  
dn:cn=admin,dc=example,dc=com
```

Note:

A simple bind without some sort of transport security mechanism is **clear text**, meaning the credentials are transmitted in the clear. You should [add Transport Layer Security \(TLS\) support](#) (page 476) to your OpenLDAP server as soon as possible.

```
$ ldapwhoami -Y EXTERNAL -H ldapi:/// -Q
```

```
dn:gidNumber=1000+uidNumber=1000,cn=peercred,cn=external,cn=auth
```

```
$ sudo ldapwhoami -Y EXTERNAL -H ldapi:/// -Q
```

```
dn:gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
```

uid *gid*

Set up LDAP access control

Note:

OpenLDAP ACLs are explained in *Set up access control* (page 461)

Populate the directory

-

-

-

-

```
dn: ou=People,dc=example,dc=com
objectClass: organizationalUnit
ou: People

dn: ou=Groups,dc=example,dc=com
objectClass: organizationalUnit
ou: Groups

dn: cn=miners,ou=Groups,dc=example,dc=com
objectClass: posixGroup
cn: miners
gidNumber: 5000
memberUid: john

dn: uid=john,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: john
sn: Doe
givenName: John
cn: John Doe
displayName: John Doe
uidNumber: 10000
gidNumber: 5000
userPassword: {CRYPT}x
gecos: John Doe
loginShell: /bin/bash
homeDirectory: /home/john
```

Note:

It's important that `uid` and `gid` values in your directory do not collide with local Linux users' values as defined in `/etc/passwd` and `/etc/group`. You can use higher number ranges, such as starting at 5000 or even higher.

```
$ ldapadd -x -D cn=admin,dc=example,dc=com -W -f add_content.ldif
```

```
Enter LDAP Password: *****
adding new entry "ou=People,dc=example,dc=com"

adding new entry "ou=Groups,dc=example,dc=com"

adding new entry "cn=miners,ou=Groups,dc=example,dc=com"

adding new entry "uid=john,ou=People,dc=example,dc=com"
```

```
$ ldapsearch -x -LLL -b dc=example,dc=com '(uid=john)' cn gidNumber

dn: uid=john,ou=People,dc=example,dc=com
cn: John Doe
gidNumber: 5000
```

```
$ ldapsearch -x -LLL -b dc=example,dc=com
'(&(objectClass=posixGroup)(memberUid=john))' cn gidNumber

dn: cn=miners,ou=Groups,dc=example,dc=com
cn: miners
gidNumber: 5000
```



```
$ ldappasswd -x -D cn=admin,dc=example,dc=com -W -S  
uid=john,ou=people,dc=example,dc=com
```

```
New password:  
Re-enter new password:  
Enter LDAP Password:
```

```
$ ldapwhoami -x -D uid=john,ou=people,dc=example,dc=com -W
```

```
Enter LDAP Password:  
dn:uid=john,ou=People,dc=example,dc=com
```

Note:

Remember that simple binds are insecure and you should *add TLS support* (page 476) to your server as soon as possible!

³⁵⁰ <https://www.rfc-editor.org/rfc/rfc4515.txt>

Change the configuration

Which “admin” DN to use?

Suffix	DN for making changes	Authentication mechanism
cn=config	cn=admin,cn=config	Absent
cn=config	gidNumber=0+uidNumber=0, cn=peercred,cn=external, cn=auth	SASL EXTERNAL as root via ldapi://
dc=example, dc=com	cn=admin,dc=example,dc=com	Simple bind with passwordset dur- ing install or reconfigure

Change the “admin” password

```
$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config  
'(olcSuffix=dc=example,dc=com)' olcSuffix olcRootDN olcRootPW
```

```
dn: olcDatabase={1}mdb,cn=config  
olcSuffix: dc=example,dc=com  
olcRootDN: cn=admin,dc=example,dc=com  
olcRootPW: {SSHA}Y0UjBUUmF08TC2SePVc6waI/mfvPNkTk
```

Note:

Even though this `olcRootDN` is the administrative DN for the `dc=example,dc=com` suffix, it is stored under the `cn=config` suffix!

```
New password:  
Re-enter new password:  
{SSHA}VKrYMxLSKhONGRpC6rnASKNmXG2xHXFo
```

```
dn: olcDatabase={1}mdb,cn=config  
changetype: modify  
replace: olcRootPW  
olcRootPW: {SSHA}VKrYMxLSKhONGRpC6rnASKNmXG2xHXFo
```

```
$ ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f changerootpw.ldif
```

```
modifying entry "olcDatabase={1}mdb,cn=config"
```

Add an index

```
dn: olcDatabase={1}mdb,cn=config
add: olcDbIndex
olcDbIndex: mail eq,sub
```

```
$ sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f add_index.ldif
modifying entry "olcDatabase={1}mdb,cn=config"
```

```
$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config
'(olcDatabase={1}mdb)' olcDbIndex

dn: olcDatabase={1}mdb,cn=config
olcDbIndex: objectClass eq
olcDbIndex: cn,uid eq
olcDbIndex: uidNumber,gidNumber eq
olcDbIndex: member,memberUid eq
olcDbIndex: mail eq,sub
```

See also

To learn more about OpenLDAP indexes, check the upstream documentation at <https://www.openldap.org/doc/admin26/tuning.html#Indexes>

Add a schema

Note:

It is not trivial to remove a schema from the slapd-config database. Practice adding schemas on a test system.

```
$ sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/corba.ldif
adding new entry "cn=corba,cn=schema,cn=config"
```

Logging

*slapd-config(5)*³⁵¹

Example logging with the stats level

```
dn: cn=config
changetype: modify
replace: olcLogLevel
olcLogLevel: stats
```

```
$ sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f logging.ldif
```

```
$ sudo journalctl -eu slapd.service
```

Next steps

about access control

set up users and groups

³⁵¹ <https://manpages.ubuntu.com/manpages/resolute/man5/slapd-config.5.html>

Set up LDAP access control

ACL

installed the slapd package

frontend

Getting the ACLs

```
$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config  
'(olcDatabase={1}mdb)' olcAccess  
  
dn: olcDatabase={1}mdb,cn=config  
olcAccess: {0}to attrs=userPassword by self write by anonymous auth by * none  
olcAccess: {1}to attrs=shadowLastChange by self write by * read  
olcAccess: {2}to * by * read
```

```
$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config  
'(olcDatabase={-1}frontend)' olcAccess  
  
dn: olcDatabase={-1}frontend,cn=config  
olcAccess: {0}to * by dn.exact=gidNumber=0+uidNumber=0,cn=peercred,cn=external  
,cn=auth manage by * break  
olcAccess: {1}to dn.exact="" by * read  
olcAccess: {2}to dn.base="cn=Subschema" by * read
```

Note:

The Root *DN* always has full rights to its database and does not need to be included in any ACL.

Interpreting the results

```
olcAccess: {0}to attrs=userPassword by self write by anonymous auth by * none
olcAccess: {1}to attrs=shadowLastChange by self write by * read
```

```
to attrs=userPassword
  by self write
  by anonymous auth
  by * none

to attrs=shadowLastChange
  by self write
  by * read
```

-

DIT

-

-

-

```
to *
  by * read
```

SASL identity

```
dn.exact=gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
```

```
$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config  
'(olcDatabase={0}config)' olcAccess  
  
dn: olcDatabase={0}config,cn=config  
olcAccess: {0}to * by dn.exact=gidNumber=0+uidNumber=0,cn=peercred,  
cn=external,cn=auth manage by * break
```

-
-

Grant root passwordless access to the data tree

```
# Allow root user (via peercred) to manage the primary database (mdb)  
dn: olcDatabase={1}mdb,cn=config  
changetype: modify  
add: olcAccess  
# Inserted as the first ACL rule  
olcAccess: {0}to * by dn.exact="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth"  
manage by * break
```

```
$ sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f peercred.ldif
```

```
$ sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f add_content.ldif
```



```
$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config  
'(olcAccess=*)' olcAccess olcSuffix
```

Next steps

set up LDAP users and groups

Further reading

- [*slapd.access\(5\)*](#)³⁵²
- [access control topic](#)³⁵³

OpenLDAP replication

ability *high avail-*

[4533](#)³⁵⁵ [OpenLDAP administrator's guide](#)³⁵⁴ [RFC](#)

-
-

Important:

You **must** have Transport Layer Security (TLS) enabled already before proceeding with this guide. Please consult the [LDAP with TLS guide](#) (page 476) for details of how to set this up.

³⁵² <https://manpages.ubuntu.com/manpages/resolute/man5/slapd.access.5.html>

³⁵³ <https://openldap.org/doc/admin26/guide.html#Access%20Control>

³⁵⁴ <https://openldap.org/doc/admin26/guide.html#LDAP%20Sync%20Replication>

³⁵⁵ <https://www.rfc-editor.org/rfc/rfc4533.txt>

Provider configuration - replication user

ACL

```
dn: cn=replicator,dc=example,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: replicator
description: Replication user
userPassword: {CRYPT}x
```

passwordless access for root

```
$ sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f replicator.ldif
```

```
$ ldapadd -x -ZZ -H ldap://ldap01.example.com -D cn=admin,dc=example,dc=com
-W -f replicator.ldif
```

```
$ ldappasswd -x -ZZ -H ldap://ldap01.example.com -D
cn=admin,dc=example,dc=com -W -S cn=replicator,dc=example,dc=com
```

Note:

Adjust the server URI in the -H parameter if needed to match your deployment.

-
-

```
$ sudo ldapsearch -Q -Y EXTERNAL -H ldapi:/// -LLL -b cn=config
'(olcSuffix=dc=example,dc=com)' olcAccess

dn: olcDatabase={1}mdb,cn=config
olcAccess: {0}to attrs=userPassword by self write by anonymous auth by * none
olcAccess: {1}to attrs=shadowLastChange by self write by * read
olcAccess: {2}to * by * read
```

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcAccess
olcAccess: {0}to *
    by dn.exact="cn=replicator,dc=example,dc=com" read
    by * break
-
add: olcLimits
olcLimits: dn.exact="cn=replicator,dc=example,dc=com"
    time.soft=unlimited time.hard=unlimited
    size.soft=unlimited size.hard=unlimited
```

```
$ sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f replicator-acl-limits.ldif
```

Provider configuration - standard replication

```
# Add indexes to the frontend db.
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryCSN eq
-
add: olcDbIndex
olcDbIndex: entryUUID eq

#Load the syncprov module.
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov

# syncrepl Provider for primary db
dn: olcOverlay=syncprov,olcDatabase={1}mdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpCheckpoint: 100 10
olcSpSessionLog: 100
```

Warning:

The LDIF above has some parameters that you should review before deploying in production on your directory. In particular – `olcSpCheckpoint` and `olcSpSessionLog`. Please see the [slapo-syncprov\(5\)](https://manpages.ubuntu.com/manpages/resolute/man5/slapo-syncprov.5.html)³⁵⁶ manual page. In general, `olcSpSessionLog` should be equal to (or preferably larger than) the number of entries in your directory. Also see [ITS #8125](https://bugs.openldap.org/show_bug.cgi?id=8125)³⁵⁷ for details on an existing bug.

³⁵⁶ <https://manpages.ubuntu.com/manpages/resolute/man5/slapo-syncprov.5.html>

³⁵⁷ https://bugs.openldap.org/show_bug.cgi?id=8125

```
$ sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f provider_simple_sync.ldif
```

Consumer configuration - standard replication

*the installation steps
enable TLS*

```
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov

dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryUUID eq
-
add: olcSyncrepl
olcSyncrepl: rid=0
provider=ldap://ldap01.example.com
bindmethod=simple
binddn="cn=replicator,dc=example,dc=com" credentials=<secret>
searchbase="dc=example,dc=com"
schemachecking=on
type=refreshAndPersist retry="60 +"
starttls=critical tls_reqcert=demand
-
add: olcUpdateRef
olcUpdateRef: ldap://ldap01.example.com
```

- *hostname*
- **binddn** *DN*
-

-
-
-

Note:

A successful encrypted connection via START_TLS is being enforced in this configuration, to avoid sending the credentials in the clear across the network. See [LDAP with TLS](#) (page 476) for details on how to set up OpenLDAP with trusted SSL certificates.

```
$ sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f consumer_simple_sync.ldif
```

Provider configuration - delta replication

-
-
-

Add syncprov and accesslog overlays and DBs

```
# Add indexes to the frontend db.
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryCSN eq
-
add: olcDbIndex
olcDbIndex: entryUUID eq

#Load the syncprov and accesslog modules.
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov
-
add: olcModuleLoad
olcModuleLoad: accesslog

# Accesslog database definitions
```

(continues on next page)

(continued from previous page)

```
dn: olcDatabase={2}mdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcMdbConfig
olcDatabase: {2}mdb
olcDbDirectory: /var/lib/ldap/accesslog
olcSuffix: cn=accesslog
olcRootDN: cn=admin,dc=example,dc=com
olcDbIndex: default eq
olcDbIndex: entryCSN,objectClass,reqEnd,reqResult,reqStart
olcAccess: {0}to * by dn.exact="cn=replicator,dc=example,dc=com" read by * break
olcLimits: dn.exact="cn=replicator,dc=example,dc=com"
    time.soft=unlimited time.hard=unlimited
    size.soft=unlimited size.hard=unlimited

# Accesslog db syncprov.
dn: olcOverlay=syncprov,olcDatabase={2}mdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpNoPresent: TRUE
olcSpReloadHint: TRUE

# syncrepl Provider for primary db
dn: olcOverlay=syncprov,olcDatabase={1}mdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpCheckpoint: 100 10
olcSpSessionLog: 100

# accesslog overlay definitions for primary db
dn: olcOverlay=accesslog,olcDatabase={1}mdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcAccessLogConfig
olcOverlay: accesslog
olcAccessLogDB: cn=accesslog
olcAccessLogOps: writes
olcAccessLogSuccess: TRUE
# scan the accesslog DB every day, and purge entries older than 7 days
olcAccessLogPurge: 07+00:00 01+00:00
```

Warning:

The LDIF above has some parameters that you should review before deploying in production on your directory. In particular – `olcSpCheckpoint`, `olcSpSessionLog`. Please see the [slapo-syncprov\(5\)](https://manpages.ubuntu.com/manpages/resolute/man5/slapo-syncprov.5.html)³⁵⁸ manual page. In general, `olcSpSessionLog` should be equal to (or preferably larger than) the number of entries in your directory. Also see [ITS #8125](https://bugs.openldap.org/show_bug.cgi?id=8125)³⁵⁹ for details on an existing bug. For `olcAccessLogPurge`, please check the [slapo-accesslog\(5\)](https://manpages.ubuntu.com/manpages/resolute/man5/slapo-accesslog.5.html)³⁶⁰ manual page.

³⁵⁸ <https://manpages.ubuntu.com/manpages/resolute/man5/slapo-syncprov.5.html>

³⁵⁹ https://bugs.openldap.org/show_bug.cgi?id=8125

³⁶⁰ <https://manpages.ubuntu.com/manpages/resolute/man5/slapo-accesslog.5.html>

```
$ sudo -u openldap mkdir /var/lib/ldap/accesslog
```

```
$ sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f provider_sync.ldif
```

Consumer configuration

*the installation steps
enable TLS*

```
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov

dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryUUID eq
-
add: olcSyncrepl
olcSyncrepl: rid=0
provider=ldap://ldap01.example.com
bindmethod=simple
binddn="cn=replicator,dc=example,dc=com" credentials=<secret>
searchbase="dc=example,dc=com"
logbase="cn=accesslog"
logfilter="(&(objectClass=auditWriteObject)(reqResult=0))"
schemachecking=on
type=refreshAndPersist retry="60 +"
syncdata=accesslog
starttls=critical tls_reqcert=demand
-
```

(continues on next page)

(continued from previous page)

```
add: olcUpdateRef  
olcUpdateRef: ldap://ldap01.example.com
```

-
-
-
-
-
-

Note:

Note that a successful encrypted connection via START_TLS is being enforced in this configuration, to avoid sending the credentials in the clear across the network. See [LDAP with TLS](#) (page 476) for details on how to set up OpenLDAP with trusted SSL certificates.

```
$ sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f consumer_sync.ldif
```

Testing

```
$ ldapsearch -z1 -LLL -x -s base -b dc=example,dc=com contextCSN  
dn: dc=example,dc=com  
contextCSN: 20200423222317.722667Z#000000#000#000000
```



```
$ ldapwhoami -x -ZZ -H ldap://ldap01.example.com -D  
cn=replicator,dc=example,dc=com -W
```

```
$ ldapsearch -x -LLL -H ldap://ldap02.example.com -b dc=example,dc=com  
'(uid=john)' uid  
  
dn: uid=john,ou=People,dc=example,dc=com  
uid: john
```

Further reading

- Replication types, OpenLDAP Administrator's Guide³⁶¹
- LDAP Sync Replication - OpenLDAP Administrator's Guide³⁶²
- RFC 4533³⁶³

How to set up LDAP users and groups

have a working LDAP server

nsld

SSSD

User and group management - ldapscripts

Install ldapscripts

```
$ sudo apt install ldapscripts
```

³⁶¹ <https://openldap.org/doc/admin26/guide.html#Configuring%20the%20different%20replication%20types>

³⁶² <https://openldap.org/doc/admin26/guide.html#LDAP%20Sync%20Replication>

³⁶³ <https://www.rfc-editor.org/rfc/rfc4533.txt>

```
SERVER=ldap://ldap01.example.com
LDAPBINOPTS="-ZZ"
BINDDN='cn=admin,dc=example,dc=com'
BINDPWDFILE="/etc/ldapscripts/ldapscripts.passwd"
SUFFIX='dc=example,dc=com'
GSUFFIX='ou=Groups'
USUFFIX='ou=People'
MSUFFIX='ou=Computers'
```

Note:

Adjust **SERVER** and related **SUFFIX** options to suit your directory structure. Here, we are forcing use of **START_TLS** (-ZZ parameter). Refer to [LDAP with TLS](#) (page 476) to learn how to set up the server with TLS support.

```
$ echo -n 'password' | sudo tee /etc/ldapscripts/ldapscripts.passwd
```

```
$ sudo chmod 400 /etc/ldapscripts/ldapscripts.passwd
```

Note:

The password file must contain exactly and only the password characters, no end-of-line or anything else. The echo command above with the -n parameter achieves that by suppressing the *EOL* character \n. And in order to prevent the password from appearing in the shell history, the *echo* command line is prefixed by a space.

Manage users and groups with ldapscripts

Create a new user

```
$ sudo ldapaddgroup george
```

```
$ sudo ldapadduser george george
```

GID

Change a user's password

```
$ sudo ldapsetpasswd george

Changing password for user uid=george,ou=People,dc=example,dc=com
New Password:
Retype New Password:
Successfully set password for user uid=george,ou=People,dc=example,dc=com
```

Delete a user

```
$ sudo ldapdeleteuser george
```

Add a group

```
$ sudo ldapaddgroup qa
```

Delete a group

```
$ sudo ldapdeletigroup qa
```

Add a user to a group

```
$ sudo ldapaddusertogroup george qa
```

Remove a user from a group

```
$ sudo ldapdeleteuserfromgroup george qa
```

Manage user attributes with `ldapmodifyuser`

```
sudo ldapmodifyuser george
# About to modify the following entry :
dn: uid=george,ou=People,dc=example,dc=com
objectClass: account
objectClass: posixAccount
cn: george
uid: george
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/george
loginShell: /bin/bash
gecos: george
description: User account
userPassword:: e1NTSEF9eXF5TFcyWlhWkF1eGUybVdFWHZKRzJVMjFTSG9vcHk=

# Enter your modifications here, end with CTRL-D.
dn: uid=george,ou=People,dc=example,dc=com
replace: geocos
gecos: George Carlin
```

ldapscripts templates

```
UTEMPLATE="/etc/ldapscripts/ldapadduser.template"
```

```
$ sudo cp /usr/share/doc/ldapscripts/examples/ldapadduser.template.sample \
> /etc/ldapscripts/ldapadduser.template
```

```
dn: uid=<user>,<usuffix>,<suffix>
objectClass: inetOrgPerson
objectClass: posixAccount
cn: <user>
sn: <ask>
uid: <user>
uidNumber: <uid>
gidNumber: <gid>
homeDirectory: <home>
```

(continues on next page)

(continued from previous page)

```
loginShell: <shell>
gecos: <user>
description: User account
title: Employee
```

```
$ dpkg -L ldapscripts | grep /usr/sbin
```

Next steps

setting up Transport Layer Security (TLS)

LDAP and Transport Layer Security (TLS)

Note:

For general information on managing certificates in Ubuntu, see [About certificates](#) (page 901). For installing a custom root CA, see [Install a root CA certificate in the trust store](#) (page 252).

```
$ sudo apt install gnutls-bin ssl-cert
```

```
$ sudo certtool --generate-privkey --bits 4096 --outfile
/etc/ssl/private/mycakey.pem
```

```
cn = Example Company
ca
cert_signing_key
expiration_days = 3650
```

```
$ sudo certtool --generate-self-signed \ > --load-privkey  
/etc/ssl/private/mycakey.pem \ > --template /etc/ssl/ca.info \ > --outfile  
/usr/local/share/ca-certificates/mycacert.crt
```

Note:

The `--outfile` path is correct. We are writing the CA certificate to `/usr/local/share/ca-certificates`. This is where `update-ca-certificates` picks up trusted local CAs from. To selectively enable CAs from `/usr/share/ca-certificates`, you can run `dpkg-reconfigure ca-certificates`.

```
$ sudo update-ca-certificates  
  
Updating certificates in /etc/ssl/certs...  
1 added, 0 removed; done.  
Running hooks in /etc/ca-certificates/update.d...  
done.
```

```
$ sudo certtool --generate-privkey \ > --bits 2048 \ > --outfile  
/etc/ldap/ldap01_slapd_key.pem
```

Note:

Replace `ldap01` in the filename with your server's *hostname*. Naming the certificate and key for the host and service that will be using them will help keep things clear.

```
organization = Example Company  
cn = ldap01.example.com  
tls_www_server  
encryption_key  
signing_key  
expiration_days = 365
```

```
$ sudo certtool --generate-certificate \> --load-privkey  
/etc/ldap/ldap01_slapd_key.pem \> --load-ca-certificate  
/etc/ssl/certs/mycacert.pem \> --load-ca-privkey  
/etc/ssl/private/mycakey.pem \> --template /etc/ssl/ldap01.info \>  
--outfile /etc/ldap/ldap01_slapd_cert.pem
```

```
$ sudo chgrp openldap /etc/ldap/ldap01_slapd_key.pem
```

```
$ sudo chmod 0640 /etc/ldap/ldap01_slapd_key.pem
```

```
dn: cn=config  
add: olcTLSCACertificateFile  
olcTLSCACertificateFile: /etc/ssl/certs/mycacert.pem  
-  
add: olcTLSCertificateFile  
olcTLSCertificateFile: /etc/ldap/ldap01_slapd_cert.pem  
-  
add: olcTLSCertificateKeyFile  
olcTLSCertificateKeyFile: /etc/ldap/ldap01_slapd_key.pem
```

```
$ sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f certinfo.ldif
```

```
SLAPD_SERVICES="ldap:/// ldapi:/// ldaps://"
```

```
$ sudo systemctl restart slapd
```

```
$ ldapwhoami -x -ZZ -H ldap://ldap01.example.com  
  
anonymous
```

```
$ ldapwhoami -x -H ldaps://ldap01.example.com  
  
anonymous
```

```
$ mkdir ldap02-ssl > cd ldap02-ssl > certtool --generate-privkey \ > --bits  
2048 \ > --outfile ldap02_slapd_key.pem
```

```
organization = Example Company  
cn = ldap02.example.com  
tls_www_server  
encryption_key  
signing_key  
expiration_days = 365
```

```
$ sudo certtool --generate-certificate \ > --load-privkey  
ldap02_slapd_key.pem \ > --load-ca-certificate /etc/ssl/certs/mycacert.pem \  
> --load-ca-privkey /etc/ssl/private/mycakey.pem \ > --template ldap02.info \  
> --outfile ldap02_slapd_cert.pem
```

Note:

We had to use `sudo` to get access to the CA's private key. This means the generated certificate file is owned by root. You should change that ownership back to your regular user before copying these files over to the Consumer.

```
$ cp /etc/ssl/certs/mycacert.pem .
```

```
$ cd .. > scp -r ldap02-ssl user@consumer:
```

```
$ sudo cp ldap02_slapd_cert.pem ldap02_slapd_key.pem /etc/ldap > sudo chgrp  
openldap /etc/ldap/ldap02_slapd_key.pem > sudo chmod 0640  
/etc/ldap/ldap02_slapd_key.pem > sudo cp mycacert.pem  
/usr/local/share/ca-certificates/mycacert.crt > sudo update-ca-certificates
```



```
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/mycacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/ldap02_slapd_cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/ldap02_slapd_key.pem
```

```
$ sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f certinfo.ldif
```

```
$ ldapwhoami -x -ZZ -H ldap://ldap02.example.com
anonymous
```

```
$ ldapwhoami -x -H ldaps://ldap02.example.com
anonymous
```

Backup and restore OpenLDAP

DIT

Backup script

```
#!/bin/bash
set -euo pipefail

BACKUP_PATH=/export/backup
CONFIG_BACKUP=f${BACKUP_PATH}/config.ldif"
DATA_BACKUP="${BACKUP_PATH}/example.com.ldif"
```

(continues on next page)

(continued from previous page)

```
# create and secure backup files
touch "$CONFIG_BACKUP" "$DATA_BACKUP"
chmod 600 "$CONFIG_BACKUP" "$DATA_BACKUP"

# Backup server config
nice slapcat -b cn=config > "$CONFIG_BACKUP"
# Backup directory tree
nice slapcat -b dc=example,dc=com > "$DATA_BACKUP"

# Optionally, use a backup tool like borgbackup to store the backups off-site
```

```
$ sudo chmod +x /usr/local/bin/ldapbackup
```

Note:

These files are uncompressed text files containing everything in your directory including the tree layout, usernames, and every password. Consider making /export/backup an encrypted partition and even having the script encrypt files as it creates them.

Schedule backups with systemd

```
[Unit]
Description=LDAP backup

[Service]
Type=oneshot
ExecStart=/usr/local/bin/ldapbackup
```

```
[Unit]
Description=Run LDAP backup daily

[Timer]
OnCalendar=*-*-* 22:45:00
Persistent=true

[Install]
WantedBy=timers.target
```

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl enable --now ldapbackup.timer
```

```
$ systemctl list-timers ldapbackup.timer
```

Restore script

```
#!/bin/bash
set -euo pipefail

BACKUP_PATH=/export/backup

if [ -n "$(ls -l /var/lib/ldap/* 2>/dev/null)" ] || [ -n "$(ls -l /etc/ldap/slapd.d/* 2>/dev/null)" ]; then
    echo "Existing database found. Run the following to remove it:"
    echo "  sudo systemctl stop slapd.service"
    echo "  sudo rm -rf /etc/ldap/slapd.d/* /var/lib/ldap/*"
    exit 1
fi

sudo systemctl stop slapd.service || :
sudo slapadd -F /etc/ldap/slapd.d -b cn=config -l ${BACKUP_PATH}/config.ldif
sudo slapadd -F /etc/ldap/slapd.d -b dc=example,dc=com -l ${BACKUP_PATH}/example.com.ldif
sudo chown -R openldap:openldap /etc/ldap/slapd.d/
sudo chown -R openldap:openldap /var/lib/ldap/
sudo systemctl start slapd.service
```

How to configure OpenLDAP with passthrough authentication

Background

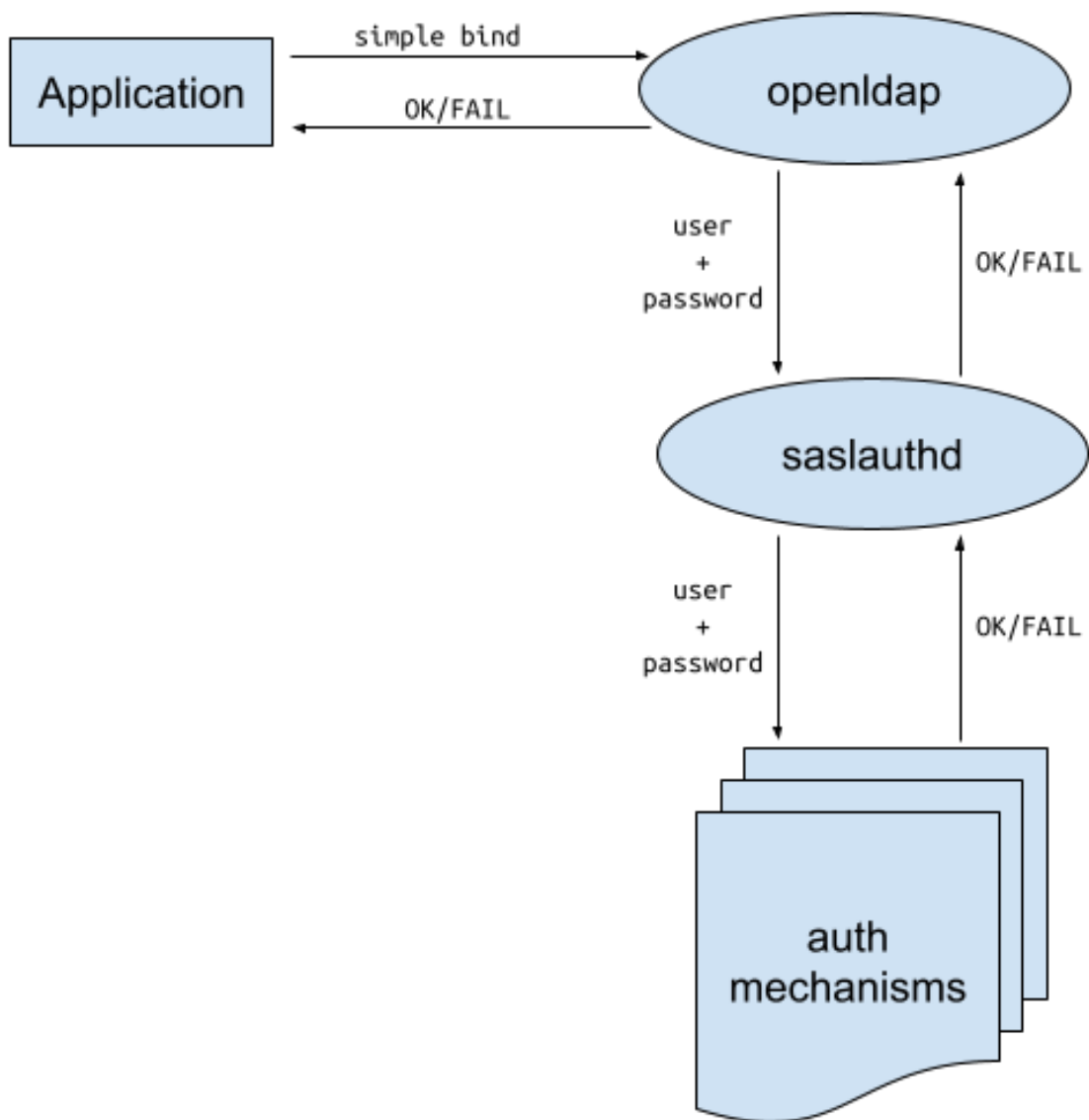
Generic Security Services Application Programming Interface³⁶⁴

setting up Transport Layer

Security (TLS)

³⁶⁴ <https://www.openldap.org/doc/admin26/sasl.html#GSSAPI>

What is passthrough authentication



Saslauthd authentication providers

-
-
-
-
-
-
-

Further reading

- [OpenLDAP.org reference](#)³⁶⁵

How to configure OpenLDAP with passthrough SASL authentication using Kerberos

Before you begin

- [How to install a Kerberos server](#)
and configure OpenLDAP *Install*
- [How to install a Kerberos server](#)
How to install a Kerberos server
- [LDAP and Transport Layer Security \(TLS\)](#)
TLS
- [How to configure Kerberos service principals](#)
How to configure Kerberos service principals

³⁶⁵ <https://www.openldap.org/doc/admin26/sasl.html>

Note:

This process is not the same as using [Generic Security Services Application Programming Interface³⁶⁶](https://www.openldap.org/doc/admin26/sasl.html#GSSAPI) (GSSAPI) to log into the LDAP server.

Rather it is using [simple authentication³⁶⁷](https://www.openldap.org/doc/admin26/security.html#%22simple%22%20method) with the OpenLDAP server so this should be over a [Transport Layer Security³⁶⁸](https://datatracker.ietf.org/wg/tls/documents/) (TLS) connection.

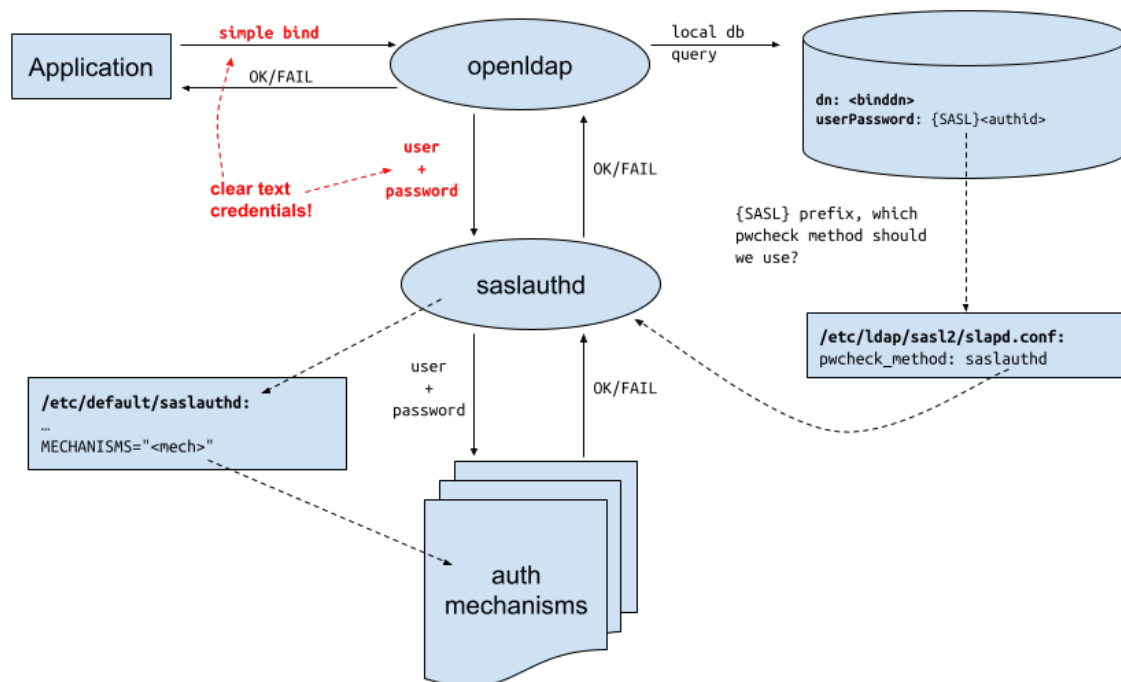
The test user we will be using is ubuntu@EXAMPLE.COM, which must exist in the Kerberos database

³⁶⁶ <https://www.openldap.org/doc/admin26/sasl.html#GSSAPI>

³⁶⁷ <https://www.openldap.org/doc/admin26/security.html#%22simple%22%20method>

³⁶⁸ <https://datatracker.ietf.org/wg/tls/documents/>

How the passthrough authentication will work



- **TLS** **LDAP and Transport Layer Security (TLS)**
-
-

Package installation

```
$ sudo apt install sasl2-bin
```

Check the hostname

```
$ hostname -f  
ldap-server.example.com
```

```
$ nslookup 10.10.17.91  
91.17.10.10.in-addr.arpa      name = ldap-server.example.com.
```

Fully Qualified Domain Name³⁶⁹

```
$ sudo vi /etc/hosts
```

```
10.10.17.91 ldap-server.example.com ldap-server
```

Create the saslauthd principal

How to configure Kerberos service principals

³⁶⁹ https://en.wikipedia.org/wiki/Fully_qualified_domain_name

```
$ sudo kadmin -p ubuntu/admin
```

```
Authenticating as principal ubuntu/admin with password.  
Password for ubuntu/admin@EXAMPLE.COM  
kadmin: addprinc -randkey host/ldap-server.example.com  
No policy specified for host/ldap-server.example.com@EXAMPLE.COM; defaulting to no policy  
Principal "host/ldap-server.example.com@EXAMPLE.COM" created.  
kadmin: ktadd host/ldap-server.example.com  
Entry for principal host/ldap-server.example.com with kvno 2, encryption type aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.  
Entry for principal host/ldap-server.example.com with kvno 2, encryption type aes128-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.
```

```
$ kadmin -p ubuntu/admin -q "addprinc -randkey host/ldap-server.example.com"
```

Note:

sudo is not needed to remotely create a new principal.

```
$ sudo kadmin -p ubuntu/admin -q "ktadd host/ldap-server.example.com"
```

```
$ sudo klist -k
```

```
Keytab name: FILE:/etc/krb5.keytab  
KVNO Principal  
-----  
  2 host/ldap-server.example.com@EXAMPLE.COM  
  2 host/ldap-server.example.com@EXAMPLE.COM
```

Configure saslauthd

```
$ sudo vi /etc/default/saslauthd
```



```
...
# Which authentication mechanisms should saslauthd use? (default: pam)
#
# Available options in this Debian package:
# getpwent -- use the getpwent() library function
# kerberos5 -- use Kerberos 5
# pam      -- use PAM
# rimap    -- use a remote IMAP server
# shadow   -- use the local shadow password file
# sasldb   -- use the local sasldb database file
# ldap     -- use LDAP (configuration is in /etc/saslauthd.conf)
#
# Only one option may be used at a time. See the saslauthd man page
# for more information.
#
# Example: MECHANISMS="pam"
MECHANISMS="kerberos5"
...
```

Important:

For Ubuntu version 22.04 and earlier “START=yes” must also be added to the default config file for saslauthd to start.

Enable and start saslauthd

```
$ sudo systemctl enable --now saslauthd
```

Test saslauthd configuration

```
$ testsaslauthd -u ubuntu -p ubuntusecret
0: OK "Success."
```

```
$ testsaslauthd -u ubuntu -p ubuntusecretwrong
0: NO "authentication failed"
```

Note:

In Ubuntu 22.04 LTS and earlier, the `/run/saslauthd` directory is restricted to members of the `sasl` group, so the `testsaslauthd` commands above need to be run as root (via `sudo`) or as a user who is in the `sasl` group.

Configure OpenLDAP

```
pwcheck_method: saslauthd
```

```
$ sudo gpasswd -a openldap sasl
```

```
$ sudo systemctl restart slapd.service
```

Change the userPassword attribute

```
dn: uid=ubuntu,dc=example,dc=com
uid: ubuntu
objectClass: account
objectClass: simpleSecurityObject
userPassword: {SSHA}S+wlmgneLDfEcwErKnY4mJngnVJMZAM5
```

bindDN

```
userPassword: {SASL}ubuntu@EXAMPLE.COM
```

Important:

Note how the username present in the userPassword attribute is independent of the bindDN used in the simple bind! If the userPassword attribute contained, say, {SASL}anotheruser@EXAMPLE.COM, OpenLDAP would ask saslauthd to authenticate anotheruser@EXAMPLE.COM, and not the user from the bindDN! Therefore, it's important to use OpenLDAP ACLs to prevent users from changing the userPassword attribute when using passthrough authentication!

```
$ ldapadd -x -D cn=admin,dc=example,dc=com -W -ZZ <<LDIF > dn:
uid=ubuntu,dc=example,dc=com > uid: ubuntu > objectClass: account >
objectClass: simpleSecurityObject > userPassword: {SASL}ubuntu@EXAMPLE.COM >
LDIF
```

Note:

Note how we don't need to add the posix attributes like user id, home directory, group, etc. All we really need is a directory entry that contains a userPassword attribute.

Test the authentication

```
$ ldapwhoami -D uid=ubuntu,dc=example,dc=com -W -x -ZZ
Enter LDAP Password:
```

```
dn:uid=ubuntu,dc=example,dc=com
```

```
ldap_bind: Invalid credentials (49)
```

Troubleshooting

```
$ sudo systemctl stop saslauthd
```

```
$ sudo /usr/sbin/saslauthd -a kerberos5 -d -m /var/run/saslauthd -n 1
```

Advanced options

-
-

```
krb5_keytab: /etc/saslauthd.keytab  
krb5_verify_principal: saslauthd
```

Further reading

- [OpenLDAP Passthrough Authentication](#)³⁷⁰
- [Cyrus SASL Password Verification](#)³⁷¹
- [Cyrus SASL slapd Configuration File](#)³⁷²
- [Kerberos Client Hostname Requirements](#)³⁷³

³⁷⁰ <https://www.openldap.org/doc/admin26/security.html#Pass-Through%20authentication>

³⁷¹ <https://www.cyrusimap.org/sasl/sasl/components.html#password-verification-services>

³⁷² <https://www.cyrusimap.org/sasl/sasl/faqs/openldap-sasl-gssapi.html>

³⁷³ https://web.mit.edu/kerberos/krb5-1.12/doc/admin/princ_dns.html

Set up an LDAP client

Lightweight Directory Access

Protocol (LDAP)³⁷⁴

Pluggable Authentication Modules (PAM)³⁷⁵

NSS

Note:

Ubuntu also supports [SSSD](https://sssd.io/)³⁷⁶ as an alternative LDAP client. SSSD provides credential caching and offline authentication. If you need those features, see [How to set up SSSD with LDAP](#) (page 214) instead.

³⁷⁶ <https://sssd.io/>

Prerequisites

- [Install and configure LDAP](#)
- [DN](#)
- [LDAP and Transport Layer Security \(TLS\)](#)
- [users/groups in LDAP](#)

Install the LDAP client daemon

```
$ sudo apt install nslcd libpam-ldapd libnss-ldapd
```

-
-

³⁷⁴ https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

³⁷⁵ https://en.wikipedia.org/wiki/Pluggable_Authentication_Module

Configure nslcd

```
# /etc/nslcd.conf
uid nslcd
gid nslcd

uri ldaps://ldap.example.com

base dc=example,dc=com

tls_reqcert demand
tls_cacertfile /etc/ssl/certs/ca-certificates.crt
```

nslcd.conf(5)³⁷⁷

Set up PAM

```
$ sudo pam-auth-update
```

```
$ sudo systemctl restart nslcd
```

Test the configuration

```
$ id <username>
```

NSS

```
$ getent passwd
```

```
$ getent group
```

NSS

³⁷⁷ <https://manpages.ubuntu.com/manpages/resolute/man5/nslcd.conf.5.html>

```
$ sudo systemctl stop nslcd.service
```

```
$ sudo nslcd -n -d
```

```
$ sudo systemctl start nslcd.service
```

Configure home directories

*passwd(5)*³⁷⁸

```
map passwd homeDirectory /home/$uid
```

*nslcd.conf(5)*³⁷⁹

Restrict login access

```
# Allow members of specific LDAP groups
+:(some-admin-group) (some-user-group):ALL
# Allow the local root user and members of the local "login" group (group name is
customizable)
+:root (login):ALL
# Deny everyone else
-:ALL:ALL
```

³⁷⁸ <https://manpages.ubuntu.com/manpages/resolute/man5/passwd.5.html>

³⁷⁹ <https://manpages.ubuntu.com/manpages/resolute/man5/nslcd.conf.5.html>

```
Name: LDAP group-based login access
Default: yes
Priority: 128
Auth-Type: Additional
Auth:
    required    pam_access.so  nodefgroup
Account-Type: Primary
Account:
    required    pam_access.so  nodefgroup
Session-Type: Additional
Session:
    required    pam_access.so  nodefgroup
```

```
$ sudo pam-auth-update --enable ldap-access --enable ldap --enable mkhomedir
```

Note:

Local users not in any listed LDAP group can be permitted to log in by adding them to a local login group. The login group is allowed to log in through `/etc/security/access.conf`, see above. Create the group and add users as needed:

```
$ sudo groupadd -r login
```

```
$ sudo usermod -aG login <username>
```

Grant sudo access to an LDAP group

*visudo(8)*³⁸⁰

```
%your-admin-group    ALL=(ALL) ALL
```

*visudo(8)*³⁸¹

```
%your-admin-group    ALL=(ALL) NOPASSWD: ALL
```

³⁸⁰ <https://manpages.ubuntu.com/manpages/resolute/man8/visudo.8.html>

³⁸¹ <https://manpages.ubuntu.com/manpages/resolute/man8/visudo.8.html>

Retrieve SSH public keys from LDAP

```
AuthorizedKeysCommand /usr/local/bin/ssh-ldap-key %u
AuthorizedKeysCommandUser nobody
```

```
$ sudo apt install python3-ldap3
```

```
#!/usr/bin/env python3
"""Fetch SSH public keys for a user from an LDAP directory."""

import argparse
import logging
import ldap3

def main():
    cli = argparse.ArgumentParser(description="SSH key fetcher from LDAP")
    cli.add_argument("username")
    cli.add_argument("--ldap-server", default="ldaps://ldap.example.com")
    cli.add_argument("--base-dn", default="dc=example,dc=com")
    cli.add_argument("--verbose", "-v", action="count", default=0)
    args = cli.parse_args()

    log_level = (logging.WARNING, logging.INFO, logging.DEBUG)[min(args.verbose, 2)]
    logging.basicConfig(level=log_level, format="%(asctime)s %(message)s")

    for key in fetch_keys(args.username, args.ldap_server, args.base_dn):
        print(key)

def fetch_keys(user, server, basedn):
    conn = ldap3.Connection(server)
    if not conn.bind():
        raise RuntimeError(f"anonymous bind to {server!r} failed")

    uid = ldap3.utils.conv.escape_filter_chars(user)
    ok = conn.search(
        basedn,
        f"(&(objectClass=inetOrgPerson)(uid={uid}))",
        attributes=["sshPublicKey"],
    )
    if not ok:
        raise RuntimeError(f"LDAP search failed: {conn.result}")
    if not conn.entries:
        return []
    if len(conn.entries) > 1:
        raise RuntimeError(f"ambiguous username: {len(conn.entries)} entries returned")
```

(continues on next page)

(continued from previous page)

```
return [
    ldap3.utils.conv.to_unicode(key)
    for key in conn.entries[0].sshPublicKey
]

if __name__ == "__main__":
    main()
```

```
$ sudo install -o root -g root -m 0755 ssh-ldap-key
/usr/local/bin/ssh-ldap-key
```

```
$ sudo systemctl restart ssh
```

Further reading

- *Install and configure LDAP*
- *How to set up SSSD with LDAP*
- *LDAP and Transport Layer Security (TLS)*
- *nsld(8)*³⁸² *nsld.conf(5)*³⁸³ *pam_access(8)*³⁸⁴ *visudo(8)*³⁸⁵

See also

- *How to set up SSSD with LDAP*
- *Introduction to OpenLDAP*

Databases

³⁸² <https://manpages.ubuntu.com/manpages/resolute/man8/nsld.8.html>

³⁸³ <https://manpages.ubuntu.com/manpages/resolute/man5/nsld.conf.5.html>

³⁸⁴ https://manpages.ubuntu.com/manpages/resolute/man8/pam_access.8.html

³⁸⁵ <https://manpages.ubuntu.com/manpages/resolute/man8/visudo.8.html>

Install and configure a MySQL server

MySQL³⁸⁶

Install MySQL

```
$ sudo apt install mysql-server
```

```
$ sudo service mysql status

mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:
  enabled)
  Active: active (running) since Tue 2019-10-08 14:37:38 PDT; 2 weeks 5 days ago
  Main PID: 2028 (mysqld)
  Tasks: 28 (limit: 4915)
  CGroup: /system.slice/mysql.service
          └─2028 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid
  Oct 08 14:37:36 db.example.org systemd[1]: Starting MySQL Community Server...
  Oct 08 14:37:38 db.example.org systemd[1]: Started MySQL Community Server.
```

```
$ sudo ss -tap | grep mysql

LISTEN  0      151             127.0.0.1:mysql      0.0.0.0:*
users:((("mysqld",pid=149190,fd=29))
LISTEN  0      70              *:33060              *:.*
users:((("mysqld",pid=149190,fd=32))
```

```
$ sudo service mysql restart
```

```
$ sudo journalctl -u mysql
```

³⁸⁶ <https://www.mysql.com/>

Configure MySQL

```
bind-address = 192.168.0.5
```

Note:

Replace 192.168.0.5 with the appropriate address, which can be determined via the `ip address show` command.

```
$ sudo systemctl restart mysql.service
```

User setup

```
$ sudo mysql -u root
```

`auth_socket`³⁸⁷

Create a new user

```
CREATE USER 'username'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'password';
```

```
CREATE USER 'username'@'localhost' IDENTIFIED WITH caching_sha2_password BY RANDOM  
PASSWORD;
```

MySQL's upstream documentation³⁸⁸

³⁸⁷ <https://dev.mysql.com/doc/mysql-secure-deployment-guide/8.0/en/secure-deployment-configure-authentication.html>

³⁸⁸ <https://dev.mysql.com/doc/mysql-secure-deployment-guide/8.0/en/secure-deployment-user-accounts.html>

```
CREATE USER 'username'@'localhost' IDENTIFIED WITH auth_socket;
```

```
CREATE USER 'username'@'localhost' IDENTIFIED WITH auth_socket AS 'system-user-username';
```

Adding user permissions

```
GRANT SELECT on *.* TO 'username'@'localhost';
```

Database engines

InnoDB³⁸⁹ MyISAM³⁹⁰

InnoDB

ACID compliant³⁹¹

³⁸⁹ <https://dev.mysql.com/doc/refman/8.0/en/innodb-storage-engine.html>

³⁹⁰ <https://dev.mysql.com/doc/refman/8.0/en/myisam-storage-engine.html>

³⁹¹ <https://en.wikipedia.org/wiki/ACID>

MyISAM

FULLTEXT

database³⁹²

MyISAM on a production

Backups

mysqldump

MySQL Shell's Dump Utility

Percona Xtrabackup

mysqldump

logical backups³⁹³

```
$ mysqldump [database name] > dump.sql
```

```
$ mysqldump -u root [database name] > dump.sql
```

³⁹² <https://www.percona.com/blog/using-mysam-in-production/>

³⁹³ https://dev.mysql.com/doc/refman/8.4/en/glossary.html#glos_logical_backup

```
$ mysql -u root [database name] < dump.sql
```

[upstream documentation](#)³⁹⁴

MySQL Shell Dump Utility

```
$ sudo apt install mysql-shell
```

```
$ mysqlsh --socket=/var/run/mysql/mysql.sock --no-password --python
```

```
util.dump_instance("/tmp/worlddump")
```

```
util.dump_tables("database name", ["table 1", "table 2"], "/tmp/taledump")
```

[dump loading utility](#)³⁹⁵

```
util.load_dump("/tmp/worlddump")
```

Note:

To restore data from a local file, `local_infile` needs to be enabled on the MySQL server. Activate this by accessing the server with the `mysql` command and entering `SET GLOBAL local_infile=1;`

[MySQL Shell dump documentation](#)³⁹⁶

³⁹⁴ <https://dev.mysql.com/doc/refman/8.4/en/mysqldump.html>

³⁹⁵ <https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-utilities-load-dump.html>

³⁹⁶ <https://dev.mysql.com/doc/mysql-shell/8.4/en/mysql-shell-utilities-dump-instance-schema.html>

Percona Xtrabackup

ical backups³⁹⁷

MySQL Enterprise Backup³⁹⁸ phys-

```
$ sudo apt install percona-xtrabackup
```

```
$ xtrabackup --backup --target-dir=/tmp/worlddump
```

```
$ sudo systemctl stop mysql > xtrabackup --prepare  
--target-dir=/tmp/worlddump > sudo rm -rf /var/lib/mysql > sudo xtrabackup  
--copy-back --target-dir=/tmp/worlddump --datadir=/var/lib/mysql > sudo chown  
-R mysql:mysql /var/lib/mysql > sudo systemctl start mysql
```

Percona's upstream documentation³⁹⁹

Advanced configuration

Creating a tuned configuration

```
$ mysqldump --all-databases --routines -u root -p > ~/fulldump.sql
```

```
$ sudo service mysql stop
```

³⁹⁷ https://dev.mysql.com/doc/refman/8.4/en/glossary.html#glos_physical_backup

³⁹⁸ <https://www.mysql.com/products/enterprise/backup.html>

³⁹⁹ <https://docs.percona.com/percona-xtrabackup/8.0/>


```
$ sudo rsync -avz /etc/mysql /root/mysql-backup
```

```
$ sudo rm -rf /var/lib/mysql/* > sudo mysqld --initialize > sudo chown -R  
mysql: /var/lib/mysql > sudo service mysql start
```

```
$ cat ~/fulldump.sql | mysql
```

```
$ sudo apt install pv
```

```
$ pv ~/fulldump.sql | mysql
```

Note:

This is not necessary for all `my.cnf` changes. Most of the variables you can change to improve performance are adjustable even whilst the server is running. As with anything, make sure to have a good backup copy of your config files and data before making changes.

MySQL Tuner

MySQL Tuner⁴⁰⁰

```
$ sudo apt install mysqltuner
```

official MySQL documentation⁴⁰¹

⁴⁰⁰ <https://github.com/major/MySQLTuner-perl>

⁴⁰¹ <https://dev.mysql.com/doc/>

```
----- Recommendations -----
General recommendations:
  Run OPTIMIZE TABLE to defragment tables for better performance
  Increase table_cache gradually to avoid file descriptor limits
Variables to adjust:
  key_buffer_size (> 1.4G)
  query_cache_size (> 32M)
  table_cache (> 64)
  innodb_buffer_pool_size (>= 22G)
```

Further reading

- [MySQL Developers portal](#)⁴⁰²
- [MySQL Getting Started with SQL: A Hands-On Approach for Beginners](#)⁴⁰³
- [MySQL Getting Started with SQL: A Hands-On Approach for Beginners](#)⁴⁰⁴

Install and configure PostgreSQL

PostgreSQL⁴⁰⁵

Install PostgreSQL

```
$ sudo apt install postgresql
```

⁴⁰² <https://dev.mysql.com/doc/>

⁴⁰³ <https://www.oreilly.com/library/view/getting-started-with/9781491938607/>

⁴⁰⁴ <https://www.oreilly.com/library/view/sql-in-a/9780596155322/>

⁴⁰⁵ <https://www.postgresql.org/>

Configure PostgreSQL

Administrator's Guide⁴⁰⁶

PostgreSQL

Tip:

To configure *IDENT* authentication, add entries to the `/etc/postgresql/*/main/pg_ident.conf` file. There are detailed comments in the file to guide you.

```
listen_addresses = '*'
```

Note:

To listen on all IPv4 interfaces, set `listen_addresses` to `'0.0.0.0'`, while `:::` will listen on all IPv6 interfaces. `'*'` will cause PostgreSQL to listen on all available network interfaces, both IPv4 and IPv6.

mentation⁴⁰⁷

PostgreSQL docu-

```
$ sudo -u postgres psql template1
```

```
ALTER USER postgres with encrypted password 'your_password';
```

hostssl template1	postgres	192.168.1.1/24	scram-sha-256

⁴⁰⁶ <https://www.postgresql.org/docs/current/admin.html>

⁴⁰⁷ <https://www.postgresql.org/docs/>

Note:

The config statement `hostssl` used here will reject TCP connections that would not use SSL. PostgreSQL in Ubuntu has the SSL feature built in and configured by default, so it works right away. On your PostgreSQL server this uses the certificate created by the `ssl-cert` package which is suitable for testing, but for production use you should replace it with a certificate from a trusted Certificate Authority (CA). See [Obtain TLS certificates](#) (page 245) for how to get a certificate from Let's Encrypt or set up your own CA.

```
$ sudo systemctl restart postgresql.service
```

Warning:

The above configuration is not complete by any means. Please refer to the [PostgreSQL Administrator's Guide](#)⁴⁰⁸ to configure more parameters.

⁴⁰⁸ <https://www.postgresql.org/docs/current/admin.html>

```
$ sudo apt install postgresql-client > psql --host your-servers-dns-or-ip  
--username postgres --password --dbname template1
```

Streaming replication

(WAL) XLOG⁴⁰⁹

```
$ sudo -u postgres createuser --replication -P -e replicator
```

```
listen_addresses = '*'  
wal_level = replica
```

⁴⁰⁹ <https://www.postgresql.org/docs/current/wal.html>

```
host replication replicator <IP address of the standby> scram-sha-256
```

```
$ sudo systemctl restart postgresql
```

```
$ sudo systemctl stop postgresql
```

```
hot_standby = on
```

```
$ sudo su - postgres > # backup the current content of the standby server
(update the version of your postgres accordingly) > cp -R
/var/lib/postgresql/14/main /var/lib/postgresql/14/main_bak > # remove all
the files in the data directory > rm -rf /var/lib/postgresql/14/main/* >
pg_basebackup -h <IP address of the main server> -D
/var/lib/postgresql/14/main -U replicator -P -v -R
```

- *hostname*
-
-
-
-
-

```
$ sudo systemctl start postgresql
```

```
$ sudo -u postgres psql -c "select * from pg_stat_replication;"
```

ability, Load Balancing, and Replication⁴¹¹

replication⁴¹⁰

High Avail-

⁴¹⁰ <https://www.postgresql.org/docs/current/runtime-config-replication.html>

⁴¹¹ <https://www.postgresql.org/docs/current/high-availability.html>

```
$ sudo -u postgres createdb test # on the main server
```

```
$ sudo -u postgres psql -c "\l" # on the standby server
```

Backups

tor's Guide⁴¹²

PostgreSQL Administra-

Further reading

- PostgreSQL Administrator's Guide⁴¹³

```
$ sudo apt install postgresql-doc
```

dblink

- On Approach for Beginners⁴¹⁴ Getting Started with SQL: A Hands-
SQL in a Nutshell⁴¹⁵

See also

- *Introduction to databases*

⁴¹² <https://www.postgresql.org/docs/current/backup.html>

⁴¹³ <https://www.postgresql.org/docs/current/admin.html>

⁴¹⁴ <https://www.oreilly.com/library/view/getting-started-with/9781491938607/>

⁴¹⁵ <https://www.oreilly.com/library/view/sql-in-a/9780596155322/>

Storage and backups

- *Storage*
- *Backups and version control*

Storage

- *Logical Volume Management (LVM)*
- *iSCSI initiator and an iSCSI target.*

How to manage logical volumes

explanation of logical volume management (LVM)

Create the physical volume

```
$ sudo pvcreate /dev/sda1
```

Create the volume group

```
$ sudo vgcreate foo /dev/sda1
```

Create a logical volume

```
$ sudo lvcreate --name bar --size 5g foo
```

Resize a partition

```
$ sudo lvextend --resizefs --size +5g foo/bar
```

filesystem

Move a partition

```
$ sudo pvmove --name bar /dev/sda1
```


Create a snapshot

```
$ sudo lvcreate --snapshot --name lv_snapshot --size 5g foo/bar
```

```
$ sudo lvconvert --merge foo/lv_snapshot
```

iSCSI initiator (or client)

Wikipedia iSCSI Definition⁴¹⁶:

iSCSI an acronym for **Internet Small Computer Systems Interface** , an [Internet Protocol](#)⁴¹⁷ (IP)-based storage networking standard for linking data storage facilities. It provides [block-level access](#)⁴¹⁸ to [storage devices](#)⁴¹⁹ by carrying [SCSI](#)⁴²⁰ commands over a [TCP/IP](#)⁴²¹ network.

iSCSI is used to facilitate data transfers over [intranets](#)⁴²² and to manage storage over long distances. It can be used to transmit data over [local area networks](#)⁴²³ (LANs), [wide area networks](#)⁴²⁴ (WANs), or the [Internet](#)⁴²⁵ and can enable location-independent data storage and retrieval.

The [protocol](#)⁴²⁶ allows clients (called *initiators*) to send SCSI commands (*CDBs*⁴²⁷) to storage devices (*targets*) on remote servers. It is a [storage area network](#)⁴²⁸ (SAN) protocol, allowing organizations to consolidate storage into [storage arrays](#)⁴²⁹ while providing clients (such as database and web servers) with the illusion of locally attached SCSI disks.

It mainly competes with [Fibre Channel](#)⁴³⁰ , but unlike traditional *Fibre Channel*, which usually requires dedicated cabling, iSCSI can be run over long distances using existing network infrastructure.

⁴¹⁶ <https://en.wikipedia.org/wiki/ISCSI>

⁴¹⁷ https://en.wikipedia.org/wiki/Internet_Protocol

⁴¹⁸ https://en.wikipedia.org/wiki/Block-level_storage

⁴¹⁹ https://en.wikipedia.org/wiki/Computer_data_storage

⁴²⁰ <https://en.wikipedia.org/wiki/SCSI>

⁴²¹ <https://en.wikipedia.org/wiki/TCP/IP>

⁴²² <https://en.wikipedia.org/wiki/Intranet>

⁴²³ https://en.wikipedia.org/wiki/Local_area_network

⁴²⁴ https://en.wikipedia.org/wiki/Wide_area_network

⁴²⁵ <https://en.wikipedia.org/wiki/Internet>

⁴²⁶ [https://en.wikipedia.org/wiki/Protocol_\(computing\)](https://en.wikipedia.org/wiki/Protocol_(computing))

⁴²⁷ https://en.wikipedia.org/wiki/SCSI_CDB

⁴²⁸ https://en.wikipedia.org/wiki/Storage_area_network

⁴²⁹ https://en.wikipedia.org/wiki/Storage_array

⁴³⁰ https://en.wikipedia.org/wiki/Fibre_Channel

Note:

It is assumed that **you already have an iSCSI target on your local network** and have the appropriate rights to connect to it. The instructions for setting up a target vary greatly between hardware providers, so consult your vendor documentation to configure your specific iSCSI target.

Network Interfaces Configuration

netplan.io⁴³¹

```
/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
```

```
{config: disabled}
```

```
/etc/netplan/50-cloud-init.yaml
```

```
network:
  ethernets:
    enp5s0:
      match:
        macaddress: 00:16:3e:af:c4:d6
      set-name: eth0
      dhcp4: true
      dhcp-identifier: mac
    enp6s0:
      match:
        macaddress: 00:16:3e:50:11:9c
      set-name: iscsi01
      dhcp4: true
      dhcp-identifier: mac
      dhcp4-overrides:
        route-metric: 300
    enp7s0:
      match:
        macaddress: 00:16:3e:b3:cc:50
      set-name: iscsi02
      dhcp4: true
      dhcp-identifier: mac
      dhcp4-overrides:
        route-metric: 300
  version: 2
  renderer: networkd
```

If you have only a single interface for the iSCSI network, make sure to follow the same instructions, but only consider the **iscsi01** interface command line examples.

⁴³¹ <https://netplan.readthedocs.io/en/latest/examples/>

iSCSI Initiator Install

```
$ sudo apt install open-iscsi
```

-
-

iSCSI Initiator Configuration

/etc/iscsi/iscsid.conf

```
### startup settings

## will be controlled by systemd, leave as is
iscsid.startup = /usr/bin/iscsidnode.startup = manual

### chap settings

# node.session.auth.authmethod = CHAP

## authentication of initiator by target (session)
# node.session.auth.username = username
# node.session.auth.password = password

# discovery.sendtargets.auth.authmethod = CHAP

## authentication of initiator by target (discovery)
# discovery.sendtargets.auth.username = username
# discovery.sendtargets.auth.password = password

### timeouts

## control how much time iscsi takes to propagate an error to the
## upper layer. if using multipath, having 0 here is desirable
## so multipath can handle path errors as quickly as possible
## (and decide to queue or not if missing all paths)
node.session.timeo.replacement_timeout = 0

node.conn[0].timeo.login_timeout = 15
node.conn[0].timeo.logout_timeout = 15

## interval for a NOP-Out request (a ping to the target)
node.conn[0].timeo.noop_out_interval = 5

## and how much time to wait before declaring a timeout
node.conn[0].timeo.noop_out_timeout = 5
```

(continues on next page)

(continued from previous page)

```
## default timeouts for error recovery logics (lu & tgt resets)
node.session.err_timeo.abort_timeout = 15
node.session.err_timeo.lu_reset_timeout = 30
node.session.err_timeo.tgt_reset_timeout = 30

### retry

node.session.initial_login_retry_max = 8

### session and device queue depth

node.session.cmds_max = 128
node.session.queue_depth = 32

### performance

node.session.xmit_thread_priority = -20
```

```
$ systemctl restart iscsid.service
```

```
/etc/iscsi/initiatorname.iscsi
```

```
InitiatorName=iqn.2004-10.com.ubuntu:01 60 73517884c3
```

iSCSI Network Configuration

•

```
$ iscsi01_ip=$(ip -4 -o addr show iscsi01 | sed -r 's:.*
([0-9]{1,3}\.){3}[0-9]{1,3})/.*:\1:')

```

```
$ iscsi02_ip=$(ip -4 -o addr show iscsi02 | sed -r 's:.*
([0-9]{1,3}\.){3}[0-9]{1,3})/.*:\1:')

```

```
$ iscsi01_mac=$(ip -o link show iscsi01 | sed -r 's:.*\s+link/ether  
([0-f]{2}(\:|))\{6\}.*:\1:g')
```

```
$ iscsi02_mac=$(ip -o link show iscsi02 | sed -r 's:.*\s+link/ether  
([0-f]{2}(\:|))\{6\}.*:\1:g')
```

•

```
$ sudo iscsiadm -m iface -I iscsi01 --op=new  
New interface iscsi01 added
```

```
$ sudo iscsiadm -m iface -I iscsi01 --op=update -n iface.hwaddress -v  
$iscsi01_mac  
iscsi01 updated.
```

```
$ sudo iscsiadm -m iface -I iscsi01 --op=update -n iface.ipaddress -v  
$iscsi01_ip  
iscsi01 updated.
```

•

```
$ sudo iscsiadm -m iface -I iscsi02 --op=new  
New interface iscsi02 added
```

```
$ sudo iscsiadm -m iface -I iscsi02 --op=update -n iface.hwaddress -v  
$iscsi02_mac  
iscsi02 updated.
```

```
$ sudo iscsiadm -m iface -I iscsi02 --op=update -n iface.ipaddress -v  
$iscsi02_ip  
iscsi02 updated.
```

•

```
$ sudo iscsiadm -m discovery -I iscsi01 --op=new --op=del --type  
sendtargets --portal storage.iscsi01  
10.250.94.99:3260,1 iqn.2003-01.org.linux-iscsi.storage.x8664:sn.2c084c8320ca
```

```
$ sudo iscsiadm -m discovery -I iscsi02 --op=new --op=del --type
sendtargets --portal storage.iscsi02

10.250.93.99:3260,1 iqn.2003-01.org.linux-iscsi.storage.x8664:sn.2c084c8320ca
```

```
$ sudo iscsiadm -m node --op=update -n node.conn[0].startup -v automatic
```

```
$ sudo iscsiadm -m node --op=update -n node.startup -v automatic
```

```
$ systemctl enable open-iscsi
```

```
Synchronizing state of open-iscsi.service with SysV service script with /lib/
systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable open-iscsi
Created symlink /etc/systemd/system/iscsi.service → /lib/systemd/system/open-
iscsi.service.
Created symlink /etc/systemd/system/sysinit.target.wants/open-iscsi.service →
/lib/systemd/system/open-iscsi.service.
```

```
$ systemctl enable iscsid
```

```
Synchronizing state of iscsid.service with SysV service script with /lib/
systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable iscsid
Created symlink /etc/systemd/system/sysinit.target.wants/iscsid.service → /
lib/systemd/system/iscsid.service.
```

```
$ systemctl restart iscsid.service
```

```
$ sudo iscsiadm -m node --loginall=automatic
```

```
Logging in to [iface: iscsi02, target: iqn.2003-01.org.linux-iscsi.storage.
x8664:sn.2c084c8320ca, portal: 10.250.93.99,3260] (multiple)
Logging in to [iface: iscsi01, target: iqn.2003-01.org.linux-iscsi.storage.
x8664:sn.2c084c8320ca, portal: 10.250.94.99,3260] (multiple)
Login to [iface: iscsi02, target: iqn.2003-01.org.linux-iscsi.storage.
x8664:sn.2c084c8320ca, portal: 10.250.93.99,3260] successful.
Login to [iface: iscsi01, target: iqn.2003-01.org.linux-iscsi.storage.
x8664:sn.2c084c8320ca, portal: 10.250.94.99,3260] successful.
```

Accessing the Logical Units (or LUNs)

dmesg

dmesg

```
[ 166.840694] scsi 7 0 0:4: Direct-Access    LIO-ORG    TCMU device >      0002 PQ: 0
ANSI: 5
[ 166.840892] scsi 8 0 0:4: Direct-Access    LIO-ORG    TCMU device >      0002 PQ: 0
ANSI: 5
[ 166.841741] sd 7 0 0 4: Attached scsi generic sg2 type 0
[ 166.841808] sd 8 0 0 4: Attached scsi generic sg3 type 0
[ 166.842278] scsi 7 0 0:3: Direct-Access    LIO-ORG    TCMU device >      0002 PQ: 0
ANSI: 5
[ 166.842571] scsi 8 0 0:3: Direct-Access    LIO-ORG    TCMU device >      0002 PQ: 0
ANSI: 5
[ 166.843482] sd 8 0 0 3: Attached scsi generic sg4 type 0
[ 166.843681] sd 7 0 0 3: Attached scsi generic sg5 type 0
[ 166.843706] sd 8 0 0 4: [sdd] 2097152 512-byte logical blocks: > (1.07 GB/1.00 GiB)
[ 166.843884] scsi 8 0 0:2: Direct-Access    LIO-ORG    TCMU device >      0002 PQ: 0
ANSI: 5
[ 166.843971] sd 8 0 0 4: [sdd] Write Protect is off
[ 166.843972] sd 8 0 0 4: [sdd] Mode Sense: 2f 00 00 00
[ 166.844127] scsi 7 0 0:2: Direct-Access    LIO-ORG    TCMU device >      0002 PQ: 0
ANSI: 5
[ 166.844232] sd 7 0 0 4: [sdc] 2097152 512-byte logical blocks: > (1.07 GB/1.00 GiB)
[ 166.844421] sd 8 0 0 4: [sdd] Write cache: enabled, read cache: > enabled, doesn't
support DPO or FUA
[ 166.844566] sd 7 0 0 4: [sdc] Write Protect is off
[ 166.844568] sd 7 0 0 4: [sdc] Mode Sense: 2f 00 00 00
[ 166.844846] sd 8 0 0 2: Attached scsi generic sg6 type 0
[ 166.845147] sd 7 0 0 4: [sdc] Write cache: enabled, read cache: > enabled, doesn't
support DPO or FUA
[ 166.845188] sd 8 0 0 4: [sdd] Optimal transfer size 65536 bytes
[ 166.845527] sd 7 0 0 2: Attached scsi generic sg7 type 0
[ 166.845678] sd 8 0 0 3: [sde] 2097152 512-byte logical blocks: > (1.07 GB/1.00 GiB)
[ 166.845785] scsi 8 0 0:1: Direct-Access    LIO-ORG    TCMU device >      0002 PQ: 0
ANSI: 5
[ 166.845799] sd 7 0 0 4: [sdc] Optimal transfer size 65536 bytes
[ 166.845931] sd 8 0 0 3: [sde] Write Protect is off
[ 166.845933] sd 8 0 0 3: [sde] Mode Sense: 2f 00 00 00
[ 166.846424] scsi 7 0 0:1: Direct-Access    LIO-ORG    TCMU device >      0002 PQ: 0
ANSI: 5
[ 166.846552] sd 8 0 0 3: [sde] Write cache: enabled, read cache: > enabled, doesn't
support DPO or FUA
[ 166.846708] sd 7 0 0 3: [sdf] 2097152 512-byte logical blocks: > (1.07 GB/1.00 GiB)
[ 166.847024] sd 8 0 0 1: Attached scsi generic sg8 type 0
[ 166.847029] sd 7 0 0 3: [sdf] Write Protect is off
[ 166.847031] sd 7 0 0 3: [sdf] Mode Sense: 2f 00 00 00
[ 166.847043] sd 8 0 0 3: [sde] Optimal transfer size 65536 bytes
[ 166.847133] sd 8 0 0 2: [sdg] 2097152 512-byte logical blocks: > (1.07 GB/1.00 GiB)
[ 166.849212] sd 8 0 0 2: [sdg] Write Protect is off
[ 166.849214] sd 8 0 0 2: [sdg] Mode Sense: 2f 00 00 00
[ 166.849711] sd 7 0 0 3: [sdf] Write cache: enabled, read cache: > enabled, doesn't
support DPO or FUA
[ 166.849718] sd 7 0 0 1: Attached scsi generic sg9 type 0
```

(continues on next page)

(continued from previous page)

```
[ 166.849721] sd 7 0 0 2: [sdh] 2097152 512-byte logical blocks: > (1.07 GB/1.00 GiB)
[ 166.853296] sd 8 0 0 2: [sdg] Write cache: enabled, read cache: > enabled, doesn't
support DPO or FUA
[ 166.853721] sd 8 0 0 2: [sdg] Optimal transfer size 65536 bytes
[ 166.853810] sd 7 0 0 2: [sdh] Write Protect is off
[ 166.853812] sd 7 0 0 2: [sdh] Mode Sense: 2f 00 00 00
[ 166.854026] sd 7 0 0 3: [sdf] Optimal transfer size 65536 bytes
[ 166.854431] sd 7 0 0 2: [sdh] Write cache: enabled, read cache: > enabled, doesn't
support DPO or FUA
[ 166.854625] sd 8 0 0 1: [sdi] 2097152 512-byte logical blocks: > (1.07 GB/1.00 GiB)
[ 166.854898] sd 8 0 0 1: [sdi] Write Protect is off
[ 166.854900] sd 8 0 0 1: [sdi] Mode Sense: 2f 00 00 00
[ 166.855022] sd 7 0 0 2: [sdh] Optimal transfer size 65536 bytes
[ 166.855465] sd 8 0 0 1: [sdi] Write cache: enabled, read cache: > enabled, doesn't
support DPO or FUA
[ 166.855578] sd 7 0 0 1: [sdj] 2097152 512-byte logical blocks: > (1.07 GB/1.00 GiB)
[ 166.855845] sd 7 0 0 1: [sdj] Write Protect is off
[ 166.855847] sd 7 0 0 1: [sdj] Mode Sense: 2f 00 00 00
[ 166.855978] sd 8 0 0 1: [sdi] Optimal transfer size 65536 bytes
[ 166.856305] sd 7 0 0 1: [sdj] Write cache: enabled, read cache: > enabled, doesn't
support DPO or FUA
[ 166.856701] sd 7 0 0 1: [sdj] Optimal transfer size 65536 bytes
[ 166.859624] sd 8 0 0 4: [sdd] Attached SCSI disk
[ 166.861304] sd 7 0 0 4: [sdc] Attached SCSI disk
[ 166.864409] sd 8 0 0 3: [sde] Attached SCSI disk
[ 166.864833] sd 7 0 0 3: [sdf] Attached SCSI disk
[ 166.867906] sd 8 0 0 2: [sdg] Attached SCSI disk
[ 166.868446] sd 8 0 0 1: [sdi] Attached SCSI disk
[ 166.871588] sd 7 0 0 1: [sdj] Attached SCSI disk
[ 166.871773] sd 7 0 0 2: [sdh] Attached SCSI disk
```

Introduction to device mapper multipathing

("multipath")

```
$ apt-get install multipath-tools
```

```
$ sudo multipath -r
```

```
$ sudo multipath -ll
```

```

mpathd (360014051a042fb7c41c4249af9f2cfbc) dm-3 LIO-ORG,TCMU device
size=1.0G features='0' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=1 status=active
|  '- 7:0:0:4 sde 8:64  active ready running
`-+- policy='service-time 0' prio=1 status=enabled
   '- 8:0:0:4 sdc 8:32  active ready running
mpathc (360014050d6871110232471d8bcd155a3) dm-2 LIO-ORG,TCMU device
size=1.0G features='0' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=1 status=active
|  '- 7:0:0:3 sdf 8:80  active ready running
`-+- policy='service-time 0' prio=1 status=enabled
   '- 8:0:0:3 sdd 8:48  active ready running
mpathb (360014051f65c6cb11b74541b703ce1d4) dm-1 LIO-ORG,TCMU device
size=1.0G features='0' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=1 status=active
|  '- 7:0:0:2 sdh 8:112 active ready running
`-+- policy='service-time 0' prio=1 status=enabled
   '- 8:0:0:2 sdg 8:96  active ready running
mpatha (36001405b816e24fcab64fb88332a3fc9) dm-0 LIO-ORG,TCMU device
size=1.0G features='0' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=1 status=active
|  '- 7:0:0:1 sdj 8:144 active ready running
`-+- policy='service-time 0' prio=1 status=enabled
   '- 8:0:0:1 sdi 8:128 active ready running

```

```

.
-
-
.
-
-
.
-
-
.
-
-

```

Warning:

Do not use this in production without checking appropriate multipath configuration options in the **Device Mapper Multipathing** section. The *default multipath configuration* is sub-optimal for regular usage.

-

-

-

-

-

```
$ sudo fdisk /dev/mapper/mpatha

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x92c0322a.

Command (m for help): p
Disk /dev/mapper/mpatha: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 65536 bytes
Disklabel type: dos
Disk identifier: 0x92c0322a

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-2097151, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097151, default 2097151):

Created a new partition 1 of type 'Linux' and of size 1023 MiB.

Command (m for help): w
The partition table has been altered.
```

- *filesystem*

```
$ sudo mkfs.ext4 /dev/mapper/mpatha-part1
```

```
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 261888 4k blocks and 65536 inodes
Filesystem UUID: cdb70b1e-c47c-47fd-9c4a-03db6f038988
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

•

```
$ sudo mount /dev/mapper/mpatha-part1 /mnt
```

•

```
$ ls /mnt

lost+found
```

Further reading

- [iscsid](#)⁴³²
- [iscsi.conf](#)⁴³³
- [iscsid.conf](#)⁴³⁴
- [iscsi.service](#)⁴³⁵
- [iscsid.service](#)⁴³⁶
- [Open-iSCSI](#)⁴³⁷
- [Debian Open-iSCSI](#)⁴³⁸

⁴³² <https://linux.die.net/man/8/iscsid>

⁴³³ <https://linux.die.net/man/5/iscsi.conf>

⁴³⁴ <https://github.com/open-iscsi/open-iscsi/blob/master/etc/iscsid.conf>

⁴³⁵ <https://github.com/open-iscsi/open-iscsi/blob/master/etc/systemd/iscsi.service.template>

⁴³⁶ <https://github.com/open-iscsi/open-iscsi/blob/master/etc/systemd/iscsid.service.template>

⁴³⁷ <http://www.open-iscsi.com/>

⁴³⁸ <https://wiki.debian.org/SAN/iSCSI/open-iscsi>

See also

- [About Logical Volume Management \(LVM\)](#)

Backups and version control

-
-
-
-

Backup utilities

Bacula⁴³⁹ rsnapshot⁴⁴⁰

Tool	Back up	Backup Method	Install and configure
Bacula	Multiple systems over a network	Incremental backups	how to install and configure Bacula (page 525)
rsnapshot	Single system	Periodic “snapshots” of files locally or remotely with SSH	how to install and configure rsnapshot (page 547)

⁴³⁹ <https://www.bacula.org/>
⁴⁴⁰ <https://rsnapshot.org/>

How to install and configure Bacula

Bacula⁴⁴¹

Bacula components

-
-
-
-
-
-

Installing the Server Components

-
-
-
-

⁴⁴¹ <https://www.bacula.org/>

-

```
$ sudo apt install postgresql
```

Note:

Refer to *MySQL databases* (page 498) and *PostgreSQL databases* (page 505) for more details on these powerful databases.

```
$ sudo apt install bacula
```

Configuring the Server

-
-
-
-

Director

-
-
-
-
-

The Director resource

```
Director {  
  Name = bacula-server-dir  
  DIRport = 9101  
  QueryFile = "/etc/bacula/scripts/query.sql"  
  WorkingDirectory = "/var/lib/bacula"  
  PidDirectory = "/run/bacula"  
  Maximum Concurrent Jobs = 20  
  Password = "<randomly generated>"  
  Messages = Daemon  
  #DirAddress = 127.0.0.1  
}
```

-
-
-

Tip:

For more details about all the options of the Director resource, please check the upstream [Director Resource](#)⁴⁴² documentation.

⁴⁴² https://www.bacula.org/15.0.x-manuals/en/main/Configuring_Director.html#SECTION0023200000000000000000

The FileSet resource

```
FileSet {
  Name = "Home Set"
  Include {
    Options {
      signature = SHA256
      aclsupport = yes
      xattrsupport = yes
    }
    Options {
      wilddir = "/home/*/Downloads"
      wildfile = "*.iso"
      exclude = yes
    }
    File = /home
  }
}
```

-

-

—

-

Tip:

For more details about all the options of the FileSet resource, please check the upstream [FileSet Resource](https://www.bacula.org/15.0.x-manuals/en/main/Configuring_Director.html#SECTION00237000000000000000)⁴⁴³ documentation.

⁴⁴³ https://www.bacula.org/15.0.x-manuals/en/main/Configuring_Director.html#SECTION00237000000000000000

The Client resource

```
# Client (File Services) to backup
Client {
    Name = bacula-server-fd
    Address = bacula-server.lxd # use the real hostname instead of "localhost"
    FDPort = 9102
    Catalog = MyCatalog
    Password = "<randomly generated>"
    File Retention = 60 days
    Job Retention = 6 months
    AutoPrune = yes
}
```

-
-
-
-
-

Tip:

For more details about all the options of the Client resource, please check the upstream [Client Resource⁴⁴⁴](https://www.bacula.org/15.0.x-manuals/en/main/Configuring_Director.html#SECTION0023130000000000000000) documentation.

⁴⁴⁴ https://www.bacula.org/15.0.x-manuals/en/main/Configuring_Director.html#SECTION0023130000000000000000

The Pool resource

```
Pool {
  Name = File
  Pool Type = Backup
  Recycle = yes           # Bacula can automatically recycle Volumes
  AutoPrune = yes        # Prune expired volumes
  Volume Retention = 365 days # one year
  Maximum Volume Bytes = 50G # Limit Volume size
  Maximum Volumes = 100     # Limit number of Volumes in Pool
  Label Format = "Vol-"     # Auto label
}
```

-
-
-
-
-

Tip:

For more details about all the options of the Pool resource, please check the upstream [Pool Resource](#)⁴⁴⁵ documentation.

⁴⁴⁵ https://www.bacula.org/15.0.x-manuals/en/main/Configuring_Director.html#SECTION002316000000000000000000

The Storage resource

```
Storage {
  Name = FileBackup
  Address = bacula-server.lxd
  SDPort = 9103
  # For this password, use:
  # sudo grep ^SDPASSWD /etc/bacula/common_default_passwords
  Password = "<SDPASSWD value>"
  Device = FileBackup
  Media Type = File
}
```

-
-
-
-
-

Tip:

For more details about all the options of the `Pool` resource, please check the upstream [Storage Resource](https://www.bacula.org/15.0.x-manuals/en/main/Configuring_Director.html#blb:StorageResource)⁴⁴⁶ documentation.

⁴⁴⁶ https://www.bacula.org/15.0.x-manuals/en/main/Configuring_Director.html#blb:StorageResource

```
Storage {
  Name = bacula-server-sd
  SDPort = 9103
  WorkingDirectory = "/var/lib/bacula"
  Pid Directory = "/run/bacula"
  Plugin Directory = "/usr/lib/bacula"
  Maximum Concurrent Jobs = 20
  Encryption Command = "/etc/bacula/scripts/key-manager.py getkey"
  #SDAddress = 127.0.0.1
}
```

-

-

```
Device {  
  Name = FileBackup  
  Media Type = File  
  Archive Device = /storage/backups  
  Random Access = yes  
  Automatic Mount = yes  
  Removable Media = no  
  Always Open = no  
  Label Media = yes  
}
```

-

-

-

-

```
Director {  
  Name = bacula-server-dir  
  Password = "<randomly generated>"  
}
```

-

-

```
$ sudo systemctl restart bacula-sd.service
```

The Job resource

-
-
-
-

```
JobDefs {  
  Name = "DefaultJob"  
  Type = Backup  
  Level = Incremental  
  Client = bacula-server-fd  
  FileSet = "Home Set"  
  Schedule = "WeeklyCycle"  
  Storage = FileBackup  
  Messages = Standard  
  Pool = File  
  SpoolAttributes = yes  
  Priority = 10  
  Write Bootstrap = "/var/lib/bacula/%c.bsr"  
}
```

-
-
-
-
-
-

```
Job {  
  Name = "DirectorHomeBackup"  
  JobDefs = "DefaultJob"  
}
```

Tip:

A job has many attributes that can only be specified once. This means that jobs are pretty much specific to what is being backed up, and from where, among other things. It therefore helps to come up with a naming convention.

The upstream documentation has a section about [Naming Resources](https://docs.baculasystems.com/BEPlanningAndPreparation/BackupPolicy/NamingResources/index.html)⁴⁴⁷ with some suggestions.

⁴⁴⁷ <https://docs.baculasystems.com/BEPlanningAndPreparation/BackupPolicy/NamingResources/index.html>

```
Job {  
  Name = "RestoreFiles"  
  Type = Restore  
  Client = bacula-server-fd  
  Storage = FileBackup  
  # The FileSet and Pool directives are not used by Restore Jobs  
  # but must not be removed  
  FileSet = "Home Set"  
  Pool = File  
  Messages = Standard  
  Where = /storage/restore  
}
```

-
-
-
-
-
-

Tip:

For more details about all the options of the Job resource, please check the upstream [Job Resource](https://www.bacula.org/15.0.x-manuals/en/main/Configuring_Director.html#blb:JobResource)⁴⁴⁸ documentation.

⁴⁴⁸ https://www.bacula.org/15.0.x-manuals/en/main/Configuring_Director.html#blb:JobResource

Storage daemon

```
$ sudo mkdir -m 0700 /storage /storage/backups /storage/restore
```

```
$ sudo chown bacula: -R /storage
```

Tip:

For more details about the Storage daemon configuration options, please check the upstream [Storage Daemon](https://www.bacula.org/15.0.x-manuals/en/main/Storage_Daemon_Configuratio.html)⁴⁴⁹ documentation.

⁴⁴⁹ https://www.bacula.org/15.0.x-manuals/en/main/Storage_Daemon_Configuratio.html

File daemon

```
FileDaemon {  
    Name = bacula-server-fd  
    FDport = 9102  
    WorkingDirectory = /var/lib/bacula  
    Pid Directory = /run/bacula  
    Maximum Concurrent Jobs = 20  
    Plugin Directory = /usr/lib/bacula  
    #FDAddress = 127.0.0.1  
}
```

```
$ sudo systemctl restart bacula-fd.service
```


Tip:

For more details about the File daemon configuration, please check the upstream [File Daemon⁴⁵⁰](#) documentation.

⁴⁵⁰ https://www.bacula.org/15.0.x-manuals/en/main/Client_File_daemon_Configur.html

Console

tion⁴⁵¹

Console Configura-

```
$ sudo adduser <username> bacula
```

Warning:

Be mindful of who is added to the bacula group: members of this group are able to read all the data that is being backed up!

Cleaning up

-

-

-

⁴⁵¹ https://www.bacula.org/15.0.x-manuals/en/main/Console_Configuration.html

```
$ sudo systemctl restart bacula-dir.service bacula-fd.service
bacula-sd.service
```

Running a backup job

```
$ sudo bconsole
```

```
Connecting to Director localhost:9101
1000 OK: 10002 bacula-server-dir Version: 15.0.3 (25 March 2025)
Enter a period to cancel a command.
*
```

Command	Description
=====	=====
run	Run a job

Arguments:

```
job=<job-name> client=<client-name>
fileset=<FileSet-name> level=<level-keyword>
storage=<storage-name> where=<directory-prefix>
when=<universal-time-specification> pool=<pool-name>
nextpool=<next-pool-name> comment=<text> accurate=<bool> spooldata=<bool> yes
```

When at a prompt, entering a period cancels the command.

```
*run
Using Catalog "MyCatalog"
A job name must be specified.
The defined Job resources are:
    1: HomeBackup
    2: BackupCatalog
    3: RestoreFiles
Select Job resource (1-3): 1
Run Backup job
JobName: HomeBackup
Level: Incremental
Client: bacula-server-fd
FileSet: Home Set
Pool: File (From Job resource)
Storage: FileBackup (From Job resource)
When: 2025-10-20 20:21:03
Priority: 10
OK to run? (Yes/mod/no): yes
Job queued. JobId=7
```

-
-
-
-
-
-
-
-

```
+-----+-----+-----+-----+-----+-----+-----+
+-----+
| jobid | name      | starttime      | type | level | jobfiles | jobbytes |
| jobstatus |
+-----+-----+-----+-----+-----+-----+-----+
+-----+
|      7 | HomeBackup | 2025-10-20 20:21:11 | B    | I      |          0 |          0 | T
|
+-----+-----+-----+-----+-----+-----+-----+
+-----+
```

Tip:

For a list of status and error codes, check the upstream [Job status and Error codes tables](https://www.bacula.org/15.0.x-manuals/en/main/Job_status_Error_codes.html#blb:director:job:status)⁴⁵².

⁴⁵² https://www.bacula.org/15.0.x-manuals/en/main/Job_status_Error_codes.html#blb:director:job:status

```
+-----+
+-----+
| logtext
|
+-----+
+-----+
| bacula-server-dir JobId 7: Start Backup JobId 7, Job=HomeBackup.2025-10-20_20.21.
08_03
|
| bacula-server-dir JobId 7: Connected to Storage "FileBackup" at bacula-server.
lxd:9103 with TLS
|
| bacula-server-dir JobId 7: Using Device "FileBackup" to write.
|
...
Build OS:          x86_64-pc-linux-gnu ubuntu 25.10
JobId:             7
Job:               HomeBackup.2025-10-20_20.21.08_03
Backup Level:      Incremental, since=2025-10-20 18:11:00
Client:            "bacula-server-fd" 15.0.3 (25Mar25) x86_64-pc-linux-gnu,
ubuntu,25.10
FileSet:           "Home Set" 2025-10-20 16:27:31
Pool:              "File" (From Job resource)
Catalog:           "MyCatalog" (From Client resource)
Storage:           "FileBackup" (From Job resource)
...
Non-fatal FD errors: 0
SD Errors:          0
FD termination status: OK
SD termination status: OK
Termination:        Backup OK
|
...
```

```
-rw-r----- 1 bacula tape 345K Oct 20 20:21 /storage/backups/Vol-0001
```

Restoring a backup

First you select one or more JobIds that contain files to be restored. You will be presented several methods of specifying the JobIds. Then you will be allowed to select which files from those JobIds are to be restored.

To select the JobIds, you have the following choices:

1: List last 20 Jobs run

...

5: Select the most recent backup for a client

...

Select item: (1-14): 5

Automatically selected Client: bacula-server-fd

Automatically selected FileSet: Home Set

```
+-----+-----+-----+-----+-----+-----+
| jobid | level | jobfiles | jobbytes | starttime           | volumename |
+-----+-----+-----+-----+-----+-----+
|    4  | F    |    266  | 312,756 | 2025-10-20 17:34:43 | Vol-0001   |
|    5  | I    |     3   |    32   | 2025-10-20 18:11:00 | Vol-0001   |
+-----+-----+-----+-----+-----+-----+
```

You have selected the following JobIds: 4,5

...

You are now entering file selection mode where you add (mark) and remove (unmark) files to be restored. No files are initially added, unless you used the "all" keyword on the command line.

Enter "done" to leave this mode.

cwd is: /

\$

```
$ dir
```

```
drwxr-xr-x  1 root    root          12 2025-10-20 14:03:44 /home/
```

```
$ cd home/ubuntu
```

```
cwd is: /home/ubuntu/
```

```
$ dir
```

```
-rw----- 1 ubuntu ubuntu          32 2025-10-20 18:10:51 /home/ubuntu/.
bash_history
-rw-r--r-- 1 ubuntu ubuntu        220 2025-10-20 14:03:50 /home/ubuntu/.
bash_logout
-rw-r--r-- 1 ubuntu ubuntu       3830 2025-10-20 14:03:50 /home/ubuntu/.
bashrc
-rw-r--r-- 1 ubuntu ubuntu        807 2025-10-20 14:03:44 /home/ubuntu/.
profile
drwx----- 1 ubuntu ubuntu         30 2025-10-20 14:03:45 /home/ubuntu/.
ssh/
-rw-rw-r-- 1 ubuntu ubuntu         0 2025-10-20 18:10:50 /home/ubuntu/
this-is-on-the-server.txt
```

```
$ mark .tmux.conf
```

```
1 file marked.
```

```
$ done
```

```
Bootstrap records written to /var/lib/bacula/bacula-server-dir.restore.1.bsr
```

```
The Job will require the following (*=>InChanger):
```

Volume(s)	Storage(s)	SD Device(s)
=====		
Vol-0001	FileBackup	FileBackup

```
Volumes marked with "*" are in the Autochanger.
```

```
1 file selected to be restored.
```

```
Run Restore job
```

```
JobName:      RestoreFiles
Bootstrap:    /var/lib/bacula/bacula-server-dir.restore.1.bsr
Where:        /storage/restore
Replace:      Always
FileSet:      Home Set
Backup Client: bacula-server-fd
Restore Client: bacula-server-fd
Storage:      FileBackup
When:         2025-10-20 21:02:37
Catalog:      MyCatalog
Priority:      10
Plugin Options: *None*
OK to run? (Yes/mod/no):
```

```
OK to run? (Yes/mod/no): yes
Job queued. JobId=8
*
```

```
-rw-r--r-- 1 ubuntu ubuntu 2.4K Oct 20 14:03 /storage/restore/home/ubuntu/.tmux.conf
```

```
OK to run? (Yes/mod/no): mod
Parameters to modify:
  1: Level
  2: Storage
  3: Job
  4: FileSet
  5: Restore Client
  6: When
  7: Priority
  8: Bootstrap
  9: Where
 10: File Relocation
 11: Replace
 12: JobId
 13: Plugin Options
Select parameter to modify (1-13): 9
Please enter the full path prefix for restore (/ for none): /
Run Restore job
JobName:      RestoreFiles
Bootstrap:    /var/lib/bacula/bacula-server-dir.restore.2.bsr
Where:
Replace:      Always
FileSet:      Home Set
Backup Client: bacula-server-fd
Restore Client: bacula-server-fd
Storage:      FileBackup
When:         2025-10-20 21:11:55
Catalog:      MyCatalog
Priority:      10
Plugin Options: *None*
OK to run? (Yes/mod/no): yes
Job queued. JobId=9
```


Adding a client

```
$ sudo apt install bacula-fd
```

```
Director {  
    Name = bacula-server-dir # same as Director's Name on the Director server  
    Password = "<randomly generated>"  
}
```

-
-

```
FileDaemon {  
    Name = workstation1-fd  
    FDport = 9102          # where we listen for the director  
    WorkingDirectory = /var/lib/bacula  
    Pid Directory = /run/bacula  
    Maximum Concurrent Jobs = 20  
    Plugin Directory = /usr/lib/bacula  
    #FDAddress = 127.0.0.1 # default is to listen on all interfaces  
}
```

```
Messages {  
    Name = Standard  
    director = bacula-server-dir = all, !skipped, !restored, !verified, !saved  
}
```

```
$ sudo systemctl restart bacula-fd.service
```

```
Client {
    Name = workstation1-fd
    Address = workstation1.lxd
    FdPort = 9102
    Catalog = MyCatalog
    Password = "<to be filled in>" # password from bacula-fd.conf on workstation1-fd
    File Retention = 60 days
    Job Retention = 6 months
    AutoPrune = yes
}
```

-
-
-

```
Job {
    Name = "BackupWorkstation"
    JobDefs = "DefaultJob"
    Client = workstation1-fd
}
```

```
$ sudo systemctl restart bacula-dir.service
```

```
*list clients
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"
+-----+-----+-----+-----+
| clientid | name           | fileretention | jobretention |
+-----+-----+-----+-----+
| 1 | bacula-server-fd | 5,184,000 | 15,552,000 |
| 2 | workstation1-fd  | 5,184,000 | 15,552,000 |
+-----+-----+-----+-----+
```

```
*run
Using Catalog "MyCatalog"
A job name must be specified.
The defined Job resources are:
    1: HomeBackup
    2: BackupWorkstation
    3: BackupCatalog
    4: RestoreFiles
Select Job resource (1-4): 2
Run Backup job
JobName: BackupWorkstation
Level: Incremental
Client: workstation1-fd
FileSet: Home Set
Pool: File (From Job resource)
Storage: FileBackup (From Job resource)
When: 2025-10-21 21:02:48
Priority: 10
OK to run? (Yes/mod/no): yes
Job queued. JobId=14
You have messages.
```

```
*restore
...
    5: Select the most recent backup for a client
...
Select item: (1-14): 5
Defined Clients:
    1: bacula-server-fd
    2: workstation1-fd
Select the Client (1-2): 2
...
$ cd /home/ubuntu
cwd is: /home/ubuntu/
$ dir this*
-rw-rw-r-- 1 ubuntu ubuntu      0 2025-10-21 21:02:08 /home/ubuntu/
this-is-workstation1.txt
```

Further reading

- [Bacula documentation](#)⁴⁵³
- [Bacula home page](#)⁴⁵⁴

⁴⁵³ <https://www.bacula.org/documentation/documentation/>

⁴⁵⁴ <https://www.bacula.org/>

How to install and configure rsnapshot

`rsnapshot`⁴⁵⁵ *filesystem*

Install rsnapshot

```
$ sudo apt-get install rsnapshot
```

OpenSSH documentation

Configure rsnapshot

```
snapshot_root    /var/cache/rsnapshot/
```

Number of backups to keep

```
retain daily 6
retain weekly 7
retain monthly 4
```

⁴⁵⁵ <https://rsnapshot.org/>

Remote machine access

```
ssh_args      -p 22222
```

What to backup

```
backup /home/      localhost/  
backup /etc/       localhost/  
backup /usr/local/ localhost/
```

```
backup root@example.com:/home/ example.com/ +rsync_long_args=-bwlimit=16,  
exclude=core  
backup root@example.com:/etc/   example.com/ exclude=mtab,exclude=core
```

*rsnapshot(1)*⁴⁵⁶

Test configuration

```
$ sudo rsnapshot configtest
```

```
$ sudo rsnapshot -t daily
```

```
$ sudo rsnapshot daily
```

⁴⁵⁶ <https://manpages.ubuntu.com/manpages/resolute/man1/rsnapshot.1.html>

Scheduling backups

```
0 4 * * *      root    /usr/bin/rsnapshot daily
0 3 * * 1      root    /usr/bin/rsnapshot weekly
0 2 1 * *      root    /usr/bin/rsnapshot monthly
```

-
-
-

Backups - Shell Scripts

Further reading

- [rsnapshot official web page](#)⁴⁵⁷
- *rsnapshot(1)*⁴⁵⁸
- *rsync(1)*⁴⁵⁹

Shell scripts

- *how to use shell scripts for backups*
- *An example of archive rotation with shell scripts*

⁴⁵⁷ <https://rsnapshot.org/>

⁴⁵⁸ <https://manpages.ubuntu.com/manpages/resolute/man1/rsnapshot.1.html>

⁴⁵⁹ <https://manpages.ubuntu.com/manpages/resolute/man1/rsync.1.html>

How to back up using shell scripts

Network File System (NFS)

The shell script

Advanced Bash-Scripting Guide⁴⁶⁰

-

-

-

-

-

Network File System (NFS)

-

-

-

-

-

-

⁴⁶⁰ <https://tldp.org/LDP/abs/html/>

```
#!/bin/bash
#####
#
# Backup to NFS mount script.
#
#####

# What to backup.
backup_files="/home /var/spool/mail /etc /root /boot /opt"

# Where to backup to.
dest="/mnt/backup"

# Create archive filename.
day=$(date +%A)
hostname=$(hostname -s)
archive_file="$hostname-$day.tgz"

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"
date
echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
echo
echo "Backup finished"
date

# Long listing of files in $dest to check file sizes.
ls -lh $dest
```

Running the script

Run from a terminal

```
$ chmod u+x backup.sh
```

```
$ sudo ./backup.sh
```


Run with cron

```
# m h dom mon dow  command
```

-
-
-
-
-
-

```
$ sudo crontab -e
```

Note:

Using `sudo` with the `crontab -e` command edits the `root` user's crontab. This is necessary if you are backing up directories only the root user has access to.

```
# m h dom mon dow  command
0 0 * * * bash /usr/local/bin/backup.sh
```

Note:

The `backup.sh` script will need to be copied to the `/usr/local/bin/` directory in order for this entry to run properly. The script can reside anywhere on the file system, simply change the script path appropriately.

Restoring from the archive

-

```
$ tar -tzvf /mnt/backup/host-Monday.tgz
```

-

```
$ tar -xzf /mnt/backup/host-Monday.tgz -C /tmp etc/hosts
```

-

```
$ cd / > sudo tar -xzf /mnt/backup/host-Monday.tgz
```

Note:

This will overwrite the files currently on the file system.

Archive rotation shell script

simple backup shell script

Rotating NFS archives

-

-

-

```
#!/bin/bash
#####
#
# Backup to NFS mount script with
# grandparent-parent-child rotation.
#
#####

# What to backup.
backup_files="/home /var/spool/mail /etc /root /boot /opt"

# Where to backup to.
dest="/mnt/backup"

# Setup variables for the archive filename.
day=$(date +%A)
hostname=$(hostname -s)

# Find which week of the month 1-4 it is.
day_num=$(date +%d)
if (( $day_num <= 7 )) then
    week_file="$hostname-week1.tgz"
elif (( $day_num > 7 && $day_num <= 14 )) then
    week_file="$hostname-week2.tgz"
elif (( $day_num > 14 && $day_num <= 21 )) then
    week_file="$hostname-week3.tgz"
elif (( $day_num > 21 && $day_num < 32 )) then
    week_file="$hostname-week4.tgz"
fi

# Find if the Month is odd or even.
month_num=$(date +%m)
month=$(expr $month_num % 2)
if [ $month -eq 0 ]; then
    month_file="$hostname-month2.tgz"
else
    month_file="$hostname-month1.tgz"
fi

# Create archive filename.
if [ $day_num == 1 ]; then
    archive_file=$month_file
elif [ $day != "Saturday" ]; then
    archive_file="$hostname-$day.tgz"
else
    archive_file=$week_file
fi

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"
date
echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
```

(continues on next page)

(continued from previous page)

```
echo
echo "Backup finished"
date

# Long listing of files in $dest to check file sizes.
ls -lh $dest/
```

Running the script

Backing up to tape drives

```
#!/bin/bash
#####
#
# Backup to tape drive script.
#
#####

# What to backup.
backup_files="/home /var/spool/mail /etc /root /boot /opt"

# Where to backup to.
dest="/dev/st0"

# Print start status message.
echo "Backing up $backup_files to $dest"
date
echo

# Make sure the tape is rewound.
mt -f $dest rewind

# Backup the files using tar.
tar czf $dest $backup_files

# Rewind and eject the tape.
mt -f $dest rewoffl

# Print end status message.
echo
echo "Backup finished"
date
```

Note:

The default device name for a SCSI tape drive is `/dev/st0`. Use the appropriate device path for your system.

```
$ mt -f /dev/st0 rewind
```

```
$ tar -xzf /dev/st0 -C /tmp etc/hosts
```

Further reading

- [Advanced Bash-Scripting Guide](#)⁴⁶¹
- [cron how-to wiki page](#)⁴⁶²
- [GNU tar manual](#)⁴⁶³
- [Backup Rotation Scheme](#)⁴⁶⁴
- - [cpio](#)⁴⁶⁵
 - [dd](#)⁴⁶⁶
 - [rsnapshot](#)⁴⁶⁷
Tools - rsnapshot
 - [rsync\(1\)](#)⁴⁶⁸

⁴⁶¹ <https://tldp.org/LDP/abs/html/>

⁴⁶² <https://help.ubuntu.com/community/CronHowto>

⁴⁶³ <http://www.gnu.org/software/tar/manual/index.html>

⁴⁶⁴ http://en.wikipedia.org/wiki/Backup_rotation_scheme

⁴⁶⁵ <http://www.gnu.org/software/cpio/>

⁴⁶⁶ <http://www.gnu.org/software/coreutils/>

⁴⁶⁷ <https://rsnapshot.org/>

⁴⁶⁸ <https://manpages.ubuntu.com/manpages/resolute/man1/rsync.1.html>

Version control

- *etckeeper*
- *Install gitolite*

etckeeper

etckeeper⁴⁶⁹

Install etckeeper

```
$ sudo apt install etckeeper
```

Initialize etckeeper

Note:

It is possible to undo this by entering the following command:

```
$ sudo etckeeper uninit
```

⁴⁶⁹ <https://etckeeper.branchable.com/>

Configure autocommit frequency

```
$ sudo etckeeper commit "Reason for configuration change"
```

```
$ sudo etckeeper vcs log /etc/passwd
```

```
$ sudo apt install postfix
```

```
[master 5a16a0d] committing changes in /etc made by "apt install postfix"
Author: Your Name <xyz@example.com>
36 files changed, 2987 insertions(+), 4 deletions(-)
create mode 100755 init.d/postfix
create mode 100644 insserv.conf.d/postfix
create mode 100755 network/if-down.d/postfix
create mode 100755 network/if-up.d/postfix
create mode 100644 postfix/dynamicmaps.cf
create mode 100644 postfix/main.cf
create mode 100644 postfix/main.cf.proto
create mode 120000 postfix/makedefs.out
create mode 100644 postfix/master.cf
create mode 100644 postfix/master.cf.proto
create mode 100755 postfix/post-install
create mode 100644 postfix/postfix-files
create mode 100755 postfix/postfix-script
create mode 100755 ppp/ip-down.d/postfix
create mode 100755 ppp/ip-up.d/postfix
create mode 120000 rc0.d/K01postfix
create mode 120000 rc1.d/K01postfix
create mode 120000 rc2.d/S01postfix
create mode 120000 rc3.d/S01postfix
create mode 120000 rc4.d/S01postfix
create mode 120000 rc5.d/S01postfix
create mode 120000 rc6.d/K01postfix
create mode 100755 resolvconf/update-libc.d/postfix
create mode 100644 rsyslog.d/postfix.conf
create mode 120000 systemd/system/multi-user.target.wants/postfix.service
create mode 100644 ufw/applications.d/postfix
```

```
$ sudo etckeeper vcs status
```

```
$ sudo etckeeper vcs diff
```

```
$ sudo etckeeper commit "added new host"
```

Further reading

- [etckeeper](https://etckeeper.branchable.com/)⁴⁷⁰
- [the Git website](https://git-scm.com/)⁴⁷¹

How to set up Gitolite

Gitolite

*"bare git"*⁴⁷²

SSH-key

Install a Gitolite server

```
$ sudo apt install gitolite3
```

⁴⁷⁰ <https://etckeeper.branchable.com/>

⁴⁷¹ <https://git-scm.com/>

⁴⁷² <https://git-scm.com/docs/gitglossary#Documentation/gitglossary.txt-aiddefbarerepositoryabarerepository>

Configure Gitolite

```
$ sudo useradd --system --home /home/git --create-home git
```

SSH-key
SSH keys in our OpenSSH

guide

SSH-key

```
$ cp ~/.ssh/id_ed25519.pub /tmp/admin.pub
```

```
$ sudo -i -u git gitolite setup -pk /tmp/admin.pub
```

```
•  
•  
•  
  -  
•  
•
```

```
$ ssh git@$yourserver  
  
hello admin, this is git@your-gitolite-server running gitolite3  
  
R W      gitolite-admin  
R W      testing
```

Managing Gitolite users and repositories

```
$ git clone git@$yourserver:gitolite-admin.git
```

```
$ git commit
```

```
$ git push
```

-
-

Adding a Gitolite user

nsswitch

Deleting a Gitolite user

Managing repositories

Example configuration

```
# Gitolite config
# Users are created by having their public key in keydir/$username.pub

# Group creation
@bestproject      = name1 name2
@projectwatchers  = name3 @bestproject

# This repo itself
repo    gitolite-admin
RW+     =   admin
R       =   alice

# A repo with access to anybody
repo    testing
RW+     = @all

# A repo with special privileges, to tags and branches
repo    some/awesome/project
RW      =   alice @bestproject
RW+     =   bob
RW+    dev/ = @bestproject
R       = @projectwatchers carol
# bestproject members and alice can push code (but not force-push)
# bestproject members can force-push branches starting with dev/
# bob can forcepush anything
# projectwatchers and carol have readonly access
```

page 1⁴⁷³

page 2⁴⁷⁴

Using your server

```
$ git clone git@$server:some/awesome/project.git
```

```
$ git remote add $somename git@$server:some/awesome/project.git
```

⁴⁷³ <https://gitolite.com/gitolite/conf.html>

⁴⁷⁴ <https://gitolite.com/gitolite/conf-2.html>

Further reading

- [Gitolite's code repository](#)⁴⁷⁵
- [Gitolite's documentation](#)⁴⁷⁶
- [Gitolite Essentials](#)⁴⁷⁷
- [Git homepage](#)⁴⁷⁸

See also

- [Backup and restore OpenLDAP](#)

See also

- [Data and storage](#)
- [Mail services](#)
-
-
-

Mail services

- 1.
- 2.
- 3.
- 4.

⁴⁷⁵ <https://github.com/sitaramc/gitolite>

⁴⁷⁶ <https://gitolite.com/gitolite/>

⁴⁷⁷ <https://www.packtpub.com/en-us/product/gitolite-essentials-9781783282388>

⁴⁷⁸ <https://git-scm.com/>

Mail User Agent

Thunderbird⁴⁷⁹

this short guide⁴⁸⁰

Mail Transfer Agent

Postfix⁴⁸¹

sendmail⁴⁸²

- *Install Postfix*

Exim4⁴⁸³

- *Install Exim4*

Install and configure Postfix

Note:

This guide does not cover setting up Postfix *Virtual Domains*. For information on Virtual Domains and other advanced configurations see the references list at the end of this page.

Install Postfix

Postfix⁴⁸⁴

```
$ sudo apt install postfix
```

Warning:

The mail-stack-delivery metapackage has been deprecated in Focal. The package still exists for compatibility reasons, but won't setup a working email system.

⁴⁷⁹ <https://www.thunderbird.net/>

⁴⁸⁰ <https://snapcraft.io/install/thunderbird/ubuntu>

⁴⁸¹ <https://www.postfix.org/>

⁴⁸² <https://www.authsmtp.com/sendmail/index.html>

⁴⁸³ <https://www.exim.org/>

⁴⁸⁴ <https://www.postfix.org/>

Configure Postfix

-
-
-
-

```
$ sudo dpkg-reconfigure postfix
```

-
-
-
-
-
-
-
-
-

Configure mailbox format

```
$ sudo postconf -e 'home_mailbox = Maildir/'
```

SMTP authentication

Configure SMTP authentication

```
$ sudo postconf -e 'smtpd_sasl_type = dovecot' > sudo postconf -e  
'smtpd_sasl_path = private/auth' > sudo postconf -e 'smtpd_sasl_local_domain  
=' > sudo postconf -e 'smtpd_sasl_security_options = noanonymous' > sudo  
postconf -e 'broken_sasl_auth_clients = yes' > sudo postconf -e  
'smtpd_sasl_auth_enable = yes' > sudo postconf -e  
'smtpd_recipient_restrictions = \ >  
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
```

Note:

The `smtpd_sasl_path` config parameter is a path relative to the Postfix queue directory.

Configure TLS

Obtain TLS certificates

```
$ sudo postconf -e 'smtp_tls_security_level = may' > sudo postconf -e  
'smtpd_tls_security_level = may' > sudo postconf -e  
'smtp_tls_note_starttls_offer = yes' > sudo postconf -e  
'smtpd_tls_chain_files =  
/etc/ssl/private/server.key,/etc/ssl/certs/server.crt' > sudo postconf -e
```

```
'smtpd_tls_loglevel = 1' > sudo postconf -e 'smtpd_tls_received_header = yes'  
> sudo postconf -e 'myhostname = mail.example.com'
```

```
$ sudo postconf -e 'smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem'
```

Obtain TLS certificates

Outcome of initial configuration

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete  
# version  
  
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)  
biff = no  
  
# appending .domain is the MUA's job.  
append_dot_mydomain = no  
  
# Uncomment the next line to generate "delayed mail" warnings  
#delay_warning_time = 4h  
  
myhostname = server1.example.com  
alias_maps = hash:/etc/aliases  
alias_database = hash:/etc/aliases  
myorigin = /etc/mailname  
mydestination = server1.example.com, localhost.example.com, localhost  
relayhost =  
mynetworks = 127.0.0.0/8  
mailbox_command = procmail -a "$EXTENSION"  
mailbox_size_limit = 0  
recipient_delimiter = +  
inet_interfaces = all  
smtpd_sasl_local_domain =  
smtpd_sasl_auth_enable = yes  
smtpd_sasl_security_options = noanonymous  
broken_sasl_auth_clients = yes  
smtpd_recipient_restrictions =  
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination  
smtpd_tls_auth_only = no  
smtpd_tls_security_level = may  
smtpd_tls_security_level = may  
smtpd_tls_note_starttls_offer = yes  
smtpd_tls_key_file = /etc/ssl/private/smtpd.key  
smtpd_tls_cert_file = /etc/ssl/certs/smtpd.crt  
smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem  
smtpd_tls_loglevel = 1  
smtpd_tls_received_header = yes
```

(continues on next page)

(continued from previous page)

```
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
```

```
$ sudo systemctl restart postfix.service
```

SASL

RFC2554⁴⁸⁵

SASL⁴⁸⁶

```
mynetworks = 127.0.0.0/8, [::1]/128
```

Configure SASL

```
$ sudo apt install dovecot-core
```

```
service auth {
    # auth_socket_path points to this userdb socket by default. It's typically
    # used by dovecot-lda, doveadm, possibly imap process, etc. Its default
    # permissions make it readable only by root, but you may need to relax these
    # permissions. Users that have access to this socket are able to get a list
    # of all usernames and get results of everyone's userdb lookups.
    unix_listener auth-userdb {
        #mode = 0600
        #user =
        #group =
    }

    # Postfix smtp-auth
    unix_listener /var/spool/postfix/private/auth {
        mode = 0660
        user = postfix
        group = postfix
    }
}
```

⁴⁸⁵ <https://www.ietf.org/rfc/rfc2554.txt>

⁴⁸⁶ <https://www.ietf.org/rfc/rfc2222.txt>

```
auth_mechanisms = plain
```

```
auth_mechanisms = plain login
```

```
$ sudo systemctl restart dovecot.service
```

Test your setup

```
$ telnet mail.example.com 25
```

```
$ ehlo mail.example.com
```

```
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250 8BITMIME
```

Troubleshooting

Escaping chroot

```
smtp      inet  n       -       -       -       smtpd
```

```
smtp      inet  n       -       n       -       smtpd
```

```
$ sudo service postfix restart
```

SMTPS

```
smtps      inet  n       -       -       -       smtpd
-o smtpd_tls_wrappermode=yes
-o smtpd_sasl_auth_enable=yes
-o smtpd_client_restrictions=permit_sasl_authenticated,reject
-o milter_macro_daemon_name=ORIGINATING
```

Log viewing

```
$ tail -f /var/log/mail.err
```

Increase logging detail

```
$ sudo postconf -e 'smtpd_tls_loglevel = 4'
```

```
$ sudo systemctl reload postfix.service
```

Logging mail delivery

```
$ sudo postconf -e 'debug_peer_list = problem.domain'
```

```
$ sudo systemctl reload postfix.service
```

Increase daemon verbosity

```
smtp      unix  -      -      -      -      -      smtp -v
```

```
$ sudo systemctl reload postfix.service
```

Log SASL debug info

```
auth_debug=yes  
auth_debug_passwords=yes
```

```
$ sudo systemctl reload dovecot.service
```

Note:

Some of the options above can drastically increase the amount of information sent to the log files. Remember to return the log level back to normal after you have corrected the problem – then reload the appropriate daemon for the new configuration to take effect.

Further reading

- [Postfix website](http://www.postfix.org/documentation.html)⁴⁸⁷
- [Postfix: The Definitive Guide](https://www.oreilly.com/library/view/postfix-the-definitive/0596002122/)⁴⁸⁸
- [Ubuntu Wiki Postfix](https://help.ubuntu.com/community/Postfix)⁴⁸⁹
- [Debian Wiki Postfix](https://wiki.debian.org/Postfix)⁴⁹⁰
[Postfix Tutorials](https://wiki.debian.org/Postfix/Tutorials)⁴⁹¹
- [set up mailman3 with postfix](https://docs.mailman3.org/projects/mailman/en/latest/src/mailman/docs/mta.html)⁴⁹²

⁴⁸⁷ <http://www.postfix.org/documentation.html>

⁴⁸⁸ <https://www.oreilly.com/library/view/postfix-the-definitive/0596002122/>

⁴⁸⁹ <https://help.ubuntu.com/community/Postfix>

⁴⁹⁰ <https://wiki.debian.org/Postfix>

⁴⁹¹ <https://wiki.debian.org/Postfix/Tutorials>

⁴⁹² <https://docs.mailman3.org/projects/mailman/en/latest/src/mailman/docs/mta.html>

Install and configure Exim4

Install Exim4

Exim4⁴⁹³

```
$ sudo apt install exim4
```

Configure Exim4

```
$ sudo dpkg-reconfigure exim4-config
```

Note:

The default configuration layout for Exim4 is the single configuration file one.

```
/etc/exim4/  
├── conf.d  
│   ├── acl  
│   ├── auth  
│   ├── main  
│   ├── retry  
│   ├── rewrite  
│   ├── router  
│   └── transport
```

```
$ sudo update-exim4.conf
```

⁴⁹³ <https://www.exim.org/>

Warning:

You should never manually edit the configuration file `/var/lib/exim4/config.autogenerated`, because it is updated automatically every time you run `update-exim4.conf`. Any changes you make to it will eventually be lost.

```
$ sudo systemctl restart exim4
```

Start the Exim4 daemon

```
$ sudo service exim4 start
```

SMTP authentication

-
-

Warning:

All configuration steps shown from now on will assume a split-configuration mode for Exim4. If you have selected the non-split mode, then all commands that edit a configuration file under `/etc/exim4/conf.d` in the sections below should be replaced with editing the single file `/etc/exim4/exim4.conf.template`.

Enabling TLS

```
$ sudo /usr/share/doc/exim4-base/examples/exim-gencert
```

Warning:

This will install a self-signed certificate. If deploying this system in production, obtain a trusted certificate instead. See [Obtain TLS certificates](#) (page 245) for how to get a certificate, or how to set up your own Certificate Authority (CA) for internal deployments.

```
MAIN_TLS_ENABLE = yes
```

Authenticating existing Linux users

Note:

To manage local Linux users, please refer to [User management](#) (page 154).

```
plain_saslauthd_server:
  driver = plaintext
  public_name = PLAIN
  server_condition = ${if saslauthd{${auth2}{${auth3}}}{1}{0}}
  server_set_id = $auth2
  server_prompts = :
  .ifndef AUTH_SERVER_ALLOW_NOTLS_PASSWORDS
  server_advertise_condition = ${if eq{${tls_cipher}{}{}}{}}{*}}
  .endif

login_saslauthd_server:
  driver = plaintext
  public_name = LOGIN
  server_prompts = "Username:: : Password::"
  # don't send system passwords over unencrypted connections
  server_condition = ${if saslauthd{${auth1}{${auth2}}}{1}{0}}
```

(continues on next page)

(continued from previous page)

```
server_set_id = $auth1
.ifdef AUTH_SERVER_ALLOW_NOTLS_PASSWORDS
server_advertise_condition = ${if eq{$tls_cipher}{*}}{*}}
.endif
```

```
$ sudo gpasswd -a Debian-exim sasl
```

```
$ sudo update-exim4.conf
```

```
$ sudo systemctl restart exim4
```

```
$ sudo apt install sasl2-bin
```

```
MECHANISMS="pam"
```

Note:

In Ubuntu 22.04 Jammy and earlier, we also need to add `START="yes"` to `/etc/default/saslauthd`.

```
$ sudo systemctl enable saslauthd
```

```
$ sudo systemctl start saslauthd
```


Authenticating arbitrary users

```
plain_server:
  driver = plaintext
  public_name = PLAIN
  server_condition = "${if crypteq[$auth3]{$extract[1]:}{$lookup[$auth2]lsearch
[CONFDIR/passwd]{$value}{*:*)}}}{1}{0}}"
  server_set_id = $auth2
  server_prompts = :
  .ifndef AUTH_SERVER_ALLOW_NOTLS_PASSWORDS
  server_advertise_condition = ${if eq{$tls_in_cipher}{}}{*}}
  .endif

login_server:
  driver = plaintext
  public_name = LOGIN
  server_prompts = "Username:: : Password::"
  server_condition = "${if crypteq[$auth2]{$extract[1]:}{$lookup[$auth1]lsearch
[CONFDIR/passwd]{$value}{*:*)}}}{1}{0}}"
  server_set_id = $auth1
  .ifndef AUTH_SERVER_ALLOW_NOTLS_PASSWORDS
  server_advertise_condition = ${if eq{$tls_in_cipher}{}}{*}}
  .endif
```

Warning:

DO NOT enable both these and the `_saslauthd_server` variants (from “Authenticating existing Linux users” above) at the same time!

```
username:crypted-password:cleartext-password
```

```
$ sudo /usr/share/doc/exim4-base/examples/exim-adduser
```

```
User: ubuntu
Password: ubuntusecret
```

```
ubuntu:$1$ZvPA$HTddFobmJD1vURtJHBmbw/:ubuntusecret
```

```
$ sudo chown root:Debian-exim /etc/exim4/passwd
```

```
$ sudo chmod 0640 /etc/exim4/passwd
```

-
-
-

Warning:

The `/usr/share/doc/exim4-base/examples/exim-adduser` serves mostly as an example and is not able to handle many scenarios. For example, it won't check if the username you are providing already exists in the `passwd` file, which can lead to multiple entries for the same user, with unpredictable results.

```
$ sudo update-exim4.conf
```

```
$ sudo systemctl restart exim4
```

Note:

There is no need to restart Exim4 after making changes to the `/etc/exim4/passwd` file.

Troubleshooting

```
$ testsaslauthd -u ubuntu -p ubuntusecret
```

```
0: OK "Success."
```

```
$ sudo apt install cyrus-clients --no-install-recommends
```

*cyrus-smtpdtest(1)*⁴⁹⁴

```
/usr/lib/cyrus/bin/smtpdtest -t "" -a ubuntu -w ubuntusecret n-exim.lxd
```

-
-
-
-

⁴⁹⁴ <https://manpages.ubuntu.com/manpages/resolute/man1/cyrus-smtpdtest.1.html>

```
S: 220 n-exim ESMTP Exim 4.97 Ubuntu Mon, 23 Jun 2025 21:11:59 +0000
C: EHLO smtptest
S: 250-n-exim Hello n-exim.lxd [10.10.17.9]
S: 250-SIZE 52428800
S: 250-8BITMIME
S: 250-PIPELINING
S: 250-PIPECONNECT
S: 250-CHUNKING
S: 250-STARTTLS
S: 250-PRDR
S: 250 HELP
C: STARTTLS
S: 220 TLS go ahead
verify error:num=18:self-signed certificate
TLS connection established: TLSv1.3 with cipher TLS_AES_256_GCM_SHA384 (256/256
bits)
C: EHLO smtptest
S: 250-n-exim Hello n-exim.lxd [10.10.17.9]
S: 250-SIZE 52428800
S: 250-8BITMIME
S: 250-PIPELINING
S: 250-PIPECONNECT
S: 250-AUTH PLAIN LOGIN
S: 250-CHUNKING
S: 250-PRDR
S: 250 HELP
C: AUTH LOGIN
S: 334 VXNlcm5hbWU6
C: dWJ1bnR1
S: 334 UGFzc3dvcmQ6
C: dWJ1bnR1c2VjcmV0
S: 235 Authentication succeeded
Authenticated.
Security strength factor: 256
```

-
-
-
-

Tip:

Want to obtain the original username and password back from the base64 encoded values? Feed those values to the base64 -d tool. Example, using the value from the session above:

```
$ echo -n dWJ1bnR1c2Vjc2V0 | base64 -d; echo
```

```
ubuntusecret
```

To test the PLAIN mechanism, add the -m plain command-line option:

```
/usr/lib/cyrus/bin/smtptest -t "" -a ubuntu -w ubuntusecret -m plain n-exim.lxd
```

In the new output, PLAIN was selected:

```
S: 220 n-exim ESMTP Exim 4.97 Ubuntu Mon, 23 Jun 2025 21:15:39 +0000
C: EHLO smtptest
S: 250-n-exim Hello n-exim.lxd [10.10.17.9]
S: 250-SIZE 52428800
S: 250-8BITMIME
S: 250-PIPELINING
S: 250-PIPECONNECT
S: 250-CHUNKING
S: 250-STARTTLS
S: 250-PRDR
S: 250 HELP
C: STARTTLS
S: 220 TLS go ahead
verify error:num=18:self-signed certificate
TLS connection established: TLSv1.3 with cipher TLS_AES_256_GCM_SHA384 (256/256
bits)
C: EHLO smtptest
S: 250-n-exim Hello n-exim.lxd [10.10.17.9]
S: 250-SIZE 52428800
S: 250-8BITMIME
S: 250-PIPELINING
S: 250-PIPECONNECT
S: 250-AUTH PLAIN LOGIN
S: 250-CHUNKING
S: 250-PRDR
S: 250 HELP
C: AUTH PLAIN AHVidW50dQB1YnVudHVzZWNYZXQ=
S: 235 Authentication succeeded
Authenticated.
Security strength factor: 256
```

Troubleshooting tips

Here are some troubleshooting tips.

Permissions

- If using saslauthd: Can the Debian-exim user read and write to the saslauthd socket in /run/saslauthd/mux socket?
- If using /etc/exim4/passwd: Can the Debian-exim user read this file?

580 of 1096

Config

Mail Delivery Agent

Dovecot⁴⁹⁸
Maildir⁵⁰⁰

mbx⁴⁹⁹

- *Install Dovecot*

Install and configure Dovecot

Install Dovecot

```
$ sudo apt install dovecot-imapd dovecot-pop3d
```

Configure Dovecot

```
!include_try /usr/share/dovecot/protocols.d/*.protocol
```

web site⁵⁰¹

the Dovecot

⁴⁹⁸ <https://dovecot.org/>

⁴⁹⁹ <https://en.wikipedia.org/wiki/Mbox>

⁵⁰⁰ <https://en.wikipedia.org/wiki/Maildir>

⁵⁰¹ <https://doc.dovecot.org/main/>

Restart the Dovecot daemon

```
$ sudo service dovecot restart
```

```
bhuvan@rainbow:~$ telnet localhost pop3
```

```
Trying 127.0.0.1...  
Connected to localhost.localdomain.  
Escape character is '^]'.  
+OK Dovecot ready.
```

Dovecot SSL configuration

```
$ sudo openssl req -new -x509 -days 1000 -nodes -out  
"/etc/dovecot/dovecot.pem" \ > -keyout "/etc/dovecot/private/dovecot.pem"
```

```
ssl_cert = </etc/dovecot/private/dovecot.pem  
ssl_key = </etc/dovecot/private/dovecot.key
```

See also

See [Obtain TLS certificates](#) (page 245) for how to get a trusted certificate or set up your own CA.

Configure a firewall for an email server

-
-
-
-

Further reading

- [Dovecot website](#)⁵⁰²
- [Dovecot manual](#)⁵⁰³
- [Dovecot Ubuntu Wiki](#)⁵⁰⁴

Web services section

-
-
-

Web services

Introduction to web services

Proxy servers

⁵⁰² <https://dovecot.org/>

⁵⁰³ <https://doc.dovecot.org/latest/>

⁵⁰⁴ <https://help.ubuntu.com/community/Dovecot>

How to install a Squid server

refer to this guide

Install Squid

```
$ sudo apt install squid
```

Configure Squid

Protect the original config file

```
$ sudo cp /etc/squid/squid.conf /etc/squid/squid.conf.original
```

```
$ sudo chmod a-w /etc/squid/squid.conf.original
```

Change TCP port

```
http_port 8888
```

Set the hostname

hostname

```
visible_hostname weezie
```

Configure the memory cache

```
cache_mem 512 MB  
memory_replacement_policy lru
```

Configure on-disk cache

```
cache_dir <Type> <Directory-Name> <Size-in-MB> <L1-Dirs> <L2-Dirs> [options]
```

```
cache_dir ufs /var/spool/squid 10000 16 256
```

-
-
-
-

Configure cached objects size limits

```
maximum_object_size 512 MB
minimum_object_size 0 KB
maximum_object_size_in_memory 512 KB
```

Configure cached objects lifetime

```
refresh_pattern regex min percent max [options]
```

```
refresh_pattern -i \.(gif|jpg|png|css|js)$ 10080 90% 43200
refresh_pattern . 0 20% 4320
```

Caching HTTPS content

[Squid HTTPS documentation](#)⁵⁰⁵

Other caching configuration options

[Squid configuration guide](#)⁵⁰⁶

⁵⁰⁵ <https://wiki.squid-cache.org/Features/HTTPS>

⁵⁰⁶ <https://www.squid-cache.org/Doc/config/>

Access control

- *ACL*

```
acl fortytwo_network src 192.168.42.0/24
```

-

```
http_access allow fortytwo_network
```

-

```
acl biz_network src 10.1.42.0/24  
acl biz_hours time M T W T F 9 00-17 00
```

-

```
http_access allow biz_network biz_hours
```

Restart the Squid server

```
$ sudo squid -k parse
```

```
$ sudo systemctl restart squid.service
```

Note:

If a formerly customized squid3 was used to set up the spool at `/var/log/squid3` to be a mount point, but otherwise kept the default configuration, the upgrade will fail. The upgrade tries to rename/move files as needed, but it can't do so for an active mount point. In that case you will need to adapt either the mount point or the config in `/etc/squid/squid.conf` so that they match. The same applies if the **include** config statement was used to pull in more files from the old path at `/etc/squid3/`. In those cases you should move and adapt your configuration accordingly.

Troubleshooting

```
$ sudo squid -v
```

```
$ sudo systemctl status squid
```

```
$ sudo cat /var/log/squid/cache.log
```

```
$ sudo cat /var/log/squid/access.log
```

```
$ curl http://localhost:8888/squid-internal-mgr/info
```

-
-
-
-
-

Further reading

- [The Squid Website](#)⁵⁰⁷
- [Ubuntu Wiki page on Squid](#)⁵⁰⁸

Web servers

Apache2

How to install Apache2

The Apache HTTP Server⁵⁰⁹

Install apache2

```
$ sudo apt install apache2
```

Configure apache2

Files

-

⁵⁰⁷ <http://www.squid-cache.org/>

⁵⁰⁸ <https://help.ubuntu.com/community/Squid>

⁵⁰⁹ <https://httpd.apache.org/>

Note:

Historically, the main Apache2 configuration file was `httpd.conf`, named after the “httpd” daemon. In other distributions (or older versions of Ubuntu), the file might be present. In modern releases of Ubuntu, all configuration options have been moved to `apache2.conf` and the below referenced directories and `httpd.conf` no longer exists.

-
-
-

Directories

-
-
-
-
-
-

Detailed configuration

- [*Apache2 configuration settings*](#)
- [*how to extend Apache2*](#)

Further reading

- [Apache2 Documentation⁵¹⁰](#)
- [Apache Cookbook⁵¹¹](#)

How to configure Apache2 settings

installed Apache2

Basic directives

Note:

The directives set for a virtual host only apply to that particular virtual host. If a directive is set server-wide and not defined in the virtual host settings, the default setting is used. For example, you can define a Webmaster email address and not define individual email addresses for each virtual host.

```
$ sudo cp /etc/apache2/sites-available/000-default.conf  
/etc/apache2/sites-available/mynewsite.conf
```

⁵¹⁰ <https://httpd.apache.org/docs/2.4/>

⁵¹¹ <https://www.oreilly.com/library/view/apache-cookbook-2nd/9780596529949/>

The ServerAdmin directive

Found in `/etc/apache2/sites-available`

The Listen directive

Found in `/etc/apache2/ports.conf`

-
-
-

The ServerName directive (optional)

FQDN

The ServerAlias directive

`ServerAlias *.mynewsite.com`

The DocumentRoot directive

```
$ sudo a2ensite mynewsite
```

```
$ sudo systemctl restart apache2.service
```

Note:

Be sure to replace `mynewsite` with a more descriptive name for the VirtualHost. One method is to name the file after the **ServerName** directive of the VirtualHost.

```
$ sudo a2dissite mynewsite
```

```
$ sudo systemctl restart apache2.service
```

Apache2 server default settings

The DirectoryIndex

The ErrorDocument

CustomLog and ErrorLog

The Options directive

```
<Directory /var/www/html/mynewsite>  
...  
</Directory>
```

-

Caution:

Most files should not be run as CGI scripts. This would be very dangerous. CGI scripts should be kept in a directory separate from and outside your **DocumentRoot**, and only this directory should have the ExecCGI option set. This is the default, and the default location for CGI scripts is `/usr/lib/cgi-bin`.

-

[Apache SSI documentation \(Ubuntu community\)](https://help.ubuntu.com/community/ServerSideIncludes)⁵¹²

-

⁵¹² <https://help.ubuntu.com/community/ServerSideIncludes>

-

Caution:

For security reasons, this should usually not be set, and certainly should not be set on your DocumentRoot directory. Enable this option carefully on a per-directory basis **only** if you are certain you want users to see the entire contents of the directory.

-

[Apache2 documentation on this option](#)⁵¹³

-

Apache2 daemon settings

-

-

-

Warning:

Unless you know exactly what you are doing, do not set the User directive to root. Using root as the User will create large security holes for your Web server.

-

⁵¹³ https://httpd.apache.org/docs/2.4/mod/mod_negotiation.html#multiviews

Extending Apache2

how to extend

Apache2

Further reading

- [Apache2 Documentation](#)⁵¹⁴
- [Apache Cookbook](#)⁵¹⁵

How to use Apache2 modules

Installing and handling modules

```
$ sudo apt install libapache2-mod-wsgi-py3
```

```
$ sudo a2dismod wsgi
```

```
$ sudo systemctl restart apache2.service
```

```
$ sudo a2enmod wsgi
```

⁵¹⁴ <https://httpd.apache.org/docs/2.4/>

⁵¹⁵ <https://www.oreilly.com/library/view/apache-cookbook-2nd/9780596529949/>

```
$ sudo systemctl restart apache2.service
```

Configure Apache2 for HTTPS

```
$ sudo a2enmod ssl
```

Note:

For more information on generating a key and obtaining a certificate see [Certificates](#) (page 901).

```
$ sudo a2ensite default-ssl
```

Note:

The directories `/etc/ssl/certs` and `/etc/ssl/private` are the default locations. If you install the certificate and key in another directory make sure to change `SSLCertificateFile` and `SSLCertificateKeyFile` appropriately.

```
$ sudo systemctl restart apache2.service
```

Sharing write permission

```
$ sudo chgrp -R webmasters /var/www/html
```

```
$ sudo chmod -R g=rwX /var/www/html/
```

Warning:

The apache2 daemon will run as the www-data user, which has a corresponding www-data group. These **should not** be granted write access to the document root, as this would mean that vulnerabilities in Apache or the applications it is serving would allow attackers to overwrite the served content.

Further reading

- [Apache2 Documentation](#)⁵¹⁶
- [Apache Cookbook](#)⁵¹⁷
- [Install Apache2](#)
- [Configure Apache2](#)
- [Extend Apache2 with modules](#)

Nginx

How to install nginx

Apache

⁵¹⁶ <https://httpd.apache.org/docs/2.4/>

⁵¹⁷ <http://shop.oreilly.com/product/9780596529949.do>

Install nginx

```
$ sudo apt update && sudo apt install nginx
```

Verify nginx is running

```
$ sudo systemctl status nginx

■ nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Sun 2023-08-20 01:04:22 UTC; 53s ago
     Docs: man:nginx(8)
   Process: 28210 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process
   on; (code=exited, status=0/SUCCESS)
   Process: 28211 ExecStart=/usr/sbin/nginx -g daemon on; master_process on;
   (code=exited, status=0/SUCCESS)
    Main PID: 28312 (nginx)
      Tasks: 13 (limit: 76969)
     Memory: 13.1M
        CPU: 105ms
    CGroup: /system.slice/nginx.service
            └─28312 "nginx: master process /usr/sbin/nginx -g daemon on; master_
process on;"
            └─28314 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" ""
" "" "" "" ""
...
```


Restarting nginx

```
$ sudo systemctl restart nginx
```

Enable/disable nginx manually

```
$ sudo systemctl disable nginx
```

```
$ sudo systemctl enable nginx
```

The default nginx homepage

```
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further
configuration is
required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.
```

Setting up nginx

- [How to configure nginx](#)
- [How to use nginx modules](#)

Further reading

- [nginx documentation](#)⁵¹⁸
-

How to configure nginx

installed nginx

Server blocks

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /var/www/html;  
  
    # Add index.php to the list if you are using PHP  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name _;  
  
    location / {  
        # First attempt to serve request as file, then  
        # as directory, then fall back to displaying a 404.  
        try_files $uri $uri/ =404;  
    }  
}
```

filesystem

```
server {  
    listen      80;  
    root        /srv/my-site/html;  
    index       index.html;  
    server_name my-site.org www.my-site.org;  
  
    location / {
```

(continues on next page)

⁵¹⁸ <https://nginx.org/en/docs/>

(continued from previous page)

```
        try_files $uri $uri/ =404;  
    }  
}
```

```
$ sudo mkdir -p /srv/my-site/html
```

```
$ sudo chmod -R 755 /srv/my-site/html
```

```
$ echo "<html><body><h1>My Site!</h1></body></html>" >  
/srv/my-site/html/index.html
```

```
$ sudo systemctl reload nginx
```

```
$ www-browser www.my-site.org
```

Multi-site hosting

```
$ sudo ln -s /etc/nginx/sites-available/<your-domain>  
/etc/nginx/sites-enabled/
```

```
$ sudo rm /etc/nginx/sites-enabled/default
```

SSL and HTTPS

```
server {  
    listen            80;  
    server_name       our-site.org www.our-site.org;  
    return            301 https://$host$request_uri;  
}  
  
server {  
    listen            443 ssl;  
  
    root              /srv/our-site/html;  
    index             index.html;  
  
    server_name       our-site.org www.our-site.org;  
  
    ssl_certificate    our-site.org.crt;  
    ssl_certificate_key our-site.org.key;  
    ssl_protocols     TLSv1 TLSv1.1 TLSv1.2 TLSv1.3;  
    ssl_ciphers       HIGH:!aNULL:!MD5;  
    ssl_session_timeout 15m;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

Obtain TLS certificates

OpenSSL

GnuTLS

Setting up nginx

- [How to use nginx modules](#)

Further reading

- [nginx's beginner's guide](#)⁵¹⁹ *FastCGI*
- [nginx documentation](#)⁵²⁰

How to use nginx modules

Available modules

```
$ nginx -V 2>&1 | tr -- - '\n' | grep _module
http_ssl_module
http_stub_status_module
http_realip_module
...
http_image_filter_module=dynamic
http_perl_module=dynamic
http_xslt_module=dynamic
stream_geoip_module=dynamic
```

```
$ apt policy libnginx-mod-http-image-filter
```

⁵¹⁹ https://nginx.org/en/docs/beginners_guide.html

⁵²⁰ https://nginx.org/en/docs/http/configuring_https_servers.html

```
libnginx-mod-http-image-filter:
  Installed: (none)
  Candidate: 1.24.0-1ubuntu1
  Version table:
    1.24.0-1ubuntu1 500
      500 http://archive.ubuntu.com/ubuntu mantic/main amd64 Packages
```

```
$ sudo apt install libnginx-mod-http-image-filter

...
The following NEW packages will be installed:
  libnginx-mod-http-image-filter
0 upgraded, 1 newly installed, 0 to remove and 34 not upgraded.
...
Triggering nginx reload
...
```

Enabling and disabling dynamic modules

```
$ ls /etc/nginx/modules-*

/etc/nginx/modules-available:

/etc/nginx/modules-enabled:
50-mod-http-image-filter.conf
```

```
$ sudo mv /etc/nginx/modules-enabled/50-mod-http-image-filter.conf
/etc/nginx/modules-available/
```

```
$ service nginx restart
```

Configuring modules

```
## /etc/nginx/modules-enabled/50-mod-http-image-filter.conf
load_module modules/nginx_http_image_filter_module.so;
```

```
location /img/ {
    image_filter resize 240 360;
    image_filter rotate 180;
    image_filter_buffer 16M;
    error_page 415 = /415.html;
}
```

Further reading

- [nginx documentation](https://nginx.org/en/docs/)⁵²¹
-
- *Install nginx*
- *Configure nginx*
- *Extend nginx with modules*

Web programming

⁵²¹ <https://nginx.org/en/docs/>

How to install and configure PHP

PHP⁵²²

Prerequisites

- [our Apache guide](#)
- [our MySQL guide](#)

Install PHP

```
$ sudo apt install php libapache2-mod-php
```

Install optional packages

-

```
$ sudo apt install php-cli
```

-

```
$ sudo apt install php-cgi
```

-

```
$ sudo apt install php-mysql
```

-

```
$ sudo apt install php-pgsql
```

⁵²² <https://www.php.net/>

Configure PHP

```
$ sudo systemctl restart apache2.service
```

Test your setup

```
<?php
phpinfo();
?>
```

Docu-

mentRoot

Further reading

- [the php.net documentation](https://www.php.net/docs.php)⁵²³
-
- [Apache MySQL PHP Ubuntu Wiki](https://help.ubuntu.com/community/ApacheMySQLPHP)⁵²⁴

How to install and configure Ruby on Rails

[Ruby on Rails](https://rubyonrails.org/)⁵²⁵

⁵²³ <https://www.php.net/docs.php>

⁵²⁴ <https://help.ubuntu.com/community/ApacheMySQLPHP>

⁵²⁵ <https://rubyonrails.org/>

Prerequisites

- [our Apache guide](#)
- [our MySQL guide](#)

Install rails

```
$ sudo apt install rails
```

```
$ sudo apt install --no-install-recommends rails
```

Configure the web server

DocumentRoot

```
DocumentRoot /path/to/rails/application/public
```

```
<Directory "/path/to/rails/application/public">
    Options Indexes FollowSymlinks MultiViews ExecCGI
    AllowOverride All
    Order allow,deny
    allow from all
    AddHandler cgi-script .cgi
</Directory>
```

```
$ sudo a2enmod rewrite
```

```
$ sudo chown -R www-data:www-data /path/to/rails/application/public
```

```
$ sudo chown -R www-data:www-data /path/to/rails/application/tmp
```

```
$ RAILS_ENV=production rake assets:precompile
```

Configure the database

```
# Mysql
production
  adapter: mysql2
  username: user
  password: password
  host: 127.0.0.1
  database: app
```

```
$ RAILS_ENV=production rake db:create > RAILS_ENV=production rake db:migrate
```

daemonize

Further reading

- [Ruby on Rails](#)⁵²⁶
- [Agile Development with Rails](#)⁵²⁷

⁵²⁶ <https://rubyonrails.org/>

⁵²⁷ <https://pragprog.com/titles/rails7/agile-web-development-with-rails-7/>

See also

- [Web services](#)

[Graphics](#)

Graphics

On-system GPU

NVIDIA drivers installation

Warning:

NVIDIA drivers installed from sources outside of those listed in this guide could potentially overwrite those provided by ubuntu-drivers and may break secure boot.

NVIDIA drivers releases

1. [NVIDIA drivers releases](#) on the NVIDIA website⁵²⁸
2. [NVIDIA drivers releases](#) *ERD* in the NVIDIA documentation⁵²⁹

⁵²⁸ <https://www.nvidia.com/en-us/drivers/unix/>

⁵²⁹ <https://docs.nvidia.com/datacenter/tesla/index.html>

Check driver versions

```
$ cat /proc/driver/nvidia/version
```

The recommended way (ubuntu-drivers tool)

Ensure your system and kernel is up-to-date

```
$ sudo apt update && sudo apt upgrade
```

Check the available drivers for your hardware

```
$ sudo ubuntu-drivers list
```

```
$ sudo ubuntu-drivers list --gpgpu
```

```
nvidia-driver-470  
nvidia-driver-470-server  
nvidia-driver-535  
nvidia-driver-535-open  
nvidia-driver-535-server  
nvidia-driver-535-server-open  
nvidia-driver-550  
nvidia-driver-550-open  
nvidia-driver-550-server  
nvidia-driver-550-server-open
```

Installing the drivers for generic use (e.g. desktop and gaming)

```
$ sudo ubuntu-drivers install
```

```
$ sudo ubuntu-drivers install nvidia:535
```

Installing the drivers on servers and/or for computing purposes

```
$ sudo ubuntu-drivers install --gpgpu
```

```
$ sudo ubuntu-drivers install --gpgpu nvidia:535-server
```

```
$ sudo apt install nvidia-utils-535-server
```

Optional step

```
$ sudo apt install nvidia-fabricmanager-535 libnvidia-nscq-535
```

Note:

While `nvidia-fabricmanager` and `libnvidia-nscq` do not have the same `-server` label in their name, they are really meant to match the `-server` drivers in the Ubuntu archive. For example, `nvidia-fabricmanager-535` will match the `nvidia-driver-535-server` package version (not the `nvidia-driver-535` package).

Manual driver installation (using APT)

Installing the kernel modules

Installing the pre-compiled NVIDIA modules for your kernel

```
$ sudo apt install  
linux-modules-nvidia-${DRIVER_BRANCH}${SERVER}-${LINUX_FLAVOUR}
```

ABI

```
$ sudo apt-cache policy  
linux-modules-nvidia-${DRIVER_BRANCH}${SERVER}-${uname -r}
```

```
$ sudo apt install linux-modules-nvidia-${DRIVER_BRANCH}${SERVER}-${uname  
-r}
```

Building your own kernel modules using the NVIDIA DKMS package

DKMS

```
$ sudo apt install linux-headers-${LINUX_FLAVOUR}
```

```
$ sudo apt-cache policy linux-headers-$(uname -r)
```

```
$ sudo apt install linux-headers-$(uname -r)
```

```
$ sudo apt install nvidia-dkms-${DRIVER_BRANCH}${SERVER}
```

```
$ sudo ubuntu-drivers install --include-dkms
```

Installing the user-space drivers and the driver libraries

```
$ sudo apt install nvidia-driver-${DRIVER_BRANCH}${SERVER}
```

(Optional) Installing Fabric Manager and the NSCQ library

```
$ sudo apt install nvidia-fabricmanager-${DRIVER_BRANCH}  
libnvidia-nscq-${DRIVER_BRANCH}
```

Note:

While `nvidia-fabricmanager` and `libnvidia-nscq` do not have the same `-server` label in their name, they are really meant to match the `-server` drivers in the Ubuntu archive. For example, `nvidia-fabricmanager-535` will match the `nvidia-driver-535-server` package version (not the `nvidia-driver-535` package).

Switching between pre-compiled and DKMS modules

- 1.
- 2.

Uninstalling the NVIDIA drivers

```
$ sudo apt --purge remove '*nvidia*${DRIVER_BRANCH}*
```

```
$ apt-mark showmanual | grep nvidia`.
```

```
$ sudo apt autoremove
```

Transitional packages to new driver branches

current support matrix⁵³⁰

Troubleshooting

Driver/library version mismatch error

`nvidia-smi`⁵³¹

```
Failed to initialize NVML: Driver/library version mismatch
```

⁵³⁰ <https://docs.nvidia.com/datacenter/tesla/drivers/index.html>

⁵³¹ <https://developer.nvidia.com/system-management-interface>

```
$ sudo dmesg
```

```
NVRM: API mismatch: the client has the version 570.172.08, but  
NVRM: this kernel module has the version 570.158.01. Please  
NVRM: make sure that this kernel module and all NVIDIA driver  
NVRM: components have the same version.
```

No devices were found error

`nvidia-smi`⁵³²

```
No devices were found
```

`nouveau`⁵³³

```
$ lsmod | grep nouveau
```

```
$ echo "blacklist nouveau" | sudo tee /etc/modprobe.d/disable-nouveau.conf >  
echo "options nouveau modeset=0" | sudo tee -a  
/etc/modprobe.d/disable-nouveau.conf > sudo rmmod nouveau || true > sudo  
update-initramfs -u
```

Virtual GPU

⁵³² <https://developer.nvidia.com/system-management-interface>

⁵³³ <https://nouveau.freedesktop.org/>

GPU virtualisation with QEMU/KVM

Graphics

frontend

-

-

libvirt section

ease-of-use and capability⁵³⁴

-

virgil3d⁵³⁵
Kernels >= 4.4⁵³⁶

GL

set up for device passthrough⁵³⁷

-

IOMMU

GPUs

```
-device vfio-pci,host=05:00.0,bus=1,addr=00.0,multifunction=on,x-vga=on -device vfio-pci,host=05:00.1,bus=1,addr=00.1
```

-

```
-display gtk,gl=on -device vfio-pci,syfsdev=/sys/bus/pci/devices/0000:00:02.0/4dd511f6-ec08-11e8-b839-2f163dde3b3,display=on,rombar=0
```

about vGPU at ⁵³⁸

Ubuntu GPU mdev evaluation⁵³⁹

⁵³⁴ <https://cpaelzer.github.io/blogs/006-mediated-device-to-pass-parts-of-your-gpu-to-a-guest/>

⁵³⁵ <https://virgil3d.github.io/>

⁵³⁶ <https://www.kraxel.org/blog/2016/09/using-virtio-gpu-with-libvirt-and-spice/>

⁵³⁷ <https://www.kernel.org/doc/Documentation/vfio-mediated-device.txt>

⁵³⁸ <https://www.kraxel.org/blog/2018/04/vgpu-display-support-finally-merged-upstream/>

⁵³⁹ <https://cpaelzer.github.io/blogs/006-mediated-device-to-pass-parts-of-your-gpu-to-a-guest/>

Nvidia⁵⁴¹ AMD

libvirt section
BIOS

Preparing the input-output memory management unit (IOMMU)

options you can enable/configure⁵⁴² IOMMU fea-
 ture⁵⁴³

Note:

The card used in all examples below e.g. when filtering for or assigning PCI IDs, is an NVIDIA V100 on PCI ID 41.00.0

```
$ lspci | grep 3D
41:00.0 3D controller: NVIDIA Corporation GV100GL [Tesla V100 PCIe 16GB] (rev a1)
```

DMAR

```
$ dmesg | grep -i -e DMAR -e IOMMU
[ 3.509232] iommu: Default domain type: Translated
...
[ 4.516995] pci 0000:00:01.0: Adding to iommu group 0
...
[ 4.702729] perf/amd_iommu: Detected AMD IOMMU #0 (2 banks, 4 counters/bank).
```

```
$ dmesg | grep -i -e DMAR -e IOMMU | grep $(lspci | awk '/ 3D / {print $1}' )
[ 4.598150] pci 0000:41:00.0: Adding to iommu group 66
```

⁵⁴⁰ https://github.com/intel/gvt-linux/wiki/GVTg_Setup_Guide

⁵⁴¹ <https://docs.nvidia.com/vgpu/latest/grid-vgpu-user-guide/index.html>

⁵⁴² <https://www.kernel.org/doc/html/latest/admin-guide/kernel-parameters.html?highlight=iommu>

⁵⁴³ <https://www.kernel.org/doc/html/latest/arch/x86/iommu.html>

```
$ find /sys/kernel/iommu_groups/ -name "*$(lspci | awk '/ 3D / {print $1}')"

/sys/kernel/iommu_groups/66/devices/0000:41:00.0
```

```
$ ll /sys/kernel/iommu_groups/66/devices/

lrwxrwxrwx 1 root root 0 Jan  3 06:57 0000:41:00.0 -> ../../../../devices/
pci0000:40/0000:40:03.1/0000:41:00.0/
```

Note:

Some older or non-server boards tend to group devices in one IOMMU group, which isn't very useful as it means you'll need to pass "all or none of them" to the same guest.

Preparations for PCI and mediated devices pass-through – block host drivers

```
$ echo "blacklist nouveau" | sudo tee /etc/modprobe.d/blacklist-nouveau.conf
> echo "options nouveau modeset=0" | sudo tee -a
/etc/modprobe.d/blacklist-nouveau.conf > sudo update-initramfs -u > sudo
reboot ``
```

You can check which kernel modules are loaded and available via ``lspci -v``:

```
```{terminal}
:copy:
:user:
:host:
:dir:
lspci -v | grep -A 10 " 3D "

41:00.0 3D controller: NVIDIA Corporation GV100GL [Tesla V100 PCIe 16GB] (rev a1)
...
Kernel modules: nvidiafb, nouveau
```

Kernel driver in use: nouveau

## Preparations for mediated devices pass-through - driver

supported and available from Nvidia<sup>544</sup>

ration of vGPU on Ubuntu<sup>545</sup>

installation and configu-

```
$ lsmod | grep nvidia

nvidia_vgpu_vfio 53248 38
nvidia 35282944 586 nvidia_vgpu_vfio
mdev 24576 2 vfio_mdev,nvidia_vgpu_vfio
drm 491520 6 drm_kms_helper,drm_vram_helper,nvidia
```

<sup>544</sup> <https://docs.nvidia.com/vgpu/latest/product-support-matrix/ubuntu.html>

<sup>545</sup> <https://docs.nvidia.com/vgpu/latest/grid-vgpu-user-guide/index.html#ubuntu-install-configure-vgpu>

**Note:**

While it works without a vGPU manager, to get the full capabilities you'll need to configure the **vGPU manager (that came with above package)**<sup>546</sup> and a license server so that each guest can get a license for the vGPU provided to it. Please see **Nvidia's documentation for the license server**<sup>547</sup>. While not officially supported on Linux (as of Q1 2022), it's worthwhile to note that it runs fine on Ubuntu with `sudo apt install unzip default-jre tomcat9 liblog4j2-java libslf4j-java` using `/var/lib/tomcat9` as the server path in the license server installer.

It's also worth mentioning that the Nvidia license server went **EOL on 31 July 2023**<sup>548</sup>. At that time, it was replaced by the **NVIDIA License System**<sup>549</sup>.

<sup>546</sup> <https://docs.nvidia.com/vgpu/latest/grid-vgpu-user-guide/index.html#install-vgpu-package-ubuntu>

<sup>547</sup> <https://docs.nvidia.com/vgpu/latest/grid-license-server-user-guide/index.html>

<sup>548</sup> <https://docs.nvidia.com/vgpu/news/vgpu-software-license-server-eol-notice/index.html>

<sup>549</sup> <https://docs.nvidia.com/license-system/latest/nvidia-license-system-quick-start-guide/index.html>

```
general status
$ systemctl status nvidia-vgpu-mgr
 Loaded: loaded (/lib/systemd/system/nvidia-vgpu-mgr.service; enabled; vendor preset:
 Active: active (running) since Tue 2021-09-14 07:30:19 UTC; 3min 58s ago
 Process: 1559 ExecStart=/usr/bin/nvidia-vgpu-mgr (code=exited, status=0/SUCCESS)
 Main PID: 1564 (nvidia-vgpu-mgr)
 Tasks: 1 (limit: 309020)
 Memory: 1.1M
 CGroup: /system.slice/nvidia-vgpu-mgr.service
 └─1564 /usr/bin/nvidia-vgpu-mgr

Sep 14 07:30:19 node-watt systemd[1]: Starting NVIDIA vGPU Manager Daemon...
Sep 14 07:30:19 node-watt systemd[1]: Started NVIDIA vGPU Manager Daemon.
Sep 14 07:30:20 node-watt nvidia-vgpu-mgr[1564]: notice: vmiop_env_log: nvidia-vgpu-mgr
daemon started

Entries when a guest gets a vGPU passed
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: (0x0): gpu-pci-id :
0x4100
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: (0x0): vgpu_type :
Quadro
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: (0x0): Framebuffer:
0x1dc000000
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: (0x0): Virtual Device
Id: 0x1db4:0x1252
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: (0x0): FRL Value: 60
FPS
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: ##### vGPU Manager
Information: #####
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: Driver Version: 470.68
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: (0x0): vGPU supported
range: (0x70001, 0xb0001)
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: (0x0): Init frame copy
engine: syncing...
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: (0x0): vGPU migration
enabled
```

(continues on next page)

(continued from previous page)

```
Sep 14 08:29:50 node-watt nvidia-vgpu-mgr[2866]: notice: vmiop_log: display_init inst: 0
successful

Entries when a guest grabs a license
Sep 15 06:55:50 node-watt nvidia-vgpu-mgr[4260]: notice: vmiop_log: (0x0): vGPU license
state: Unlicensed (Unrestricted)
Sep 15 06:55:52 node-watt nvidia-vgpu-mgr[4260]: notice: vmiop_log: (0x0): vGPU license
state: Licensed

In the guest the card is then fully recognized and enabled
$ nvidia-smi -a | grep -A 2 "Licensed Product"
 vGPU Software Licensed Product
 Product Name : NVIDIA RTX Virtual Workstation
 License Status : Licensed
```

mediated device<sup>550</sup>

*firmware*

NVIDIA documentation<sup>551</sup>

```
$ sudo mdevctl types
```

```
...
nvidia-300
 Available instances: 0
 Device API: vfio-pci
 Name: GRID V100-8C
 Description: num_heads=1, frl_config=60, framebuffer=8192M, max_resolution=4096x2160,
max_instance=2
```

```
$ $ sudo mdevctl define --parent 0000:41:00.0 --type nvidia-300
bc127e23-aaaa-4d06-a7aa-88db2dd538e0
```

```
$ $ sudo mdevctl define --parent 0000:41:00.0 --type nvidia-300
1360ce4b-2ed2-4f63-abb6-8cdb92100085
```

<sup>550</sup> <https://www.kernel.org/doc/html/latest/driver-api/vfio-mediated-device.html>

<sup>551</sup> <https://docs.nvidia.com/vgpu/latest/grid-vgpu-user-guide/index.html#ubuntu-install-configure-vgpu>



```
$ $ sudo mdevctl start --parent 0000:41:00.0 --uuid
bc127e23-aaaa-4d06-a7aa-88db2dd538e0
```

```
$ $ sudo mdevctl start --parent 0000:41:00.0 --uuid
1360ce4b-2ed2-4f63-abb6-8cdb92100085
```

```
$ sudo mdevctl list -d

bc127e23-aaaa-4d06-a7aa-88db2dd538e0 0000:41:00.0 nvidia-108 manual (active)
1360ce4b-2ed2-4f63-abb6-8cdb92100085 0000:41:00.0 nvidia-108 manual (active)
```

## Passing through PCI or mediated devices

```
<hostdev mode='subsystem' type='pci' managed='yes'>
 <source>
 <address domain='0x0000' bus='0x41' slot='0x00' function='0x0' />
 </source>
</hostdev>
```

```
<hostdev mode='subsystem' type='mdev' managed='no' model='vfio-pci' display='on'>
 <source>
 <address uuid='634fc146-50a3-4960-ac30-f09e5cedc674' />
 </source>
</hostdev>
```

part of the guest definition<sup>552</sup>

### Note:

This works great on Focal, but `type='none'` as well as `display='off'` weren't available on Bionic. If this level of control is required one would need to consider using the [Ubuntu Cloud Archive](https://wiki.ubuntu.com/OpenStack/CloudArchive)<sup>553</sup> or [Server-Backports](https://launchpad.net/~canonical-server/+archive/ubuntu/server-backports)<sup>554</sup> for a newer stack of the virtualisation components.

<sup>553</sup> <https://wiki.ubuntu.com/OpenStack/CloudArchive>

<sup>554</sup> <https://launchpad.net/~canonical-server/+archive/ubuntu/server-backports>

<sup>552</sup> <https://libvirt.org/formatdomain.html#usb-pci-scsi-devices>

s390x vfio-ccw<sup>555</sup>

- *Virtual GPU (vGPU) with QEMU/KVM*

*Virtualisation*

## Virtualisation

*Introduction to virtualization*

## Virtual machines

## LXD containers and virtual machines

LXD<sup>556</sup>

interface<sup>557</sup>

REST APIs<sup>558</sup>

built-in graphical user

to a cluster<sup>559</sup>

such as a MicroCloud<sup>560</sup>

## Online resources

the official LXD documentation<sup>561</sup>

Ubuntu Discourse forum<sup>562</sup>

YouTube channel<sup>563</sup>

---

<sup>555</sup> <https://docs.kernel.org/arch/s390/vfio-ccw.html>

<sup>556</sup> <https://canonical.com/lxd>

<sup>557</sup> [https://canonical.com/lxd/docs/latest/howto/access\\_ui/](https://canonical.com/lxd/docs/latest/howto/access_ui/)

<sup>558</sup> [https://canonical.com/lxd/docs/latest/restapi\\_landing/](https://canonical.com/lxd/docs/latest/restapi_landing/)

<sup>559</sup> <https://canonical.com/lxd/docs/latest/explanation/clusters/>

<sup>560</sup> <https://canonical.com/microcloud>

<sup>561</sup> <https://canonical.com/lxd/docs/default/>

<sup>562</sup> <https://discourse.ubuntu.com/c/lxd/126>

<sup>563</sup> <https://www.youtube.com/c/LXDvideos>

## Installation

```
$ sudo snap install lxd
```

## Configuration

```
$ lxd init
```

- [Btrfs](#)<sup>564</sup> [ZFS](#)<sup>565</sup>
- 
- 

```
$ gpasswd -a joe lxd
```

[How to initialize LXD](#)<sup>566</sup>

## Creating your first container

---

<sup>564</sup> <https://btrfs.readthedocs.io/>

<sup>565</sup> [https://openzfs.org/wiki/Main\\_Page](https://openzfs.org/wiki/Main_Page)

<sup>566</sup> <https://canonical.com/lxd/docs/latest/howto/initialize/>

## Creating a container

- 
- 
- 
- 

```
$ lxc launch remote:image containername
```

```
$ lxc launch --vm remote:image vmname
```

```
$ lxc init remote:image containername
```

```
$ lxc image list ubuntu:
```

```
$ lxc image info ubuntu:noble
```

```
$ lxc image info ubuntu:lts/arm64
```

```
$ lxc launch ubuntu:noble n1
```

```
$ lxc list
```

```
$ lxc info n1
```

```
$ lxc exec n1 -- sudo -i -u ubuntu
```

```
$ lxc shell n1
```

## LXD server configuration

```
$ lxc list
```

```
$ lxc config set core.https_address :8443
```

## Authentication

```
$ lxc remote add r1 r1.example.com:8443
```

```
$ lxc config trust add r1 certfile.crt
```

## Backing store

*filesystem*

Storage configuration<sup>567</sup>

## Container configuration

```
$ lxc config device add c1 opt-mount disk source=/opt path=/opt
```

```
$ lxc help config
```

```
$ lxc config edit c1
```

---

<sup>567</sup> <https://canonical.com/lxd/docs/latest/explanation/storage/>

## Profiles

```
$ lxc config device add c1 eth1 none
```

## Nesting

```
$ lxc config set container1 security.nesting true
```

## Limits

- 
- 
- 
- 
- 

the configuration documentation<sup>568</sup>

---

<sup>568</sup> [https://canonical.com/lxd/docs/latest/reference/instance\\_options/](https://canonical.com/lxd/docs/latest/reference/instance_options/)

## UID mappings and privileged containers

*GIDs*  
*subuid(5)*<sup>569</sup>

```
$ lxc config set c1 security.privileged true
```

## AppArmor

## Seccomp

## Raw LXC configuration

*lxc.*

*container.conf(5)*<sup>570</sup>

---

<sup>569</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/subuid.5.html>

<sup>570</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/lxc.container.conf.5.html>



## Snapshots

```
$ lxc move c1 final-beta
```

```
$ lxc snapshot c1 snapshot-name
```

```
$ lxc restore c1 snapshot-name
```

```
$ lxc copy c1/snapshot-name new-container
```

## Publishing images

```
$ lxc publish c1/snapshot-name --alias foo-2.0
```

```
$ lxc publish c1/snapshot-name --alias foo-2.0 public=true
```

```
$ lxc image edit foo-2.0
```

## Image export and import

```
$ lxc image export foo-2.0 foo-2.0.tar.gz
```

```
$ lxc image import foo-2.0.tar.gz --alias foo-2.0 --public
```

## Troubleshooting

```
$ journalctl -u lxd
```

```
$ lxc info c1 --show-log
```

## How to create a VM with Multipass

Multipass<sup>571</sup>

```
$ sudo snap install multipass
```

```
$ sudo apt update && sudo apt install snapd
```

## Find available images

---

<sup>571</sup> <https://canonical.com/multipass>

| Image                            | Aliases      | Version  | Description          |
|----------------------------------|--------------|----------|----------------------|
| core                             | core16       | 20200818 | Ubuntu Core 16       |
| core18                           |              | 20211124 | Ubuntu Core 18       |
| core20                           |              | 20230119 | Ubuntu Core 20       |
| core22                           |              | 20230717 | Ubuntu Core 22       |
| 22.04                            | jammy        | 20260515 | Ubuntu 22.04 LTS     |
| 24.04                            | noble        | 20260518 | Ubuntu 24.04 LTS     |
| 25.10                            | questing     | 20260520 | Ubuntu 25.10         |
| 26.04                            | resolute,lts | 20260520 | Ubuntu 26.04 LTS     |
| charm-dev                        |              | latest   | A development and    |
| testing environment for charmers |              |          |                      |
| docker                           |              | latest   | A Docker environment |
| with Portainer and related tools |              |          |                      |
| minikube                         |              | latest   | minikube is local    |
| Kubernetes                       |              |          |                      |

## Launch a fresh instance of Ubuntu 26.04 LTS

```
$ multipass launch 26.04
Launched: cleansing-guanaco
```

## Check out the running instances

```
$ multipass list
```

| Name              | State   | IPv4         | Image            |
|-------------------|---------|--------------|------------------|
| cleansing-guanaco | Running | 10.140.26.17 | Ubuntu 26.04 LTS |

## Learn more about the VM instance you just launched

```
$ multipass info cleansing-guanaco
```

```
Name: cleansing-guanaco
State: Running
Snapshots: 0
IPv4: 10.140.26.17
Release: Ubuntu 26.04 LTS
Image hash: dc5b5a43c267 (Ubuntu 26.04 LTS)
CPU(s): 1
Load: 0.74 0.35 0.13
Disk usage: 2.2GiB out of 4.8GiB
Memory usage: 347.4MiB out of 950.8MiB
Mounts: --
```

### Connect to a running instance

```
$ multipass shell cleansing-guanaco

Welcome to Ubuntu 26.04 LTS (GNU/Linux 7.0.0-15-generic x86_64)
(...)

ubuntu@cleansing-guanaco:~$
```

### Disconnect from the instance

### Run commands inside an instance from outside

```
$ multipass exec cleansing-guanaco -- lsb_release -a

No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 26.04 LTS
Release: 26.04
Codename: resolute
```

## Stop or start an instance

```
$ multipass stop cleansing-guanaco
```

```
$ multipass start cleansing-guanaco
```

## Delete the instance

```
$ multipass delete cleansing-guanaco
```

```
$ multipass list
```

| Name              | State   | IPv4 | Image         |
|-------------------|---------|------|---------------|
| cleansing-guanaco | Deleted | --   | Not Available |

```
$ multipass purge
```

```
$ multipass list
```

```
No instances found.
```

## Integrate with the rest of your virtualisation

*ESXi*  
*uvtool*

*Libvirt*

*QEMU KVM*  
*virt-manager*<sup>572</sup>

*guide*<sup>573</sup>

---

<sup>572</sup> <https://virt-manager.org/>

<sup>573</sup> <https://documentation.ubuntu.com/multipass/latest/how-to-guides/customise-multipass/set-up-a-graphical-interface/>

```
$ sudo multipass set local.driver=libvirt
```

#### Note:

If you are having issues interacting with Multipass after switching to the libvirt driver, check if there is a restriction by [AppArmor](#), for example. AppArmor may have a default policy which restricts the multipass service from interacting with the [libvirt service](#)<sup>574</sup>. So you need to add an explicit permission that allows it.

<sup>574</sup> <https://libvirt.org/daemons.html>

[virt-manager](#)<sup>575</sup>

```
$ multipass launch lts
```

```
Launched: engaged-amberjack
```

```
$ virsh list
```

| Id    | Name              | State   |
|-------|-------------------|---------|
| ----- | -----             | -----   |
| 15    | engaged-amberjack | running |

Mul-

[tipass drivers documentation](#)<sup>576</sup>

## Get help

```
$ multipass help
```

```
$ multipass help <command>
```

```
$ multipass help --all
```

[Multipass documentation](#)<sup>577</sup>

<sup>575</sup> <https://virt-manager.org/>

<sup>576</sup> <https://documentation.ubuntu.com/multipass/latest/how-to-guides/customise-multipass/set-up-the-driver/>

<sup>577</sup> <https://documentation.ubuntu.com/multipass/latest/>

## Create cloud image VMs with UVtool

### Creating virtual machines using uvtool

#### Install uvtool packages

- 
- 

```
$ sudo apt -y install uvtool uvtool-libvirt
```

#### Note:

You might need to log out and log in again to use libvirt without root privilege. If not, you might encounter a Permission denied error in the later steps.

#### Get the Ubuntu cloud image with uvt-simplestreams-libvirt

```
$ uvt-simplestreams-libvirt --verbose sync arch=amd64
```

```
$ uvt-simplestreams-libvirt query
```

```
release=bionic arch=amd64 label=daily (20191107)
release=focal arch=amd64 label=daily (20191029)
...
```

```
$ uvt-simplestreams-libvirt sync release=DISTRO-SHORT-CODENAME arch=amd64
```

```
$ uvt-simplestreams-libvirt sync --source
http://cloud-images.ubuntu.com/daily [... further options]
```

### Create a valid SSH key

```
$ ssh-keygen

Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.
The key fingerprint is:
4d:ba:5d:57:c9:49:ef:b5:ab:71:14:56:6e:2b:ad:9b ubuntu@DISTRO-SHORT-CODENAMES
The key's randomart image is:
+--[RSA 2048]-----+
| ..|
| o.=|
| . **|
| + o+=|
| S=|
| o . .+ .|
| . . o o |
| * |
| E |
+-----+

```



## Create the VM using `uvt-kvm`

```
$ uvt-kvm create firsttest
```

```
$ uvt-kvm create secondtest release=DISTRO-SHORT-CODENAME
```

```
$ uvt-kvm wait secondtest
```

## Connect to the running VM

```
$ uvt-kvm ssh secondtest
```

```
$ uvt-kvm ip secondtest
192.168.122.199
```

```
$ ssh -i ~/.ssh/id_rsa ubuntu@192.168.122.199
[...]
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@secondtest:~$
```

## Get the list of running VMs

## Destroy your VM

```
$ uvt-kvm destroy secondtest
```

### Note:

Unlike libvirt's `destroy` or `undefine` actions, this will (by default) also remove the associated virtual storage files.

## More `uv`-`kvm` options

- 
- 
- 
- 
- 
- 

*uv*-*kvm*(1)<sup>578</sup>

## QEMU

### Note:

Please bear in mind that invoking QEMU manually may sometimes require your user to be part of the `kvm` group.

<sup>578</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/uv-kvm.1.html>

## Virtualisation with QEMU

QEMU<sup>579</sup>

KVM<sup>580</sup>

command line interface<sup>581</sup>

monitor<sup>582</sup>

*libvirt*

## Install QEMU/KVM

```
$ kvm-ok
```

### Note:

The command `kvm-ok` is part of the `cpu-checker` package. Install it with `sudo apt install cpu-checker` if it is not already present.

```
$ sudo apt-get install qemu-system
```

## Boot a VM

```
$ qemu-system-x86_64 -m 4G -enable-kvm -cdrom
http://releases.ubuntu.com/noble/ubuntu-24.04.3-live-server-amd64.iso
```

### Caution:

This example is just for illustration purposes - it is not generally recommended without verifying the checksums. [Multipass](#) (page 633) and [UVTool](#) (page 638) are much better ways to get actual guests easily.

<sup>579</sup> [https://wiki.qemu.org/Main\\_Page](https://wiki.qemu.org/Main_Page)

<sup>580</sup> [https://www.linux-kvm.org/page/Main\\_Page](https://www.linux-kvm.org/page/Main_Page)

<sup>581</sup> <https://qemu-project.gitlab.io/qemu/system/invoke.html>

<sup>582</sup> <https://qemu-project.gitlab.io/qemu/system/monitor.html>

## VNC

### Create a virtual disk

```
$ qemu-img create -f qcow2 disk.qcow 5G
```

[virt-manager](https://virt-manager.org/)<sup>583</sup>

[libvirt](https://libvirt.org/)<sup>584</sup>

```
$ qemu-system-x86_64 options image[s]
```

[QEMU manpage](https://manpages.ubuntu.com/manpages/resolute/man1/qemu-img.1.html)<sup>585</sup> [qemu-img\(1\)](https://manpages.ubuntu.com/manpages/resolute/man1/qemu-img.1.html)<sup>586</sup>

[QEMU documentation](https://www.qemu.org/documentation/)<sup>587</sup>

### Create QEMU VMs with up to 1024 vCPUs

---

<sup>583</sup> <https://virt-manager.org/>

<sup>584</sup> <https://libvirt.org/>

<sup>585</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/qemu-system.1.html>

<sup>586</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/qemu-img.1.html>

<sup>587</sup> <https://www.qemu.org/documentation/>

## Requirement

| QEMU version | Ubuntu release           | Supported machine types                                                             |
|--------------|--------------------------|-------------------------------------------------------------------------------------|
| QEMU 8.2.1+  | Ubuntu 24.04 LTS (Noble) | ubuntu (Native support, also supports Jammy and Mantic machine types)               |
| QEMU 8.0.4   | Ubuntu 23.10 (Mantic)    | pc-q35-mantic-maxcpus, pc-i440fx-mantic-maxcpus (Also supports Jammy machine types) |
| QEMU 6.2     | Ubuntu 22.04 LTS (Jammy) | pc-q35-jammy-maxcpus, pc-i440fx-jammy-maxcpus                                       |

## Configuration by Ubuntu release

### 24.04 Noble

```
$ qemu-system-x86_64 -M ubuntu,accel=kvm,kernel-irqchip=split -device intel-iommu,intremap=on -smp cpus=300,maxcpus=300 ...
```

### 23.10 Mantic

```
$ qemu-system-x86_64 -M pc-q35-mantic-maxcpus,accel=kvm,kernel-irqchip=split -device intel-iommu,intremap=on -smp cpus=300,maxcpus=300 ...
```

## 22.04 Jammy

```
$ qemu-system-x86_64 -M pc-q35-jammy-maxcpus,accel=kvm,kernel-irqchip=split
-device intel-iommu,intremap=on -smp cpus=300,maxcpus=300 ...
```

## Boot ARM64 virtual machines on QEMU

### Note:

This requires Ubuntu 20.04 or greater

## Install QEMU to run ARM64 virtual machines

```
$ sudo apt install qemu-system-arm
```

## Create necessary support files

*EFI*

```
$ truncate -s 64m varstore.img
```

```
$ truncate -s 64m efi.img
```

```
$ dd if=/usr/share/qemu-efi-aarch64/QEMU_EFI.fd of=efi.img conv=notrunc
```

## Fetch the Ubuntu cloud image

Ubuntu cloud image<sup>588</sup>

## Run QEMU natively on an ARM64 host

```
$ qemu-system-aarch64 \ > -enable-kvm \ > -m 1024 \ > -cpu host \ > -M virt \ > -nographic \ > -drive if=pflash,format=raw,file=efi.img,readonly=on \ > -drive if=pflash,format=raw,file=varstore.img \ > -drive if=none,file=jammy-server-cloudimg-arm64.img,id=hd0 \ > -device virtio-blk-device,drive=hd0 -netdev type=tap,id=net0 \ > -device virtio-net-device,netdev=net0
```

## Run an emulated ARM64 VM on x86

```
$ qemu-system-aarch64 \ > -m 2048 \ > -cpu max \ > -M virt \ > -nographic \ > -drive if=pflash,format=raw,file=efi.img,readonly=on \ > -drive if=pflash,format=raw,file=varstore.img \ > -drive if=none,file=jammy-server-cloudimg-arm64.img,id=hd0 \ > -device virtio-blk-device,drive=hd0 \ > -netdev type=tap,id=net0 \ > -device virtio-net-device,netdev=net0
```

---

<sup>588</sup> <https://cloud-images.ubuntu.com/>

## Troubleshooting

### No output and no response

## Further reading

- *Virtualizing graphics using QEMU/KVM*
- *Using QEMU to create a microvm*

## Confidential Computing with AMD

- *AMD-SME*
- *AMD-SEV*
- *AMD-SEV-ES*
- *AMD-SEV-SNP*

*AMD-SEV-SNP*



## Host configuration

AMD's documentation<sup>589</sup>

BMC

```
$ cat /sys/module/kvm_amd/parameters/sev
```

```
Y
```

```
$ cpuid -1 -l 0x8000001f
```

CPU:

AMD Secure Encryption (0x8000001f):

```
SME: secure memory encryption support = true
SEV: secure encrypted virtualize support = true
VM page flush MSR support = false
SEV-ES: SEV encrypted state support = true
SEV-SNP: SEV secure nested paging = true
```

```
$ qemu-system-x86_64 \ > -enable-kvm \ > -nographic \ > -machine q35 -smp 6
-m 6G \ > -drive "if=virtio,format=qcow2,file=disk.img" \ > -net
nic,model=e1000 -net user,hostfwd=tcp::2222-:22 \ > -cpu EPYC-v4 \ > -machine
memory-encryption=sev0,vmport=off \ > -object
memory-backend-memfd,id=ram1,size=6G,share=true,prealloc=false \ > -machine
memory-backend=ram1 \ > -object sev-snp-guest,id=sev0,cbitpos=47,reduced-
phys-bits=1,kernel-hashes=on,policy=0x30000 \ > -kernel ./vmlinuz \ > -append
"root=/dev/vda1 console=ttyS0" \ > -bios /usr/share/ovmf/OVMF.amdsev.fd
```

•

EDK II<sup>590</sup>

```
-object sev-snp-guest,id=sev0,cbitpos=47,reduced-phys-bits=1,kernel-hashes=on \
```

•

•

•

<sup>589</sup> <https://docs.amd.com/v/u/en-US/58207-using-sev-with-amd-epyc-processors>

<sup>590</sup> [https://en.wikipedia.org/wiki/TianoCore\\_EDK\\_II](https://en.wikipedia.org/wiki/TianoCore_EDK_II)

- 

QEMU documentation<sup>591</sup>

tion<sup>592</sup> AMD SEV<sup>593</sup>

invoca-

## Guest configuration

- <https://cloud-images.ubuntu.com/releases/noble/release/ubuntu-24.04-server-cloudimg-amd64.img>
- <https://cloud-images.ubuntu.com/releases/noble/release/unpacked/>

snpguest<sup>594</sup>

```
$ sudo ./snpguest report --random attestation-report.bin request-file.txt
```

## Confidential Computing with Intel Trust Domain Extensions (TDX)

Intel TDX<sup>595</sup>

---

<sup>591</sup> <https://www.qemu.org/docs/master/system/i386/amd-memory-encryption.html>

<sup>592</sup> <https://www.qemu.org/docs/master/system/invocation.html>

<sup>593</sup> <https://www.qemu.org/docs/master/system/i386/amd-memory-encryption.html#sevapi>

<sup>594</sup> <https://github.com/virtee/snpguest>

<sup>595</sup> <https://www.intel.com/content/www/us/en/developer/tools/trust-domain-extensions/overview.html>

Supported hardware

| Processor                                                           | Code Name       | TDX Version | Module |
|---------------------------------------------------------------------|-----------------|-------------|--------|
| 4th Gen Intel Xeon Scalable Processors (select SKUs with Intel TDX) | Sapphire Rapids | 1.5.x       |        |
| 5th Gen Intel Xeon Scalable Processors                              | Emerald Rapids  | 1.5.x       |        |
| Intel Xeon 6 Processors with E-Cores                                | Sierra Forest   | 1.5.x       |        |
| Intel Xeon 6 Processors with P-Cores                                | Granite Rapids  | 2.0.x       |        |

Configure host

BIOS

Note:

The following is a sample BIOS configuration. The necessary BIOS settings or menus may differ based on the platform used. Reach out to your OEM/ODM or independent BIOS vendor for platform-specific instructions.

- 
- 
- 
- 
- 
- 
- 
-

```
$ cat /proc/cmdline
```

```
BOOT_IMAGE=/boot/vmlinuz-6.19.0-3-generic root=UUID=f6ce4201-d6d1-4505-bdf5-019ee4fe842e ro nohibernate kvm_intel.tdx=1
```

```
$ sudo dmesg | grep -i tdx
```

```
[1.824045] virt/tdx: BIOS enabled: private KeyID range [32, 64)
[1.824049] virt/tdx: Disable ACPI S3. Turn off Intel TDX in the BIOS to use ACPI S3.
[24.195089] virt/tdx: 1050644 KB allocated for PAMT
[24.195098] virt/tdx: module initialized
```

## Install virtualization stack

```
$ sudo apt update
```

```
$ sudo apt install \ > qemu-system-x86 \ > ovmf-inteltdx \ >
libvirt-daemon-system \ > libvirt-clients
```

- 
- 
- 
- 

```
$ ls -l /usr/share/ovmf/OVMF.inteltdx.ms.fd
```

```
-rw-r--r-- 1 root root 4194304 Jan 15 12:00 /usr/share/ovmf/OVMF.inteltdx.ms.fd
```

## Create a guest image

```
$ wget https://cloud-images.ubuntu.com/resolute/current/resolute-server-cloudimg-amd64.img
```

```
$ sudo apt install cloud-image-utils
```

```
$ cat > user-data.yaml << 'EOF' > #cloud-config > chpasswd: > list: | >
root:root > expire: False > ssh_pwauth: True > disable_root: false > EOF
```

```
$ cloud-localds user-data.img user-data.yaml
```

### Note:

For production use, set a strong password and configure SSH key-based authentication instead of password authentication.

## Launch a TD guest using QEMU

```
$ qemu-system-x86_64 \ > -accel kvm \ > -smp 32 \ > -m 16G \ > -cpu host \ >
-object
'{"qom-type":"tdx-guest","id":"tdx","quote-generation-socket":{"type":
"vsock", "cid":"2","port":"4050"}}' \ > -object
memory-backend-ram,id=mem0,size=16G \ > -machine
q35,kernel_irqchip=split,confidential-guest-support=tdx,memory-backend=mem0 \
> -bios /usr/share/ovmf/OVMF.inteltdx.ms.fd \ > -nographic \ > -nodefaults \
> -vga none \ > -drive
file=resolute-server-cloudimg-amd64.img,if=none,id=virtio-disk0 \ > -device
virtio-blk-pci,drive=virtio-disk0 \ > -drive
file=user-data.img,if=none,id=cloud-init,format=raw \ > -device
virtio-blk-pci,drive=cloud-init \ > -serial stdio
```

•

- 

- 

- 

- 

- 

- 

- 

QEMU documentation<sup>596</sup>

## Verify Intel TDX enablement in the guest

```
$ sudo dmesg | grep tdx
[0.000000] tdx: Guest detected
[11.162378] systemd[1]: Detected confidential virtualization tdx.
```

```
$ ls -l /dev/tdx_guest
crw----- 1 root root 10, 125 Jan 1 00:00 /dev/tdx_guest
```

## Launch a TDX VM with libvirt

```
$ cat > tdx-vm.xml << 'EOF' > <domain type='kvm'
xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'> >
<name>tdx-guest</name> > <memory unit='GiB'>16</memory> > <memoryBacking> >
<source type="anonymous"/> > <access mode="private"/> > </memoryBacking> >
<vcpu placement="static">16</vcpu> > <os> <type arch='x86_64'
machine='q35'>hvm</type> > <loader type='rom'
readonly='yes'>/usr/share/ovmf/OVMF.inteltdx.ms.fd</loader> > <boot
dev='hd'> > </os> > <features> > <acpi/> > <apic/> > <ioapic driver='qemu'> >
```

<sup>596</sup> <https://www.qemu.org/docs/master/system/i386/tdx.html>

```
> </features> > <clock offset='utc'> > <timer name='hpet' present='no' /> >
</clock> > <cpu mode='host-passthrough'> > <topology sockets='1' cores='16'
threads='1' /> > </cpu> > <devices> >
<emulator>/usr/bin/qemu-system-x86_64</emulator> > <disk type="file"
device="disk"> > <driver name="qemu" type="qcow2"/> > <source
file="/var/lib/libvirt/images/resolute-server-cloudimg-amd64.img"/> > <target
dev="vda" bus="virtio"/> > </disk> > <console type='pty'> > <target
type='virtio' port='1' /> > </console> > </devices> > <launchSecurity
type='tdx'> > <policy>0x10000000</policy> > <quoteGenerationService> >
<SocketAddress type='vsock' cid='2' port='4050' /> > </quoteGenerationService>
> </launchSecurity> > </domain> > EOF
```

```
$ sudo virsh define tdx-vm.xml
```

```
$ sudo virsh start tdx-guest
```

```
$ sudo virsh console tdx-guest
```

- 
- 
- 
- 
- 

## Manage the VM

```
$ sudo virsh list --all
```

```
$ sudo virsh shutdown tdx-guest
```

```
$ sudo virsh destroy tdx-guest
```

```
$ sudo virsh undefine tdx-guest
```

## Troubleshooting Intel TDX

- 

```
$ sudo apt install msr-tools
```

- 

```
$ sudo rdmsr 0x982 -f 1:1 1
```

- 

```
$ sudo rdmsr 0x1401 -f 11:11 1
```

- 

```
$ echo $((0x$(sudo rdmsr 0x87 -f 63:32)))
```

- 

```
$ echo $((0x$(sudo rdmsr 0x87 -f 63:32)))
```

- 

```
$ sudo rdmsr 0xa0
```



updating to the latest Intel TDX

Module<sup>597</sup>

#### Note:

If you chose to “Update Intel TDX Module via Binary Deployment”, make sure you’re using the correct Intel TDX Module version for your hardware. See the [Supported hardware](#) (page 650) table.

## Virtualization Hardware Enablement Stack

Ubuntu Resolute 26.04<sup>598</sup>

- **qemu-hwe**<sup>599</sup>
- **libvirt-hwe**<sup>600</sup>
- **edk2-hwe**<sup>601</sup>
- **seabios-hwe**<sup>602</sup> *BIOS*

## Helper package

```
root@enticed-cichlid:~# apt install ubuntu-helper-virt-hwe
```

```
root@enticed-cichlid:~# ubuntu_virt_helper
Installed variant: none
```

```
1 packages:
- ubuntu-helper-virt-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe)
```

<sup>597</sup> [https://cc-enabling.trustedservices.intel.com/intel-tdx-enabling-guide/04/hardware\\_setup/#deploy-specific-intel-tdx-module-version](https://cc-enabling.trustedservices.intel.com/intel-tdx-enabling-guide/04/hardware_setup/#deploy-specific-intel-tdx-module-version)

<sup>598</sup> <https://releases.ubuntu.com/resolute/>

<sup>599</sup> <https://launchpad.net/ubuntu/+source/qemu-hwe>

<sup>600</sup> <https://launchpad.net/ubuntu/+source/libvirt-hwe>

<sup>601</sup> <https://launchpad.net/ubuntu/+source/edk2-hwe>

<sup>602</sup> <https://launchpad.net/ubuntu/+source/seabios-hwe>

## Impact on current usage

```
root@enticed-cichlid:~# apt install qemu-system-x86
```

```
root@enticed-cichlid:~# ubuntu_virt_helper --verbose
Installed variant: base
```

15 packages:

- ovmf (2025.11-3ubuntu7, src:edk2, auto)
- ovmf-amdsev (2025.11-3ubuntu7, src:edk2, auto)
- ovmf-generic (2025.11-3ubuntu7, src:edk2, auto)
- ovmf-inteltdx (2025.11-3ubuntu7, src:edk2, auto)
- qemu-block-extra (1:10.2.1+ds-1ubuntu3, src:qemu, auto)
- qemu-system-common (1:10.2.1+ds-1ubuntu3, src:qemu, auto)
- qemu-system-data (1:10.2.1+ds-1ubuntu3, src:qemu, auto)
- qemu-system-gui (1:10.2.1+ds-1ubuntu3, src:qemu, auto)
- qemu-system-modules-opengl (1:10.2.1+ds-1ubuntu3, src:qemu, auto)
- qemu-system-modules-spice (1:10.2.1+ds-1ubuntu3, src:qemu, auto)
- qemu-system-x86 (1:10.2.1+ds-1ubuntu3, src:qemu, manual)
- qemu-utils (1:10.2.1+ds-1ubuntu3, src:qemu, auto)
- seabios (1.17.0-1ubuntu1, src:seabios, auto)
- ubuntu-helper-virt-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, manual)
- ubuntu-virt (1:10.2.1+ds-1ubuntu3, src:qemu, auto)

## Uncontrolled variant switches and dependency management implications

```
root@enticed-cichlid:~# apt install qemu-utils-hwe
The following packages were automatically installed and are no longer required:
 adwaita-icon-theme libflac14
 ...
 x11-common
Use 'sudo apt autoremove' to remove them.
```

```
Installing:
 qemu-utils-hwe
```

```
Installing dependencies:
 qemu-block-extra-hwe ubuntu-virt-hwe
```

(continues on next page)

(continued from previous page)

```
REMOVING:
 ovmf ovmf-inteltdx qemu-system-data qemu-system-modules-spice
seabios
 ovmf-amdsev qemu-block-extra qemu-system-gui qemu-system-x86
ubuntu-virt
 ovmf-generic qemu-system-common qemu-system-modules-opengl qemu-utils

Summary:
 Upgrading: 0, Installing: 3, Removing: 14, Not Upgrading: 9
 Download size: 2496 kB
 Freed space: 118 MB

Continue? [Y/n]
```

```
Get:1 http://archive.ubuntu.com/ubuntu resolute/main amd64 ubuntu-virt-hwe all 1:10.2.
1+ds-1ubuntu4 [11.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu resolute/main amd64 qemu-utils-hwe amd64 1:10.2.
1+ds-1ubuntu4 [2389 kB]
Get:3 http://archive.ubuntu.com/ubuntu resolute/main amd64 qemu-block-extra-hwe amd64
1:10.2.1+ds-1ubuntu4 [95.4 kB]
Fetched 2496 kB in 1s (3358 kB/s)
Virt: switch of the ubuntu-virt[-hwe] stack is detected and there are orphaned removals
(no counterpart being installed):
- ovmf
- ovmf-amdsev
- ovmf-generic
- ovmf-inteltdx
- qemu-system-x86
- qemu-system-modules-spice
- qemu-system-common
- qemu-system-data
- qemu-system-gui
- qemu-system-modules-opengl
- seabios
Virt: Please install them back and use the ubuntu-virt-helper script to switch between
ubuntu-virt[-hwe] packages.
(Reading database ... 96412 files and directories currently installed.)
Removing ovmf (2025.11-3ubuntu7) ...
Removing ovmf-amdsev (2025.11-3ubuntu7) ...
Removing ovmf-generic (2025.11-3ubuntu7) ...
```

```
root@enticed-cichlid:~# ubuntu_virt_helper --verbose
Installed variant: hwe
```

(continues on next page)

(continued from previous page)

```
4 packages:
- qemu-block-extra-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, auto)
- qemu-utils-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, manual)
- ubuntu-helper-virt-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, manual)
- ubuntu-virt-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, auto)
```

## Proper variant switch using the helper script

```
root@enticed-cichlid:~# ubuntu_virt_helper switch
Switching from base to hwe variant...
Installing:
 ovmf-amdsev-hwe ovmf-inteltdx-hwe qemu-system-data-hwe qemu-system-
modules-spice-hwe seabios-hwe
 ovmf-generic-hwe qemu-block-extra-hwe qemu-system-gui-hwe qemu-system-
x86-hwe ubuntu-virt-hwe
 ovmf-hwe qemu-system-common-hwe qemu-system-modules-opengl-hwe qemu-utils-hwe

Suggested packages:
 samba vde2 passt

REMOVING:
 ovmf ovmf-generic qemu-block-extra qemu-system-data qemu-system-modules-
opengl qemu-system-x86 seabios
 ovmf-amdsev ovmf-inteltdx qemu-system-common qemu-system-gui qemu-system-modules-
spice qemu-utils ubuntu-virt

Summary:
 Upgrading: 0, Installing: 14, Removing: 14, Not Upgrading: 9
 Download size: 25.5 MB
 Freed space: 7168 B

Continue? [Y/n]
```

```
root@enticed-cichlid:~# ubuntu_virt_helper --verbose
Installed variant: hwe

15 packages:
- ovmf-amdsev-hwe (2025.11-3ubuntu8, src:edk2-hwe, auto)
```

(continues on next page)

(continued from previous page)

```
- ovmf-generic-hwe (2025.11-3ubuntu8, src:edk2-hwe, auto)
- ovmf-hwe (2025.11-3ubuntu8, src:edk2-hwe, auto)
- ovmf-inteltdx-hwe (2025.11-3ubuntu8, src:edk2-hwe, auto)
- qemu-block-extra-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, auto)
- qemu-system-common-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, auto)
- qemu-system-data-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, auto)
- qemu-system-gui-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, auto)
- qemu-system-modules-opengl-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, auto)
- qemu-system-modules-spice-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, auto)
- qemu-system-x86-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, manual)
- qemu-utils-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, auto)
- seabios-hwe (1:17.0-1ubuntu2, src:seabios-hwe, auto)
- ubuntu-helper-virt-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, manual)
- ubuntu-virt-hwe (1:10.2.1+ds-1ubuntu4, src:qemu-hwe, auto)
```

- *LXD containers and virtual machines*
- *Create VMs with Multipass*
- *Create cloud image VMs with UVtool*
- *QEMU*
- *Confidential Computing with AMD*
- *Confidential Computing with Intel*
- *Hardware Enablement Virtualization Stack*

## VM tooling

### Libvirt

libvirt library<sup>603</sup>

Kernel-based Virtual Machine (KVM)<sup>604</sup>

```
$ kvm-ok
```

#### Note:

This command is part of the package `cpu-checker`, you might need to install it because it is not installed by default.

<sup>603</sup> <https://libvirt.org/>

<sup>604</sup> [https://www.linux-kvm.org/page/Main\\_Page](https://www.linux-kvm.org/page/Main_Page)

**Note:**

On many computers with processors supporting hardware-assisted virtualisation, you must activate an option in the *BIOS* to enable it.

## Virtual networking

SLiRP<sup>605</sup>

how to configure a bridge<sup>606</sup>  
netplan.io documentation<sup>607</sup>

## Install libvirt

```
$ sudo apt update
```

```
$ sudo apt install qemu-kvm libvirt-daemon-system
```

```
$ sudo adduser $USER libvirt
```

**Note:**

If you add the current user, log out and back in for the new group membership to take effect.

<sup>605</sup> <https://en.wikipedia.org/wiki/Slirp>

<sup>606</sup> <https://netplan.readthedocs.io/en/latest/netplan-yaml/#properties-for-device-type-bridges>

<sup>607</sup> <https://netplan.readthedocs.io/en/latest/>

- 
- 
- 

*UVTool*

*Multipass*

*Graphical User Interface (GUI)*

*Virtual*

*Machine Manager / Viewer*

## Virtual machine management

- *Multipass*
- *UVTool*
- *virt-\* tools*
- *OpenStack*<sup>608</sup>

## Manage VMs with virsh

- 
- 
- 
- 
- 

```
$ virsh list
```

```
$ virsh start <guestname>
```

```
$ virsh autostart <guestname>
```

```
$ virsh reboot <guestname>
```

---

<sup>608</sup> <https://ubuntu.com/openstack>

```
$ virsh save <guestname> save-my.state
```

- 

```
$ virsh restore save-my.state
```

- 

```
$ virsh shutdown <guestname>
```

- 

```
$ virsh attach-disk <guestname> /dev/cdrom /media/cdrom
```

- 

```
$ virsh edit <guestname>
```

XML representation that defines the guest<sup>609</sup>

### *Virtual Machine Manager / Viewer*

#### Note:

If virsh (or other vir\* tools) connects to something other than the default qemu-kvm/system hypervisor, you can find alternatives for the --connect option using `man virsh` or the [libvirt docs](https://libvirt.org/docs)<sup>610</sup>.

---

<sup>610</sup> <https://libvirt.org/uri.html>

### system and session scope

```
$ virsh --connect qemu:///system
```

- 

- 

---

<sup>609</sup> <https://libvirt.org/formatdomain.html>



## SLiRP networking<sup>611</sup>

### See also

There is more information about this topic in the [libvirt FAQ](#)<sup>612</sup> and this [blog post](#)<sup>613</sup> about the topic.

## Migration

- [Offline migration](#)<sup>614</sup>
- [Live migration](#)<sup>615</sup>
- [Postcopy migration](#)<sup>616</sup>

```
$ virsh migrate --help
```

[Ubuntu Wiki documentation on](#)

## KVM migration<sup>617</sup>

---

<sup>611</sup> <https://en.wikipedia.org/wiki/Slirp>

<sup>612</sup> <https://wiki.libvirt.org/FAQ.html#what-is-the-difference-between-qemu-system-and-qemu-session-which-one-should-i-use>

<sup>613</sup> <https://blog.wikichoon.com/2016/01/qemusystem-vs-qemusession.html>

<sup>614</sup> <https://libvirt.org/migration.html#offline-migration>

<sup>615</sup> <https://libvirt.org/migration.html>

<sup>616</sup> <https://wiki.qemu.org/Features/PostCopyLiveMigration>

<sup>617</sup> <https://wiki.ubuntu.com/QemuKVMMigration>

## CPU model and topology

- 
- 
- 
- 

```
example snippet from qemu command line generated by libvirt
... -cpu Haswell-noTSX-IBRS,vmx=on,pdcm=off,... vmx-entry-load-efer=on,vmx-eptp-
switching=on ...
```

```
<capabilities>
 <host>
 <uuid>10303837-3831-584d-5135-323430354a3f</uuid>
 <cpu>
 <arch>x86_64</arch>
 <model>Haswell-noTSX-IBRS</model>
 <vendor>Intel</vendor>
 <microcode version='73' />
 <signature family='6' model='63' stepping='2' />
 <counter name='tsc' frequency='2397195000' scaling='no' />
 <topology sockets='1' dies='1' cores='6' threads='2' />
 <maxphysaddr mode='emulate' bits='46' />
 <feature name='vme' />
 <feature name='ds' />
 <feature name='acpi' />
 <feature name='ss' />
```

upstream CPU model and features documentation<sup>618</sup>

<sup>618</sup> <https://libvirt.org/formatcaps.html#host-cpu-model-and-features>

## Host CPU capabilities

### Device passthrough/hotplug

- *hotplug*
- 

guest description<sup>619</sup>

```
<hostdev mode='subsystem' type='usb' managed='yes'>
 <source>
 <vendor id='0x0b6d' />
 <product id='0x3880' />
 </source>
</hostdev>
```

```
<hostdev mode='subsystem' type='pci' managed='yes'>
 <source>
 <address domain='0x0000' bus='0x04' slot='0x10' function='0x0' />
 </source>
</hostdev>
```

#### Note:

Getting the virtual function is device-dependent and cannot be fully covered here. In general, it involves setting up an *IOMMU*, registering via *VFIO*<sup>620</sup>, and sometimes requesting a number of VFs.

<sup>620</sup> <https://www.kernel.org/doc/Documentation/vfio.txt>

ppc64el<sup>621</sup>

```
$ sudo modprobe vfio-pci
identify device
$ lspci -n -s 0005:01:01.3
0005:01:01.3 0200 10df:e228 (rev 10)
register and request VFs
```

(continues on next page)

<sup>619</sup> <https://libvirt.org/formatdomain.html>

<sup>621</sup> <https://wiki.debian.org/ppc64el>

(continued from previous page)

```
$ echo 10df e228 | sudo tee /sys/bus/pci/drivers/vfio-pci/new_id
$ echo 4 | sudo tee /sys/bus/pci/devices/0005\:01\:00 0/sriov_numvfs
```

```
virsh attach-device <guestname> <device-xml>
Use the Device in the Guest
virsh detach-device <guestname> <device-xml>
```

## Access QEMU Monitor via libvirt

QEMU Monitor<sup>622</sup>

```
$ virsh qemu-monitor-command --hmp focal-test-log 'drive_add 0
if=none,file=/var/lib/libvirt/images/test.img,format=raw,id=disk1'
```

```
$ virsh qemu-monitor-command --hmp y-ipns 'info registers'

RAX=00ffffc000000000 RBX=ffff8f0f5d5c7e48 RCX=0000000000000000 RDX=ffffea00007571c0
RSI=0000000000000000 RDI=ffff8f0fdd5c7e48 RBP=ffff8f0f5d5c7e18 RSP=ffff8f0f5d5c7df8
[...]
```

## Huge pages

*huge pages*

64-bit CPUs<sup>623</sup>

32-bit CPUs<sup>624</sup>

x86\_64<sup>625</sup>

<sup>622</sup> <https://en.wikibooks.org/wiki/QEMU/Monitor>

<sup>623</sup> [https://en.wikipedia.org/wiki/64-bit\\_computing](https://en.wikipedia.org/wiki/64-bit_computing)

<sup>624</sup> [https://en.wikipedia.org/wiki/32-bit\\_computing](https://en.wikipedia.org/wiki/32-bit_computing)

<sup>625</sup> <https://en.wikipedia.org/wiki/X86-64>

1 byte each<sup>626</sup>

Memory Management Unit (MMU)<sup>627</sup>

*Translation Lookaside Buffer (TLB)*

*memory to the host*      *pre-allocated*      *map the VM's*

transparent huge pages<sup>628</sup>

#### Tip:

When using device passthrough on very large guests, there is an extra benefit of using huge pages, as it is faster to do the initial memory clear on the VFIO *DMA* pin.

<sup>626</sup> [https://en.wikipedia.org/wiki/Memory\\_address](https://en.wikipedia.org/wiki/Memory_address)

<sup>627</sup> [https://en.wikipedia.org/wiki/Memory\\_management\\_unit](https://en.wikipedia.org/wiki/Memory_management_unit)

<sup>628</sup> <https://www.kernel.org/doc/html/next/admin-guide/mm/transhuge.html>

## Huge page allocation

```
$ echo 256 | sudo tee /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
```

```
$ cat /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
256
```

`default_hugepagesz`<sup>629</sup>

```
$ grep Hugepagesize /proc/meminfo
Hugepagesize: 2048 kB
```

```
$ tail /sys/kernel/mm/hugepages/hugepages-*/nr_hugepages`
==> /sys/kernel/mm/hugepages/hugepages-1048576kB/nr_hugepages <==
0
==> /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages <==
2
```

Numa node<sup>630</sup>

boot or runtime<sup>631</sup>  
kernel documentation<sup>632</sup>

`hugeadm`<sup>633</sup>

```
$ apt install libhugetlbfs-bin
```

<sup>629</sup> <https://www.kernel.org/doc/html/v5.4/admin-guide/kernel-parameters.html>

<sup>630</sup> <https://www.kernel.org/doc/html/v5.4/vm/numa.html>

<sup>631</sup> <https://www.kernel.org/doc/Documentation/vm/hugetlbpage.txt>

<sup>632</sup> <https://www.kernel.org/doc/Documentation/vm/hugetlbpage.txt>

<sup>633</sup> <https://linux.die.net/man/8/hugeadm>

```
$ hugeadm --list-all-mounts
```

| Mount Point    | Options                 |
|----------------|-------------------------|
| /dev/hugepages | rw,relatime,pagesize=2M |

```
$ hugeadm --explain
```

## Huge page usage in libvirt

```
<memoryBacking>
 <hugepages/>
</memoryBacking>
```

libvirt memory backing documentation<sup>635</sup> Numa nodes<sup>634</sup>

## Controlling addressing bits

### -hpb machine types

provided special machine-types<sup>636</sup>

*HPB*

<sup>634</sup> <https://www.kernel.org/doc/html/v5.4/vm/numa.html>

<sup>635</sup> <https://libvirt.org/formatdomain.html#memory-backing>

<sup>636</sup> <https://bugs.launchpad.net/bugs/1776189>

```
$ cat /proc/cpuinfo | grep '^address sizes'

...
an older server with a E5-2620
address sizes : 46 bits physical, 48 bits virtual
a laptop with an i7-8550U
address sizes : 39 bits physical, 48 bits virtual
```

## maxphysaddr guest configuration

CPU

model and topology section<sup>637</sup>

```
<maxphysaddr mode='passthrough' />
```

```
<maxphysaddr mode='emulate' bits='42' />
```

```
<maxphysaddr mode='passthrough' limit='41' />
```

## AppArmor isolation

AppArmor rules for a guest<sup>638</sup>

- 
- 
- 

<sup>637</sup> <https://libvirt.org/formatdomain.html#cpu-model-and-topology>

<sup>638</sup> <https://gitlab.com/apparmor/apparmor/-/wikis/Libvirt#implementation-overview>



- 

- 

rules<sup>639</sup>

- 

deny

guest XML<sup>640</sup>

## Sharing files between Host<->Guest

```
<memoryBacking>
 <access mode='shared' />
</memoryBacking>
```

```
<memoryBacking>
 <hugepages>
 <page size='2048' unit='KiB' />
 </hugepages>
 <access mode='shared' />
</memoryBacking>
```

```
<filesystem type='mount' accessmode='passthrough'>
 <driver type='virtiofs' />
 <source dir='/var/guests/h-virtiofs' />
 <target dir='myfs' />
</filesystem>
```

```
$ sudo mount -t virtiofs myfs /mnt/
```

<sup>639</sup> <https://gitlab.com/apparmor/apparmor/-/wikis/FAQ#what-is-default-deny-allow-listing>

<sup>640</sup> <https://libvirt.org/formatdomain.html>

**Note:**

While virtiofs works with  $\geq 20.10$  (Groovy),  $\geq 21.04$  (Hirsute) made it more convenient, especially in small environments (no hard requirement to specify guest Numa topology or use huge pages). If you need to set up on 20.10 or want more details, the libvirt [knowledge-base about virtiofs](https://libvirt.org/kbase/virtiofs.html)<sup>642</sup> provides additional information.

<sup>642</sup> <https://libvirt.org/kbase/virtiofs.html>

**Resources**

- [KVM home page](#)<sup>643</sup>
- [libvirt home page](#)<sup>644</sup>
  - [domains](#)<sup>645</sup> [storage](#)<sup>646</sup>
- [Ubuntu Wiki KVM](#)<sup>647</sup>
- [linux-kvm](#)<sup>648</sup>
- [Introduction to Memory Management in Linux](#)<sup>649</sup>

**Virtual Machine Manager**

[Virtual Machine Manager](#)<sup>650</sup>

<sup>641</sup> <https://libvirt.org/formatdomain.html#filesystems>

<sup>643</sup> [https://linux-kvm.org/page/Main\\_Page](https://linux-kvm.org/page/Main_Page)

<sup>644</sup> <https://libvirt.org/>

<sup>645</sup> <https://libvirt.org/formatdomain.html>

<sup>646</sup> <https://libvirt.org/formatstorage.html>

<sup>647</sup> <https://help.ubuntu.com/community/KVM>

<sup>648</sup> [https://www.linux-kvm.org/page/How\\_to\\_assign\\_devices\\_with\\_VT-d\\_in\\_KVM#Assigning\\_the\\_device](https://www.linux-kvm.org/page/How_to_assign_devices_with_VT-d_in_KVM#Assigning_the_device)

<sup>649</sup> <https://www.youtube.com/watch?v=7aONIVSXiJ8>

<sup>650</sup> <https://virt-manager.org/>

## Install virt-manager

```
$ sudo apt install virt-manager
```

*Graphical User Interface (GUI)*

```
$ virt-manager
```

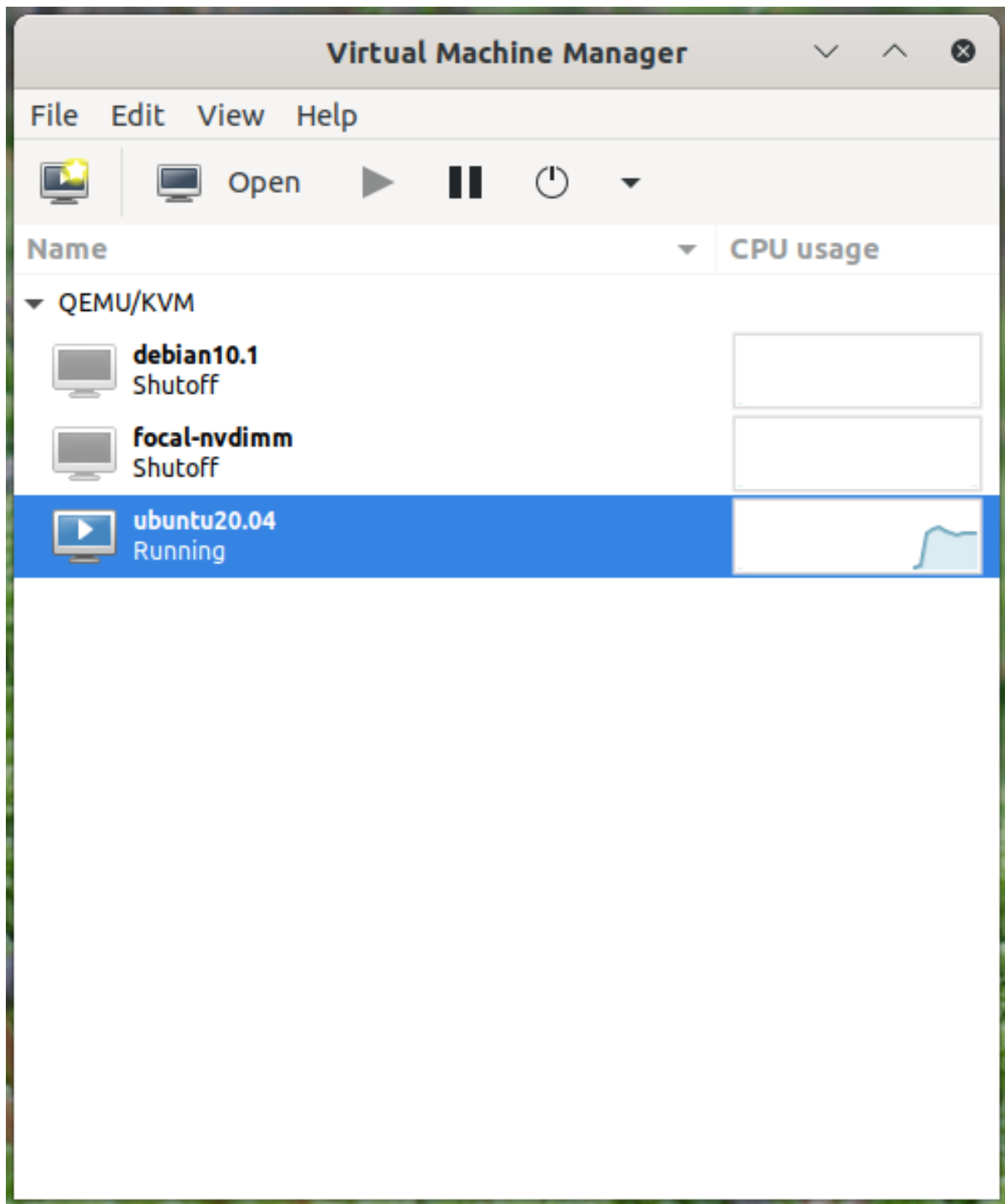
```
$ virt-manager -c qemu+ssh://virtnode1.mydomain.com/system
```

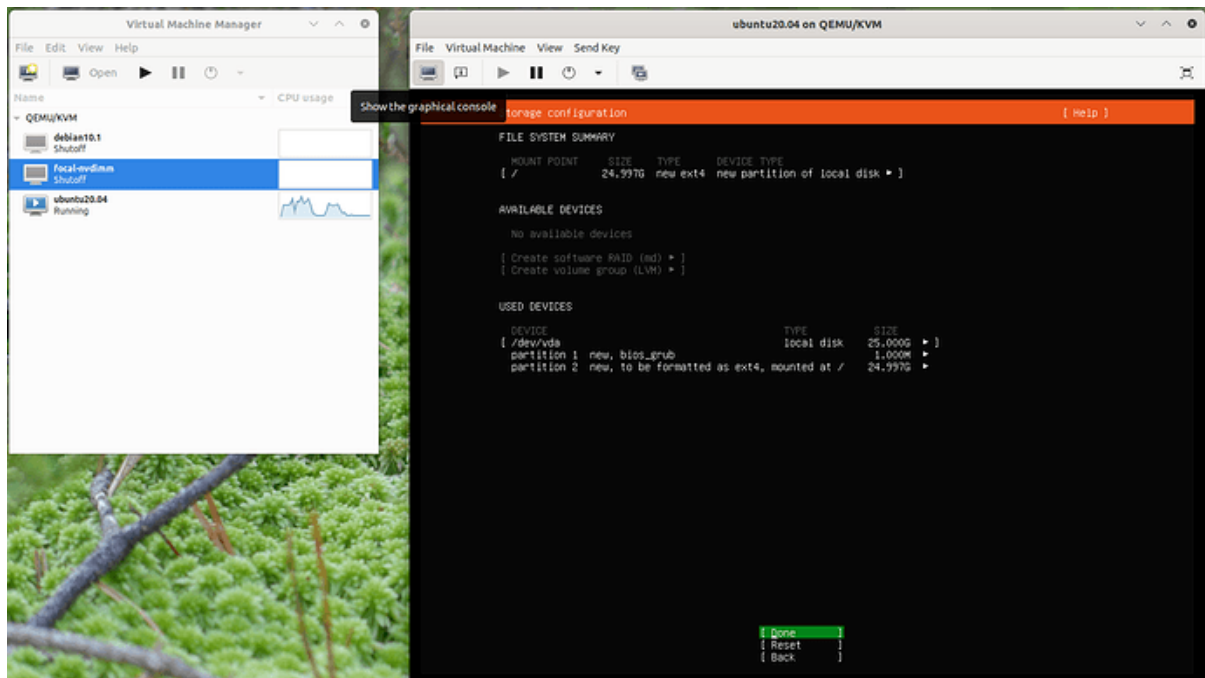
### Note:

The above example assumes that SSH connectivity between the management system and the target system has already been configured, and uses **SSH keys** for authentication. SSH keys are needed because libvirt sends the password prompt to another process. See our guide on OpenSSH for details on [how to set up SSH keys](#) (page 237).

## Use virt-manager to manage guests

### Guest lifecycle

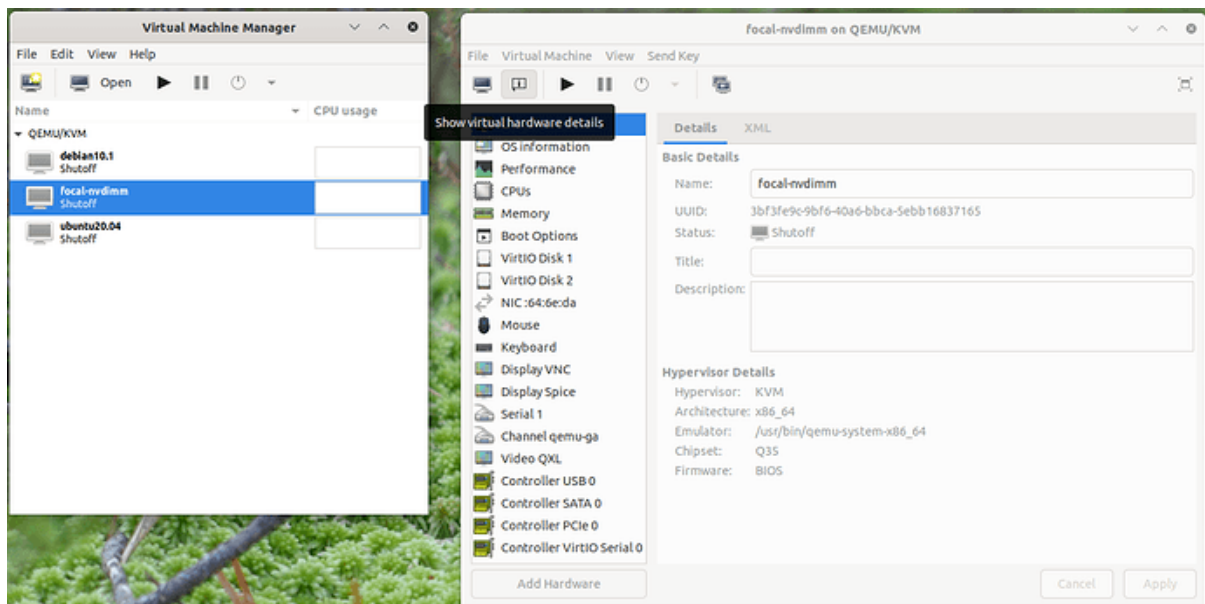




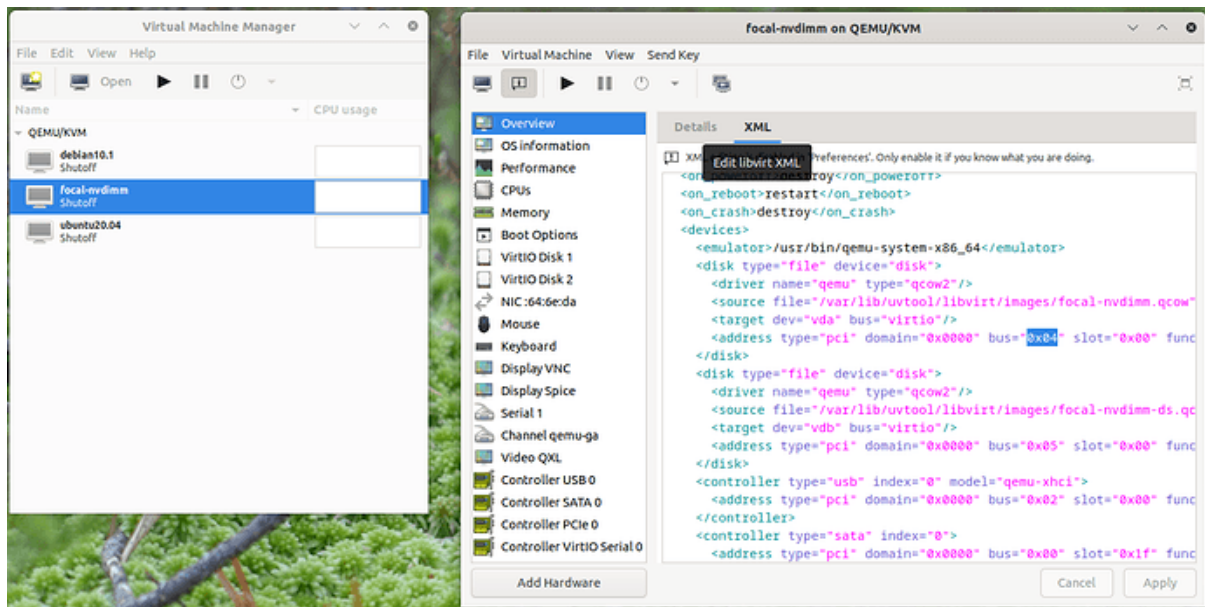
## Guest modification

tion<sup>651</sup>

guest representa-



<sup>651</sup> <https://libvirt.org/formatdomain.html>



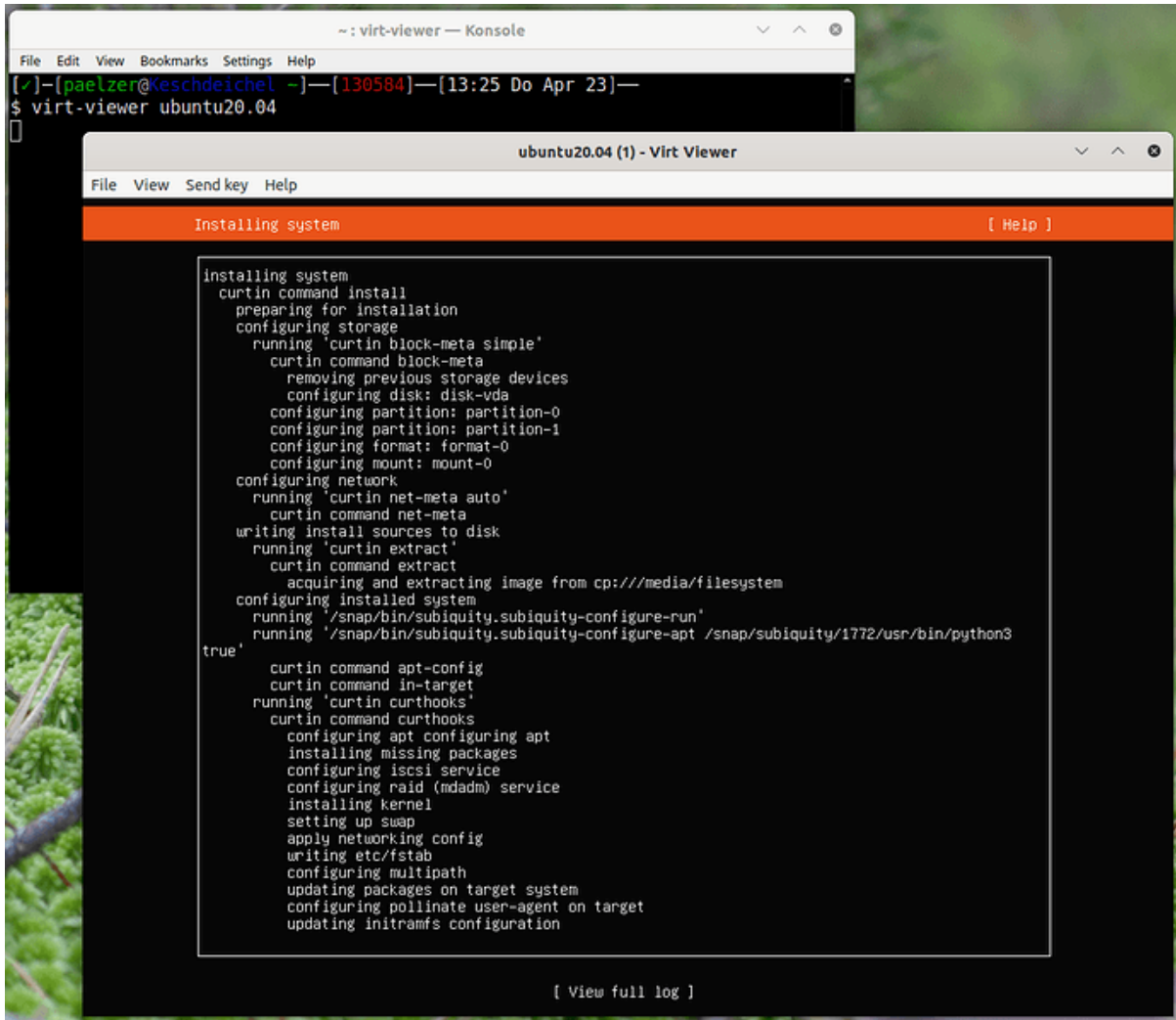
*libvirt-client*

## Virtual Machine Viewer (virt-viewer)

### Install virt-viewer

```
$ sudo apt install virt-viewer
```

```
$ virt-viewer <guestname>
```



```

~: virt-viewer — Konsole
File Edit View Bookmarks Settings Help
[✓]—[paelzer@keschdeichel ~]—[130584]—[13:25 Do Apr 23]—
$ virt-viewer ubuntu20.04

ubuntu20.04 (1) - Virt Viewer
File View Send key Help

Installing system [Help]

installing system
 curtin command install
 preparing for installation
 configuring storage
 running 'curtin block-meta simple'
 curtin command block-meta
 removing previous storage devices
 configuring disk: disk-vda
 configuring partition: partition-0
 configuring partition: partition-1
 configuring format: format-0
 configuring mount: mount-0
 configuring network
 running 'curtin net-meta auto'
 curtin command net-meta
 writing install sources to disk
 running 'curtin extract'
 curtin command extract
 acquiring and extracting image from cp:///media/filesystem
 configuring installed system
 running '/snap/bin/subiquity.subiquity-configure-run'
 running '/snap/bin/subiquity.subiquity-configure-apt /snap/subiquity/1772/usr/bin/python3
true'
 curtin command apt-config
 curtin command in-target
 running 'curtin curthooks'
 curtin command curthooks
 configuring apt configuring apt
 installing missing packages
 configuring iscsi service
 configuring raid (mdadm) service
 installing kernel
 setting up swap
 apply networking config
 writing etc/fstab
 configuring multipath
 updating packages on target system
 configuring pollinate user-agent on target
 updating initramfs configuration

[View full log]

```

```
$ virt-viewer -c qemu+ssh://virtnode1.mydomain.com/system <guestname>
```

## virt-install

### Install virt-install

```
$ sudo apt install virtinst
```

```
$ virt-install \ > --name web_devel \ > --ram 8192 \ > --disk
path=/home/doug/vm/web_devel.img,bus=virtio,size=50 \ > --cdrom
focal-desktop-amd64.iso \ > --network network=default,model=virtio \ >
--graphics vnc,listen=0.0.0.0 \ > --noautoconsole \ > --hvm \ > --vcpus=4
```

*virt-install(1)*<sup>652</sup>

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 

---

<sup>652</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/virt-install.1.html>



## virt-clone

```
$ virt-clone --auto-clone --original focal
```

- 
- 

### Caution:

Please be aware that this is a full clone, therefore any secrets, keys, and for example, `/etc/machine-id`, will be shared. This will cause issues with security and anything that needs to identify the machine like [DHCP](#). You most likely want to edit those afterward and de-duplicate them as needed.

## Resources

- [KVM](#)<sup>653</sup>
- [libvirt home page](#)<sup>654</sup>
- [Virtual Machine Manager](#)<sup>655</sup>

## How to enable nested virtualization

### Important:

Nested virtualization is enabled by default on Ubuntu. If you are using Ubuntu, it's unlikely that you will need to manually enable the feature. If you check (using the steps below) and discover that nested virtualization is enabled, then you will not need to do anything further.

<sup>653</sup> [https://linux-kvm.org/page/Main\\_Page](https://linux-kvm.org/page/Main_Page)

<sup>654</sup> <https://libvirt.org/>

<sup>655</sup> <https://virt-manager.org/>

## Check if nested virtualization is enabled

*AMD*

```
$ lsmod | grep -i kvm

kvm_intel 204800 0
kvm 1347584 1 kvm_intel
```

## If the module is loaded

```
$ cat /sys/module/kvm_intel/parameters/nested
```

```
$ cat /sys/module/kvm_amd/parameters/nested
```

## If the module is not loaded

```
$ modprobe kvm-intel nested=1
```

```
$ modprobe kvm-amd nested=1
```

## Enable nested virtualization

```
$ sudo modprobe -r <module>
```

```
$ sudo modprobe <module>
```

### Note:

Make sure to stop all running VMs that might use the kernel module and prevent its reload.

```
$ sudo modprobe -r kvm-intel
```

```
$ sudo modprobe kvm-intel
```

```
$ cat /sys/module/kvm_intel/parameters/nested
```

## Check and enable nested virtualization inside a VM

```
$ egrep "svm|vmx" /proc/cpuinfo
```

- 
-

- 

on the OpenStack wiki<sup>656</sup>

```
<cpu mode='host-model' check='partial' />
```

- 

## Limitations of nested virtualization

- *How to use the libvirt library with virsh*
- *How to use virt-manager and other virt\* tools*
- *How to enable nested virtualisation*

## Ubuntu in other virtual environments

### How to set up Ubuntu on Hyper-V

type 1 hypervisor<sup>657</sup> *ESXi*

- 

- 

- 

for Linux<sup>658</sup>

Windows Subsystem

## Hyper-V system requirements

system requirements for Hyper-V<sup>659</sup>

- 

- 

*BIOS*

- 

<sup>656</sup> <https://wiki.openstack.org/wiki/LibvirtXMLCPUModel>

<sup>657</sup> <https://en.wikipedia.org/wiki/Hypervisor#Classification>

<sup>658</sup> <https://learn.microsoft.com/windows/wsl/about>

<sup>659</sup> <https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/system-requirements-for-hyper-v-on-windows>

•

## Install Hyper-V

graded<sup>660</sup>

would need to be up-

## Install Hyper-V on Windows client

### Install Hyper-V graphically

- 1.
- 2.
- 3.
- 4.
- 5.

### Install Hyper-V using PowerShell

- 1.
- 2.

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

- 3.

## Install Hyper-V on Windows Server

```
Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -Restart
```

installing the Hyper-V role on Windows Server<sup>661</sup>

<sup>660</sup> <https://support.microsoft.com/en-us/windows/upgrade-windows-home-to-windows-pro-ef34d520-e73f-3198-c525-d1a218>

<sup>661</sup> <https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/get-started/install-the-hyper-v-role-on-windows-server>

## Install Ubuntu on Hyper-V

### Using Quick Create

#### Note:

Quick Create is only available on Windows 10 and Windows 11 **client** editions. It is not available on Windows Server. If you are running Windows Server, use the [New Virtual Machine Wizard](#) (page 686) method instead.

1.

2.

- 

- 

3.

4.

- 

- 

5.

6.

7.

8.

## Using the New Virtual Machine Wizard

1. [official Ubuntu source](#)<sup>662</sup>
- 2.
- 3.
- 4.
5.
  - 
  - 
  - 
  - 
  - 
  -
- 6.
- 7.
- 8.
- 9.

### Further reading

[creating a virtual machine in Hyper-V](#)<sup>663</sup>

- [Setting up Ubuntu on Hyper-V](#)

---

<sup>662</sup> <https://ubuntu.com/download/server>

<sup>663</sup> <https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/get-started/create-a-virtual-machine-in-hyper-v>

## See also

- [Virtualisation and containers](#)

### [Containers](#)

- 
- 

## Containers

- [LXD](#)
- [Docker for sysadmins](#)
- [How to run rocks on your server](#)

## Docker for system admins

## Installation

```
$ sudo apt-get install -y docker.io docker-compose
```

### Note:

This guide uses the Docker packages provided by Ubuntu (`docker.io`), which are supported by Canonical. The instructions and examples below are written for these packages. Canonical also provides a [Docker snap](#)<sup>664</sup>. This guide does not cover the snap installation path.

---

<sup>664</sup> <https://snapcraft.io/docker>



## Configuring storage

### How to configure volumes

- 

```
$ docker volume create my-vol

my-vol
```

- 

```
$ docker volume ls

DRIVER VOLUME NAME
local my-vol
```

- 

```
$ docker volume inspect my-vol

[
 {
 "CreatedAt": "2023-10-25T00:53:24Z",
 "Driver": "local",
 "Labels": null,
 "Mountpoint": "/var/lib/docker/volumes/my-vol/_data",
 "Name": "my-vol",
 "Options": null,
 "Scope": "local"
 }
]
```

- 

```
$ docker run --name web-server -d \ > --mount source=my-vol,target=/app
 \ > ubuntu/apache2

0709c1b632801fddd767deddda0d273289ba423e9228cc1d77b2194989e0a882
```

- 

```
$ docker inspect web-server --format '{{ json .Mounts }}' | jq .
```

```
[
 {
 "Type": "volume",
 "Name": "my-vol",
 "Source": "/var/lib/docker/volumes/my-vol/_data",
 "Destination": "/app",
 "Driver": "local",
 "Mode": "z",
 "RW": true,
 "Propagation": ""
 }
]
```

- 

```
$ docker stop web-server
```

```
$ docker rm web-server
```

```
$ docker volume rm my-vol
```

## How to configure bind mounts

- 

```
$ docker run -d \ > --name web-server \ > --mount
type=bind,source="$(pwd)",target=/app \ > ubuntu/apache2
6f5378e34d6c6811702e16d047a5a80f18adbd9d8a14b11050ae3c3353bf8d2a
```

- 

```
$ docker inspect web-server --format '{{ json .Mounts }}' | jq .
[
 {
 "Type": "bind",
 "Source": "/root",
 "Destination": "/app",
 "Mode": "",
 "RW": true,
 "Propagation": "rprivate"
 }
]
```

-

```
$ docker stop web-server
```

```
$ docker rm web-server
```

## How to configure tmpfs

- 

```
$ docker run --name web-server -d \ > --mount type=tmpfs,target=/app \ > ubuntu/apache2
```

```
03483cc28166fc5c56317e4ee71904941ec5942071e7c936524f74d732b6a24c
```

- 

```
$ docker inspect web-server --format '{{ json .Mounts }}' | jq .
```

```
[
 {
 "Type": "tmpfs",
 "Source": "",
 "Destination": "/app",
 "Mode": "",
 "RW": true,
 "Propagation": ""
 }
]
```

## Choosing the right storage drivers

- 

```
$ docker info | grep "Storage Driver"
```

```
Storage Driver: overlayfs
```

- 

```
$ apt install zfsutils-linux -y # Install ZFS
```

```
$ fallocate -l 5G /zfs-pool.img # Create a 5GB file
```

```
$ zpool create mypool /zfs-pool.img # Create a ZFS pool
```

```
$ zfs create -o mountpoint=/var/lib/docker mypool/docker # Create a ZFS dataset and mount it to dockers directory, "/var/lib/docker".
```

```
$ zfs list # Verify that it mounted successfully
```

| NAME          | USED | AVAIL | REFER | MOUNTPOINT      |
|---------------|------|-------|-------|-----------------|
| mypool        | 162K | 4.36G | 24K   | /mypool         |
| mypool/docker | 39K  | 4.36G | 39K   | /var/lib/docker |

- 

- 1.

```
$ systemctl stop docker
```

- 2.

```
$ vim /etc/docker/daemon.json
```

```
{
 "storage-driver": "zfs"
}
```

- 3.

```
$ systemctl restart docker
```

- 

```
$ docker info | grep "Storage Driver"
```

```
Storage Driver: zfs
```

## Configuring networking

- 

```
$ docker network create --driver bridge my-net
```

```
D84efaca11d6f643394de31ad8789391e3ddf29d46faecf0661849f5ead239f7
```

-

```
$ docker network ls
```

| NETWORK ID   | NAME   | DRIVER | SCOPE |
|--------------|--------|--------|-------|
| 1f55a8891c4a | bridge | bridge | local |
| 9ca94be2c1a0 | host   | host   | local |
| d84efaca11d6 | my-net | bridge | local |
| 5d300e6a07b1 | none   | null   | local |

```
$ docker network inspect my-net
```

```
[
 {
 "Name": "my-net",
 "Id": "d84efaca11d6f643394de31ad8789391e3ddf29d46faecf0661849f5ead239f7",
 "Created": "2023-10-25T22:18:52.972569338Z",
 "Scope": "local",
 "Driver": "bridge",
 "EnableIPv6": false,
 "IPAM": {
 "Driver": "default",
 "Options": {},
 "Config": [
 {
 "Subnet": "172.18.0.0/16",
 "Gateway": "172.18.0.1"
 }
]
 },
 "Internal": false,
 "Attachable": false,
 "Ingress": false,
 "ConfigFrom": {
 "Network": ""
 },
 "ConfigOnly": false,
 "Containers": {},
 "Options": {},
 "Labels": {}
 }
]
```

## Connecting a new container to an existing network

```
$ docker run -d --name c1 --network my-net ubuntu/apache2
C7aa78f45ce3474a276ca3e64023177d5984b3df921aadf97e221da8a29a891e
```

•

```
$ docker inspect c1 --format '{{ json .NetworkSettings }}' | jq .
{
 "Bridge": "",
 "SandboxID":
 "ee1cc10093fd5d4a30c056cef47abbfa564e770272e1e5f681525fdd85555",
 "HairpinMode": false,
 "LinkLocalIPv6Address": "",
 "LinkLocalIPv6PrefixLen": 0,
 "Ports": {
 "80/tcp": null
 },
 "SandboxKey": "/var/run/docker/netns/ee1cc10093fd",
 "SecondaryIPAddresses": null,
 "SecondaryIPv6Addresses": null,
 "EndpointID": "",
 "Gateway": "",
 "GlobalIPv6Address": "",
 "GlobalIPv6PrefixLen": 0,
 "IPAddress": "",
 "IPPrefixLen": 0,
 "IPv6Gateway": "",
 "MacAddress": "",
 "Networks": {
 "my-net": {
 "IPAMConfig": null,
 "Links": null,
 "Aliases": ["c7aa78f45ce3"],
 "NetworkID":
 "d84efaca11d6f643394de31ad8789391e3ddf29d46faecf0661849f5ead239f7",
 "EndpointID":
 "1cb76d44a484d302137bb4b042c8142db8e931e0c63f44175a1aa75ae8af9cb5",
 "Gateway": "172.18.0.1",
 "IPAddress": "172.18.0.2",
 "IPPrefixLen": 16,
 "IPv6Gateway": "",
 "GlobalIPv6Address": "",
 "GlobalIPv6PrefixLen": 0,
 "MacAddress": "02:42:ac:12:00:02",
 "DriverOpts": null
 }
 }
}
```

## Connecting a running container to an existing network

- 

- 1.

```
$ docker run -d --name c2 ubuntu/nginx
Fea22fbb6e3685eae28815f3ad8c8a655340ebcd6a0c13f3aad0b45d71a20935
```

- 2.

```
$ docker network connect my-net c2
```

```
$ docker inspect c2 --format '{{ json .NetworkSettings }}' | jq .
```

```
{
 "Bridge": "",
 "SandboxID":
"82a7ea6efd679dffcc3e4392e0e5da61a8cccf33dd78eb5381c9792a4c01f366",
 "HairpinMode": false,
 "LinkLocalIPv6Address": "",
 "LinkLocalIPv6PrefixLen": 0,
 "Ports": {
 "80/tcp": null
 },
 "SandboxKey": "/var/run/docker/netns/82a7ea6efd67",
 "SecondaryIPAddresses": null,
 "SecondaryIPv6Addresses": null,
 "EndpointID":
"490c15cf3bcb149dd8649e3ac96f71addd13f660b4ec826dc39e266184b3f65b",
 "Gateway": "172.17.0.1",
 "GlobalIPv6Address": "",
 "GlobalIPv6PrefixLen": 0,
 "IPAddress": "172.17.0.3",
 "IPPrefixLen": 16,
 "IPv6Gateway": "",
 "MacAddress": "02:42:ac:11:00:03",
 "Networks": {
 "bridge": {
 "IPAMConfig": null,
 "Links": null,
 "Aliases": null,
 "NetworkID":
"1f55a8891c4a523a288aca8881dae0061f9586d5d91c69b3a74e1ef3ad1bfcf4",
 "EndpointID":
"490c15cf3bcb149dd8649e3ac96f71addd13f660b4ec826dc39e266184b3f65b",
 "Gateway": "172.17.0.1",
 "IPAddress": "172.17.0.3",
 "IPPrefixLen": 16,
 "IPv6Gateway": "",
 "GlobalIPv6Address": "",
 "GlobalIPv6PrefixLen": 0,
 "MacAddress": "02:42:ac:11:00:03",
 "DriverOpts": null
 },
 "my-net": {
 "IPAMConfig": {},
 "Links": null,
 "Aliases": ["fea22fbb6e36"],
 "NetworkID":
"d84efaca11d6f643394de31ad8789391e3ddf29d46faecf0661849f5ead239f7",
 "EndpointID":
"17856b7f6902db39ff6ab418f127d75d8da597fdb8af0a6798f35a94be0cb805",
 "Gateway": "172.18.0.1",
 "IPAddress": "172.18.0.3",
 "IPPrefixLen": 16,
 "IPv6Gateway": "",
 "GlobalIPv6Address": "",
 "GlobalIPv6PrefixLen": 0,
 "MacAddress": "02:42:ac:12:00:03",
 "DriverOpts": {}
 }
 }
}
```



## Modifying the default network “bridge”

- 

```
$ vim /etc/docker/daemon.json

{
 "bip": "192.168.1.1/24",
 "fixed-cidr": "192.168.1.0/25",
 "fixed-cidr-v6": "2001:db8::/64",
 "mtu": 1500,
 "default-gateway": "192.168.1.254",
 "default-gateway-v6": "2001:db8:abcd::89",
 "dns": ["10.20.1.2", "10.20.1.3"]
}
```

- 

```
$ systemctl restart docker
```

- 

```
$ docker network inspect bridge
```

## Exposing a container port to the host

- 

-

- 

- 

```
$ docker run -d --name web-server --publish 8080:80 ubuntu/nginx
f451aa1990db7d2c9b065c6158e2315997a56a764b36a846a19b1b96ce1f3910
```

- 

```
$ docker inspect web-server --format '{{ json .NetworkSettings.Ports }}'
| jq .
{
 "80/tcp": [
 {
 "HostIp": "0.0.0.0",
 "HostPort": "8080"
 },
 {
 "HostIp": ":::",
 "HostPort": "8080"
 }
]
}
```

*hostname*

*DNS*

## Managing logs

### Modifying the logging driver via the docker daemon file

- 

```
$ vim /etc/docker/daemon.json

{
 "log-driver": "journald"
}
```

### Modifying the logging driver when creating a container

- 

```
$ docker run -d --name web-server --log-driver=journald ubuntu/nginx
1c08b667f32d8b834f0d9d6320721e07de5f22168cfc8a024d6e388daf486dfa
```

- 

```
$ docker inspect web-server --format '{{ json .HostConfig.LogConfig }}'
| jq .
{
 "Type": "journald",
 "Config": {}
}
```

- 

```
$ docker logs web-server

/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to
perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-
templates.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
```

```
{
 "log-driver": "syslog",
 "log-opts": {
 "cache-disabled": "true",
 "syslog-address": "udp://1.2.3.4:1111"
 }
}
```

```
$ docker logs web-server
```

```
Error response from daemon: configured logging driver does not support reading
```

## Further reading

- *Docker storage, networking, and logging*

## How to run rocks on your server

## Deploying rocks with Docker

rocks<sup>666</sup>

*Docker*

```
$ sudo apt-get install -y docker.io docker-compose
```

<sup>665</sup> <https://docs.docker.com/engine/logging/configure/>

<sup>666</sup> <https://documentation.ubuntu.com/rockcraft/latest/explanation/rocks/>

possible parameters<sup>667</sup>

“Usage” section<sup>668</sup>

```
$ sudo docker run -d --name grafana-container -e TZ=UTC -p 3000:3000
ubuntu/grafana:10.3.3-22.04_stable

Unable to find image 'ubuntu/grafana:10.3.3-22.04_stable' locally
10.3.3-22.04_stable: Pulling from ubuntu/grafana
bccd10f490ab: Already exists
549078d9d057: Pull complete
6ef870aa8500: Pull complete
2b475da7ccbd: Pull complete
Digest: sha256:df566ef90ecb14267a459081949ee7b6693fa573b97a7134a9a6722207275caa
Status: Downloaded newer image for ubuntu/grafana:10.3.3-22.04_stable
356e623ef2c16bc7d810bddccad8d7980f9c633aefc3a88bc8761eac4e1b1c50
```

- 
- 
- 
- 
- 

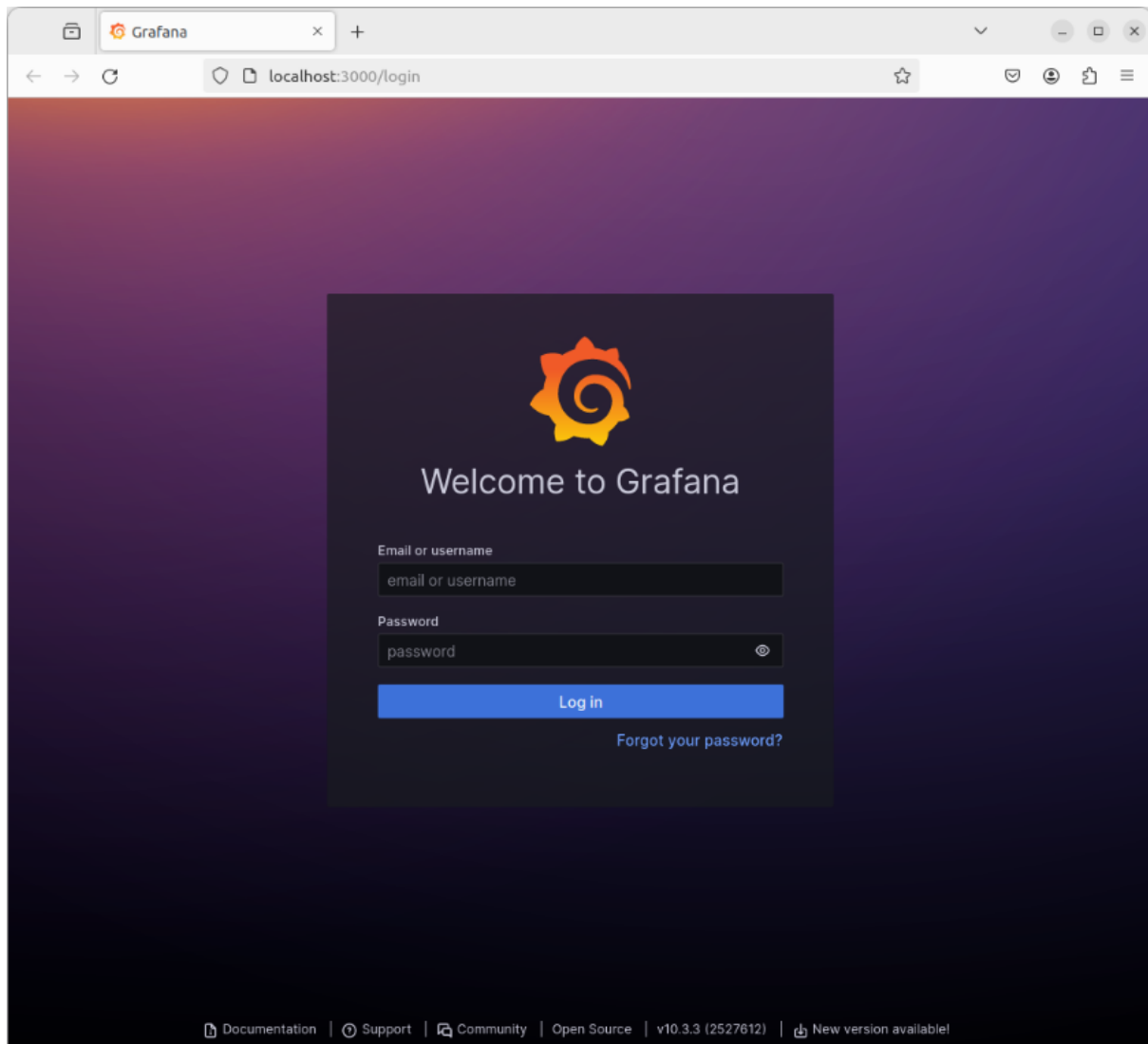
snap “channels”<sup>669</sup>

---

<sup>667</sup> <https://docs.docker.com/reference/cli/docker/container/run/>

<sup>668</sup> <https://hub.docker.com/r/ubuntu/grafana>

<sup>669</sup> <https://snapcraft.io/docs/channels>



```
$ curl -s http://localhost:3000/login | grep "<title>"
<title>Grafana</title>
```

```
$ sudo docker ps
```

| CONTAINER ID | IMAGE                              | PORTS         | COMMAND                                   | NAMES             | CREATED        |
|--------------|------------------------------------|---------------|-------------------------------------------|-------------------|----------------|
| 356e623ef2c1 | ubuntu/grafana:10.3.3-22.04_stable |               | "/bin/pebble enter -..."                  |                   | 17 minutes ago |
|              |                                    | Up 17 minutes | 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp | grafana-container |                |

```
$ sudo docker stop grafana-container
```

```
$ sudo docker rm grafana-container
```

Grafana rock's documentation<sup>670</sup>

## Multi-container deployment

Grafana<sup>672</sup>

Prometheus<sup>671</sup>

```
global:
 scrape_interval: 1m

scrape_configs:
 - job_name: 'prometheus'
 scrape_interval: 1m
 static_configs:
 - targets: ['localhost:9090']
```

```
services:
 grafana:
 image: ubuntu/grafana:10.3.3-22.04_stable
 container_name: grafana-container
 environment:
 TZ: UTC
 ports:
 - "3000:3000"
 prometheus:
 image: ubuntu/prometheus:2.49.1-22.04_stable
 container_name: prometheus-container
 environment:
 TZ: UTC
 ports:
 - "9090:9090"
```

(continues on next page)

<sup>670</sup> <https://hub.docker.com/r/ubuntu/grafana>

<sup>671</sup> <https://hub.docker.com/r/ubuntu/prometheus>

<sup>672</sup> <https://hub.docker.com/r/ubuntu/grafana>

**volumes**

```
- ./prometheus.yml:/etc/prometheus/prometheus.yml
```

```
$ sudo docker compose up -d

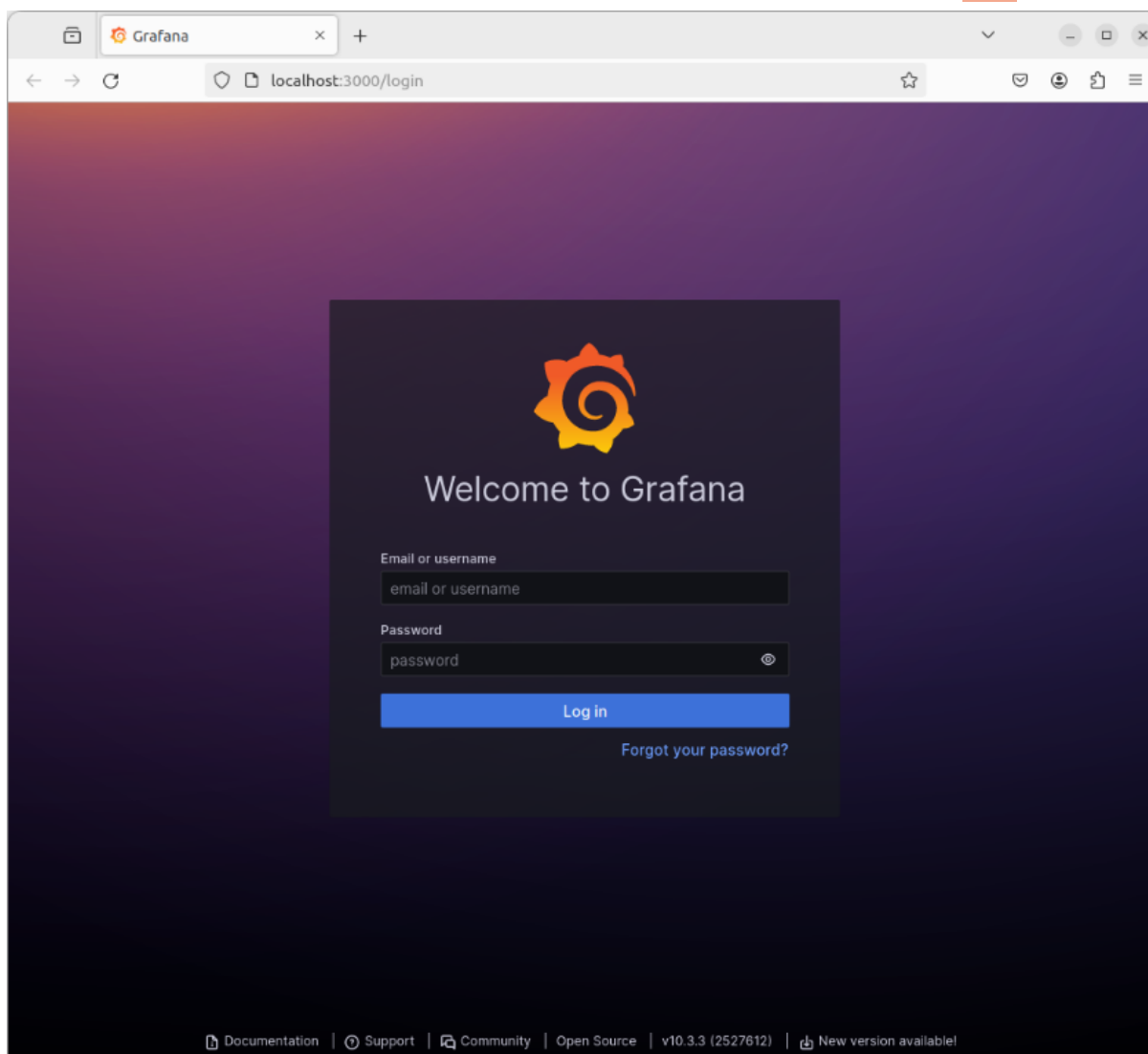
[+] Running 10/10
 ✓ grafana Pulled
 ✓ bccd10f490ab Already exists
 ✓ 549078d9d057 Pull complete
 ✓ 6ef870aa8500 Pull complete
 ✓ 2b475da7ccbd Pull complete
 ✓ prometheus Pulled
 ✓ a8b1c5f80c2d Already exists
 ✓ f021062473aa Pull complete
 ✓ 9c6122d12d1d Pull complete
 ✓ 274b56f68abe Pull complete
[+] Running 3/3
 ✓ Network compose_default Created
 ✓ Container prometheus-container Started
 ✓ Container grafana-container Started
```

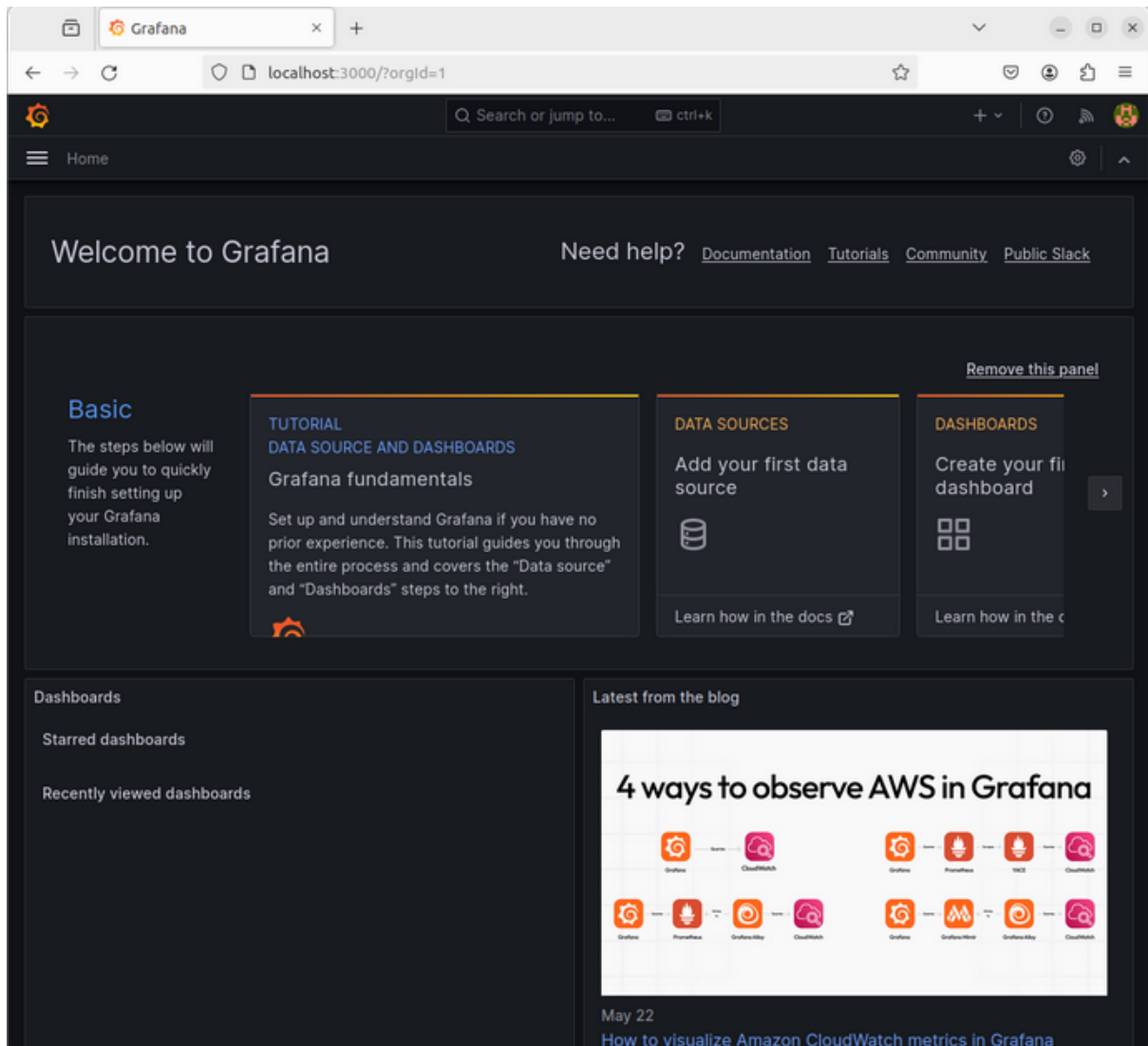
```
$ sudo docker compose ps
```

| NAME                 | SERVICE    | IMAGE                                 | STATUS       | PORTS                                     | COMMAND                 |
|----------------------|------------|---------------------------------------|--------------|-------------------------------------------|-------------------------|
| grafana-container    | grafana    | ubuntu/grafana:10.3.3-22.04_stable    | Up 3 seconds | 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp | "/bin/pebble enter -... |
| prometheus-container | prometheus | ubuntu/prometheus:2.49.1-22.04_stable | Up 3 seconds | 0.0.0.0:9090->9090/tcp, :::9090->9090/tcp | "/bin/pebble enter -... |

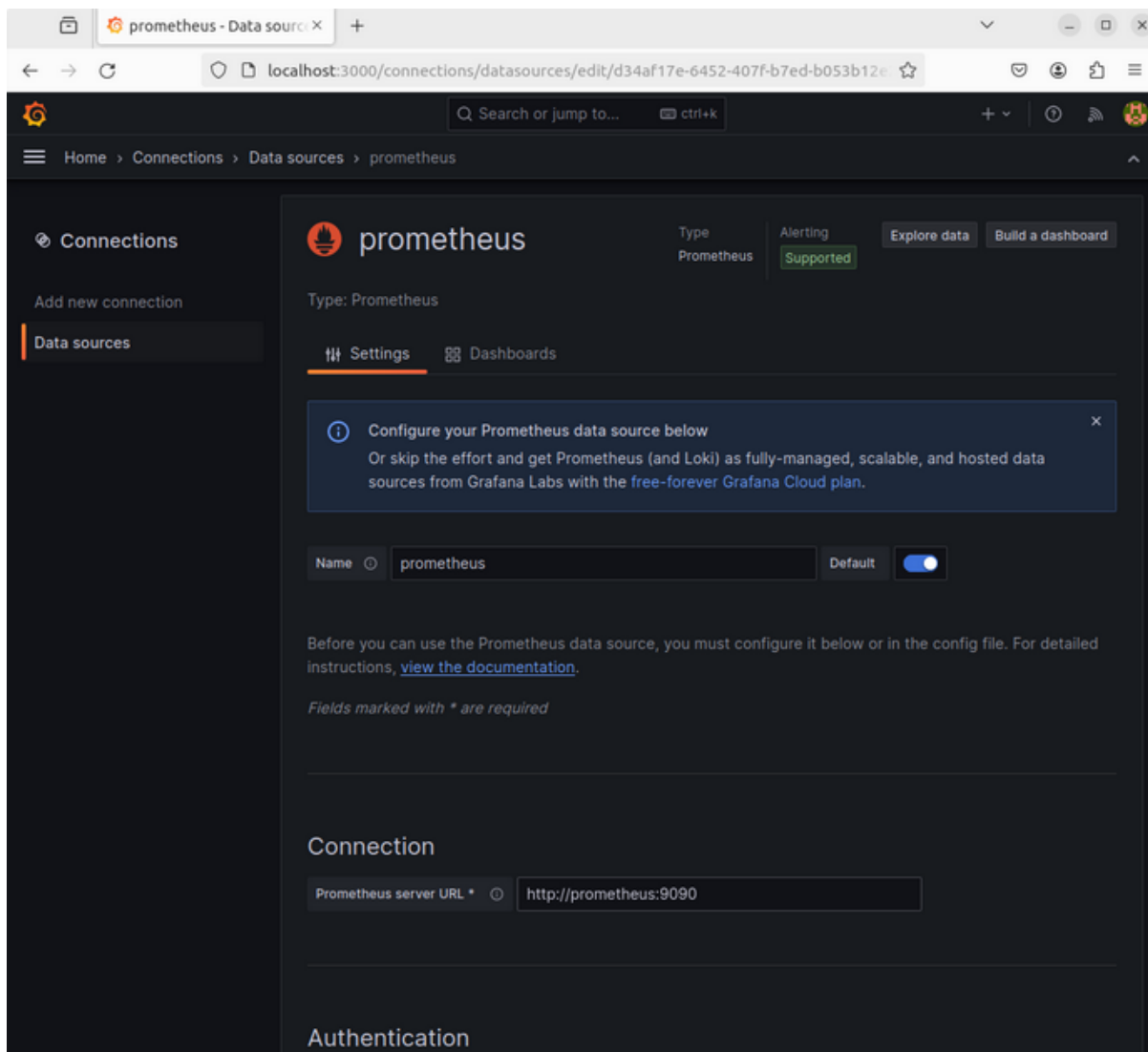
<http://localhost:3000>







<http://prometheus:9090>



The screenshot shows the Grafana web interface for configuring a Prometheus data source. The browser address bar indicates the URL is `localhost:3000/connections/datasources/edit/d34af17e-6452-407f-b7ed-b053b12e`. The left sidebar shows the navigation menu with 'Connections' and 'Data sources' (the latter is highlighted). The main content area is titled 'prometheus' and includes tabs for 'Settings' (active) and 'Dashboards'. A notification box at the top of the settings area provides instructions on how to configure the data source or mentions Grafana Cloud. Below this, the 'Name' field is set to 'prometheus' and the 'Default' toggle is turned on. A text block explains that configuration is required before using the data source, with a link to the documentation. The 'Connection' section contains a 'Prometheus server URL' field with the value `http://prometheus:9090`. The 'Authentication' section is partially visible at the bottom.

prometheus - Data source

localhost:3000/connections/datasources/edit/d34af17e-6452-407f-b7ed-b053b12e

Search or jump to... ctrl+k

Home > Connections > Data sources > prometheus

Connections

Add new connection

Data sources

prometheus

Type: Prometheus

Alerting: Supported

Explore data Build a dashboard

Type: Prometheus

Settings Dashboards

Configure your Prometheus data source below

Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the [free-forever Grafana Cloud plan](#).

Name prometheus Default

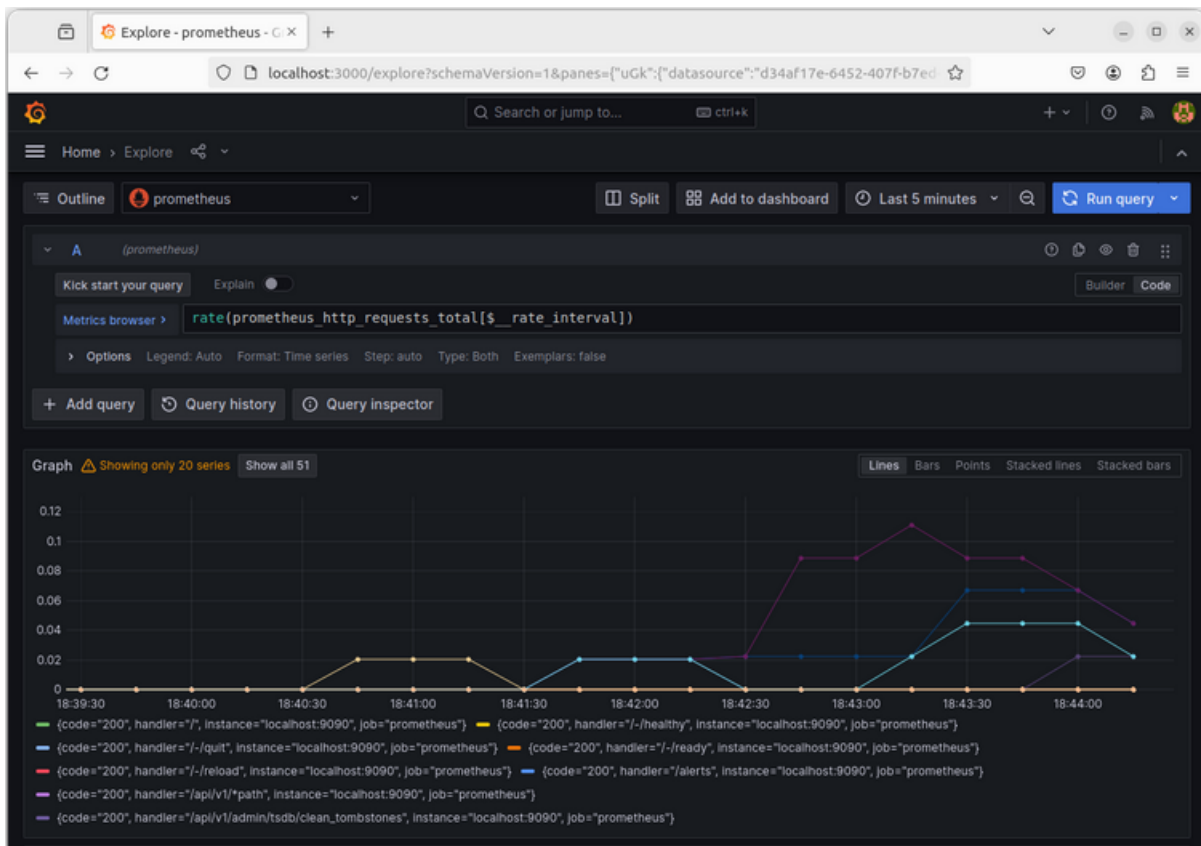
Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#).

Fields marked with \* are required

Connection

Prometheus server URL \* http://prometheus:9090

Authentication



## Next steps

microk8s<sup>673</sup>

## See also

- [Virtualisation and containers](#)

## High Availability

<sup>673</sup> <https://canonical.com/microk8s/docs>

## High availability

- *Install a Distributed Replicated Block Device*

### Distributed Replicated Block Device (DRBD)

#### Install DRBD

```
$ sudo apt install drbd-utils
```

#### Note:

If you are using the **virtual kernel** as part of a virtual machine you will need to manually compile the drbd module. It may be easier to install the `linux-modules-extra-$(uname -r)` package inside the virtual machine.

#### Configure DRBD

*filesystem*

*DNS*

*guide to DNS*

```
global { usage-count no; }
common { syncer { rate 100M; } }
resource r0 {
 protocol C;
 startup {
 wfc-timeout 15;
 degr-wfc-timeout 60;
 }
 net {
 cram-hmac-alg sha1;
 shared-secret "secret";
 }
}
```

(continues on next page)

(continued from previous page)

```
}
on drbd01 {
 device /dev/drbd0;
 disk /dev/sdb1;
 address 192.168.0.1 7788;
 meta-disk internal;
}
on drbd02 {
 device /dev/drbd0;
 disk /dev/sdb1;
 address 192.168.0.2 7788;
 meta-disk internal;
}
}
```

#### Note:

There are many other options in `/etc/drbd.conf`, but for this example the default values are enough.

```
$ scp /etc/drbd.conf drbd02:~
```

```
$ sudo mv drbd.conf /etc/
```

```
$ sudo drbdadm create-md r0
```

```
$ sudo systemctl start drbd.service
```

```
$ sudo drbdadm -- --overwrite-data-of-peer primary all
```

```
$ watch -n1 cat /proc/drbd
```

```
$ sudo mkfs.ext3 /dev/drbd0
```

```
$ sudo mount /dev/drbd0 /srv
```

## Testing

```
$ sudo cp -r /etc/default /srv
```

```
$ sudo umount /srv
```

```
$ sudo drbdadm secondary r0
```

```
$ sudo drbdadm primary r0
```

```
$ sudo mount /dev/drbd0 /srv
```

## Further reading

- [DRBD web site](#)<sup>674</sup>
- [\*drbd.conf\(5\)\*](#)<sup>675</sup>
- [\*drbdadm\(8\)\*](#)<sup>676</sup>

<sup>674</sup> <https://linbit.com/linbit-software-download-page-for-linstor-and-drbd-linux-driver/>

<sup>675</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/drbd.conf.5.html>

<sup>676</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/drbdadm.8.html>

## See also

- [High Availability](#)
- [High availability](#)

## Observability

Canonical Observability Stack<sup>677</sup>

## Observability

Canonical Observability Stack<sup>678</sup>

- [How to migrate from LMA to COS Lite](#)<sup>679</sup>
- [Integrate COS Lite with un-charmed applications](#)<sup>680</sup>
- [The OpenTelemetry Collector snap](#)<sup>681</sup>

## Classical options

## Set up your LMA stack

### Note:

#### LMA to COS

The LMA stack is being succeeded by the Canonical Observability Stack (COS). While the current LMA still works, most users are recommended to consider COS instead. For more information, refer to [this COS topic](#)<sup>682</sup>. In environments with more limited resources, there is also [COS lite](#)<sup>683</sup>.

<sup>682</sup> <https://charmhub.io/topics/canonical-observability-stack/>

<sup>683</sup> <https://charmhub.io/topics/canonical-observability-stack/editions/lite>

<sup>677</sup> <https://charmhub.io/topics/canonical-observability-stack>

<sup>678</sup> <https://documentation.ubuntu.com/observability/track-2/>

<sup>679</sup> <https://documentation.ubuntu.com/observability/track-2/how-to/migrate-lma-to-cos-lite/>

<sup>680</sup> <https://documentation.ubuntu.com/observability/track-2/how-to/integrating-cos-lite-with-uncharmed-applications/>

<sup>681</sup> <https://snapcraft.io/opentelemetry-collector>



## Architectural overview

Telegraf<sup>684</sup>

Prometheus<sup>685</sup>

Grafana<sup>686</sup>

Prometheus Alertmanager<sup>687</sup>

*Alertmanager*

## Getting started

### Note:

For clarity, we'll refer to these two hosts as named: `workload` and `monitor`. If you use other *hostnames*, substitute your preferred names as we go through this guide.

<sup>684</sup> <https://docs.influxdata.com/telegraf/v1/>

<sup>685</sup> <https://prometheus.io/docs/introduction/overview/>

<sup>686</sup> <https://grafana.com/docs/>

<sup>687</sup> <https://prometheus.io/docs/alerting/latest/alertmanager/>

## Ports

| Prometheus   | monitor:9090  |
|--------------|---------------|
| Alertmanager | monitor:9093  |
| Grafana      | monitor:3000  |
| Telegraf     | workload:9273 |

## Set up the Workload node

```
$ lxc launch ubuntu:20.10 workload
```

```
Creating workload
Starting workload
```

```
$ lxc exec workload -- bash
```

```
workload:~# apt update && apt install telegraf
```

```
[[inputs.cpu]]
 percpu = true
 totalcpu = true
 collect_cpu_time = false
 report_active = false
```

```
[[outputs.prometheus_client]]
 listen = "workload:9273"
 metric_version = 2

#[outputs.influxdb]
```

```
workload:~# systemctl restart telegraf
workload:~# systemctl status telegraf
[] telegraf.service - The plugin-driven server agent for reporting metrics into InfluxDB
 Loaded: loaded (/lib/systemd/system/telegraf.service; enabled; vendor preset:
 enabled)
 Active: active (running) since Sat 2020-10-31 02:17:57 UTC; 6s ago
 Docs: https://github.com/influxdata/telegraf
 Main PID: 2562 (telegraf)
 Tasks: 17 (limit: 77021)
 Memory: 42.2M
 CGroup: /system.slice/telegraf.service
 └─2562 /usr/bin/telegraf -config /etc/telegraf/telegraf.conf -
config-directory /etc/telegraf/telegraf.d

...I! Loaded inputs: swap system cpu disk diskio kernel mem processes
...I! Loaded outputs: http prometheus_client
...I! [agent] Config: Interval:10s, Quiet:false, Hostname:"workload", Flush Interval:10s
...I! [outputs.prometheus_client] Listening on http://127.0.0.1:9273/metrics
```

```
workload:~# wget -O- http://workload:9273/metrics

HELP cpu_usage_guest Telegraf collected metric
TYPE cpu_usage_guest gauge
cpu_usage_guest{cpu="cpu-total",host="workload"} 0
cpu_usage_guest{cpu="cpu0",host="workload"} 0
cpu_usage_guest{cpu="cpu1",host="workload"} 0
cpu_usage_guest{cpu="cpu10",host="workload"} 0
...
cpu_usage_idle{cpu="cpu-total",host="workload"} 92.74914376428686
cpu_usage_idle{cpu="cpu0",host="workload"} 86.72897196325539
cpu_usage_idle{cpu="cpu1",host="workload"} 90.11857707405758
cpu_usage_idle{cpu="cpu10",host="workload"} 95.95141700494543
```

## Set up the Monitor node

```
$ lxc launch ubuntu:20.10 monitor
```

```
Creating monitor
Starting monitor
```

```
$ lxc exec monitor -- bash
```

```
monitor:~# ip addr | grep 'inet .* global'
inet 10.69.244.104/24 brd 10.69.244.255 scope global dynamic eth0
```

```
monitor:~# wget -O- http://workload:9273/metrics
```

```
monitor:~# apt install snapd
```

## Install Prometheus

```
monitor:~# snap install prometheus
monitor:~# snap install prometheus-alertmanager
```

```
monitor:~# snap services
Service Startup Current Notes
lxd.activate enabled inactive -
lxd.daemon enabled inactive socket-activated
prometheus.prometheus enabled active -
prometheus-alertmanager.alertmanager enabled active -
```

```
visualizer:~# ss -tulpn | grep prometheus
tcp LISTEN 0 128 *:9090 *:.* users:(("prometheus",pid=618,fd=8))
```

```
monitor:~# journalctl | grep "prometheus.*config"
...
...msg="Completed loading of configuration file" filename=/var/snap/prometheus/32/
prometheus.yml
```

```
scrape_configs:
 # The job name is added as a label `job=<job_name>` to any timeseries scraped from this
 # config.
 - job_name: 'prometheus'

 # metrics_path defaults to '/metrics'
 # scheme defaults to 'http'.

 static_configs:
 - targets: ['localhost:9090']

 - job_name: 'telegraf'
 static_configs:
 - targets: ['workload:9273']
```

```
monitor:~# snap restart prometheus
```

Prometheus Time Series Collection and Processing Server - Mozilla Firefox

Prometheus Time Series x +

10.69.244.104:9090/graph

Prometheus Alerts Graph Status ▾ Help

☐ Enable query history [Try experimental React UI](#)

Expression (press Shift+Enter for newlines)

Execute - insert metric at cursor - ▾

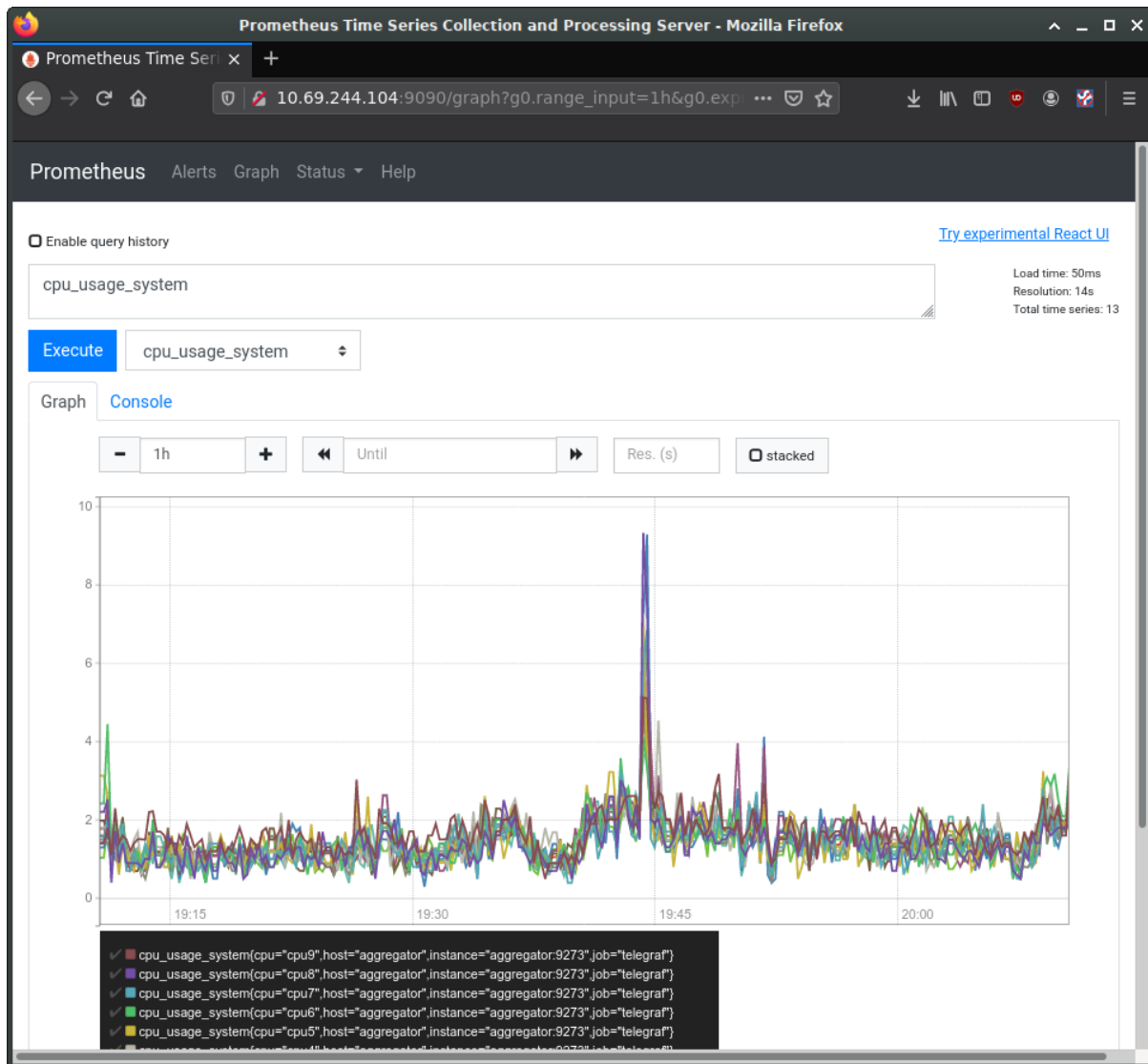
Graph Console

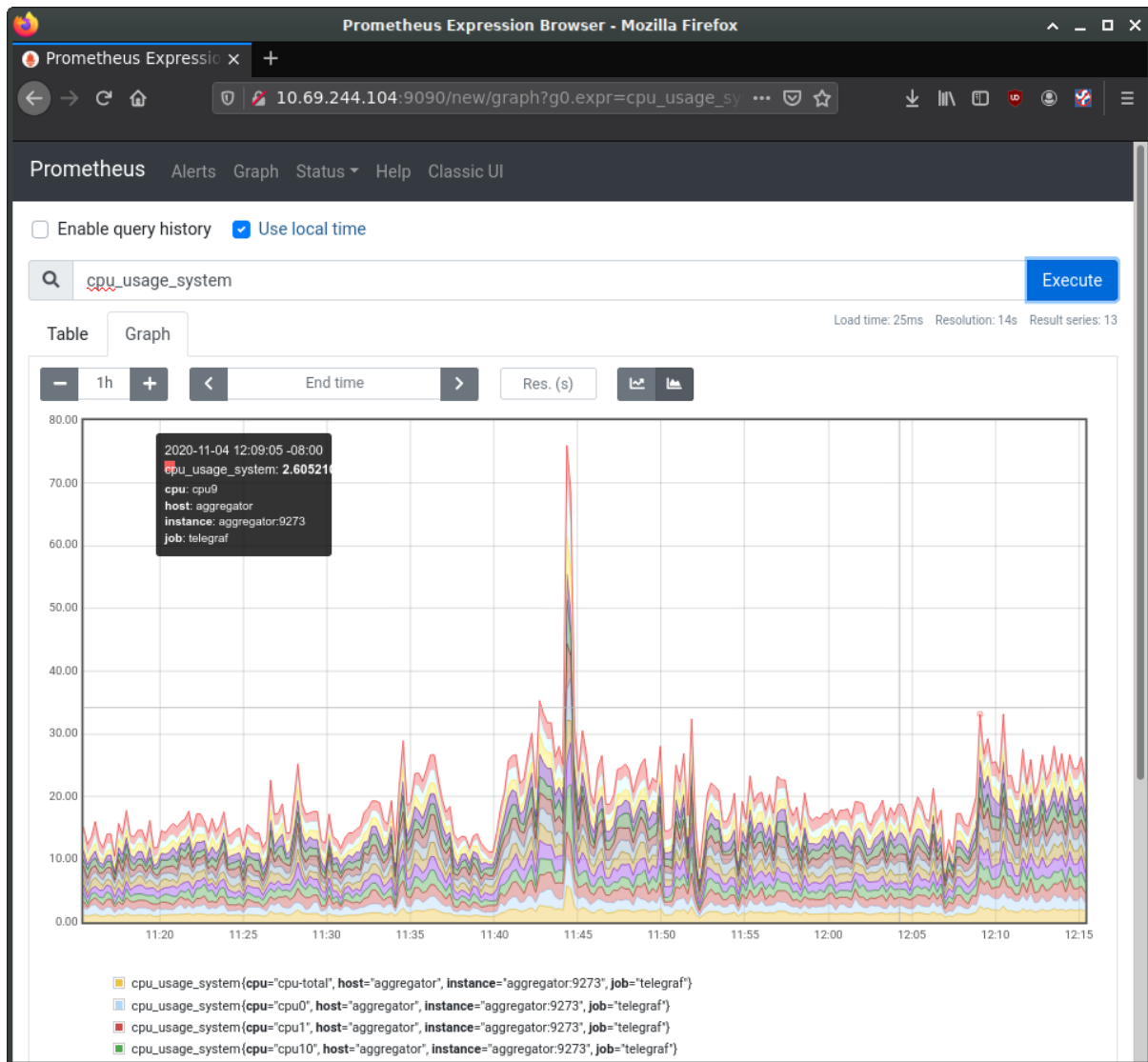
◀ Moment ▶

| Element | Value |
|---------|-------|
| no data |       |

[Remove Graph](#)

Add Graph





## Configure Alertmanager

```
/var/snap/prometheus/current/prometheus.yml
#...
Alertmanager configuration
alerting:
 alertmanagers:
 - static_configs:
 - targets:
 - 127.0.0.1:9093
rule_files:
 - 'alerts.yml'
```

```
/var/snap/prometheus/current/alerts.yml
```

(continues on next page)



(continued from previous page)

```
groups:
- name: demo-alerts
 rules:
 - alert: HighLoad
 expr: node_load1 > 2.0
 for: 60m
 labels:
 severity: normal
 annotations:
 description: '{{ $labels.instance }} of job {{ $labels.job }} is under high
load.'
 summary: Instance {{ $labels.instance }} under high load.
 value: '{{ $value }}'

 - alert: InstanceDown
 expr: up == 0
 for: 5m
 labels:
 severity: major
 annotations:
 summary: "Instance {{ $labels.instance }} down"
 description: "{{ $labels.instance }} of job {{ $labels.job }} has been down for
more than 5 minutes."
```

```
/var/snap/prometheus-alertmanager/current/alertmanager.yml
global:
 resolve_timeout: 5m

route:
 group_by: ['alertname']
 group_wait: 10s
 group_interval: 10s
 repeat_interval: 1h

inhibit_rules:
- source_match:
 severity: 'critical'
 target_match:
 severity: 'warning'
 equal: ['alertname', 'dev', 'instance']
```

```
workload:~# snap restart prometheus-alertmanager
```

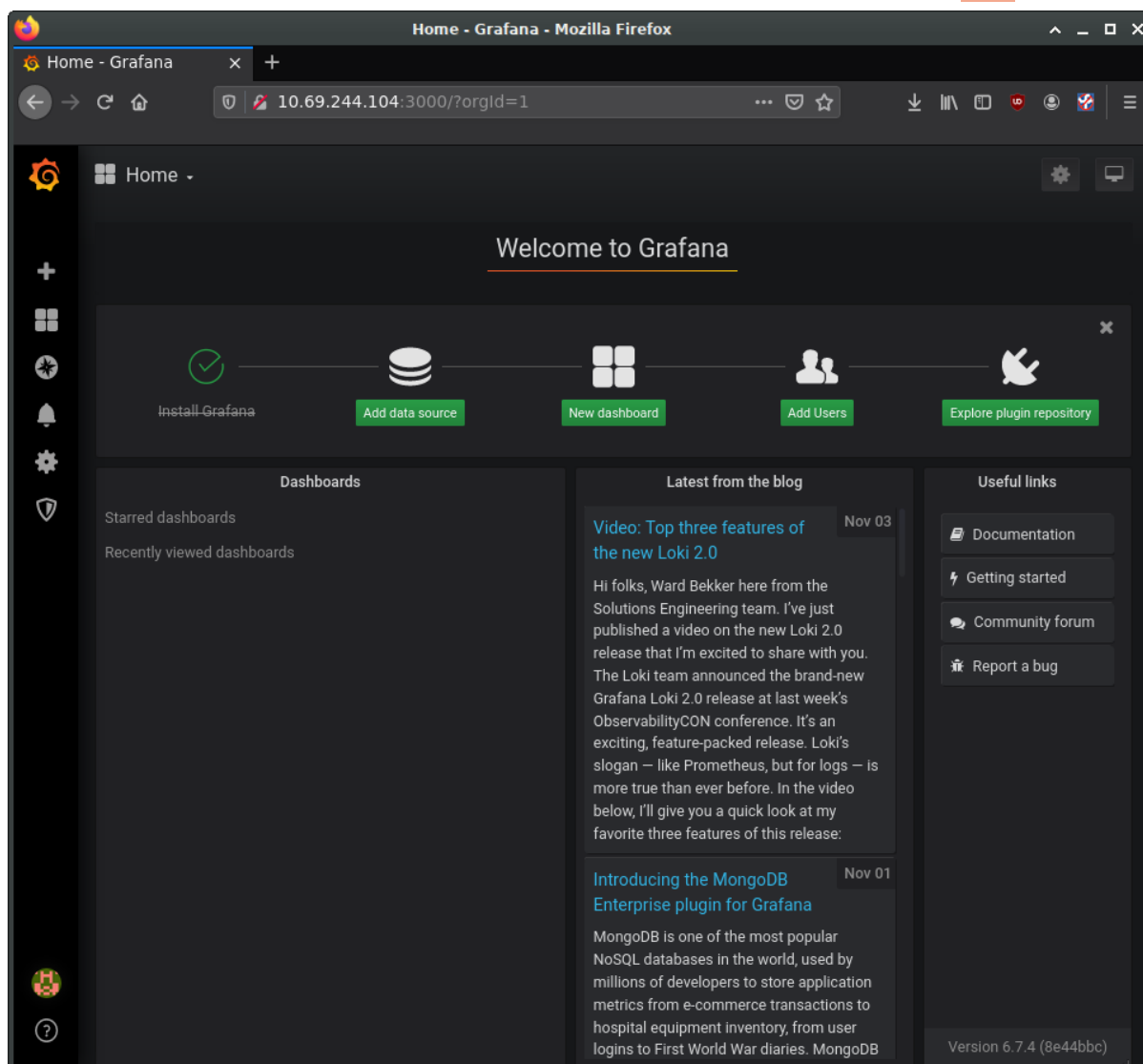
## Install Grafana

```
monitor:~# snap install grafana
grafana 6.7.4 from Alvaro Uria (aluria) installed
```

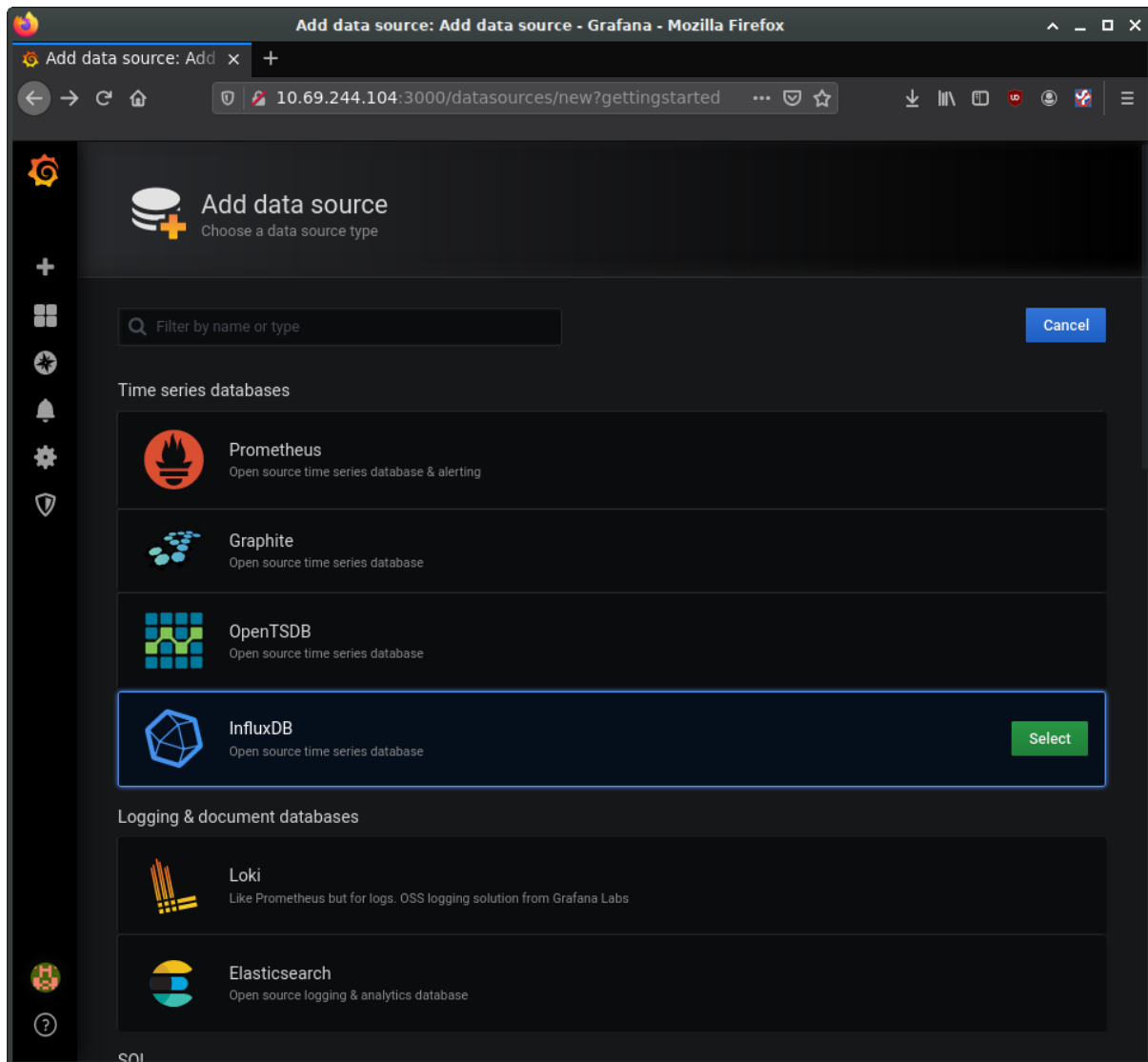
```
ss -tulpn | grep grafana
tcp LISTEN 0 128 *:3000 *:.* users:((
"grafana-server",pid=1449,fd=10))
```

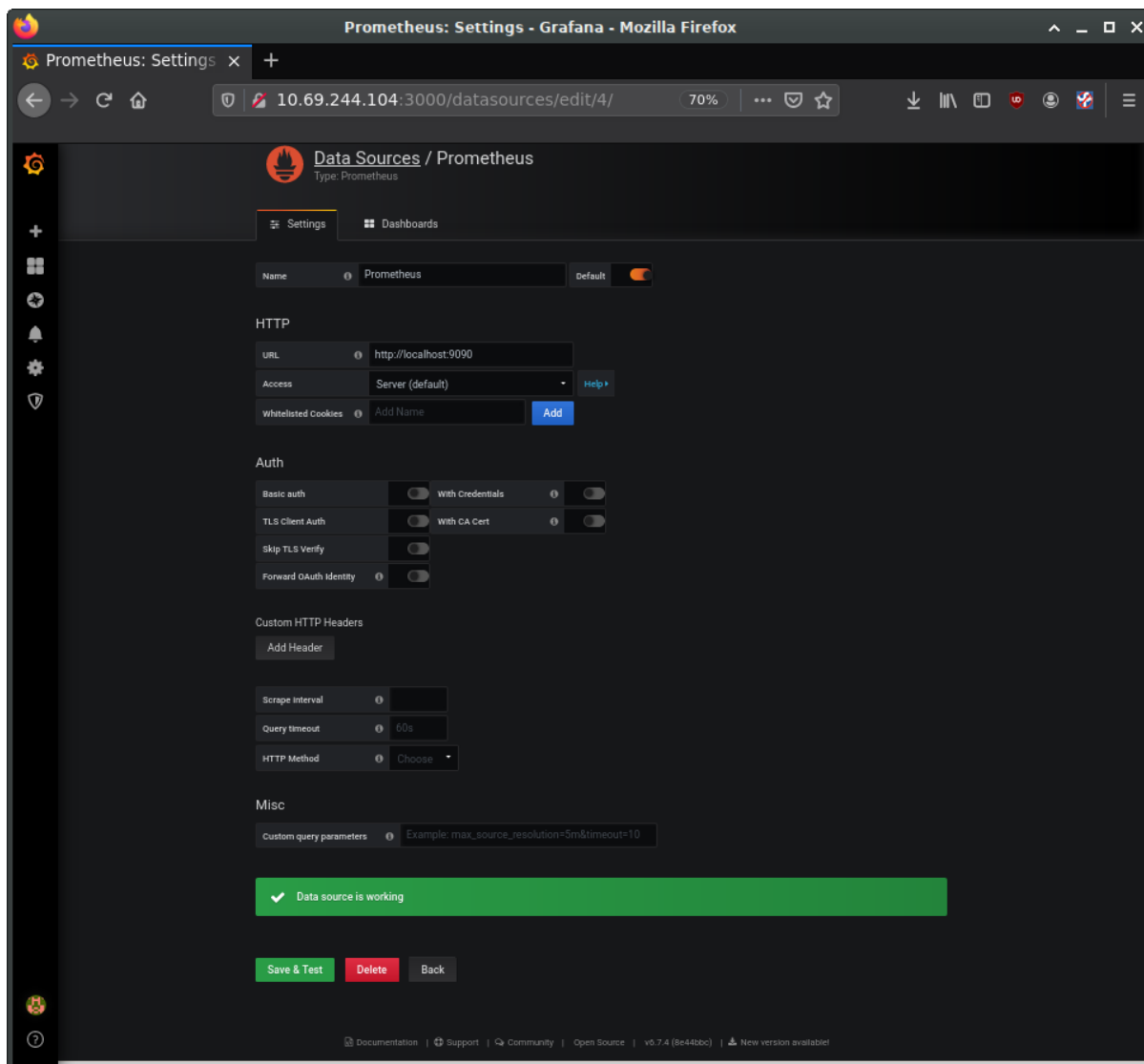
```
monitor:~# journalctl | grep "grafana.*conf"
... msg="Config loaded from" logger=settings file=/snap/grafana/36/conf/defaults.ini
... msg="Config overridden from Environment variable" logger=settings var="GF_PATHS_
PROVISIONING=/var/snap/grafana/common/conf/provisioning"
... error="open /var/snap/grafana/common/conf/provisioning/datasources: no such file or
directory"
...
```

```
$ firefox http://10.69.244.104:3000
```



- 
- 
-





The screenshot shows the Prometheus Settings page in Grafana, accessed via Mozilla Firefox. The browser address bar shows the URL `10.69.244.104:3000/datasources/edit/4/`. The page title is "Prometheus: Settings - Grafana - Mozilla Firefox".

The main content area is titled "Data Sources / Prometheus" with the subtitle "Type: Prometheus". Below this, there are tabs for "Settings" and "Dashboards".

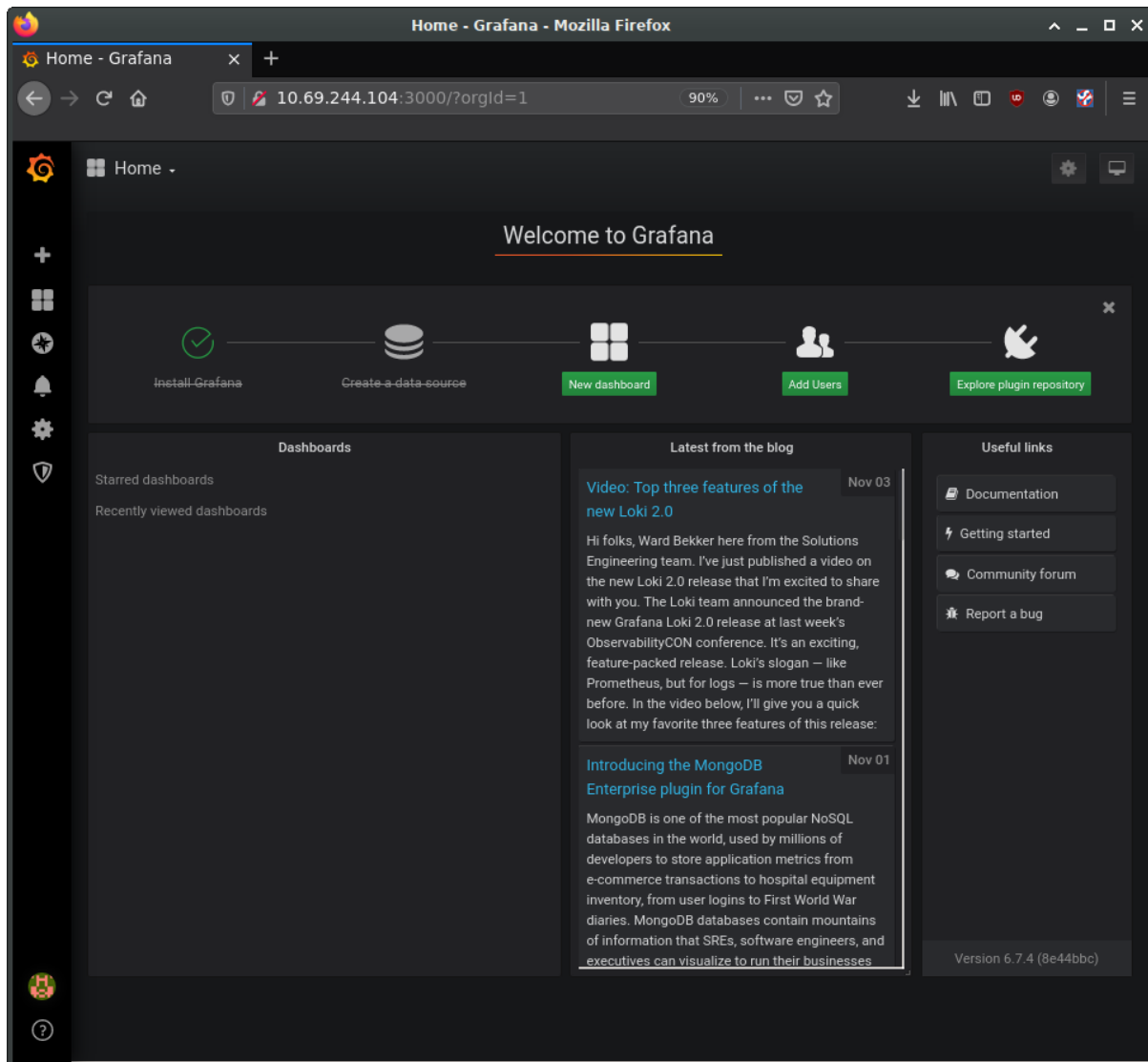
The "Settings" tab is active, showing the following configuration options:

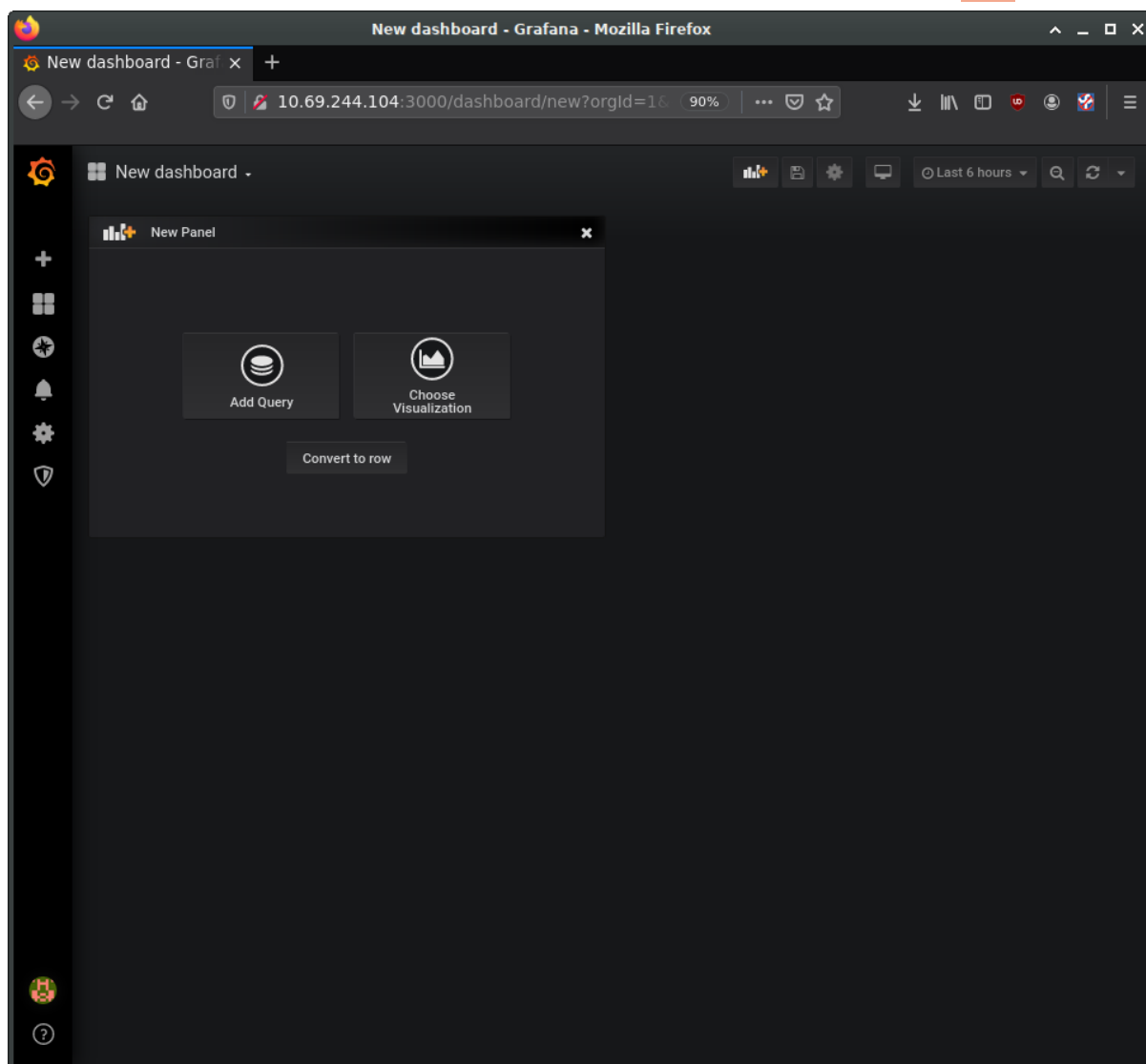
- Name:** Prometheus (Default is selected)
- HTTP:**
  - URL:** http://localhost:9090
  - Access:** Server (default) (Help icon)
  - Whitelisted Cookies:** Add Name (Add button)
- Auth:**
  - Basic auth:** With Credentials (toggle off)
  - TLS Client Auth:** With CA Cert (toggle off)
  - Skip TLS Verify:** (toggle off)
  - Forward OAuth Identity:** (toggle off)
- Custom HTTP Headers:** Add Header
- Scrape interval:** (input field)
- Query timeout:** 60s
- HTTP Method:** Choose
- Misc:** Custom query parameters (Example: max\_source\_resolution=5m&timeout=10)

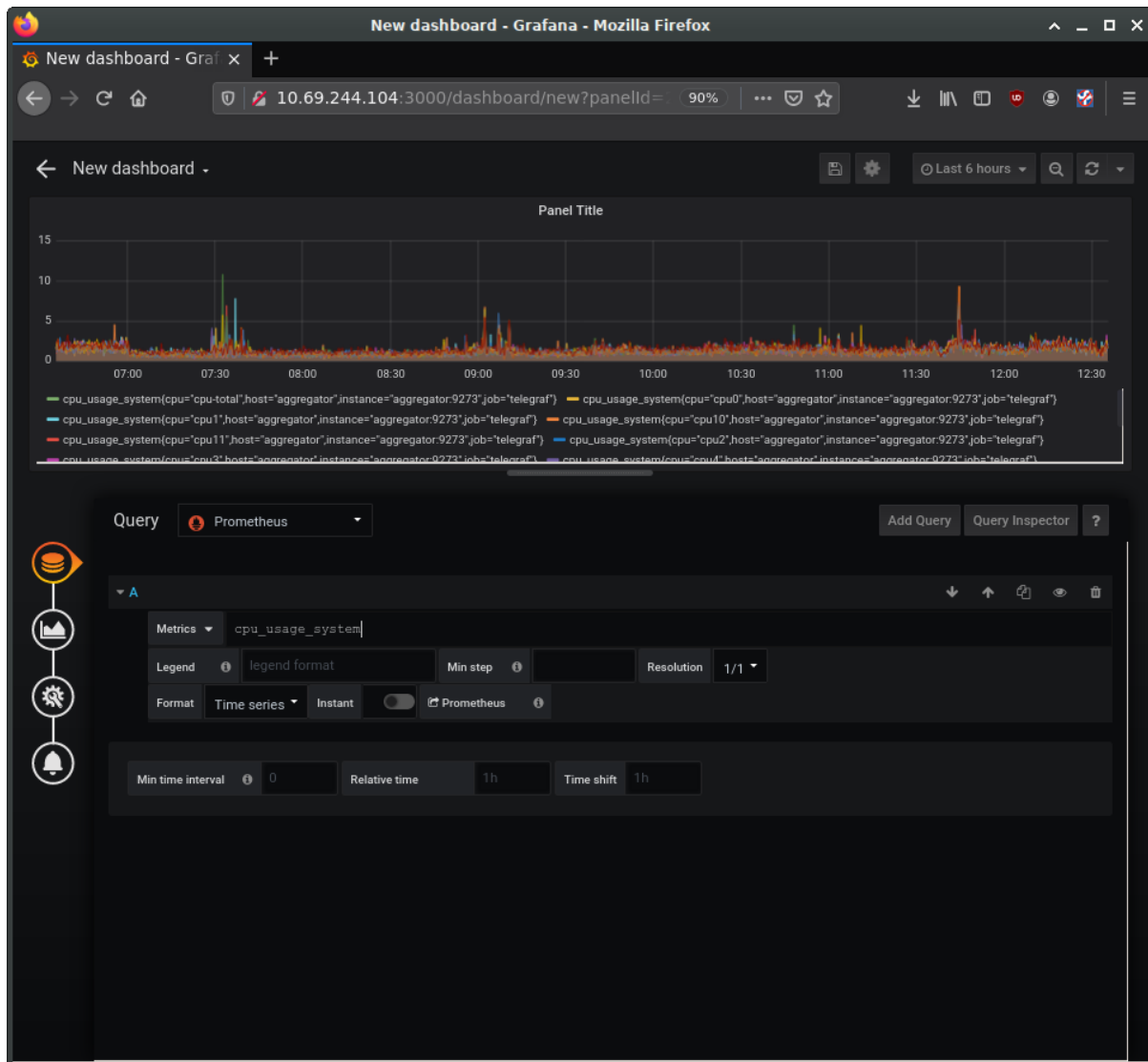
A green status bar at the bottom of the settings area indicates "Data source is working".

At the bottom of the page, there are buttons for "Save & Test", "Delete", and "Back".

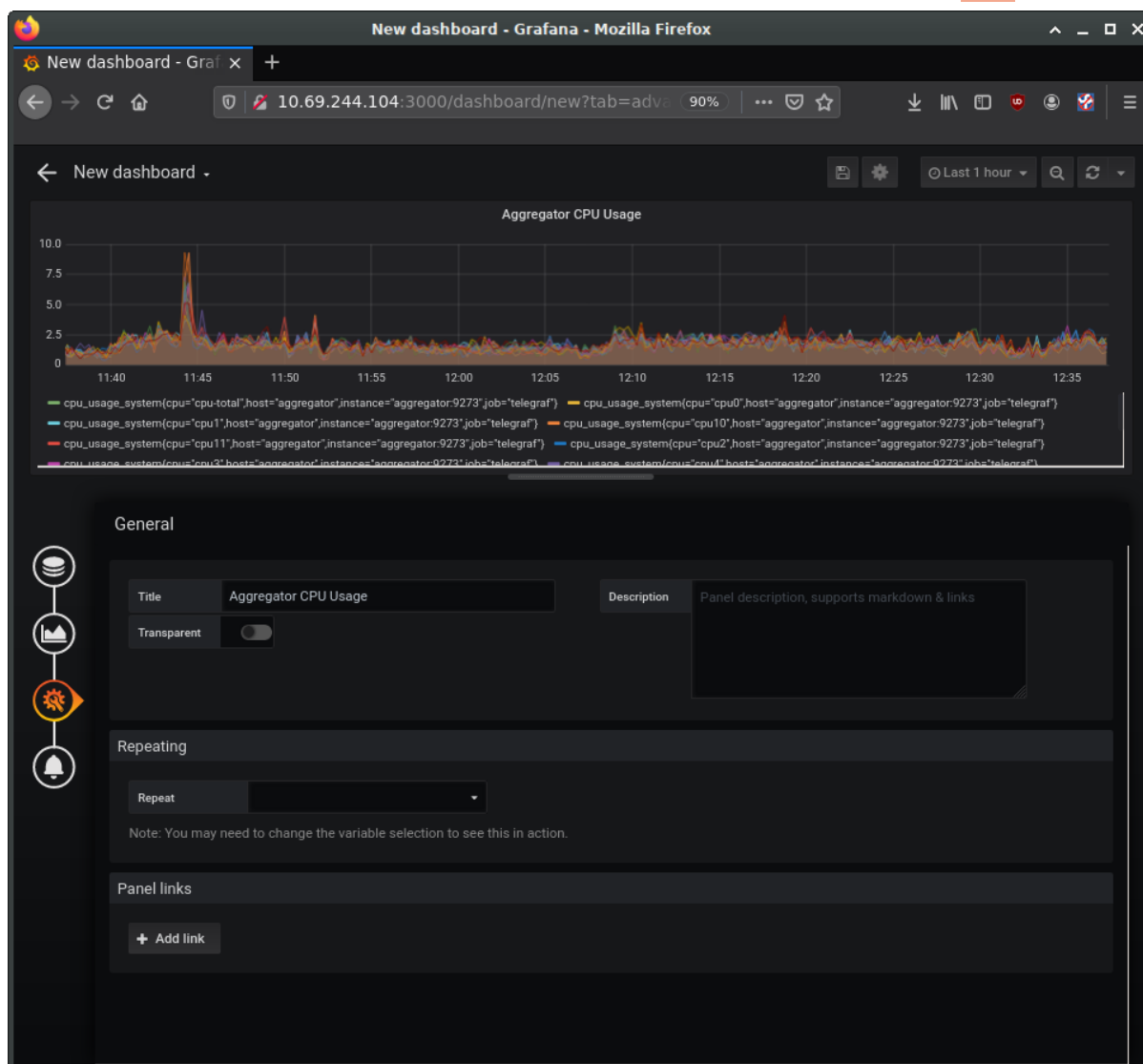
The footer of the page contains links for Documentation, Support, Community, Open Source, version information (v6.7.4 (8e44bdc)), and a note about a new version being available.











New dashboard - Grafana - Mozilla Firefox

10.69.244.104:3000/dashboard/new?tab=adv

New dashboard -

Aggregator CPU Usage

Save dashboard as...

Dashboard name  
Aggregator

Folder  
General

Copy tags  
☐

Save Cancel

General

Title  
Aggregator CPU Usage

Description  
Panel description, supports markdown & links

Transparent  
☐

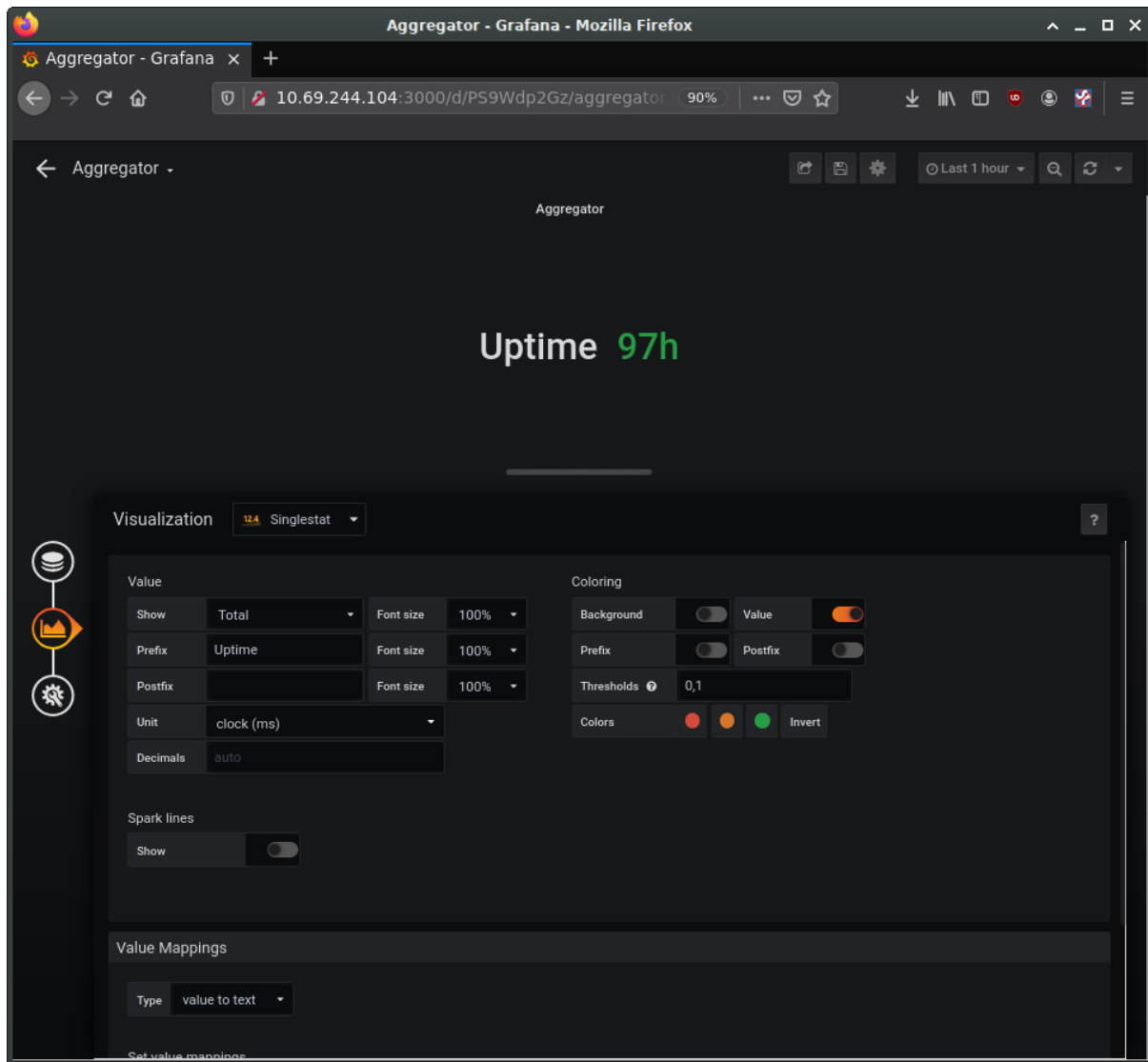
Repeating

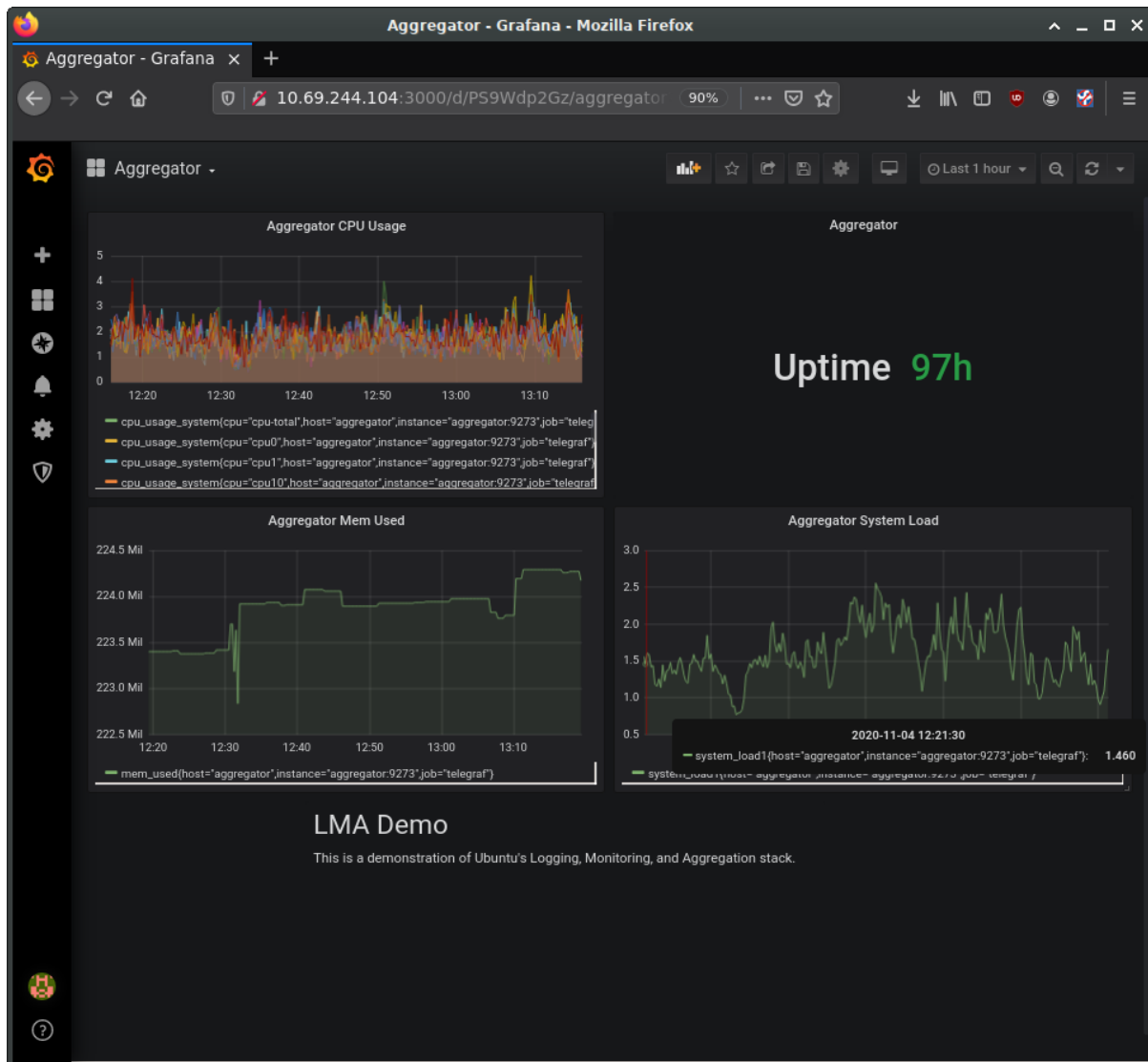
Repeat  
[Dropdown]

Note: You may need to change the variable selection to see this in action.

Panel links

+ Add link





## How to install and configure Logwatch

Logwatch<sup>688</sup>

## Install Logwatch

```
$ sudo apt install logwatch
```

<sup>688</sup> <https://sourceforge.net/projects/logwatch/>

## Configure Logwatch

```
$ sudo cp /usr/share/logwatch/default.conf/logwatch.conf /etc/logwatch/conf/
```

```
Output = mail
MailTo = me@mydomain.org
MailFrom = logwatch@host1.mydomain.org
Detail = Low
Service = All
```

```
Service = "-http"
Service = "-eximstats"
```

```
$ sudo logwatch --detail Low --range today
```

```
Logwatch 7.4.3 (12/07/16)
Processing Initiated: Fri Apr 24 16:58:14 2020
Date Range Processed: today
 (2020-Apr-24)
 Period is day.

Detail Level of Output: 0
Type of Output/Format: stdout / text
Logfiles for Host: `host1.mydomain.org`
#####

----- pam_unix Begin -----

sudo:
Sessions Opened:
 bryce -> root: 1 Time(s)

----- pam_unix End -----

----- rsnapshot Begin -----

ERRORS:
/usr/bin/rsnapshot hourly: completed, but with some errors: 5 Time(s)
/usr/bin/rsync returned 127 while processing root@host2:/etc/: 5 Time(s)
/usr/bin/rsync returned 127 while processing root@host2:/home/: 5 Time(s)
/usr/bin/rsync returned 127 while processing root@host2:/proc/uptime: 5 Time(s)
/usr/bin/rsync returned 127 while processing root@host3:/etc/: 5 Time(s)
/usr/bin/rsync returned 127 while processing root@host3:/home/: 5 Time(s)
/usr/bin/rsync returned 127 while processing root@host3:/proc/uptime: 5 Time(s)

----- rsnapshot End -----

----- SSHD Begin -----

Users logging in through sshd:
 bryce:
 192.168.1.123 (`host4.mydomain.org`): 1 time

----- SSHD End -----

----- Sudo (secure-log) Begin -----

bryce => root
\-----
/bin/bash - 1 Time(s).

----- Sudo (secure-log) End -----

----- Disk Space Begin -----
```

| Filesystem  | Size | Used | Avail | Use% | Mounted on                     |
|-------------|------|------|-------|------|--------------------------------|
| /dev/sdc1   | 220G | 19G  | 190G  | 9%   | /                              |
| /dev/loop1  | 157M | 157M | 0     | 100% | /snap/gnome-3-28-1804/110      |
| /dev/loop11 | 1.0M | 1.0M | 0     | 100% | /snap/gnome-logs/81            |
| /dev/md5    | 9.1T | 7.3T | 1.8T  | 81%  | /srv/Products                  |
| /dev/md6    | 9.1T | 5.6T | 3.5T  | 62%  | /srv/Archives                  |
| /dev/loop14 | 2.0M | 2.0M | 0     | 100% | /snap/gnome-system-monitor/127 |

## Further reading

- *logwatch(8)*<sup>689</sup>

## How to install and configure Munin

Munin<sup>690</sup>  
*hostnames*

## Prerequisites

*Apache2 will need to be installed*

## Install munin and munin-node

```
$ sudo apt install munin
```

```
$ sudo apt install munin-node
```

## Configure munin

```
First our "normal" host.
[server02]
 address 172.18.100.101
```

### Note:

Replace server02 and 172.18.100.101 with the actual hostname and IP address for your server.

<sup>689</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/logwatch.8.html>

<sup>690</sup> <https://munin-monitoring.org/>

```
Require ip 172.18.100.1
```

**Note:**

Replace 172.18.100.1 with the actual IP address for the host you want to allow.

## Configure munin-node

```
allow ^172\.18\.100\.100$
```

**Note:**

Replace ^172\.18\.100\.100\$ with the IP address for your munin server.

```
$ sudo systemctl restart munin-node.service
```

## Test the setup

## Additional Plugins

*DNS DHCP*

```
$ sudo apt install munin-plugins-extra
```



**Note:**

The `munin` package recommends `munin-node`, which recommends `munin-plugins-extra`. This means that in the most common use cases, you will already have `munin-plugins-extra` installed after following the instructions in this page.

**Further reading**

- [Munin](#)<sup>691</sup>
- [Munin Documentation](#)<sup>692</sup>

**How to install and configure Nagios Core 3****Note:**

Nagios Core 3 has been deprecated and is now replaced by Nagios Core 4. The `nagios3` package was last supported in Bionic, so subsequent releases should use `nagios4` instead.

*hostnames*

**Install `nagios3` on `server01`**

```
$ sudo apt install nagios3 nagios-nrpe-plugin
```

```
$ sudo htpasswd /etc/nagios3/htpasswd.users nagiosadmin
```

---

<sup>691</sup> <https://munin-monitoring.org/>

<sup>692</sup> <https://munin.readthedocs.io/en/latest/>

```
$ sudo htpasswd /etc/nagios3/htpasswd.users steve
```

## Install nagios-nrpe-server on server02

```
$ sudo apt install nagios-nrpe-server
```

### Note:

NRPE allows you to execute local checks on remote hosts. There are other ways of accomplishing this through other Nagios plugins, as well as other checks.

## Configuration overview

- 
- 
- 
- 

*DNS*

*hostgroup*

### Note:

See these guides for details on [setting up Apache](#) (page 1000), [Domain Name Service](#) (page 299), and [MySQL](#) (page 498).

- 
- 
- 
-

•

## Configure Nagios

### Create host config file for server02

```
$ sudo cp /etc/nagios3/conf.d/localhost_nagios2.cfg \ > /etc/nagios3/conf.d/server02.cfg
```

#### Note:

In all command examples, replace “server01”, “server02”, 172.18.100.100, and 172.18.100.101 with the host names and IP addresses of your servers.

### Edit the host config file

```
define host{
 use generic-host ; Name of host template to use
 host_name server02
 alias Server 02
 address 172.18.100.101
}

check DNS service.
define service {
 use generic-service
 host_name server02
 service_description DNS
 check_command check_dns!172.18.100.101
}
```

```
$ sudo systemctl restart nagio3.service
```

## Add service definition

```
check MySQL servers.
define service {
 hostgroup_name mysql-servers
 service_description MySQL
 check_command check_mysql_cmdlinecred!nagios!secret!$HOSTADDRESS$
 use generic-service
 notification_interval 0 ; set > 0 if you want to be renotified
}
```

```
MySQL hostgroup.
define hostgroup {
 hostgroup_name mysql-servers
 alias MySQL servers
 members localhost, server02
}
```

```
$ mysql -u root -p -e "create user nagios identified by 'secret';"
```

### Note:

The nagios user will need to be added to all hosts in the **mysql-servers** hostgroup.

```
$ sudo systemctl restart nagios3.service
```

## Configure NRPE

```
NRPE disk check.
define service {
 use generic-service
 host_name server02
 service_description nrpe-disk
 check_command check_nrpe_1arg!check_all_disks!172.18.100.101
}
```

```
allowed_hosts=172.18.100.100
```

```
command[check_all_disks]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -e
```

```
$ sudo systemctl restart nagios-nrpe-server.service
```

```
$ sudo systemctl restart nagios3.service
```

## Further reading

- [the Nagios website](https://www.nagios.org/)<sup>693</sup>
- [Nagios Core Documentation](https://library.nagios.com/products/nagios-core/documentation/)<sup>694</sup> [Nagios Core 3 Documentation](https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/toc.html)<sup>695</sup>
- [list of books](https://www.nagios.org/about/propaganda/books-2/)<sup>696</sup>
- [Nagios Ubuntu Wiki](https://help.ubuntu.com/community/Nagios3)<sup>697</sup>

## How to use Nagios with Munin

### Note:

Nagios Core 3 has been deprecated and is now replaced by Nagios Core 4. The `nagios3` package was last supported in Bionic, so subsequent releases should use `nagios4` instead.

<sup>693</sup> <https://www.nagios.org/>

<sup>694</sup> <https://library.nagios.com/products/nagios-core/documentation/>

<sup>695</sup> <https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/toc.html>

<sup>696</sup> <https://www.nagios.org/about/propaganda/books-2/>

<sup>697</sup> <https://help.ubuntu.com/community/Nagios3>

## Install Nagios 3

### On server01

```
$ sudo apt install nagios3 nagios-nrpe-plugin
```

```
$ sudo htpasswd /etc/nagios3/htpasswd.users nagiosadmin
```

```
$ sudo htpasswd /etc/nagios3/htpasswd.users steve
```

### On server02

```
$ sudo apt install nagios-nrpe-server
```

#### Note:

NRPE allows you to run local checks on remote hosts. There are other ways of accomplishing this, including through other Nagios plugins.

## Configuration overview

- 
- 
-

- 

*DNS*

*host group*

#### Note:

See these additional guides for details on setting up [Apache](#) (page 589), [Domain Name Service \(DNS\)](#) (page 299), and [MySQL](#) (page 498).

- 
- 
- 
- 
- 

## Configure Nagios

```
$ sudo cp /etc/nagios3/conf.d/localhost_nagios2.cfg \ >
/etc/nagios3/conf.d/server02.cfg
```

#### Note:

In the above and following command examples, replace “server01”, “server02”, 172.18.100.100, and 172.18.100.101 with the host names and IP addresses of your servers.

```
define host{
 use generic-host ; Name of host template to use
 host_name server02
```

(continues on next page)

(continued from previous page)

```
 alias Server 02
 address 172.18.100.101
 }

check DNS service.
define service {
 use generic-service
 host_name server02
 service_description DNS
 check_command check_dns!172.18.100.101
}
```

```
$ sudo systemctl restart nagios3.service
```

```
check MySQL servers.
define service {
 hostgroup_name mysql-servers
 service_description MySQL
 check_command check_mysql_cmdlinecred!nagios!secret!$HOSTADDRESS
 use generic-service
 notification_interval 0 ; set > 0 if you want to be renotified
}
```

```
MySQL hostgroup.
define hostgroup {
 hostgroup_name mysql-servers
 alias MySQL servers
 members localhost, server02
}
```

```
$ mysql -u root -p -e "create user nagios identified by 'secret';"
```

#### Note:

The nagios user will need to be added all hosts in the mysql-servers host group.

```
$ sudo systemctl restart nagios3.service
```



```
NRPE disk check.
define service {
 use generic-service
 host_name server02
 service_description nrpe-disk
 check_command check_nrpe_1arg!check_all_disks!172.18.100.101
}
```

```
allowed_hosts=172.18.100.100
```

```
command[check_all_disks]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -e
```

```
$ sudo systemctl restart nagios-nrpe-server.service
```

```
$ sudo systemctl restart nagios3.service
```

## Install Munin

*setting up Apache*

### On server01

```
$ sudo apt install munin
```

### On server02

```
$ sudo apt install munin-node
```

## Configure Munin on server01

```
First our "normal" host.
[server02]
 address 172.18.100.101
```

### Note:

Replace server02 and 172.18.100.101 with the actual hostname and IP address of your server.

## Configure munin-node on server02

```
allow ^172\.18\.100\.100$
```

### Note:

Replace ^172\.18\.100\.100\$ with IP address for your Munin server.

```
$ sudo systemctl restart munin-node.service
```

### Note:

Since this is a new install it may take some time for the graphs to display anything useful.

## Additional plugins

### *DHCP*

```
$ sudo apt install munin-plugins-extra
```

## Further reading

- [Munin](#)<sup>698</sup> [Nagios](#)<sup>699</sup>
- [Munin Documentation](#)<sup>700</sup>
- [Nagios Online Documentation](#)<sup>701</sup>
- [list of books](#)<sup>702</sup>
- [Nagios Ubuntu Wiki](#)<sup>703</sup>

*These debugging pages*

## Debugging

- *About debuginfod*
- *Debug symbol packages*
- *Introduction to eBPF*
- *Kernel crash dump*

## About debuginfod

`debuginfod`<sup>704</sup>

*manually install*

---

<sup>698</sup> <https://munin-monitoring.org/>

<sup>699</sup> <https://www.nagios.org/>

<sup>700</sup> <https://munin.readthedocs.io/en/latest/>

<sup>701</sup> <https://www.nagios.org/documentation/>

<sup>702</sup> <https://www.nagios.org/about/propaganda/books-2/>

<sup>703</sup> <https://help.ubuntu.com/community/Nagios3>

<sup>704</sup> <https://sourceware.org/elfutils/Debuginfod.html>

## Using the service

supported Ubuntu releases<sup>705</sup>

GNU Debugger (GDB)<sup>706</sup>

```
export DEBUGINFOD_URLS="https://debuginfod.ubuntu.com"
```

*GDB*

```
set debuginfod enabled on
```

## Example session with GDB

```
$ gdb -q program

Reading symbols from program...

This GDB supports auto-downloading debuginfo from the following URLs:
https://debuginfod.ubuntu.com
Enable debuginfod for this session? (y or [n])
```

```
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
Downloading 0.20 MB separate debug info for /home/ubuntu/program
Reading symbols from /home/ubuntu/.cache/debuginfod_client/
c0fbda15a807f880e9d0b2dcc635eeeb1f0f728e/debuginfo...
(gdb)
```

---

<sup>705</sup> <https://releases.ubuntu.com/>

<sup>706</sup> <https://www.sourceware.org/gdb/>

## Opting out of the service

```
unset DEBUGINFOD_URLS
```

```
set debuginfod enabled off
```

## How does debuginfod find the debug symbols for the binary I am debugging?

```
$ readelf -j .note.gnu.build-id /bin/bash
```

Displaying notes found in: .note.gnu.build-id

| Owner                                              | Data size  | Description                                 |
|----------------------------------------------------|------------|---------------------------------------------|
| GNU                                                | 0x00000014 | NT_GNU_BUILD_ID (unique build ID bitstring) |
| Build ID: 7426cc87c256cb386b52891bbe4300f2f5bab8c6 |            |                                             |

## Can ddeb packages co-exist with debuginfod?

## Can I use debuginfod with my own binary that links against system libraries?

## Bug reports

[Launchpad for ubuntu-debuginfod](#)<sup>707</sup>

## Debug symbol packages

[Valgrind](#)<sup>709</sup>

[backtrace](#)<sup>708</sup>

## Debuginfod

[\(GDB\)](#)<sup>710</sup>

[GNU Debugger](#)

*to our Debuginfod guide*

## Getting -dbgsym.ddeb packages

## Import the signing key

[signing key](#)<sup>711</sup>

```
$ sudo apt install ubuntu-dbgsym-keyring
```

<sup>707</sup> <https://bugs.launchpad.net/ubuntu-debuginfod/+filebug>

<sup>708</sup> <https://wiki.ubuntu.com/Backtrace>

<sup>709</sup> <https://wiki.ubuntu.com/Valgrind>

<sup>710</sup> <https://www.sourceware.org/gdb/>

<sup>711</sup> [https://help.ubuntu.com/community/Repositories/Ubuntu#Authentication\\_Tab](https://help.ubuntu.com/community/Repositories/Ubuntu#Authentication_Tab)

## Create a ddebs.list file

```
$ echo "Types: deb > URIs: http://ddebs.ubuntu.com/ > Suites: $(lsb_release
-cs) $(lsb_release -cs)-updates $(lsb_release -cs)-proposed > Components:
main restricted universe multiverse > Signed-by:
/usr/share/keyrings/ubuntu-dbg-sym-keyring.gpg" | \ > sudo tee -a
/etc/apt/sources.list.d/ddebs.sources
```

ries<sup>713</sup>

this article<sup>712</sup>

adding other repositories

```
$ deb http://ddebs.ubuntu.com focal main restricted universe multiverse
```

### Note:

Make sure you replace focal with the Ubuntu release name you're using.

## Update package list

```
$ sudo apt-get update
```

## Manual install of debug packages

```
$ sudo apt-get install PACKAGE-dbg-sym
```

```
$ sudo apt-get install xserver-xorg-core-dbg-sym
```

```
$ sudo apt-get install libc6-dbg
```

<sup>712</sup> <https://help.ubuntu.com/community/Repositories/Ubuntu>

<sup>713</sup> [https://help.ubuntu.com/community/Repositories/Ubuntu#Adding\\_Other\\_Repositories](https://help.ubuntu.com/community/Repositories/Ubuntu#Adding_Other_Repositories)

## The debian-goodies tool

```
$ sudo apt install debian-goodies
```

```
$ find-dbgsym-packages [core_path|running_pid|binary_path]
```

## “debs” versus “ddebs”

their own separate repository<sup>714</sup>

---

<sup>714</sup> <http://ddebs.ubuntu.com>



## Kernel crash dump

- 
- 
- 
- 
- 

## Kernel crash dump mechanism

## Installation

```
$ sudo apt install kdump-tools
```

```
-----| Configuring kdump-tools |-----

If you choose this option, the kdump-tools mechanism will be enabled. A
reboot is still required in order to enable the crashkernel kernel
parameter.

Should kdump-tools be enabled by default?

 <Yes> <No>
-----|
```

**kdump is enabled by default where applicable**

- 
- 
- 
- 

—  
—

## Disabling kdump

```
$ sudo apt remove kdump-tools
```

```
USE_KDUMP=0
```

## Activating config changes

## Verify if kdump is ready

```
$ kdump-config show

DUMP_MODE: kdump
USE_KDUMP: 1
KDUMP_COREDIR: /var/crash
crashkernel addr: 0x60000000
 /var/lib/kdump/vmlinuz: symbolic link to /boot/vmlinuz-6.18.0-8-generic
kdump initrd:
 /var/lib/kdump/initrd.img: symbolic link to /var/lib/kdump/initrd.img-6.18.0-8-generic
current state: ready to kdump

crashkernel suggested size: 417M

kexec command:
 /sbin/kexec -p -s --command-line="BOOT_IMAGE=/vmlinuz-6.18.0-8-generic
root=UUID=0a86b691-f733-4cb0-9c5c-b88e0ef9e212 ro console=tty1 console=ttyS0 reset_
devices systemd.unit=kdump-tools-dump.service nr_cpus=1 irqpoll usbcore.nousb" --
initrd=/var/lib/kdump/initrd.img /var/lib/kdump/vmlinuz
```

## Reserving Memory

```
$ cat /proc/cmdline

BOOT_IMAGE=/vmlinuz-6.18.0-8-generic
root=UUID=0a86b691-f733-4cb0-9c5c-b88e0ef9e212 ro console=tty1 console=ttyS0
crashkernel=2G-4G:320M,4G-32G:512M,32G-64G:1024M,64G-128G:2048M,128G-:4096M
```

```
crashkernel=<range1>:<size1>[,<range2>:<size2> ...]@offset|
range=start-[end] 'start' is inclusive and 'end' is exclusive.
```

```
crashkernel=2G-4G:320M,4G-32G:512M,32G-64G:1024M,64G-128G:2048M,128G-:4096M
```

- 
- 
- 
- 

```
$ dmesg | grep -i crash
```

```
[0.004623] crashkernel reserved: 0x0000000060000000 - 0x0000000074000000 (320 MB)
```

## Configuration

## Local kernel crash dumps

## Remote kernel crash dumps using the SSH protocol

```

Remote dump facilities:
SSH - username and hostname of the remote server that will receive the dump
and dmesg files.
SSH_KEY - Full path of the ssh private key to be used to login to the remote
server. use kdump-config propagate to send the public key to the
remote server
HOSTTAG - Select if hostname of IP address will be used as a prefix to the
timestamped directory when sending files to the remote server.
'ip' is the default.
SSH="ubuntu@kdump-netcrash"
```

*hostname*

```
$ sudo kdump-config propagate
```

```
Need to generate a new ssh key...
The authenticity of host 'kdump-netcrash (192.168.1.74)' can't be established.
ECDSA key fingerprint is SHA256:IMp+5Y28qhbd+tevFCWrEXyKdD4dI3yN40Vlu3CBBQ4.
Are you sure you want to continue connecting (yes/no)? yes
ubuntu@kdump-netcrash's password:
propagated ssh key /root/.ssh/kdump_id_rsa to server ubuntu@kdump-netcrash
```

```
$ kdump-config show
```

```
DUMP_MODE: kdump
USE_KDUMP: 1
KDUMP_SYSCTL: kernel.panic_on_oops=1
KDUMP_COREDIR: /var/crash
crashkernel addr: 0x2c000000
/var/lib/kdump/vmlinuz: symbolic link to /boot/vmlinuz-4.4.0-10-generic
kdump initrd:
/var/lib/kdump/initrd.img: symbolic link to /var/lib/kdump/initrd.img-4.4.0-10-generic
SSH: ubuntu@kdump-netcrash
SSH_KEY: /root/.ssh/kdump_id_rsa
HOSTTAG: ip
current state: ready to kdump
```

## Remote kernel crash dumps using the NFS protocol

```
NFS - Hostname and mount point of the NFS server configured to receive
the crash dump. The syntax must be {HOSTNAME}:{MOUNTPOINT}
(e.g. remote:/var/crash)
#
NFS="kdump-netcrash:/var/crash"
```

```
$ kdump-config show

DUMP_MODE: kdump
USE_KDUMP: 1
KDUMP_SYSCTL: kernel.panic_on_oops=1
KDUMP_COREDIR: /var/crash
crashkernel addr: 0x2c000000
/var/lib/kdump/vmlinuz: symbolic link to /boot/vmlinuz-4.4.0-10-generic
kdump initrd:
/var/lib/kdump/initrd.img: symbolic link to /var/lib/kdump/initrd.img-4.4.0-10-generic
NFS: kdump-netcrash:/var/crash
HOSTTAG: hostname
current state: ready to kdump
```

## Troubleshooting

```
$ kdump-config show

DUMP_MODE: kdump
USE_KDUMP: 1
KDUMP_COREDIR: /var/crash
crashkernel addr:
 /var/lib/kdump/vmlinuz: symbolic link to /boot/vmlinuz-6.18.0-8-generic
kdump initrd:
 /var/lib/kdump/initrd.img: symbolic link to /var/lib/kdump/initrd.img-6.18.0-8-generic
current state: Not ready to kdump

crashkernel suggested size: 329M

kexec command:
 no kexec command recorded
```

```
$ systemctl status kdump-tools

...
... kdump-tools[5348]: Memory for crashkernel is not reserved
... kdump-tools[5348]: Please reserve memory by passing "crashkernel=Y@X" parameter
to kernel
... kdump-tools[5348]: Then try to loading kdump kernel
```

## Testing the crash dump mechanism

### Warning:

Testing the crash dump mechanism **will cause a system reboot**. In certain situations, this can cause data loss if the system is under heavy load. If you want to test the mechanism, make sure that the system is idle or under very light load.

```
$ cat /proc/sys/kernel/sysrq
```

**Note:**

On 24.10 and earlier the location of this file was at `/etc/sysctl.d/10-magic-sysrq.conf`

```
$ sudo sysctl -w kernel.sysrq=1
```

```
[977.208267] sysrq: Trigger a crash
[977.209313] Kernel panic - not syncing: sysrq triggered crash
[977.210684] CPU: 0 UID: 0 PID: 1567 Comm: bash Kdump: loaded Not tainted 6.18.0-8-
generic #8-Ubuntu PREEMPT(voluntary)
[977.215248] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009) /LXD, BIOS unknown 2/2/
2022
[977.216319] Call Trace:
...
```

```
[5.171504] kdump-tools[740]: Starting kdump-tools:
[5.174084] kdump-tools[747]: * running makedumpfile --dump-dmesg /proc/vmcore /var/
crash/202601210724/dmesg.202601210724
[5.185243] kdump-tools[765]: The kernel version is not supported.
[5.188411] kdump-tools[765]: The makedumpfile operation may be incomplete.
[5.191528] kdump-tools[765]: The dmesg log is saved to /var/crash/202601210724/dmesg.
202601210724.
[5.196300] kdump-tools[765]: makedumpfile Completed.
[5.199056] kdump-tools[747]: * kdump-tools: saved dmesg content in /var/crash/
202601210724
[5.204580] kdump-tools[747]: * running makedumpfile -F -c -d 31 /proc/vmcore |
compress > /var/crash/202601210724/dump-incomplete
[5.210935] kdump-tools[769]: The kernel version is not supported.
[5.214499] kdump-tools[769]: The makedumpfile operation may be incomplete.
Copying data : [100.0 %] \ eta: 0s
[7.000839] kdump-tools[769]: The dumpfile is saved to STDOUT.
[7.003347] kdump-tools[769]: makedumpfile Completed.
[7.006824] kdump-tools[747]: * kdump-tools: saved vmcore in /var/crash/202601210724
```



```
$ ll /var/crash
drwxr-xr-x 2 root root 4096 Jan 21 07:31 202601210724/
-rw-r--r-- 1 root root 28838 Jan 21 07:24 linux-image-6.18.0-8-generic-
202601210724.crash
```

```
$ ll /var/crash/202601210724
-rw----- 1 root root 82264 Jan 21 07:24 dmesg.202601210724
-rw-r--r-- 1 root root 51753823 Jan 21 07:24 dump.202601210724
```

## Analyzing a crash dump

*Getting -dbgsym.ddeb*

*packages*

### Note:

This does not strictly need to happen on the system that had the crash, it can be copied away for analysis. But the examples are simplified to use e.g. `uname -r` to automatically select the right kernel - when running on a different system these might need to be adapted.

```
$ apt-get install linux-image-$(uname -r)-dbgsym
```

```
$ crash /usr/lib/debug/boot/vmlinux-6.18.0-8-generic
/var/crash/202601210724/dump.202601210724
```

```
...
 KERNEL: /usr/lib/debug/boot/vmlinux-6.18.0-8-generic
 DUMPFILE: /var/crash/202601210724/dump.202601210724 [PARTIAL DUMP]
 CPUS: 1
 DATE: Thu Jan 1 00:00:00 UTC 1970
 UPTIME: 00:16:16
 LOAD AVERAGE: 0.00, 0.00, 0.00
 TASKS: 157
 NODENAME: r-vm
 RELEASE: 6.18.0-8-generic
 VERSION: #8-Ubuntu SMP PREEMPT_DYNAMIC Wed Dec 17 16:14:24 UTC 2025
 MACHINE: x86_64 (3792 Mhz)
 MEMORY: 3 GB
 PANIC: "Kernel panic - not syncing: sysrq triggered crash"
 PID: 1567
 COMMAND: "bash"
 TASK: ffff89cd07908000 [THREAD_INFO: ffff89cd07908000]
 CPU: 0
 STATE: TASK_RUNNING (PANIC)

crash>
```

PANIC: "Kernel panic - not syncing: sysrq triggered crash"

## Further reading

- [kdump kernel documentation](https://www.kernel.org/doc/Documentation/kdump/kdump.txt)<sup>715</sup>
- [Analyzing Linux Kernel Crash](https://www.dedoimedo.com/computers/crash-analyze.html)<sup>716</sup>

## Introduction to eBPF

eBPF<sup>717</sup>

<sup>715</sup> <https://www.kernel.org/doc/Documentation/kdump/kdump.txt>

<sup>716</sup> <https://www.dedoimedo.com/computers/crash-analyze.html>

<sup>717</sup> <https://ebpf.io/>

## How eBPF works

- 
- 
- 

## eBPF in Ubuntu Server

enhance the ap-  
plication developer and sysadmin experience in Ubuntu<sup>718</sup>

```
$ dpkg -L bpftrace bpfcc-tools | grep -E '/s?bin/.*$' | xargs -n1 basename
```

<sup>718</sup> <https://discourse.ubuntu.com/t/spec-include-performance-tooling-in-ubuntu/43134>

```
bashreadline.bt
```

### Example - Determine what commands are executed

- 
- 

```
$ sudo execsnoop-bpfcc -Uu root -T
```

```
$ sudo apt install --reinstall vim-nox
```

| TIME     | UID  | PCOMM | PID     | PPID    | RET | ARGS                                          |
|----------|------|-------|---------|---------|-----|-----------------------------------------------|
| 10:58:07 | 1000 | sudo  | 1323101 | 1322857 | 0   | /usr/bin/sudo apt install --reinstall vim-nox |
| 10:58:10 | 0    | apt   | 1323107 | 1323106 | 0   | /usr/bin/apt install --reinstall vim-nox      |
| 10:58:10 | 0    | dpkg  | 1323108 | 1323107 | 0   | /usr/bin/dpkg --print-foreign-architectures   |

(continues on next page)

(continued from previous page)

```
...
10:58:12 0 sh 1323134 1323107 0 /bin/sh -c /usr/sbin/dpkg-preconfigure
--apt || true
...
10:58:13 0 tar 1323155 1323152 0 /usr/bin/tar -x -f --warning=no-
timestamp
10:58:14 0 vim-nox.prerm 1323157 1323150 0 /var/lib/dpkg/info/vim-nox.prerm
upgrade 2:9.1.0016-1ubuntu7.3
10:58:14 0 dpkg-deb 1323158 1323150 0 /usr/bin/dpkg-deb --fsys-tarfile /var/
cache/apt/archives/vim-nox_2%3a9.1.0016-1ubuntu7.3_amd64.deb
...
10:58:14 0 update-alternat 1323171 1323163 0 /usr/bin/update-alternatives --install
/usr/bin/vimdiff vimdiff /usr/bin/vim.nox 40
...
10:58:17 0 snap 1323218 1323217 0 /usr/bin/snap advise-snap --from-apt
10:58:17 1000 git 1323224 1323223 0 /usr/bin/git rev-parse --abbrev-ref
HEAD
10:58:17 1000 git 1323226 1323225 0 /usr/bin/git status --porcelain
10:58:17 1000 vte-urlencode-c 1323227 1322857 0 /usr/libexec/vte-urlencode-cwd
```

## eBPF can be modified to your needs

### Example - Find out which files QEMU is loading

- 
- 

```
$ sudo /usr/sbin/opensnoop-bpfcc --full-path --name qemu-system-x86
```

```
$ lxc launch ubuntu-daily:n n-vm-test --ephemeral --vm
```

```
1308728 qemu-system-x86 -1 2 PID COMM FD ERR PATH
/snap/lxd/current/zfs-2.2/lib/glibc-hwcaps/x86-64-v3/libpixman-1.so.0
1308728 qemu-system-x86 -1 2 /snap/lxd/current/zfs-2.2/lib/glibc-hwcaps/x86-64-v2/
libpixman-1.so.0
1308728 qemu-system-x86 -1 2 /snap/lxd/current/zfs-2.2/lib/tls/haswell/x86_64/
libpixman-1.so.0
...
1313104 qemu-system-x86 58 0 /sys/dev/block/230:16/queue/zoned
1313104 qemu-system-x86 20 0 /dev/fd/4
```

### Focusing on a specific file type

```
$ sudo cp /usr/sbin/opensnoop-bpfcc /usr/sbin/opensnoop-bpfcc.new
```

```
$ $ sudo vim /usr/sbin/opensnoop-bpfcc.new
```

```
...
```

```
$ $ diff -Naur /usr/sbin/opensnoop-bpfcc /usr/sbin/opensnoop-bpfcc.new
```

```

--- /usr/sbin/opensnoop-bpfcc 2025-10-24 12:17:00.000000000 +0000
+++ /usr/sbin/opensnoop-bpfcc.new 2026-01-19 10:45:13.620367441 +0000
@@ -40,6 +40,7 @@
 ./opensnoop -u 1000 # only trace UID 1000
 ./opensnoop -d 10 # trace for 10 seconds only
 ./opensnoop -n main # only print process names containing "main"
+ ./opensnoop -c path # only print paths containing "fname"
 ./opensnoop -e # show extended fields
 ./opensnoop -f O_WRONLY -f O_RDWR # only print calls for writing
 ./opensnoop -F # show full path for an open file with
relative path
@@ -71,6 +72,9 @@
 parser.add_argument("-n", "--name",
 type=ArgString,
 help="only print process names containing this name")
+parser.add_argument("-c", "--contains",
+ type=ArgString,
+ help="only print paths containing this string (implies --full-path)")
 parser.add_argument("--ebpf", action="store_true",
 help=argparse.SUPPRESS)
 parser.add_argument("-e", "--extended_fields", action="store_true",
@@ -83,6 +87,8 @@
 help="size of the perf ring buffer "
 "(must be a power of two number of pages and defaults to 64)")
 args = parser.parse_args()
+if args.contains is not None:
+ args.full_path = True
 debug = 0
 if args.duration:
 args.duration = timedelta(seconds=int(args.duration))
@@ -478,6 +484,12 @@
 if args.name and bytes(args.name) not in event.comm:
 skip = True

+ paths = entries[event.id]
+ paths.reverse()
+ entire_path = os.path.join(*paths)
+ if args.contains and bytes(args.contains) not in entire_path:
+ skip = True
+
 if not skip:
 if args.timestamp:
 delta = event.ts - initial_ts
@@ -502,9 +514,7 @@
 if not args.full_path:
 printb(b"%s" % event.name)
 else:
- paths = entries[event.id]
- paths.reverse()
- printb(b"%s" % os.path.join(*paths))
+ printb(b"%s" % entire_path)

 if args.full_path:
 try:

```

```
$ sudo /usr/sbin/opensnoop-bpffcc.new --contains '.bin' --name
qemu-system-x86
```

| PID     | COMM            | FD | ERR | PATH                                             |
|---------|-----------------|----|-----|--------------------------------------------------|
| 1316661 | qemu-system-x86 | 21 | 0   | /snap/lxd/current/share/qemu//kvmvapic.bin       |
| 1316661 | qemu-system-x86 | 39 | 0   | /snap/lxd/current/share/qemu//vgabios-virtio.bin |
| 1316661 | qemu-system-x86 | 39 | 0   | /snap/lxd/current/share/qemu//vgabios-virtio.bin |

Use for other purposes, the limit is your imagination

```
$ sudo /usr/sbin/opensnoop-bpffcc.new --name 'apache2' --contains '/etc'
```

| PID     | COMM    | FD | ERR | PATH                                                 |
|---------|---------|----|-----|------------------------------------------------------|
| 1319357 | apache2 | 3  | 0   | /etc/apache2/apache2.conf                            |
| 1319357 | apache2 | 4  | 0   | /etc/apache2/mods-enabled                            |
| 1319357 | apache2 | 4  | 0   | /etc/apache2/mods-enabled/access_compat.load         |
| ...     |         |    |     |                                                      |
| 1319357 | apache2 | 4  | 0   | /etc/apache2/ports.conf                              |
| 1319357 | apache2 | 4  | 0   | /etc/apache2/conf-enabled                            |
| 1319357 | apache2 | 4  | 0   | /etc/apache2/conf-enabled/charset.conf               |
| 1319357 | apache2 | 4  | 0   | /etc/apache2/conf-enabled/localized-error-pages.conf |
| ...     |         |    |     |                                                      |
| 1319357 | apache2 | 4  | 0   | /etc/apache2/sites-enabled/000-default.conf          |

## eBPF's limitations

Possibly lost 84 samples

same discussion<sup>719</sup>

<sup>719</sup> <https://github.com/iovisor/bcc/issues/1033>



## Conclusion

## Further reading

- [Introduction to eBPF video, given at the Ubuntu summit 2024](#)<sup>720</sup>
- [what is eBPF](#)<sup>721</sup>
- [upstream documentation](#)<sup>722</sup>
- [list of eBPF based solutions](#)<sup>723</sup>

## Further reading

- [The Ubuntu Project documentation](#)<sup>724</sup>
- [Glossary](#)
- [Command-line cheat sheet](#)

[contributions page](#)

|                   |                   |                   |                   |                   |                   |                   |
|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| <a href="#">A</a> | <a href="#">B</a> | <a href="#">C</a> | <a href="#">D</a> | <a href="#">E</a> | <a href="#">F</a> |                   |
| <a href="#">G</a> | <a href="#">H</a> | <a href="#">I</a> | <a href="#">J</a> | <a href="#">K</a> | <a href="#">L</a> | <a href="#">M</a> |
|                   | <a href="#">N</a> | <a href="#">O</a> | <a href="#">P</a> | <a href="#">Q</a> | <a href="#">R</a> | <a href="#">S</a> |
|                   | <a href="#">T</a> | <a href="#">U</a> | <a href="#">V</a> | <a href="#">W</a> | <a href="#">X</a> | <a href="#">Y</a> |
|                   | <a href="#">Z</a> |                   |                   |                   |                   |                   |

<sup>720</sup> <https://www.youtube.com/live/byPpJW5l6pg?t=30314s>

<sup>721</sup> <https://ebpf.io/what-is-ebpf/>

<sup>722</sup> <https://docs.kernel.org/bpf/>

<sup>723</sup> <https://ebpf.io/applications/>

<sup>724</sup> <https://documentation.ubuntu.com/project/contributors/debugging/>

## A

### ABI Application Binary Interface

*API*

### ACL Access Control List

- *the ACL manual page*<sup>725</sup>

- *OpenLDAP* *Kerberos*

### AD Active Directory

### ADSys

*AD*  
*SSSD*

- *the ADSys documentation*<sup>726</sup>

- *Group Policy Object* *SSSD*

### AES Advanced Encryption Standard

- 

### Alertmanager

---

<sup>725</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/acl.5.html>

<sup>726</sup> <https://documentation.ubuntu.com/adsys/stable/>

- the Alertmanager documentation<sup>727</sup>

- [Prometheus](#)

## ALUA

### Asymmetric Logical Unit Access

- [MPIO SCSI SAN](#)

## AMD

### Advanced Micro Devices

- 
- 
- 

- 

## AMD-SME

### AMD Secure Memory Encryption

- 

## AMD-SEV

### AMD Secure Encrypted Virtualization

- 

## AMD-SEV-ES

### AMD Secure Encrypted Virtualization - Encrypted State

[AMD-SEV](#)

---

<sup>727</sup> <https://prometheus.io/docs/alerting/latest/alertmanager/>

- 

## **AMD-SEV-SNP**

### **AMD Secure Encrypted Virtualization - Secure Nested Paging**

*AMD-SEV-ES*

- 

## **Ansible**

- [The Ansible website](#)<sup>728</sup>

- 

## **Apache2**

*SSL TLS*

- [The Apache project documentation](#)<sup>729</sup>

- 

## **API**

### **Application Programming Interface**

---

<sup>728</sup> <https://docs.ansible.com/>

<sup>729</sup> <https://httpd.apache.org/docs/2.4/>

## AppArmor

*Mandatory Access Control (MAC)*

- The AppArmor website<sup>730</sup>

- 

## Apport

- The Apport Wiki Page<sup>731</sup>

- 

## APT

### Advanced Package Tool

- *Install and manage packages*

## armhf

### ARM hard-float

- 

## ARP

### Address Resolution Protocol

*MAC ad-*

*dress*

---

<sup>730</sup> <https://apparmor.net/>

<sup>731</sup> <https://wiki.ubuntu.com/Apport>

- 

**ASCII**  
**American Standard Code for Information Interchange**

**async**  
**asynchronous**

- 

**attestation**

- 

- 

**Authenticator**

*bind*

- 

**autocommit**

- [autocommit behavior in Python's sqlite3 module](#)<sup>732</sup>

- 

## **autodetect**

- 

## **autoinstall**

- [The Autoinstall documentation](#)<sup>733</sup>

- 

## **autorid**

*UID*   *GID*

- [The autorid Samba Wiki](#)<sup>734</sup>

- 

## **AWS**

### **Amazon Web Services**

---

<sup>732</sup> <https://docs.python.org/3/library/sqlite3.html#transaction-control>

<sup>733</sup> <https://canonical-subiquity.readthedocs-hosted.com/en/latest/intro-to-autoinstall.html>

<sup>734</sup> [https://wiki.samba.org/index.php/Idmap\\_config\\_autorid](https://wiki.samba.org/index.php/Idmap_config_autorid)

- The AWS documentation<sup>735</sup>

- 

## B

**backend**

**Backports**

**Backtrace**

**BDC**

**Backup Domain Controller**

*Samba server* *PDC*

**bind**

**bindDN**

*DN*

*bind*

**BIOS**

**Basic Input/Output System**

*UEFI*

**BMC**

**Baseboard Management Controller**

**bootloader**

*BIOS* *UEFI*

---

<sup>735</sup> <https://docs.aws.amazon.com/>



**BOOTP**  
**Bootstrap Protocol**

**bootstrap**

**btrfs**  
**B-tree File System**

## **C**

**CA**  
**Certificate Authority**

**CAC**  
**Common Access Card**

**CARP**  
**Cache Array Routing Protocol**

**CCID**  
**Chip Card Interface Device**

**CDB**  
**Command Descriptor Block**

**CGNAT**  
**Carrier-Grade Network Address Translation**

**CGI**  
**Common Gateway Interface**

**checksums**

**chrony**

**chroot**

**CIDR**  
**Classless Inter-Domain Routing**

**CIFS**  
**Common Internet File System**

**CIS**  
**Center for Internet Security**

**CLVM**  
**Clustered Logical Volume Manager**

**CMS**  
**Configuration Management System**

**CN**  
**Common Name**

**colocation**

**conffile**

**config**

**connectionless**

**containerization**

**CPU**  
**Central Processing Unit**

**CRL**  
**Certificate Revocation List**

**crypto**  
**cryptographic**

**CSR**  
**Certificate Signing Request**

**CVE**  
**Common Vulnerabilities and Exposures**

## D

### DAC

Discretionary access control

- 

**daemonize**

### DARPA

Defense Advanced Research Projects Agency

### DAS

Direct Attached Storage

- *NAS*

- *SAN*

### DASD

Direct Access Storage Device

- 

**Datagram**

- 

### dblink

Database link

-

**DC**  
**Domain Component**

**DDNS**  
**Dynamic Domain Name System**

- 

**deb822**

- [RFC 822](#)<sup>736</sup>
- [deb822\(5\)](#)<sup>737</sup>

**debconf**  
*configuration management system*

**deduplication**

**denylist**

- 

**DER**  
**Distinguished Encoding Rules**

**DGC**  
**Distributed Garbage Collection**

**DHCP**  
**Dynamic Host Configuration Protocol**

---

<sup>736</sup> <https://datatracker.ietf.org/doc/html/rfc822>

<sup>737</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/deb822.5.html>

**DHCPD**  
Dynamic Host Configuration Protocol Daemon

**DIT**  
Directory Information Tree

•

**DKMS**  
Dynamic Kernel Module Support

•

**DMA**  
Direct Memory Access

**DMAR**  
Direct Memory Access Remapping

*Direct Memory Access*

*virtual machines (VMs)*

**dmesg**

**DN**  
Distinguished Name

*OU    DC* *CN*

**DNS**  
Domain Name System

•

**dnsmasq** *DNS    DHCP*

## DNSSEC Domain Name System Security Extensions

*DNS*

- 

## Docker

- 

## DocumentRoot

## DoT DNS over TLS

*Transport Layer Security (TLS)*

- [RFC 7858](#)<sup>738</sup>

- 

- 

## DoH DNS over HTTPS

*HTTPS*

- [RFC 8484](#)<sup>739</sup>

- 

- 

## dpkg

---

<sup>738</sup> <https://datatracker.ietf.org/doc/html/rfc7858>

<sup>739</sup> <https://datatracker.ietf.org/doc/html/rfc8484>

**DRBD**  
Distributed Replicated Block Device

•

**DSCP**  
Differentiated Services Code Point

•

**DTLS**  
Datagram Transport Layer Security

*TLS*

*UDP*

**E**

**EAL**  
Environment Abstraction Layer

**ECKD**  
Extended Count Key Data

**EFI**  
Extensible Firmware Interface

*UEFI*

*BIOS*

**ELinks**

*GUI*

**Engenio**

*SAN*

*NAS*

**EOL**  
End of life

**ERD**  
Enterprise Ready Drivers

**ESM**  
Expanded Security Maintenance

*LTS*

**ESXi**

**F**

**failover**  
*Storage Area Network (SAN)*

**fallbacks**

**FastCGI**  
Fast Common Gateway Interface  
*CGI*

**FC**  
Fiber Channel

*Storage Area Network (SAN)*

**FDE**  
Full-Disk Encryption

**FHS**  
Filesystem Hierarchy Standard

**Fileset**



- 

**filesystem**

- 

**FTP**

**File Transfer Protocol**

- *SFTP*
- *rsync*
- *SCP*

**FIPS**

**Federal Information Processing Standard**

- *NIST<sup>740</sup>*

**Fluentd**

- 

**FQDN**

**Fully Qualified Domain Name**

*DNS*

- 

**FreeIPA**

**Free Identity, Policy, and Audit**

*LDAP*

*CA*

*SSSD*

*Kerberos*

- 

---

<sup>740</sup> [https://en.wikipedia.org/wiki/National\\_Institute\\_of\\_Standards\\_and\\_Technology](https://en.wikipedia.org/wiki/National_Institute_of_Standards_and_Technology)

## Freemove

*Internet Relay Chat (IRC)*

## frontend

- 

- 

*OS*

*GPU*

*backend*

- 

*slapd*

- 

## fsck

**File System Check**

*filesystem*

- 

## FULLTEXT

- Full-Text Search Functions<sup>741</sup>

- 

## FW

**Firmware**

- 

*RAM*

- 

*PCIe*

*GPUs*

<sup>741</sup> <https://dev.mysql.com/doc/refman/8.4/en/fulltext-search.html>

## G

**gcplogs**

•

**gcrypt**

•

**GDB**  
**GNU Debugger**

•

**GELF**  
**Graylog Extended Log Format**

*JSON*

*Graylog Logstash Fluentd*

•

**GFS2**

*filesystem*

•

**GB**

**GID**  
**Group ID**

•

**gitolite**  
*Gitolite*

- **GKE**  
**Google Kubernetes Engine**

- **GL**  
**Graphics Library**  
*API*

- **GNU**  
**GNU's Not Unix**

- **GnuTLS**  
**GNU's Not Unix Transport Layer Security**

- *SSL TLS DTLS*

- **GPG**  
**GNU Privacy Guard**

- **GPS**  
**Global Positioning System**

- *PPS*

- **GPSD**  
**GPS daemon**

- *Chrony*

**GPU**  
**Graphics Processing Unit**

•

**Graylog**

*gelf*

•

**GPO**  
**Group Policy Object**

•

**GRUB**  
**GRand Unified Bootloader**

**GSSAPI**  
**Generic Security Services Application Program Interface**  
*API*

•

**GTK**  
**GIMP Toolkit**

*graphical user interfaces (GUIs)*

•

**GUI**  
**Graphical User Interface**

**GZIP**  
**GNU Zip**

*GNU*

- 
- *fileset*

- 

- 

## H

**HA**  
**High Availability**

- 

**HBA**  
**Host Bus Adapter**

- 

**HMAC**  
**Hash-based Message Authentication Code**  
*Message Authentication Code*

- 

**HMC**  
**Hardware Management Console**

*OS*

**hostgroup**

- 

**hostname**

*IP address*

**HOTP**  
**HMAC-based One-Time Password**

*OTP*

*HMAC*

**hotplug**

**HPB**  
**Host Physical Bits**

*CPU*

**HPC**  
**High Performance Computing**

**HSG**  
**High-availability Storage Group**

**HSV**  
**Highly-available Storage Virtualization**

**HTCP**  
**Hyper Text Caching Protocol**

**HTML**  
**HyperText Markup Language**

**HTTP**  
**HyperText Transfer Protocol**

**HTTDP**  
**HyperText Transfer Protocol Daemon**

**HTTPS**  
**HyperText Transfer Protocol Secure**

**hugepage**

*Translation Lookaside Buffer (TLB)*

**HWE**  
**Hardware Enablement**

## I

**ICMP**  
Internet Control Message Protocol

**ICP**  
Internet Cache Protocol

**IDENT**  
Identification Protocol

**IMAP**  
Internet Messages Access Protocol

**init**  
initialization

**Intel-TDX**  
Intel Trust Domain Extensions

•

**I/O**  
Input/Output

**IOMMU**  
Input-Output Memory Management Unit

**IoT**  
Internet of Things

**IP**  
Internet Protocol

*datagram*

- *Networking key concepts*

•



## IP address

- *Networking key concepts*

- 

**IPC**  
**Inter-Process Communication**

**IPL**  
**Initial Program Load**

**IPMI**  
**Intelligent Platform Management Interface**

**IPP**  
**Internet Printing Protocol**

**IPSec**  
**Internet Protocol Security**

**IPVS**  
**IP Virtual Server**

**IQN**  
**iSCSI Qualified Name**

**IRC**  
**Internet Relay Chat**

**ISC**  
**Internet Systems Consortium**

**iSCSI**  
**Internet Small Computer System Interface**  
*SCSI*

- *iSCSI initiator (or client)*

-

- 

**ISO**  
**International Organization for Standardization**

**ISP**  
**Internet Service Provider**

**J**

**jitter**

- 

**journal****d**

*temd*

sys-

- *journal*d.conf(5)<sup>742</sup>
- Docker journald documentation<sup>743</sup>

- 

**JSON**  
**JavaScript Object Notation**

- The JSON website<sup>744</sup>

---

<sup>742</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/journald.conf.5.html>

<sup>743</sup> <https://docs.docker.com/engine/logging/drivers/journald/>

<sup>744</sup> <https://www.json.org/json-en.html>

## K

**KDC**  
**Key Distribution Center**

**keepalive**

**Kerberos**

**keypair**

**keyring**

**keysalt**

**keyservers**

**keytab**

**keytool**

**KVM**  
**Kernel-based Virtual Machine**

## L

**LAN**  
**Local Area Network**

**LDAP**  
**Lightweight Directory Access Protocol**

**LDIF**  
**LDAP Data Interchange Format**

**lightervisor**

**LinuxONE**

**Load-balancing**

**localhost**

*hostname*

*IP address*

•

**Log files**

**Logstash**

**Logwatch**

**LPAR**

**Logical Partition**

**LSI**

**Logic Systems Incorporated**

**LTS**

**Long-Term Support**

- [Ubuntu Releases](https://documentation.ubuntu.com/project/release-team/ubuntu-releases/)<sup>745</sup>

**LU**

**Logical Unit**

**LUA**

**Lua Scripting Language**

**LUKS**

**Linux Unified Key Setup**

**LUN**

**Logical Unit Number**

**LV**

**Logical Volume**

---

<sup>745</sup> <https://documentation.ubuntu.com/project/release-team/ubuntu-releases/>

**LVM**  
**Logical Volume Manager**

**LXC**  
**Linux Containers**

**LXD**  
**Linux Container Daemon**

## **M**

**MAAS**  
**Metal as a Service**

**MAC**  
**Message Authentication Code**

**MAC address**

**manual page**

**MCE**  
**Machine Check Exception**

**MD5**  
**Message Digest Algorithm 5**

**MDA**  
**Mail Delivery Agent**

**mdev**  
**Minimal Device Manager**

**metapackage**

**METAR**  
**Meteorological Aerodrome Report**

**microservices**

**microVMs**

**MOTD**  
**Message of the Day**

**mount points**

**MPIO**  
**Multipath Input/Output**

**MSA**  
**Modular Smart Array**

**MTA**  
**Mail Transfer Agent**

**MTU**  
**Maximum Transmission Unit**

**MUA**  
**Mail User Agent**

**Multipass**

**Multipath**

**Multiview**

**MySQL**

**N**

**nameserver**

**namespace**

**NAS**  
**Network Attached Storage**

- *DAS*

- *SAN*

**NAT**  
**Network Address Translation**

**Netboot**

**Netfilter**

**Netplan**

**NFS**  
**Network File System**

**NFV**  
**Network Functions Virtualization**

**nginx**

**NIC**  
**Network Interface Card**

**NIS**  
**Network Information Service**

**NMI**  
**Non-Maskable Interrupt**

**NRPE**  
**Nagios Remote Plugin Executor**

**NSCQ**  
**Network System Configuration Queue**

**NSS**  
**Name Service Switch**

**NTP**  
**Network Time Protocol**

**NTS**  
**Network Time Security**

**NUMA**  
**Non-Uniform Memory Access**

**Nvidia**

**NVMe**  
**Non-Volatile Memory Express**

**NVRAM**  
**Non-Volatile Random Access Memory**

**NVSwitch**  
**NVIDIA NVLink Switch**

**O**

**OCF**  
**Open Cluster Framework**

**OCFS**  
**Oracle Cluster File System**

**OCI**  
**Open Container Initiative**

**OCSF**  
**Online Certificate Status Protocol**

**OpenLDAP**

**OpenSC**  
**Open Smart Card**

**OpenSSH**

**OpenSSL**

**OpenStack**



**OpenSUSE**

**OpenVPN**

**OpenVZ**

**OpenWRT**

**OS**  
**Operating System**

**OSA**  
**Open Systems Adapter**

**OSI**  
**Open Systems Interconnection**

**OSPF**  
**Open Shortest Path First**

.

**OTE**  
**Operational Test and Evaluation**

**OTP**  
**One-Time Password**

**OU**  
**Organizational Unit**

**OverlayFS**

**OVS**  
**Open vSwitch**

## P

**PAM**  
**Pluggable Authentication Module**

**passthrough**

**PB**

*TB*

**PCI**  
**Peripheral Component Interconnect**

**PCIe**  
**Peripheral Component Interconnect Express**

**PCS**  
**Pacemaker/Corosync Stack**

**PDC**  
**Primary Domain Controller**

**PEM**  
**Privacy Enhanced Mail**

**Petitboot**

**PgSQL**

**PHP**  
**PHP: HyperText Preprocessor**

**PID**  
**Process Identifier**

**pingable**

**PIV**  
**Personal Identity Verification**

**PKCS**  
**Public-Key Cryptography Standards**

**PKI**  
**Public Key Infrastructure**

**pluggable**

**PMD**  
**Poll Mode Driver**

**pockets**

- [Ubuntu Project: pockets](#)<sup>746</sup>

**POSIX**  
**Portable Operating System Interface (for UNIX)**

**Postcopy**

**Postfix**

**PostgreSQL**

**PostScript**

**Power 8**  
**Power 9**

**PowerShell**

**PPA**  
**Personal Package Archive**

**ppc**  
**PowerPC**

**PPD**  
**PostScript Printer Description**

**PPPoE**  
**Point-to-Point Protocol over Ethernet**

---

<sup>746</sup> <https://documentation.ubuntu.com/project/how-ubuntu-is-made/concepts/package-archive/#pockets>

•  
**PPS**  
Pulse Per Second

**Preboot**

**preseed**

**Prometheus**

**proxy**

**PTP**  
Precision Time Protocol

**PTR**  
Pointer Record

**PXE**  
Preboot Execution Environment

**PXELINUX**  
PXE Linux Loader

## **Q**

**QA**  
Quality Assurance

**QDevice**  
Quorum Device

**QEMU**  
Quick Emulator

**QETH**  
QDIO Ethernet

**quickstart**

## **R**

### **RAM**

**Random Access Memory**

**RangeSize**

### **RDAC**

**Redundant Disk Array Controller**

### **RDBMS**

**Relational Database Management System**

### **RDN**

**Relative Distinguished Name**

**renderer**

### **REXX**

**Restructured Extended Executor**

### **RFC**

**Request For Comments**

### **rid**

**Relative Identifier**

### **RIP**

**Router Information Protocol**

•

### **RISC-V**

**Reduced Instruction Set Computing - Version Five**

**Rocks**

### **ROM**

**Read-Only Memory**

**rootDN**  
**Root Distinguished Name**

**rootfs**  
**Root File System**

**routable**

**RSA**  
**Rivest–Shamir–Adleman**

**rsync**  
**Remote Sync**

- *FTP*
- *SCP*
- *SFTP*

**RTC**  
**Real-Time Clock**

**runtime**

**S**

**Samba**

- *Introduction to Samba*

**SAN**  
**Storage Area Network**

- *NAS*

## sandboxed

- - 
  - 
  - 
  -
- 
- 

**SANLOCK**  
Storage Area Network Locking Daemon

*SAN*

**SASL**  
Simple Authentication and Security Layer

**SBD**  
Storage-Based Death

**sbin**  
System Binaries

## schema

**SCP**  
Secure Copy Protocol

*SSH*

- *FTP*
- *SFTP*
- *rsync*

**Scrollback**

•

**SCSI**  
**Small Computer System Interface**

**SDN**  
**Software-Defined Networking**

**seccomp**  
**Secure computing mode**

**SFTP**  
**SSH File Transfer Protocol**

*SSH*

- *FTP*
- *SCP*
- *rsync*

**SGI**  
**Silicon Graphics Inc.**

**SHA**  
**Secure Hash Algorithm**

**sharding**

**Shell**



**SHM**  
**Shared Memory**

**Shorewall**  
**Shoreline Firewall**

**SIDs**  
**Security Identifiers**

**SIMD**  
**Single Instruction, Multiple Data**

**slapd**  
**Standalone LDAP Daemon**

**SLAT**  
**Second Level Address Translation**

**smart card**

**SMB**  
**Server Message Block**

**SMS**  
**Short Message Service**

**SMTP**  
**Simple Mail Transfer Protocol**

**SMTPS**  
**SMTP Secure**

**Snap**

**snapd**

**snapshot**

**Snap store**

**SNMP**  
**Simple Network Management Protocol**

**SOA**  
**Start of Authority**

**Solaris**

**SPC**  
**SCSI Primary Commands**

**Splunk**

**SRU**  
**Stable Release Update**

**SSD**  
**Solid State Drive**

**SSH**  
**Secure Shell**

**SSH-key**  
**Secure Shell Key**

*SSH*

*OpenSSH*

*SSH*

**SSI**  
**Server-Side Includes**

**SSL**  
**Secure Sockets Layer**

**SSO**  
**Single Sign-On**

**SSSD**  
**System Security Services Daemon**

**stateful**

**STDIN**  
**Standard Input**

**STDOUT**  
Standard Output

**STDERR**  
Standard Error

**STK**  
StorageTek

**storage**

**subcommand**

**Subiquity**

**subnet**  
**subnetwork**

**substring**

**subvolume**

**sudo**  
**superuser do**

**superblock**

**symlink**

**syslog**

**systemctl**

**Systemd**

## T

**tasksel**  
Task selector

## TB

*GB*

**TCP**  
Transmission Control Protocol

*TCP*

**TDX**  
Trust domain extensions

- 
- 

**TEE**  
Trusted Execution Environment

**TFTP**  
Trivial File Transfer Protocol

*UDP*

*TCP*

**TGS**  
Ticket Granting Service

**TGT**  
Ticket Granting Ticket

**TLB**  
Translation Lookaside Buffer

*CPU*

**TLS**  
**Transport Layer Security**

*HTTPS*

**tmpfs**  
**Temporary Filesystem**

**tmux**  
**Terminal Multiplexer**

**topologies**

**TOTP**  
**Time-based One-Time Password**

**TPM**  
**Trusted Platform Module**

**Traceback**

**Traceroute**

**TTYS**  
**Teletype Terminals**

- 
- 
- 

**TXT**  
**Trusted Execution Technology**

- 
- 
- 

**U**

**UDA**  
**Unified Data Architecture**

**UDP**  
**User Datagram Protocol**

**UEFI**  
**Unified Extensible Firmware Interface**

**UFW**  
**Uncomplicated Firewall**

**UID**  
**User Identifier**

**UI**  
**User Interface**

**unicast**

**unmount**

**untrusted**

1.

2.

3.

**uptime**

**URI**

**Uniform Resource Identifier**

**URL**

**Uniform Resource Locator**

**userspace**

**USN**

**Update Sequence Number**

**usr**

**UUIDs**

**Universally Unique Identifiers**

## V

**vCPU**  
**Virtual Central Processing Unit**

*VM*

*CPU*

**VCS**  
**Version Control System**

**veth**  
**Virtual Ethernet**

**VFIO**  
**Virtual Function I/O**

*VM*

*Input-Output Memory Management Unit*

**VFS**  
**Virtual File System**

**VFes**  
**Virtual Functions**

**VG**  
**Volume Group**

*Logical Volume Manager*

**vGPU**  
**Virtual Graphics Processing Unit**

*GPU*

*VM*

**virsh**  
**Virtual Shell**

**VirtIO**  
**Virtualization I/O**



**virtual**

**virtualization**

*virtual*

**VLAN**  
**Virtual Local Area Network**

**VM**  
**Virtual Machine**

*CPU RAM*

**VNC**  
**Virtual Network Computing**

**VPN**  
**Virtual Private Network**

*IP*

**VRRP**  
**Virtual Router Redundancy Protocol**

**vsftpd**  
**Very Secure FTP Daemon**

## W

**WAL**  
**Write-Ahead Log**

**WAN**  
**Wide Area Network**

**WCCP**  
**Web Cache Communication Protocol**

**Web server**

**winbind**  
**Windows Bind**

*AD*

*Samba*

**WireGuard**

*VPN*

**WLAN**  
**Wireless Local Area Network**

**WSGI**  
**Web Server Gateway Interface**

**WWID**  
**World Wide Identifier**

*LUN*

**X**

**X.509**

*Certificate Authority*

**xhtml**

**Extensible HyperText Markup Language**

**XML**

**Extensible Markup Language**

**Y**

**YAML**

**YAML Ain't Markup Language**

**Yubikey**

**Z**

**zFCP**

**zSeries Fibre Channel Protocol**

*SCSI*

**ZFS**

**Zettabyte File System**

**z/VM**

**CLI basics**

*the terminal*

*Welcome to*

## Navigation

| Command                                | Description                                                                               | Online manual page                               |
|----------------------------------------|-------------------------------------------------------------------------------------------|--------------------------------------------------|
| <code>pwd</code>                       | Print the current working directory                                                       | <a href="#"><i>pwd(1)</i></a> <sup>747</sup>     |
| <code>ls</code>                        | List all files and directories in the current directory                                   | <a href="#"><i>ls(1)</i></a> <sup>748</sup>      |
| <code>ls -l</code>                     | Long list, with permissions, owner, size, and date                                        |                                                  |
| <code>ls -a</code>                     | List all, including hidden files (file names starting with <code>.</code> )               |                                                  |
| <code>ls -la</code> or <code>ll</code> | Long listing including hidden files                                                       |                                                  |
| <code>ls -ld &lt;dir&gt;</code>        | Show the directory entry itself, not its contents, for directory <code>&lt;dir&gt;</code> |                                                  |
| <code>ls -l &lt;file&gt;</code>        | Show file type, permissions, owner, and size for <code>&lt;file&gt;</code>                |                                                  |
| <code>cd &lt;path&gt;</code>           | Change to the specified directory                                                         | <a href="#"><i>cd(1posix)</i></a> <sup>749</sup> |
| <code>cd ..</code>                     | Move up to the parent directory                                                           |                                                  |
| <code>cd ~</code>                      | Return to the home directory                                                              |                                                  |
| <code>cd -</code>                      | Return to the previous directory                                                          |                                                  |
| <code>cd /</code>                      | Move to the root directory                                                                |                                                  |
| <code>cd ../../</code>                 | Move up two levels at once                                                                |                                                  |

<sup>747</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/pwd.1.html>

<sup>748</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/ls.1.html>

<sup>749</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/cd.1posix.html>

## Working with the filesystem

| Command                                     | Description                           | Online manual page                             |
|---------------------------------------------|---------------------------------------|------------------------------------------------|
| <code>mkdir &lt;name&gt;</code>             | Create a new directory                | <a href="#"><i>mkdir(1)</i></a> <sup>750</sup> |
| <code>rmdir &lt;name&gt;</code>             | Remove an empty directory             | <a href="#"><i>rmdir(1)</i></a> <sup>751</sup> |
| <code>tree</code>                           | Display directory structure as a tree | <a href="#"><i>tree(1)</i></a> <sup>752</sup>  |
| <code>tree -L &lt;n&gt; &lt;path&gt;</code> | Display tree to depth n               |                                                |

## Working with files

| Command                                            | Description                                                            | Online manual page                             |
|----------------------------------------------------|------------------------------------------------------------------------|------------------------------------------------|
| <code>touch &lt;file&gt;</code>                    | Create a new (empty) file, or update the timestamp of an existing file | <a href="#"><i>touch(1)</i></a> <sup>753</sup> |
| <code>cp &lt;src&gt; &lt;dest&gt;</code>           | Copy a file from a source (src) to a new destination (dest)            | <a href="#"><i>cp(1)</i></a> <sup>754</sup>    |
| <code>cp -a &lt;src&gt; &lt;dest&gt;</code>        | Copy a directory recursively, preserving permissions                   |                                                |
| <code>mv &lt;src&gt; &lt;dest&gt;</code>           | Rename a file or directory                                             | <a href="#"><i>mv(1)</i></a> <sup>755</sup>    |
| <code>mv &lt;src&gt;/file &lt;dest&gt;/file</code> | Move a file without renaming it                                        |                                                |
| <code>rm &lt;file&gt;</code>                       | Remove a file                                                          | <a href="#"><i>rm(1)</i></a> <sup>756</sup>    |
| <code>rm -r &lt;dir&gt;</code>                     | Remove a non-empty directory and all its contents                      |                                                |
| <code>cat &lt;file&gt;</code>                      | Print the contents of <file> to the screen                             | <a href="#"><i>cat(1)</i></a> <sup>757</sup>   |
| <code>less &lt;file&gt;</code>                     | View file contents one page at a time                                  | <a href="#"><i>less(1)</i></a> <sup>758</sup>  |
| <code>head -n 20 &lt;file&gt;</code>               | Print the first 20 lines of a file (default with no flag is 10)        | <a href="#"><i>head(1)</i></a> <sup>759</sup>  |
| <code>tail -n 20 &lt;file&gt;</code>               | Print the last 20 lines of a file (default with no flag is 10)         | <a href="#"><i>tail(1)</i></a> <sup>760</sup>  |
| <code>wc -l &lt;file&gt;</code>                    | Count the number of lines in a file                                    | <a href="#"><i>wc(1)</i></a> <sup>761</sup>    |

<sup>750</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/mkdir.1.html>

<sup>751</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/rmdir.1.html>

<sup>752</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/tree.1.html>

## Users, groups, and permissions

| Command                                                      | Description                                            | Online manual page                             |
|--------------------------------------------------------------|--------------------------------------------------------|------------------------------------------------|
| <code>chmod &lt;mode&gt; &lt;file&gt;</code>                 | Change file permissions (symbolic or numeric notation) | <a href="#"><i>chmod(1)</i></a> <sup>762</sup> |
| <code>chown &lt;user&gt;[:&lt;group&gt;] &lt;file&gt;</code> | Change file owner and group                            | <a href="#"><i>chown(1)</i></a> <sup>763</sup> |
| <code>umask</code>                                           | Show or set the default permissions mask               | <a href="#"><i>umask(2)</i></a> <sup>764</sup> |
| <code>sudo &lt;command&gt;</code>                            | Run a command with administrator privileges            | <a href="#"><i>sudo(8)</i></a> <sup>765</sup>  |

## Searching

| Command                                                                           | Description                                   | Online manual page                             |
|-----------------------------------------------------------------------------------|-----------------------------------------------|------------------------------------------------|
| <code>grep &lt;pattern&gt; &lt;file&gt;</code>                                    | Search for lines matching a pattern in a file | <a href="#"><i>grep(1)</i></a> <sup>766</sup>  |
| <code>grep &lt;name&gt; /etc/passwd</code>                                        | Look up a user account                        |                                                |
| <code>grep &lt;name&gt; /etc/group</code>                                         | Look up group membership                      |                                                |
| <code>find &lt;path&gt; -name &lt;name&gt; -prune -o -name &lt;pattern&gt;</code> | Search for files matching a name pattern      | <a href="#"><i>find(1)</i></a> <sup>767</sup>  |
| <code>which &lt;command&gt;</code>                                                | Show the full path to an executable command   | <a href="#"><i>which(1)</i></a> <sup>768</sup> |
| <code>file &lt;name&gt;</code>                                                    | Determine the type of a file                  | <a href="#"><i>file(1)</i></a> <sup>769</sup>  |

<sup>753</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/touch.1.html>

<sup>754</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/cp.1.html>

<sup>755</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/mv.1.html>

<sup>756</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/rm.1.html>

<sup>757</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/cat.1.html>

<sup>758</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/less.1.html>

<sup>759</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/head.1.html>

<sup>760</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/tail.1.html>

<sup>761</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/wc.1.html>

<sup>762</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/chmod.1.html>

<sup>763</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/chown.1.html>

<sup>764</sup> <https://manpages.ubuntu.com/manpages/resolute/man2/umask.2.html>

<sup>765</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/sudo-rs.8.html>

## CLI in depth

### *The command line in depth*

## Redirecting input and output

| Command                                                     | Description                                                                   |
|-------------------------------------------------------------|-------------------------------------------------------------------------------|
| <code>cmd &gt; file</code>                                  | Redirect stdout (1) to a file (overwrites existing content)                   |
| <code>cmd &gt;&gt; file</code>                              | Append stdout (1) to the end of <file> (does not overwrite existing contents) |
| <code>cmd &lt; file</code>                                  | Pass a file's contents as stdin (0) to a command (cmd)                        |
| <code>cmd &gt; /dev/null</code>                             | Discard stdout (1) entirely                                                   |
| <code>cmd 2&gt; file</code>                                 | Redirect stderr (2) to a file                                                 |
| <code>cmd &gt;&amp; file</code><br><code>2&gt;&amp;1</code> | Redirect both stdout (1) and stderr (2) to a file                             |
| <code>cmd &lt;&lt; EOF</code>                               | Here-document: feed multi-line text as stdin, ended by a line containing EOF  |

## Pipes and command chaining

| Command                                | Description                                                              |
|----------------------------------------|--------------------------------------------------------------------------|
| <code>cmd1   cmd2</code>               | Pipe stdout (1) of cmd1 to stdin (0) of cmd2                             |
| <code>cmd1; cmd2</code>                | Run commands in sequence regardless of exit status                       |
| <code>cmd1 &amp;&amp; cmd2</code>      | Run cmd2 only if cmd1 succeeds (exit code 0)                             |
| <code>cmd1    cmd2</code>              | Run cmd2 only if cmd1 fails (non-zero exit code)                         |
| <code>cmd &amp;</code>                 | Run a command in the background                                          |
| <code>{ cmd1; cmd2; } &gt; file</code> | Group commands and redirect their combined output                        |
| <code>(cmd1; cmd2)</code>              | Run commands in a subshell (side effects don't affect the current shell) |
| <code>cmd1 \</code>                    | Continue a long command on the next line                                 |

<sup>766</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/grep.1.html>

<sup>767</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/find.1.html>

<sup>768</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/which.debianutils.1.html>

<sup>769</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/file.1.html>



## Variables and environment

| Command                       | Description                                             | Online manual page                           |
|-------------------------------|---------------------------------------------------------|----------------------------------------------|
| <code>env</code>              | List all environment variables                          | <a href="#"><i>env(1)</i></a> <sup>770</sup> |
| <code>echo \$VAR</code>       | Print the value of variable VAR                         | See note below this table                    |
| <code>VAR=value</code>        | Set a shell variable (available in current shell only)  |                                              |
| <code>export VAR=value</code> | Set a variable and export it to child processes         |                                              |
| <code>unset VAR</code>        | Remove a variable                                       |                                              |
| <code>echo \$?</code>         | Print the exit status of the last command (0 = success) |                                              |
| <code>echo \$\$</code>        | Print the PID of the current shell                      |                                              |

### Note:

If you are using the bash, dash or zsh shell, `echo` is a builtin command documented under the shell manual page. For example, the [bash builtin shell section](#)<sup>771</sup>. If your shell doesn't have an `echo` builtin command, use the [\*echo\(1\)\*](#)<sup>772</sup> manual page instead.

<sup>771</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/bash.1.html#shell-builtin-commands>

<sup>772</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/echo.1.html>

<sup>770</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/env.1.html>

## Shell expansion

| Command           | Description                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------|
| ~                 | Expands to the home directory (\$HOME)                                                                                 |
| \$VAR,<br>\${VAR} | Variable expansion: replaced by the value of VAR                                                                       |
| \$(command)       | Command substitution: replaced by the output of command                                                                |
| \$((expr))        | Arithmetic expansion: evaluates integer arithmetic, e.g. <code>\$((5 + 3))</code> evaluates to 8                       |
| {a,b,c}           | Brace expansion: generates separate words, e.g. <code>{a,b,c} file</code> → <code>a file b file c file</code>          |
| {1..10}           | Range expansion: generates a sequence, e.g. <code>{1..5} file</code> → <code>1 file 2 file 3 file 4 file 5 file</code> |
| *                 | Glob: matches any sequence of characters in a filename                                                                 |
| ?                 | Glob: matches any single character in a filename                                                                       |
| [abc]             | Glob: matches any one of the listed characters in a filename                                                           |

## Getting help

| Command                             | Description                                | Online manual page           |
|-------------------------------------|--------------------------------------------|------------------------------|
| <code>man &lt;command&gt;</code>    | Open the manual page for a command         |                              |
| <code>man man</code>                | Open the manual page for man itself        | <i>man(1)</i> <sup>773</sup> |
| <code>man -k &lt;keyword&gt;</code> | Search man page descriptions for a keyword |                              |
| <code>&lt;command&gt; --help</code> | Print a brief help summary for a command   |                              |

<sup>773</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/man.1.html>

## System requirements

### Architecture

- *AMD*
- 
- 
- 
- 
- 

Ubuntu certified servers<sup>774</sup>

### Memory

### Storage

---

<sup>774</sup> <https://ubuntu.com/certified/servers>

## High availability

*corresponding commands*

### Migrate from crmsh to pcs

| Action                                         | crmsh                                                     |
|------------------------------------------------|-----------------------------------------------------------|
| Show configuration (raw XML)                   | <code>crm configure show xml</code>                       |
| Show configuration (human-friendly)            | <code>crm configure show</code>                           |
| Show cluster status                            | <code>crm status</code>                                   |
| Put a node in standby mode                     | <code>crm node standby NODE</code>                        |
| Remove a node from standby mode                | <code>crm node online NODE</code>                         |
| Set cluster property                           | <code>crm configure property PROPERTY=VALUE</code>        |
| List resource agent classes                    | <code>crm ra classes</code>                               |
| List available resource agents by standard     | <code>crm ra list ocf</code>                              |
| List available resource agents by OCF provider | <code>crm ra list ocf pacemaker</code>                    |
| List available resource agent parameters       | <code>crm ra info AGENT</code>                            |
| Show available fence agent parameters          | <code>crm ra info stonith:AGENT</code>                    |
| Create a resource                              | <code>crm configure primitive NAME AGENT params PA</code> |
| Show configuration of all resources            | <code>crm configure show</code>                           |
| Show configuration of one resource             | <code>crm configure show RESOURCE</code>                  |
| Show configuration of fencing resources        | <code>crm resource status</code>                          |
| Start a resource                               | <code>crm resource start RESOURCE</code>                  |

| Action                                              | crmsh                                                         |
|-----------------------------------------------------|---------------------------------------------------------------|
| Stop a resource                                     | <code>crm resource stop RESOURCE</code>                       |
| Remove a resource                                   | <code>crm configure delete RESOURCE</code>                    |
| Modify a resource's instance parameters             | <code>crm resource param RESOURCE set PARAMETER=VALUE</code>  |
| Delete a resource's instance parameters             | <code>crm resource param RESOURCE delete PARAMETER</code>     |
| List current resource defaults                      | <code>crm configure show type:rsc_defaults</code>             |
| Set resource defaults                               | <code>crm configure rsc_defaults OPTION=VALUE</code>          |
| List current operation defaults                     | <code>crm configure show type:op_defaults</code>              |
| Set operation defaults                              | <code>crm configure op_defaults OPTION=VALUE</code>           |
| Clear fail counts for a resource                    | <code>crm resource cleanup RESOURCE</code>                    |
| Create a colocation constraint                      | <code>crm configure colocation NAME INFINITY: RESOURCE</code> |
| Create an ordering constraint                       | <code>crm configure order NAME mandatory: RESOURCE</code>     |
| Create a location constraint                        | <code>crm configure location NAME RESOURCE 50: NODE</code>    |
| Move a resource to a specific node                  | <code>crm resource move RESOURCE NODE</code>                  |
| Move a resource away from its current node          | <code>crm resource ban RESOURCE NODE</code>                   |
| Remove any constraints created by moving a resource | <code>crm resource unmove RESOURCE</code>                     |

## See also

- [Set up a distributed replicated block device \(DRBD\)](#)
- [High Availability](#)

## Other tools

- [Terminal Multiplexers](#)
- [pam\\_motd](#)
- [sudo-rs](#)

## Terminal multiplexers

- 
- 
- 
- 
- 
- 

### tmux

*tmux(1)*<sup>775</sup>

## Control key and keybindings

- 

---

<sup>775</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/tmux.1.html>

## Session management

```
$ tmux new-session -s myproject
```

```
$ tmux list-sessions
```

```
$ tmux ls
```

```
$ tmux attach-session -t myproject
```

```
$ tmux a -t myproject
```

```
$ tmux attach-session -t myproject -d
```

```
$ tmux attach
```

```
$ tmux kill-session -t myproject
```

## Window management

- 
- 
- 
- 
- 
- 

### Note:

When you exit the shell in a window (by typing `exit` or pressing `Ctrl+d`), that window closes. If it was the last window in a session, the entire `tmux` session terminates.

## Pane management

- 
- 
- 
- 
- 

- 
- 
- 
-



•

## Console-based screen sharing

```
$ tmux new-session -s shared
```

```
$ tmux attach-session -t shared
```

```
$ tmux attach-session -t shared -r
```

```
$ tmux -S /tmp/shared-session new-session -s collaboration > chmod 777
/tmp/shared-session
```

```
$ tmux -S /tmp/shared-session attach-session -t collaboration
```

**Note:**

For multi-user sharing, ensure both users have appropriate permissions on the socket file. Consider security implications when sharing sessions between different users.

## Configuration

```
$ set -g mouse on
```

```
$ set-option -g history-limit 10000
```

```
$ tmux source-file ~/.tmux.conf
```

## Scripting and automation

*tmux(1)*<sup>776</sup>

---

<sup>776</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/tmux.1.html>

## tmux plugins

```
$ sudo apt install tmux-plugin-manager
```

### Note:

While tmux plugins can be powerful, be aware that they introduce third-party code and configuration into your system. The same considerations discussed in the [Third party repository usage](#) (page 950) documentation apply here - evaluate trustworthiness, security implications, and maintenance status before using plugins, especially in production environments.

- 
- 
- 

## Other options

### Mosh (mobile shell)

```
$ sudo apt install mosh
```

```
$ mosh user@remote-server
```

## GNU Screen

## Byobu

```
$ sudo apt install byobu
```

```
$ byobu
```

- 
- 
- 
- 

- 
- 
- 
- 
- 
-

- 
- 
- 
- 

`pam_motd`

Packages adding to MOTD

Landscape-sysinfo

- 

Landscape<sup>777</sup>

```
System load: 0.0
Usage of /: 25.7% of 8.55GB
Memory usage: 31%
Swap usage: 0%
Processes: 125
Users logged in: 0
IPv4 address for enp5s0: 10.185.198.41
IPv6 address for enp5s0: fd42:50cb:3a48:1f2c:216:3eff:fe27:c18a
```

**Note:**

You can run `landscape-sysinfo` manually at any time.

- 

*filesystem checks (fsck)*

---

<sup>777</sup> <https://ubuntu.com/landscape>

## Other default MOTD entries

- 
- 
- 
- 
- 
- ubuntu-pro-client<sup>778</sup>
- 
- 
- 
- 
- 
- 

## Customization

```
#!/bin/bash

1. Get timestamp and calculate nearest sysadmin day
NOW=$(date +%s)
THIS_YEAR=$(date +%Y)
TARGET_PREV=$(date -d "$((THIS_YEAR-1))-08-01 last friday" +%s)
TARGET_CURR=$(date -d "$THIS_YEAR-08-01 last friday" +%s)
TARGET_NEXT=$(date -d "$((THIS_YEAR+1))-08-01 last friday" +%s)

2. Loop to find the shortest distance
MIN_DIFF=999999999
CHOSEN_DAYS=0
TIMING=""
for TARGET in $TARGET_PREV $TARGET_CURR $TARGET_NEXT; do
 # Calculate difference
 DIFF=$(((TARGET - NOW) / 86400))
```

(continues on next page)

---

<sup>778</sup> [https://documentation.ubuntu.com/pro-client/en/latest/explanations/motd\\_messages/](https://documentation.ubuntu.com/pro-client/en/latest/explanations/motd_messages/)

(continued from previous page)

```
Get absolute value for comparison
if [$DIFF -lt 0] then
 ABS_DIFF=$((DIFF * -1))
 CURRENT_TIMING="ago"
else
 ABS_DIFF=$DIFF
 CURRENT_TIMING="in"
fi

Update if this is the closest date found so far
if [$ABS_DIFF -lt $MIN_DIFF] then
 MIN_DIFF=$ABS_DIFF
 CHOSEN_DAYS=$ABS_DIFF
 TIMING=$CURRENT_TIMING
fi
done

3. Report to gain some admin-love
if ["$CHOSEN_DAYS" -eq 0] then
 echo -e "System Administrator Appreciation Day is Today! Gifts welcome."
elif ["$TIMING" == "ago"] then
 echo -e "System Administrator Appreciation Day is ${CHOSEN_DAYS} days ago but we'd
 appreciate to be treated well every day."
else
 echo -e "System Administrator Appreciation Day is in ${CHOSEN_DAYS} days but we'd
 appreciate to be treated well every day."
fi
```

```
$ chmod +x /etc/update-motd.d/98-admin-reminder
```

```
$ ssh ubuntu@yoursystem
```

```
...
System Administrator Appreciation Day is 181 days ago but we'd appreciate to be
treated well every day.
```

**Note:**

Be careful as this will be executed on every login. The usual pattern to prevent that from slowing down logins is to do anything even slightly complex asynchronously and only cat pre-generated content in the actual MOTD handling. Furthermore consider using stamp files to only update at regular intervals instead of at every login. This example was meant to be kept simple, but the existing serious MOTD entries placed there by the packages make use of such, please look at those for writing your own.

## Further reading

- [update-motd\(5\)](#)<sup>779</sup>
- [weather](#)<sup>780</sup>

## sudo-rs

## Differences

1.
  - [differences-from-original-sudo](#)<sup>781</sup>
  - [aim-of-the-project](#)<sup>782</sup>
- 2.

[See more information](#)<sup>783</sup>

- 3.
4. [LDAP authentication via PAM](#)<sup>784</sup>
5. [CLI flags parity](#)<sup>785</sup>

---

<sup>779</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/update-motd.5.html>

<sup>780</sup> <https://debaday.debian.net/2007/10/04/weather-check-weather-conditions-and-forecasts-on-the-command-line/>

<sup>781</sup> <https://github.com/trifectatechfoundation/sudo-rs#differences-from-original-sudo>

<sup>782</sup> <https://github.com/trifectatechfoundation/sudo-rs?tab=readme-ov-file#aim-of-the-project>

<sup>783</sup> <https://github.com/trifectatechfoundation/sudo-rs/issues/1242>

<sup>784</sup> <https://github.com/trifectatechfoundation/sudo-rs/issues/445>

<sup>785</sup> <https://github.com/trifectatechfoundation/sudo-rs/issues/129>



### *Our security explanations*

- 
- 
- 

## **Security**

### **General configuration**

- *Introduction to security*
- *Security suggestions*

### **Introduction to security**

## Server security guidance

*security suggestions*

*security*

*advanced*

Ubuntu Security documentation<sup>786</sup>

## Ubuntu Pro

Ubuntu Pro<sup>787</sup>

Client<sup>788</sup>

Ubuntu Pro

## Vulnerability management

itory<sup>789</sup>

Main repos-

vulnerabilities<sup>790</sup>

Expanded Security Maintenance<sup>791</sup>

---

<sup>786</sup> <https://documentation.ubuntu.com/security/>

<sup>787</sup> <https://ubuntu.com/pro>

<sup>788</sup> <https://documentation.ubuntu.com/pro-client/en/latest/>

<sup>789</sup> <https://documentation.ubuntu.com/project/how-ubuntu-is-made/concepts/package-archive/>

#components

<sup>790</sup> <https://ubuntu.com/security/cves/about>

<sup>791</sup> <https://documentation.ubuntu.com/security/security-updates/esm/>

## Kernel application hardening

Livepatch here<sup>792</sup>

about

## Security Compliance and Certification

CIS<sup>794</sup>

FIPS<sup>793</sup>

## Reporting vulnerabilities

disclosure policy<sup>795</sup>

## Security suggestions

## For any Ubuntu system

## Keep your system up-to-date

1.

```
$ sudo apt update && sudo apt upgrade
```

```
$ sudo apt install unattended-upgrades
```

*unattended-upgrades(8)*<sup>796</sup>

<sup>792</sup> <https://ubuntu.com/security/livepatch/docs>

<sup>793</sup> <https://documentation.ubuntu.com/security/compliance/fips/>

<sup>794</sup> <https://documentation.ubuntu.com/security/compliance/usg/>

<sup>795</sup> <https://ubuntu.com/security/disclosure-policy>

<sup>796</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/unattended-upgrades.8.html>

2.

- 

*Package management*

- 

*understand the risks and how to minimize them.*

3.

*how to upgrade*

4. **Ubuntu Pro**<sup>797</sup>

- Expanded Security Maintenance (ESM)<sup>798</sup>

- Livepatch<sup>799</sup>

*official documentation*<sup>800</sup>

## Access Control

1.

*principle of least privilege*<sup>801</sup>

- 

- 

- 

*guide on user management*

## Network security

1.

*Firewall*

2.

*guide to*

*OpenSSH*

---

<sup>797</sup> <https://ubuntu.com/pro>

<sup>798</sup> <https://documentation.ubuntu.com/security/security-updates/esm/>

<sup>799</sup> <https://ubuntu.com/security/livepatch/docs>

<sup>800</sup> <https://documentation.ubuntu.com/pro-client/en/latest/>

<sup>801</sup> [https://en.wikipedia.org/wiki/Principle\\_of\\_least\\_privilege](https://en.wikipedia.org/wiki/Principle_of_least_privilege)

## Physical security

- *Smart card authentication*
- *Smart card authentication with SSH*
- *Console security*

## Suggestions for complex setups

Ubuntu Security<sup>802</sup>

## Advanced Access Control

1.

*introduction to OpenLDAP  
on how to set up OpenLDAP*

2.

*introduction to Kerberos*

*set up and use Kerberos*

3.

*introduction to SSSD  
how-to section*

4.

*Active Directory integration  
ADSys  
tation<sup>803</sup>*

*learn how to set it up*

ADSys documen-

5.

*to set up AppArmor*

*how*

<sup>802</sup> <https://documentation.ubuntu.com/security/>

<sup>803</sup> <https://documentation.ubuntu.com/adsys/latest/>

## Security of communications

1.

*Introduction to WireGuard VPN  
Guard VPN*

*set up Wire-*

2.

*about the available clients*

*how to install OpenVPN*

## Cryptography

*libraries*

*Introduction to cryptographic libraries*

*about crypto*

*About certificates*

*the trust store*

*Install root CA certificate in*

## Compliance and auditing

Ubuntu Security documentation<sup>804</sup>

### Authentication

- *Introduction to Kerberos*
- *Introduction to SSSD*
- *About DNSSEC*

### Introduction to Kerberos

### Overview

- 
- *hostname*
- *ACL*
- *DNS*
- 
- 

---

<sup>804</sup> <https://documentation.ubuntu.com/security/compliance/>

- 
- 
- 

### Further reading

- [MIT Kerberos](#)<sup>805</sup>
- [Libera.Chat](#)<sup>807</sup> [Ubuntu Server](#)<sup>806</sup>
- Another guide for installing Kerberos on Debian, includes PKINIT<sup>808</sup>

### Introduction to network user authentication with SSSD

[System Security Services Daemon \(SSSD\)](#)<sup>809</sup>

### Common deployment scenarios

- *Active Directory*
- *LDAP*
- *LDAP and Kerberos*

---

<sup>805</sup> <http://web.mit.edu/Kerberos/>

<sup>806</sup> <https://matrix.to/#/#server:ubuntu.com>

<sup>807</sup> <https://libera.chat/>

<sup>808</sup> <http://techpubs.spinlocksolutions.com/dklar/kerberos.html>

<sup>809</sup> <https://sssd.io/>



## Integration with PAM and NSS

## Troubleshooting

*SSSD troubleshooting guide*

## DNS Security Extensions (DNSSEC)

- 
- 
- 
- 
- 
-

## What is it?

- 
- 

Simplified 12-step DNSSEC validation process<sup>810</sup>

## DNS daemons

| Daemon          | Type                | DNSSEC support                                                                                |
|-----------------|---------------------|-----------------------------------------------------------------------------------------------|
| sys-temd-resolv | Stub solver (local) | Yes. Disabled by default. Controlled via DNSSEC=... setting in /etc/systemd/resolved.conf.d.  |
| dnsmasq         | Stub solver         | Yes. Disabled by default. Controlled via dnssec and conf-file=../trust-anchors.conf settings. |
| bind9           | Recursive Resolver  | Yes. Enabled by default via dnssec-validation auto; setting.                                  |

<sup>810</sup> <https://bind9.readthedocs.io/en/latest/dnssec-guide.html#the-12-step-dnssec-validation-process-simplified>

- 
- 

```
$ sudo mkdir -p /etc/systemd/resolved.conf.d
```

```
$ sudo tee /etc/systemd/resolved.conf.d/10-dnssec.conf >/dev/null <<EOF
[Resolve]
DNSSEC=yes
EOF
```

```
$ sudo systemctl reload systemd-resolved.service
```

```
$ dig @127.0.0.53 isc.org +dnssec
[...]
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1
```

RFC 3225<sup>811</sup>

LP: #2119652<sup>812</sup> ref2<sup>813</sup> ref3<sup>814</sup> LP: #2121483<sup>815</sup>

<sup>811</sup> <https://datatracker.ietf.org/doc/html/rfc3225>

<sup>812</sup> <https://bugs.launchpad.net/bugs/2119652>

<sup>813</sup> <https://lists.fedoraproject.org/archives/list/devel@lists.fedoraproject.org/message/AFHNUEHKC5KJVGBGSJBH2BMESUAGDF4H/>

<sup>814</sup> <https://lists.fedoraproject.org/archives/list/devel@lists.fedoraproject.org/message/P63RI3VBQ7NGL3AKMTR7PCVHVSCPYPCLF/>

<sup>815</sup> <https://bugs.launchpad.net/bugs/2121483>

**Warning:**

Be aware that enforcing DNSSEC in strict mode can lead to errors, especially for local, unsigned domains and you would only be able to reach such services by accessing them through their IP address directly. For example this could manifest itself by errors like `DNS_PROBE_FINISHED_NXDOMAIN` in your browser, when trying to access services in the local network.

- <https://internet.nl/test-connection/>
- <https://wander.science/projects/dns/dnssec-resolver-test/>

**Note:**

In case of issues with Domain Name resolution, make sure to remove relevant drop-in configs in `/etc/resolved.conf.d` and execute `systemctl reload systemd-resolved.service`. This will reset any DNSSEC configuration to the default `DNSSEC=no` and can be confirmed by executing `resolvectl dnssec`.

**New Resource Records (RRs)**

- 
- 
- 
- 

```
$ dig @1.1.1.1 +dnssec -t MX isc.org
```

```
; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @1.1.1.1 +dnssec -t MX isc.org
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51256
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

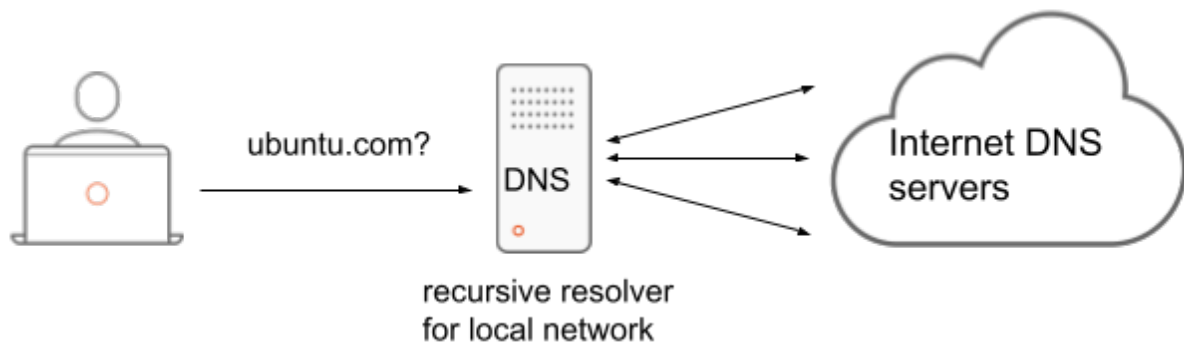
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
;; QUESTION SECTION:
;isc.org. IN MX

;; ANSWER SECTION:
isc.org. 300 IN MX 5 mx.pao1.isc.org.
isc.org. 300 IN MX 10 mx.ams1.isc.org.
isc.org. 300 IN RRSIG MX 13 2 300 20241029080338
20241015071431 27566 isc.org. LG/
cvFmZ8jLz+CM14foaCtwsyCTwKXfVBZV2jcl2UV8zV79QRLs0YXJ3 sjag1vYCqc+Q5AwUi2DB8L/
wZR6EJQ==

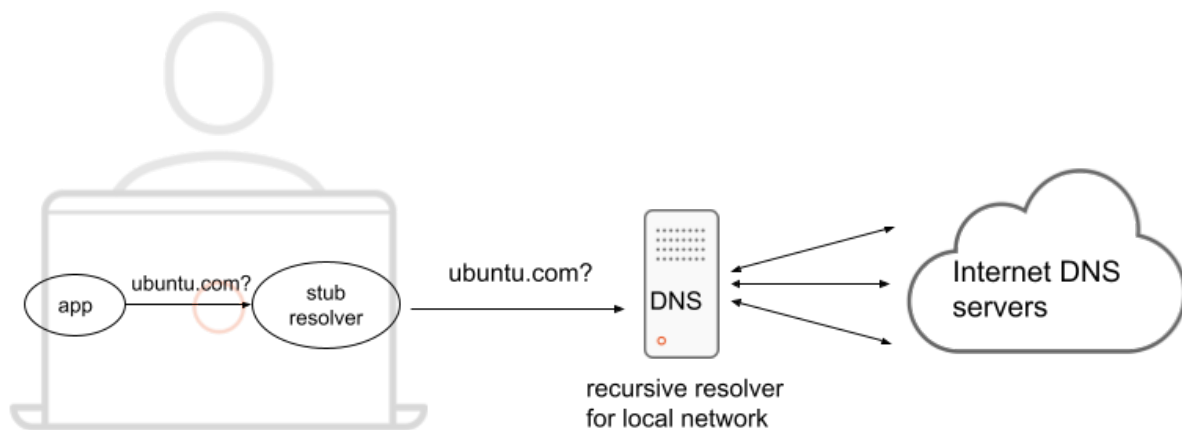
;; Query time: 199 msec
;; SERVER: 1.1.1.1#53(1.1.1.1) (UDP)
;; WHEN: Tue Oct 22 16:44:33 UTC 2024
;; MSG SIZE rcvd: 187
```

## Other uses

Where does the DNSSEC validation happen?



- 
- 



*hostname*

```
nameserver 127.0.0.53
options edns0 trust-ad
```

```
Global
 Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
 resolv.conf mode: stub

Link 12 (eth0)
 Current Scopes: DNS
```

(continues on next page)

(continued from previous page)

```

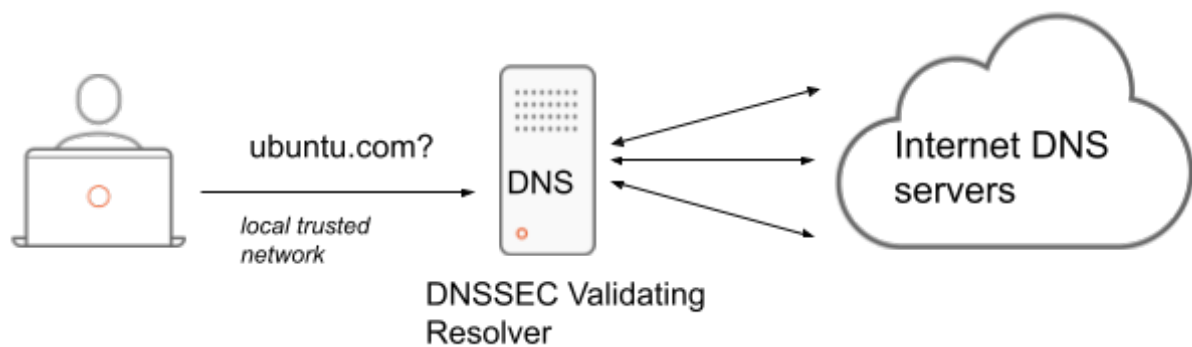
Protocols: +DefaultRoute -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
Current DNS Server: 10.10.17.1
DNS Servers: 10.10.17.1
DNS Domain: lxd

```

## DHCP

- 
- 
- 
- 

## Validating Resolver



### Note:

Up to Ubuntu 24.04 LTS (version 1:9.18.34-1) the default was explicitly stated as `dnssec-validation auto;` in `/etc/bind/named.conf.options`.

```
nameserver 127.0.0.53
options edns0 trust-ad
```

*resolv.conf(5)*<sup>816</sup>

- 
- 
- 
- 

```
nameserver 10.10.17.11
options edns0 trust-ad
```

- 
- 

---

<sup>816</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/resolv.conf.5.html>



## Local DNSSEC validation

*DNS-over-TLS (DoT)   DNS-over-HTTPS (DoH)*

```
$ resolvectl dnssec
```

```
Global: no
```

```
Link 44 (eth0): no
```

```
$ resolvectl query --type=MX isc.org
```

```
isc.org IN MX 10 mx.ams1.isc.org
```

```
-- link: eth0
```

```
isc.org IN MX 5 mx.pao1.isc.org
```

```
-- link: eth0
```

```
-- Information acquired via protocol DNS in 37.2ms.
```

```
-- Data is authenticated: no; Data was acquired via local or encrypted transport: no
```

```
-- Data from: network
```

```
$ sudo resolvectl dnssec eth0 true
```

```
$ resolvectl query --type=MX isc.org
```

```
isc.org IN MX 5 mx.pao1.isc.org -- link: eth0
isc.org IN MX 10 mx.ams1.isc.org -- link: eth0

-- Information acquired via protocol DNS in 3.0ms.
-- Data is authenticated: yes; Data was acquired via local or encrypted transport:
no
-- Data from: network
```

```
-- Data is authenticated: yes
```

## What happens when DNSSEC validation fails

```
$ resolvectl query www.dnssec-failed.org

www.dnssec-failed.org: resolve call failed: Could not resolve 'www.dnssec-failed.org
', server or network returned error SERVFAIL
```

```
Oct 22 17 14:50 n-dns named[285]: validating dnssec-failed.org/DNSKEY: no valid signature
found (DS)
Oct 22 17 14:50 n-dns named[285]: no valid RRSIG resolving 'dnssec-failed.org/DNSKEY/IN':
68.87.68.244#53
...
Oct 22 17 14:52 n-dns named[285]: broken trust chain resolving 'www.dnssec-failed.org/
AAAA/IN': 68.87.72.244#53
```

```
$ sudo resolvectl dnssec eth0 true
```

```
$ resolvectl query www.dnssec-failed.org

www.dnssec-failed.org: resolve call failed: DNSSEC validation failed: no-signature
```

```
$ host www.dnssec-failed.org
```

```
Host www.dnssec-failed.org not found: 2(SERVFAIL)
```

## Further reading

- [DNSSEC - What Is It and Why Is It Important](#)<sup>817</sup>
- [Tool to visualize the DNSSEC chain of trust of a domain](#)<sup>818</sup>
- [DANE](#)<sup>819</sup>
- [RFC 4255](#)<sup>820</sup>
- *Basic DNSSEC troubleshooting*

## Cryptography

- *Our cryptography section*
- *Certificates*
- *How to obtain certificates*

## Cryptography

### Introduction to cryptographic libraries

- 
- 
- 
- 
- 

---

<sup>817</sup> <https://www.icann.org/resources/pages/dnssec-what-is-it-why-important-2019-03-05-en>

<sup>818</sup> <https://dnsviz.net/>

<sup>819</sup> [https://en.wikipedia.org/wiki/DNS-based\\_Authentication\\_of\\_Named\\_Entities](https://en.wikipedia.org/wiki/DNS-based_Authentication_of_Named_Entities)

<sup>820</sup> <https://datatracker.ietf.org/doc/html/rfc4255>

- 
- 
- 
- *GnuTLS*
- 
- 
- *gcrypt*

**Determining which libraries are being used by an application**

**Follow dynamic links**

**Check static linking**

Look at the plugins

Examine the execution of external binaries

Indirect use of libraries or executables

Check documentation more carefully

Detailed examples of how to locate crypto libraries

Use dpkg to check package dependencies

```
$ dpkg -S /usr/bin/lynx
```

```
lynx: /usr/bin/lynx
```

```
$ dpkg -s lynx | grep -E "^(Depends|Recommends)"
```

```
Depends: libbsd0 (>= 0.0), libbz2-1.0, libc6 (>= 2.34), libgnutls30 (>= 3.7.0),
libidn2-0 (>= 2.0.0), libncursesw6 (>= 6), libtinfo6 (>= 6), zlib1g (>= 1:1.1.4),
lynx-common
Recommends: mime-support
```

## Using ldd to list dynamic libraries

```
$ ldd /usr/bin/lynx

linux-vdso.so.1 (0x00007ffffd2df000)
libz.so.1 => /lib/x86_64-linux-gnu/libz.so.1 (0x00007feb69d77000)
libbz2.so.1.0 => /lib/x86_64-linux-gnu/libbz2.so.1.0 (0x00007feb69d64000)
libidn2.so.0 => /lib/x86_64-linux-gnu/libidn2.so.0 (0x00007feb69d43000)
libncursesw.so.6 => /lib/x86_64-linux-gnu/libncursesw.so.6 (0x00007feb69d07000)
libtinfo.so.6 => /lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007feb69cd5000)
libgnutls.so.30 => /lib/x86_64-linux-gnu/libgnutls.so.30 (0x00007feb69aea000)
libbsd.so.0 => /lib/x86_64-linux-gnu/libbsd.so.0 (0x00007feb69ad0000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007feb698a8000)
libunistring.so.2 => /lib/x86_64-linux-gnu/libunistring.so.2
(0x00007feb696fe000)
libp11-kit.so.0 => /lib/x86_64-linux-gnu/libp11-kit.so.0 (0x00007feb695c3000)
libtasn1.so.6 => /lib/x86_64-linux-gnu/libtasn1.so.6 (0x00007feb695ab000)
libnettle.so.8 => /lib/x86_64-linux-gnu/libnettle.so.8 (0x00007feb69565000)
libhogweed.so.6 => /lib/x86_64-linux-gnu/libhogweed.so.6 (0x00007feb6951b000)
libgmp.so.10 => /lib/x86_64-linux-gnu/libgmp.so.10 (0x00007feb69499000)
/lib64/ld-linux-x86-64.so.2 (0x00007feb69fe6000)
libmd.so.0 => /lib/x86_64-linux-gnu/libmd.so.0 (0x00007feb6948c000)
libffi.so.8 => /lib/x86_64-linux-gnu/libffi.so.8 (0x00007feb6947f000)
```

```
$ objdump -x /usr/bin/lynx|grep NEEDED
```

```
NEEDED libz.so.1
NEEDED libbz2.so.1.0
NEEDED libidn2.so.0
NEEDED libncursesw.so.6
NEEDED libtinfo.so.6
NEEDED libgnutls.so.30
NEEDED libbsd.so.0
NEEDED libc.so.6
```

```
$ lddtree /usr/bin/lynx

lynx => /usr/bin/lynx (interpreter => /lib64/ld-linux-x86-64.so.2)
 libz.so.1 => /lib/x86_64-linux-gnu/libz.so.1
 libbz2.so.1.0 => /lib/x86_64-linux-gnu/libbz2.so.1.0
 libidn2.so.0 => /lib/x86_64-linux-gnu/libidn2.so.0
 libunistring.so.2 => /lib/x86_64-linux-gnu/libunistring.so.2
 libncursesw.so.6 => /lib/x86_64-linux-gnu/libncursesw.so.6
 libtinfo.so.6 => /lib/x86_64-linux-gnu/libtinfo.so.6
 libgnutls.so.30 => /lib/x86_64-linux-gnu/libgnutls.so.30
 libp11-kit.so.0 => /lib/x86_64-linux-gnu/libp11-kit.so.0
 libffi.so.8 => /lib/x86_64-linux-gnu/libffi.so.8
 libtasn1.so.6 => /lib/x86_64-linux-gnu/libtasn1.so.6
 libnettle.so.8 => /lib/x86_64-linux-gnu/libnettle.so.8
 libhogweed.so.6 => /lib/x86_64-linux-gnu/libhogweed.so.6
 libgmp.so.10 => /lib/x86_64-linux-gnu/libgmp.so.10
 ld-linux-x86-64.so.2 => /lib64/ld-linux-x86-64.so.2
 libbsd.so.0 => /lib/x86_64-linux-gnu/libbsd.so.0
 libmd.so.0 => /lib/x86_64-linux-gnu/libmd.so.0
 libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6
```

## Check package headers for static linking

- 
- 

```
$ dpkg -s rclone | grep -E "^(Depends|Recommends)"

Depends: libc6 (>= 2.34)
```

```
$ dpkg -s rclone | grep Built-Using
```

```
Built-Using: go-md2man-v2 (= 2.0.1+ds1-1), golang-1.18 (= 1.18-1ubuntu1), golang-bazil-fuse (= 0.0~git20160811.0.371fbbd-3), ...
```

<https://git.launchpad.net/ubuntu/+source/rclone/tree/debian/control?h=ubuntu/jammy-devel#n7>

```
$ apt-cache show golang-golang-x-crypto-dev | grep ^Source:
```

```
Source: golang-go.crypto
```

#### Note:

If there is no Source: line, then it means the name of the source package is the same as the binary package that was queried.

### Next steps

- [OpenSSL guide](#)
- [GnuTLS guide](#)
- [Troubleshooting TLS/SSL](#)

### Common crypto libraries



## OpenSSL

### Structure of the config file

*SSL config manual page<sup>821</sup>*

```
openssl_conf = <name-of-conf-section>

[<name-of-conf-section>]
ssl_conf = <name-of-ssl-section>

[<name-of-ssl-section>]
server = <name of section>
client = <name of section>
system_default = <name of section>
```

```
openssl_conf = openssl_init

[openssl_init]
ssl_conf = ssl_sect

[ssl_sect]
system_default = system_default_sect

[system_default_sect]
CipherString = DEFAULT:@SECLEVEL=2
```

*SSL\_CTX\_set\_security\_level(3)<sup>822</sup>*

---

<sup>821</sup> <https://manpages.ubuntu.com/manpages/resolute/man5ssl/config.5ssl.html>

<sup>822</sup> [https://manpages.ubuntu.com/manpages/resolute/man3/SSL\\_CTX\\_set\\_security\\_level.3.html](https://manpages.ubuntu.com/manpages/resolute/man3/SSL_CTX_set_security_level.3.html)

## Note:

In Ubuntu Jammy, TLS versions below 1.2 are **disabled** in OpenSSL's SECLEVEL=2 due to [this patch](#)<sup>823</sup>.

<sup>823</sup> <https://git.launchpad.net/ubuntu/+source/openssl/tree/debian/patches/tls1.2-min-seclevel2.patch?h=ubuntu/jammy-devel>

CLEVEL=2<sup>824</sup>

set to SE-

openssl ciphers<sup>825</sup>

```
$ openssl ciphers -s -v DEFAULT:@SECLEVEL=2
```

```
TLS_AES_256_GCM_SHA384 TLSv1.3 Kx=any Au=any Enc=AESGCM(256)
Mac=AEAD
TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any Au=any Enc=CHACHA20/
POLY1305(256) Mac=AEAD
TLS_AES_128_GCM_SHA256 TLSv1.3 Kx=any Au=any Enc=AESGCM(128)
Mac=AEAD
ECDHE - ECDSA - AES256 - GCM - SHA384 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESGCM(256)
Mac=AEAD
(...)
```

## Note:

The `openssl ciphers` command will output even ciphers that are not allowed, unless the `-s` switch is given. That option tells the command to list only **supported** ciphers.

CONF\_cmd(3)<sup>826</sup>

SSL\_

## Cipher strings, cipher suites, cipher lists

phers<sup>827</sup>

openssl ci-

<sup>824</sup> <https://git.launchpad.net/ubuntu/+source/openssl/tree/debian/rules?h=ubuntu/jammy-devel#n15>

<sup>825</sup> <https://docs.openssl.org/3.0/man1/openssl-ciphers/>

<sup>826</sup> [https://manpages.ubuntu.com/manpages/resolute/man3/SSL\\_CONF\\_cmd.3.html](https://manpages.ubuntu.com/manpages/resolute/man3/SSL_CONF_cmd.3.html)

<sup>827</sup> <https://docs.openssl.org/3.0/man1/openssl-ciphers/>

- upstream documentation<sup>828</sup>
- 

```
[system_default_sect]
CipherString = DEFAULT:@SECLEVEL=2
CipherSuites = TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
```

## openssl ciphers examples

```
$ openssl ciphers -s -v
```

<sup>828</sup> <https://docs.openssl.org/3.0/man1/openssl-ciphers/>

|                                                               |                 |          |                 |
|---------------------------------------------------------------|-----------------|----------|-----------------|
| TLS_AES_256_GCM_SHA384<br>Mac=AEAD                            | TLSv1.3 Kx=any  | Au=any   | Enc=AESGCM(256) |
| TLS_CHACHA20_POLY1305_SHA256<br>POLY1305(256) Mac=AEAD        | TLSv1.3 Kx=any  | Au=any   | Enc=CHACHA20/   |
| TLS_AES_128_GCM_SHA256<br>Mac=AEAD                            | TLSv1.3 Kx=any  | Au=any   | Enc=AESGCM(128) |
| ECDHE - ECDSA - AES256 - GCM - SHA384<br>Mac=AEAD             | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=AESGCM(256) |
| ECDHE - RSA - AES256 - GCM - SHA384<br>Mac=AEAD               | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=AESGCM(256) |
| DHE - RSA - AES256 - GCM - SHA384<br>Mac=AEAD                 | TLSv1.2 Kx=DH   | Au=RSA   | Enc=AESGCM(256) |
| ECDHE - ECDSA - CHACHA20 - POLY1305<br>POLY1305(256) Mac=AEAD | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=CHACHA20/   |
| ECDHE - RSA - CHACHA20 - POLY1305<br>POLY1305(256) Mac=AEAD   | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=CHACHA20/   |
| DHE - RSA - CHACHA20 - POLY1305<br>POLY1305(256) Mac=AEAD     | TLSv1.2 Kx=DH   | Au=RSA   | Enc=CHACHA20/   |
| ECDHE - ECDSA - AES128 - GCM - SHA256<br>Mac=AEAD             | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=AESGCM(128) |
| ECDHE - RSA - AES128 - GCM - SHA256<br>Mac=AEAD               | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=AESGCM(128) |
| DHE - RSA - AES128 - GCM - SHA256<br>Mac=AEAD                 | TLSv1.2 Kx=DH   | Au=RSA   | Enc=AESGCM(128) |
| ECDHE - ECDSA - AES256 - SHA384<br>Mac=SHA384                 | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=AES(256)    |
| ECDHE - RSA - AES256 - SHA384<br>Mac=SHA384                   | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=AES(256)    |
| DHE - RSA - AES256 - SHA256<br>Mac=SHA256                     | TLSv1.2 Kx=DH   | Au=RSA   | Enc=AES(256)    |
| ECDHE - ECDSA - AES128 - SHA256<br>Mac=SHA256                 | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=AES(128)    |
| ECDHE - RSA - AES128 - SHA256<br>Mac=SHA256                   | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=AES(128)    |
| DHE - RSA - AES128 - SHA256<br>Mac=SHA256                     | TLSv1.2 Kx=DH   | Au=RSA   | Enc=AES(128)    |
| ECDHE - ECDSA - AES256 - SHA<br>Mac=SHA1                      | TLSv1 Kx=ECDH   | Au=ECDSA | Enc=AES(256)    |
| ECDHE - RSA - AES256 - SHA<br>Mac=SHA1                        | TLSv1 Kx=ECDH   | Au=RSA   | Enc=AES(256)    |
| DHE - RSA - AES256 - SHA<br>Mac=SHA1                          | SSLv3 Kx=DH     | Au=RSA   | Enc=AES(256)    |
| ECDHE - ECDSA - AES128 - SHA<br>Mac=SHA1                      | TLSv1 Kx=ECDH   | Au=ECDSA | Enc=AES(128)    |
| ECDHE - RSA - AES128 - SHA<br>Mac=SHA1                        | TLSv1 Kx=ECDH   | Au=RSA   | Enc=AES(128)    |
| DHE - RSA - AES128 - SHA<br>Mac=SHA1                          | SSLv3 Kx=DH     | Au=RSA   | Enc=AES(128)    |
| AES256 - GCM - SHA384<br>Mac=AEAD                             | TLSv1.2 Kx=RSA  | Au=RSA   | Enc=AESGCM(256) |
| AES128 - GCM - SHA256<br>Mac=AEAD                             | TLSv1.2 Kx=RSA  | Au=RSA   | Enc=AESGCM(128) |
| AES256 - SHA256<br>Mac=SHA256                                 | TLSv1.2 Kx=RSA  | Au=RSA   | Enc=AES(256)    |
| AES128 - SHA256<br>Mac=SHA256                                 | TLSv1.2 Kx=RSA  | Au=RSA   | Enc=AES(128)    |
| AES256 - SHA<br>Mac=SHA1                                      | SSLv3 Kx=RSA    | Au=RSA   | Enc=AES(256)    |
| AES128 - SHA<br>Mac=SHA1                                      | SSLv3 Kx=RSA    | Au=RSA   | Enc=AES(128)    |

```
$ openssl ciphers -s -v 'DEFAULTS:-AES128:-SHA1'
```

|                                                         |                 |          |                 |
|---------------------------------------------------------|-----------------|----------|-----------------|
| TLS_AES_256_GCM_SHA384<br>Mac=AEAD                      | TLSv1.3 Kx=any  | Au=any   | Enc=AESGCM(256) |
| TLS_CHACHA20_POLY1305_SHA256<br>POLY1305(256) Mac=AEAD  | TLSv1.3 Kx=any  | Au=any   | Enc=CHACHA20/   |
| TLS_AES_128_GCM_SHA256<br>Mac=AEAD                      | TLSv1.3 Kx=any  | Au=any   | Enc=AESGCM(128) |
| ECDHE-ECDSA-AES256-GCM-SHA384<br>Mac=AEAD               | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=AESGCM(256) |
| ECDHE-RSA-AES256-GCM-SHA384<br>Mac=AEAD                 | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=AESGCM(256) |
| DHE-RSA-AES256-GCM-SHA384<br>Mac=AEAD                   | TLSv1.2 Kx=DH   | Au=RSA   | Enc=AESGCM(256) |
| ECDHE-ECDSA-CHACHA20-POLY1305<br>POLY1305(256) Mac=AEAD | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=CHACHA20/   |
| ECDHE-RSA-CHACHA20-POLY1305<br>POLY1305(256) Mac=AEAD   | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=CHACHA20/   |
| DHE-RSA-CHACHA20-POLY1305<br>POLY1305(256) Mac=AEAD     | TLSv1.2 Kx=DH   | Au=RSA   | Enc=CHACHA20/   |
| ECDHE-ECDSA-AES256-SHA384<br>Mac=SHA384                 | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=AES(256)    |
| ECDHE-RSA-AES256-SHA384<br>Mac=SHA384                   | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=AES(256)    |
| DHE-RSA-AES256-SHA256<br>Mac=SHA256                     | TLSv1.2 Kx=DH   | Au=RSA   | Enc=AES(256)    |
| AES256-GCM-SHA384<br>Mac=AEAD                           | TLSv1.2 Kx=RSA  | Au=RSA   | Enc=AESGCM(256) |
| AES256-SHA256<br>Mac=SHA256                             | TLSv1.2 Kx=RSA  | Au=RSA   | Enc=AES(256)    |

openssl ciphers manual page<sup>829</sup>

```
$ openssl ciphers -s -v -ciphersuites
```

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256 'DEFAULTS:-AES128:-SHA1'
```

<sup>829</sup> <https://docs.openssl.org/3.0/man1/openssl-ciphers/>

|                                                         |                 |          |                 |
|---------------------------------------------------------|-----------------|----------|-----------------|
| TLS_AES_256_GCM_SHA384<br>Mac=AEAD                      | TLSv1.3 Kx=any  | Au=any   | Enc=AESGCM(256) |
| TLS_CHACHA20_POLY1305_SHA256<br>POLY1305(256) Mac=AEAD  | TLSv1.3 Kx=any  | Au=any   | Enc=CHACHA20/   |
| ECDHE-ECDSA-AES256-GCM-SHA384<br>Mac=AEAD               | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=AESGCM(256) |
| ECDHE-RSA-AES256-GCM-SHA384<br>Mac=AEAD                 | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=AESGCM(256) |
| DHE-RSA-AES256-GCM-SHA384<br>Mac=AEAD                   | TLSv1.2 Kx=DH   | Au=RSA   | Enc=AESGCM(256) |
| ECDHE-ECDSA-CHACHA20-POLY1305<br>POLY1305(256) Mac=AEAD | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=CHACHA20/   |
| ECDHE-RSA-CHACHA20-POLY1305<br>POLY1305(256) Mac=AEAD   | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=CHACHA20/   |
| DHE-RSA-CHACHA20-POLY1305<br>POLY1305(256) Mac=AEAD     | TLSv1.2 Kx=DH   | Au=RSA   | Enc=CHACHA20/   |
| ECDHE-ECDSA-AES256-SHA384<br>Mac=SHA384                 | TLSv1.2 Kx=ECDH | Au=ECDSA | Enc=AES(256)    |
| ECDHE-RSA-AES256-SHA384<br>Mac=SHA384                   | TLSv1.2 Kx=ECDH | Au=RSA   | Enc=AES(256)    |
| DHE-RSA-AES256-SHA256<br>Mac=SHA256                     | TLSv1.2 Kx=DH   | Au=RSA   | Enc=AES(256)    |
| AES256-GCM-SHA384<br>Mac=AEAD                           | TLSv1.2 Kx=RSA  | Au=RSA   | Enc=AESGCM(256) |
| AES256-SHA256<br>Mac=SHA256                             | TLSv1.2 Kx=RSA  | Au=RSA   | Enc=AES(256)    |

## Config file examples

### Only use TLSv1.3

```
[system_default_sect]
CipherString = DEFAULT:@SECLEVEL=2
MinProtocol = TLSv1.3
```

```
$ curl https://j-server.lxd/stats
curl: (35) error:0A00042E:SSL routines::tlsv1 alert protocol version
```

```
$ wget https://j-server.lxd/stats

--2023-01-06 13:41:50-- https://j-server.lxd/stats
Resolving j-server.lxd (j-server.lxd)... 10.0.100.87
Connecting to j-server.lxd (j-server.lxd)|10.0.100.87|:443... connected.
OpenSSL: error:0A00042E:SSL routines::tlsv1 alert protocol version
Unable to establish SSL connection.
```

## Use only AES256 with TLSv1.3

```
[system_default_sect]
CipherString = DEFAULT:@SECLEVEL=2
CipherSuites = TLS_AES_256_GCM_SHA384
MinProtocol = TLSv1.3
```

*the Troubleshooting section*

```
$ sudo openssl s_server -cert j-server.pem -key j-server.key -port 443 -www
```

### Note:

Be sure to use another system for this server, or else it will be subject to the same `/etc/ssl/openssl.cnf` constraints you are testing on the client, and this can lead to very confusing results.

```
$ wget https://j-server.lxd/stats -O /dev/stdout -q | grep Cipher -w

New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Cipher : TLS_AES_256_GCM_SHA384
```

```
$ sudo openssl s_server -cert j-server.pem -key j-server.key -port 443 -www
-ciphersuites TLS_CHACHA20_POLY1305_SHA256
```

```
$ wget https://j-server.lxd/stats -O /dev/stdout
```

```
--2023-01-06 14:20:55-- https://j-server.lxd/stats
Resolving j-server.lxd (j-server.lxd)... 10.0.100.87
Connecting to j-server.lxd (j-server.lxd)|10.0.100.87|:443... connected.
OpenSSL: error:0A000410:SSL routines::ssl3 alert handshake failure
Unable to establish SSL connection.
```

## Drop AES128 entirely

```
[system_default_sect]
CipherString = DEFAULT:@SECLEVEL=2:!AES128
CipherSuites = TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
MinProtocol = TLSv1.2
```

```
$ sudo openssl s_server -cert j-server.pem -key j-server.key -port 443 -www
-tls1_2
```

```
$ wget https://j-server.lxd/stats -O /dev/stdout -q | grep Cipher -w

New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Cipher : ECDHE-RSA-AES256-GCM-SHA384
```

```
$ sudo openssl s_server -cert j-server.pem -key j-server.key -port 443 -www
-tls1_2 -cipher 'DEFAULT:!AES256:!CHACHA20'
```

```
$ wget https://j-server.lxd/stats -O /dev/stdout -q | grep Cipher -w

New, TLSv1.2, Cipher is ECDHE-RSA-AES128-GCM-SHA256
Cipher : ECDHE-RSA-AES128-GCM-SHA256
```



```
$ echo | openssl s_client -connect j-server.lxd:443 | grep -w -i cipher

4007F4F9D47F0000:error:0A000410:SSL routines:ssl3_read_bytes:ssl3 alert handshake
failure:../ssl/record/rec_layer_s3.c:1584:SSL alert number 40
New, (NONE), Cipher is (NONE)
```

```
$ echo | openssl s_client -connect j-server.lxd:443 --cipher DEFAULT:AES128
2>&1| grep -w -i cipher

New, TLSv1.2, Cipher is ECDHE-RSA-AES128-GCM-SHA256
Cipher : ECDHE-RSA-AES128-GCM-SHA256
```

## Further reading

- [OpenSSL home page](https://www.openssl.org/)<sup>830</sup>
- - [https://docs.openssl.org/3.0/man3/SSL\\_CTX\\_set\\_security\\_level/](https://docs.openssl.org/3.0/man3/SSL_CTX_set_security_level/)
  - <https://www.feistyduck.com/library/openssl-cookbook/online/openssl-command-line/understanding-security-levels.html>
- [SSL\\_CONF\\_cmd\(3\)](https://manpages.ubuntu.com/manpages/resolute/man3/SSL_CONF_cmd.3.html)<sup>831</sup>

## GnuTLS

*GnuTLS*

---

<sup>830</sup> <https://www.openssl.org/>

<sup>831</sup> [https://manpages.ubuntu.com/manpages/resolute/man3/SSL\\_CONF\\_cmd.3.html](https://manpages.ubuntu.com/manpages/resolute/man3/SSL_CONF_cmd.3.html)

## Structure of the config file

```
[global]
override-mode = blocklist

[priorities]
SYSTEM = NORMAL:-MD5

[overrides]
tls-disabled-mac = sha1
```

## The global section

- 
- 

```
[global]
override-mode = allowlist

[overrides]
secure-hash = sha256
enabled-curve = secp256r1
secure-sig = ecdsa-secp256r1-sha256
enabled-version = tls1.3
tls-enabled-cipher = aes-128-gcm
tls-enabled-mac = aead
tls-enabled-group = secp256r1
```

```
[global]
override-mode = blocklist

[overrides]
tls-disabled-cipher = aes-128-cbc
tls-disabled-cipher = aes-256-cbc
tls-disabled-mac = sha1
tls-disabled-group = group-ffdhe8192
```

disabling algorithms and protocols<sup>832</sup>

## Priority strings

defined in `debian/rules`<sup>833</sup>

```
NORMAL: -VERS-ALL:+VERS-TLS1.3:+VERS-TLS1.2:+VERS-DTLS1.2:%PROFILE_MEDIUM
```

defined in this table<sup>834</sup>

```
$ gnutls-cli --list --priority SECURE256
```

Cipher suites for SECURE256

|                                    |            |        |
|------------------------------------|------------|--------|
| TLS_AES_256_GCM_SHA384             | 0x13, 0x02 | TLS1.3 |
| TLS_CHACHA20_POLY1305_SHA256       | 0x13, 0x03 | TLS1.3 |
| TLS_ECDHE_ECDSA_AES_256_GCM_SHA384 | 0xc0, 0x2c | TLS1.2 |
| TLS_ECDHE_ECDSA_CHACHA20_POLY1305  | 0xcc, 0xa9 | TLS1.2 |
| TLS_ECDHE_ECDSA_AES_256_CCM        | 0xc0, 0xad | TLS1.2 |
| TLS_ECDHE_RSA_AES_256_GCM_SHA384   | 0xc0, 0x30 | TLS1.2 |
| TLS_ECDHE_RSA_CHACHA20_POLY1305    | 0xcc, 0xa8 | TLS1.2 |
| TLS_RSA_AES_256_GCM_SHA384         | 0x00, 0x9d | TLS1.2 |
| TLS_RSA_AES_256_CCM                | 0xc0, 0x9d | TLS1.2 |
| TLS_DHE_RSA_AES_256_GCM_SHA384     | 0x00, 0x9f | TLS1.2 |
| TLS_DHE_RSA_CHACHA20_POLY1305      | 0xcc, 0xaa | TLS1.2 |
| TLS_DHE_RSA_AES_256_CCM            | 0xc0, 0x9f | TLS1.2 |

Protocols: VERS-TLS1.3, VERS-TLS1.2, VERS-TLS1.1, VERS-TLS1.0, VERS-DTLS1.2, VERS-DTLS1.0

Ciphers: AES-256-GCM, CHACHA20-POLY1305, AES-256-CBC, AES-256-CCM

MACs: AEAD

Key Exchange Algorithms: ECDHE-ECDSA, ECDHE-RSA, RSA, DHE-RSA

Groups: GROUP-SECP384R1, GROUP-SECP521R1, GROUP-FFDHE8192

PK-signatures: SIGN-RSA-SHA384, SIGN-RSA-PSS-SHA384, SIGN-RSA-PSS-RSAE-SHA384, SIGN-ECDSA-SHA384, SIGN-ECDSA-SECP384R1-SHA384, SIGN-EdDSA-Ed448, SIGN-RSA-SHA512, SIGN-RSA-PSS-SHA512, SIGN-RSA-PSS-RSAE-SHA512, SIGN-ECDSA-SHA512, SIGN-ECDSA-SECP521R1-SHA512

<sup>832</sup> [https://www.gnutls.org/manual/html\\_node/Disabling-algorithms-and-protocols.html](https://www.gnutls.org/manual/html_node/Disabling-algorithms-and-protocols.html)

<sup>833</sup> <https://git.launchpad.net/ubuntu/+source/gnutls28/tree/debian/rules?h=ubuntu/jammy-devel#n38>

<sup>834</sup> [https://gnutls.org/manual/html\\_node/Priority-Strings.html#tab\\_003aprio\\_002dkeywords](https://gnutls.org/manual/html_node/Priority-Strings.html#tab_003aprio_002dkeywords)

```
$ gnutls-cli --list --priority SECURE256:-CHACHA20-POLY1305
```

Cipher suites for SECURE256:-CHACHA20-POLY1305

|                                    |            |        |
|------------------------------------|------------|--------|
| TLS_AES_256_GCM_SHA384             | 0x13, 0x02 | TLS1.3 |
| TLS_ECDHE_ECDSA_AES_256_GCM_SHA384 | 0xc0, 0x2c | TLS1.2 |
| TLS_ECDHE_ECDSA_AES_256_CCM        | 0xc0, 0xad | TLS1.2 |
| TLS_ECDHE_RSA_AES_256_GCM_SHA384   | 0xc0, 0x30 | TLS1.2 |
| TLS_RSA_AES_256_GCM_SHA384         | 0x00, 0x9d | TLS1.2 |
| TLS_RSA_AES_256_CCM                | 0xc0, 0x9d | TLS1.2 |
| TLS_DHE_RSA_AES_256_GCM_SHA384     | 0x00, 0x9f | TLS1.2 |
| TLS_DHE_RSA_AES_256_CCM            | 0xc0, 0x9f | TLS1.2 |

Protocols: VERS-TLS1.3, VERS-TLS1.2, VERS-TLS1.1, VERS-TLS1.0, VERS-DTLS1.2, VERS-DTLS1.0

Ciphers: AES-256-GCM, AES-256-CBC, AES-256-CCM

MACs: AEAD

Key Exchange Algorithms: ECDHE-ECDSA, ECDHE-RSA, RSA, DHE-RSA

Groups: GROUP-SECP384R1, GROUP-SECP521R1, GROUP-FFDHE8192

PK-signatures: SIGN-RSA-SHA384, SIGN-RSA-PSS-SHA384, SIGN-RSA-PSS-RSAE-SHA384, SIGN-ECDSA-SHA384, SIGN-ECDSA-SECP384R1-SHA384, SIGN-EdDSA-Ed448, SIGN-RSA-SHA512, SIGN-RSA-PSS-SHA512, SIGN-RSA-PSS-RSAE-SHA512, SIGN-ECDSA-SHA512, SIGN-ECDSA-SECP521R1-SHA512

### [priorities]

MYSET = SECURE256:-CHACHA20-POLY1305

```
$ gnutls-cli --list --priority @MYSET
```

## Verification profile (overrides)

### [overrides]

*# do not allow applications use the LOW or VERY-WEAK profiles.*

min-verification-profile = legacy

key sizes and security

parameters<sup>835</sup>

<sup>835</sup> [https://gnutls.org/manual/html\\_node/Selecting-cryptographic-key-sizes.html#tab\\_003akey\\_002dsizes](https://gnutls.org/manual/html_node/Selecting-cryptographic-key-sizes.html#tab_003akey_002dsizes)

## Practical examples

### Only use TLSv1.3

```
[global]
override-mode = blocklist

[overrides]
default-priority-string = NORMAL:-VERS-TLS-ALL:+VERS-TLS1.3
```

```
$ sudo openssl s_server -cert j-server.pem -key j-server.key -port 443
-no_tls1_3 -www
```

```
$ gnutls-cli j-server.lxd

Processed 125 CA certificate(s).
Resolving 'j-server.lxd:443'...
Connecting to '10.0.100.87:443'...
*** Fatal error: A TLS fatal alert has been received.
*** Received alert [70]: Error in protocol version
```

```
$ lftp -c "cat https://j-server.lxd/status"

cat: /status: Fatal error: gnutls_handshake: A TLS fatal alert has been received.
```

```
$ lftp -c "set ssl:priority NORMAL:+VERS-TLS-ALL; cat
https://j-server.lxd/status" | grep ^New

New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
```

```
[global]
override-mode = blocklist

[overrides]
disabled-version = tls1.1
disabled-version = tls1.2
disabled-version = tls1.0
```

```
$ lftp -c "set ssl:priority NORMAL:+VERS-TLS-ALL; cat
https://j-server.lxd/status" | grep ^New

cat: /status: Fatal error: gnutls_handshake: A TLS fatal alert has been received.
```

## Use only AES256 with TLSv1.3

### AES

```
[global]
override-mode = blocklist

[overrides]
disabled-version = tls1.1
disabled-version = tls1.2
disabled-version = tls1.0
tls-disabled-cipher = AES-128-GCM
```

```
$ sudo openssl s_server -cert j-server.pem -key j-server.key -port 443
-ciphersuites TLS_AES_128_GCM_SHA256 -www
```

```
$ gnutls-cli j-server.lxd

Processed 126 CA certificate(s).
Resolving 'j-server.lxd:443'...
Connecting to '10.0.100.87:443'...
*** Fatal error: A TLS fatal alert has been received.
*** Received alert [40]: Handshake failed
```

```
$ gnutls-cli --priority="NORMAL:+AES-128-GCM" j-server.lxd
```

```
Processed 126 CA certificate(s).
Resolving 'j-server.lxd:443'...
Connecting to '10.0.100.87:443'...
*** Fatal error: A TLS fatal alert has been received.
*** Received alert [40]: Handshake failed
```

## Further reading

- System-wide configuration<sup>836</sup>
- Priority strings<sup>837</sup>
- min-verification-profile values<sup>838</sup>
- Disabling algorithms and protocols<sup>839</sup>
- Invoking the `gnutls-cli` command line tool<sup>840</sup>

## Network Security Services (NSS)

## Config file

LP: #2016303<sup>841</sup>

- 
- 
- 

<sup>836</sup> [https://www.gnutls.org/manual/html\\_node/System\\_002dwide-configuration-of-the-library.html](https://www.gnutls.org/manual/html_node/System_002dwide-configuration-of-the-library.html)

<sup>837</sup> [https://gnutls.org/manual/html\\_node/Priority-Strings.html](https://gnutls.org/manual/html_node/Priority-Strings.html)

<sup>838</sup> [https://gnutls.org/manual/html\\_node/Selecting-cryptographic-key-sizes.html#tab\\_003akey\\_002dsizes](https://gnutls.org/manual/html_node/Selecting-cryptographic-key-sizes.html#tab_003akey_002dsizes)

<sup>839</sup> [https://www.gnutls.org/manual/html\\_node/Disabling-algorithms-and-protocols.html](https://www.gnutls.org/manual/html_node/Disabling-algorithms-and-protocols.html)

<sup>840</sup> [https://gnutls.org/manual/html\\_node/gnutls\\_002dcli-Invocation.html](https://gnutls.org/manual/html_node/gnutls_002dcli-Invocation.html)

<sup>841</sup> <https://bugs.launchpad.net/bugs/2016303>

- 

- 

- 

## Configuring the NSS policy module

```
library=
name=Policy
NSS=flags=policyOnly,moduleDB
config="disallow=<list> allow=<list> flags=<flags>"
```

- 

- 

- 

- 

- 

*DTLS*

- 

- 

- 

- 

-



- 
- 
- 
- 
- 

- 
- 

## Practical examples

### Bootstrapping the NSS database

```
$ sudo apt install libnss3-tools
```

```
$ mkdir -p ~/.pki/nssdb
```

```
$ certutil -d ~/.pki/nssdb -N
```

```
library=
name=NSS Internal PKCS #11 Module
parameters=configdir='/home/ubuntu/.pki/nssdb' certPrefix='' keyPrefix='' secmod='secmod.
db' flags= updatedir='' updateCertPrefix='' updateKeyPrefix='' updateid=''
updateTokenDescription=''
NSS=Flags=internal,critical trustOrder=75 cipherOrder=100 slotParams=(1={slotFlags=[ECC,
RSA,DSA,DH,RC2,RC4,DES,RANDOM,SHA1,MD5,MD2,SSL,TLS,AES,Camellia,SEED,SHA256,SHA512]
askpw=any timeout=30})
```

```
library=
name=NSS Internal PKCS #11 Module
parameters=configdir='/home/ubuntu/.pki/nssdb' certPrefix='' keyPrefix='' secmod='secmod.
db' flags= updatedir='' updateCertPrefix='' updateKeyPrefix='' updateid=''
updateTokenDescription=''
NSS=Flags=internal,critical trustOrder=75 cipherOrder=100 slotParams=(1={slotFlags=[ECC,
RSA,DSA,DH,RC2,RC4,DES,RANDOM,SHA1,MD5,MD2,SSL,TLS,AES,Camellia,SEED,SHA256,SHA512]
askpw=any timeout=30})
```

```
library=
name=Policy
NSS=flags=policyOnly,moduleDB
config="allow=tls-version-min=tlsl.3"
```

## Test setup

```
$ openssl req -new -x509 -days 30 -nodes -subj "/CN=localhost" -out
localhost.pem -keyout localhost.key
```

```
$ certutil -d ~/.pki/nssdb -A -a -i localhost.pem -t TCP -n localhost
```

- 
- 
- 
- 
- 

*certutil manpage<sup>842</sup>*

-  
-  
-

- 

```
$ openssl s_server -accept 4443 -cert localhost.pem -key localhost.key -www
```

## The tstclnt tool

```
$ tstclnt -d ~/.pki/nssdb -h localhost -p 4443
```

- 
- 
- 

```
$ echo "GET / HTTP/1.0" | tstclnt -d ~/.pki/nssdb -h localhost -p 4443 2>&1 |
grep ^New
New, TLSv1.3, Cipher is TLS_AES_128_GCM_SHA256
^C
```

---

<sup>842</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/certutil.1.html>

## Only use TLSv1.3

```
library=
name=Policy
NSS=flags=policyOnly,moduleDB
config="allow=tls-version-min=tls1.3"
```

```
$ openssl s_server -accept 4443 -cert localhost.pem -key localhost.key -www
-no_tls1_3
```

```
$ echo "GET / HTTP/1.0" | tstclnt -d ~/.pki/nssdb -h localhost -p 4443 2>&1 |
grep ^New > echo $?
1
```

```
$ echo "GET / HTTP/1.0" | tstclnt -d ~/.pki/nssdb -h localhost -p 4443 2>&1
tstclnt: write to SSL socket failed: SSL_ERROR_PROTOCOL_VERSION_ALERT: Peer reports
incompatible or unsupported protocol version.
```

```
$ openssl s_server -accept 4443 -cert localhost.pem -key localhost.key -www
```

```
$ echo "GET / HTTP/1.0" | tstclnt -d ~/.pki/nssdb -h localhost -p 4443 2>&1 |
grep ^New
New, TLSv1.3, Cipher is TLS_AES_128_GCM_SHA256
^C
```

## Use only AES256 with TLSv1.3

AES

```
library=
name=Policy
NSS=flags=policyOnly,moduleDB
config="disallow=aes128-gcm allow=tls-version-min=tls1.3"
```

```
$ echo "GET / HTTP/1.0" | tstclnt -d ~/.pki/nssdb -h localhost -p 4443 2>&1 |
grep ^New

New, TLSv1.3, Cipher is TLS_CHACHA20_POLY1305_SHA256
```

```
config="disallow=aes128-gcm:chacha20-poly1305 allow=tls-version-min=tls1.3"
```

```
$ echo "GET / HTTP/1.0" | tstclnt -d ~/.pki/nssdb -h localhost -p 4443 2>&1 |
grep ^New

New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
```

## Further reading

- In the source code<sup>843</sup>
- In the tests (policy)<sup>844</sup>
- In the tests (SSL policy)<sup>845</sup>

<sup>843</sup> <https://git.launchpad.net/ubuntu/+source/nss/tree/nss/lib/pk11wrap/pk11pars.c#n144>

<sup>844</sup> <https://git.launchpad.net/ubuntu/+source/nss/tree/nss/tests/policy>

<sup>845</sup> <https://git.launchpad.net/ubuntu/+source/nss/tree/nss/tests/ssl/sslpolicy.txt>

## Java cryptography configuration

*GUI*

```
$ sudo apt install default-jre-headless
```

### Config file

```
$ java -version 2>&1 | head -n 1
```

- 
- 

java.security configuration file<sup>846</sup>

```
$ java -Djava.security.properties=file://$HOME/java.security
```

<sup>846</sup> <https://git.launchpad.net/ubuntu/+source/openjdk-lts/tree/src/java.base/share/conf/security/java.security/#n520>

```
$ java -Djava.security.properties==file://$HOME/java.security
```

## Practical examples

### Test setup

```
$ openssl req -new -x509 -days 30 -nodes -subj "/CN=localhost" -out
localhost.pem -keyout localhost.key
```

```
$ sudo cp localhost.pem /usr/local/share/ca-certificates/localhost-test.crt
```

```
$ sudo update-ca-certificates
```

```
$ openssl s_server -accept 4443 -cert localhost.pem -key localhost.key | grep
^CIPHER
```

```
$ keytool -J-Djava.security.properties=file://$HOME/java.security -printcert
-sslserver localhost:4443 > /dev/null;echo $?
```

- 
- 

#### Note:

keytool is not really intended as a tool to test SSL/TLS connections, but being part of the Java packaging makes it convenient and it's enough for our purposes.

### Only use TLSv1.3

```
jdk.tls.disabledAlgorithms=TLSv1, TLSv1.1, TLSv1.2, SSLv3, SSLv2
```

```
$ keytool -J-Djava.security.properties=file://$HOME/java.security -printcert
-sslserver localhost:4443 > /dev/null;echo $?
```

```
0
```

```
$ openssl s_server -accept 4443 -key localhost.key -cert localhost.pem | grep
^CIPHER

CIPHER is TLS_AES_256_GCM_SHA384
```



```
$ keytool \ > -J-Djava.security.properties=file://$HOME/java.security \ >
-J-Djdk.tls.client.protocols=TLSv1.2 \ > -printcert -sslserver localhost:4443
```

```
keytool error: java.lang.Exception: No certificate from the SSL server
```

```
$ openssl s_server -accept 4443 -key localhost.key -cert localhost.pem
-no_tls1_3
```

```
Using default temp DH parameters
```

```
ACCEPT
```

```
ERROR
```

```
40676E75B37F0000:error:0A000102:SSL routines:tls_early_post_process_client_
```

```
hello:unsupported protocol:../ssl/statem/statem_srvr.c:1657:
```

```
shutting down SSL
```

```
CONNECTION CLOSED
```

```
$ keytool -J-Djava.security.properties=file://$HOME/java.security -printcert
-sslserver localhost:4443
```

```
keytool error: java.lang.Exception: No certificate from the SSL server
```

```
$ keytool -J-Djava.security.properties=file://$HOME/java.security -printcert
-sslserver localhost:4443 -v
```

```
keytool error: java.lang.Exception: No certificate from the SSL server
```

```
java.lang.Exception: No certificate from the SSL server
```

```
 at java.base/sun.security.tools.keytool.Main.doPrintCert(Main.java:2981)
```

```
 at java.base/sun.security.tools.keytool.Main.doCommands(Main.java:1292)
```

```
 at java.base/sun.security.tools.keytool.Main.run(Main.java:421)
```

```
 at java.base/sun.security.tools.keytool.Main.main(Main.java:414)
```

```
Caused by: javax.net.ssl.SSLHandshakeException: Received fatal alert: protocol_
version
```

```
...
```

## Prevent a specific cipher

Java Security Standard Algorithm Names<sup>847</sup>

```
jdk.tls.disabledAlgorithms=TLSv1, TLSv1.1, TLSv1.2, SSLv3, SSLv2, TLS_AES_256_GCM_SHA384
```

```
$ keytool -J-Djava.security.properties=file://$HOME/java.security -printcert
-sslserver localhost:4443 > /dev/null; echo $?
0
```

```
$ openssl s_server -accept 4443 -key localhost.key -cert localhost.pem | grep
^CIPHER
CIPHER is TLS_AES_128_GCM_SHA256
```

## Blocking cipher “elements”

```
jdk.tls.disabledAlgorithms=TLSv1, TLSv1.1, TLSv1.3, SSLv2, SSLv3
```

```
$ keytool -J-Djava.security.properties=file://$HOME/java.security -printcert
-sslserver localhost:4443 > /dev/null; echo $?
0
```

```
$ openssl s_server -accept 4443 -key localhost.key -cert localhost.pem | grep
^CIPHER
```

<sup>847</sup> <https://docs.oracle.com/en/java/javase/11/docs/specs/security/standard-names.html>

```
CIPHER is ECDHE-RSA-AES256-GCM-SHA384
```

```
jdk.tls.disabledAlgorithms=TLSv1, TLSv1.1, TLSv1.3, SSLv2, SSLv3, AES_256_GCM
```

```
$ openssl s_server -accept 4443 -key localhost.key -cert localhost.pem | grep ^CIPHER

CIPHER is ECDHE-RSA-CHACHA20-POLY1305
```

## Further reading

- [Java's Cryptographic Algorithms settings](#)<sup>848</sup>
- [Standard Algorithm Names](#)<sup>849</sup>
- [keytool upstream documentation](#)<sup>850</sup>
- [java.security file with comments](#)<sup>851</sup>

## BIND 9 DNSSEC cryptography selection

*DNS*

## DNSSEC validation

```
$ dig @127.0.0.1 isc.org +dnssec +multiline
```

<sup>848</sup> [https://www.java.com/en/configure\\_crypto.html](https://www.java.com/en/configure_crypto.html)

<sup>849</sup> <https://docs.oracle.com/en/java/javase/12/docs/specs/security/standard-names.html>

<sup>850</sup> <https://docs.oracle.com/en/java/javase/11/tools/keytool.html>

<sup>851</sup> <https://git.launchpad.net/ubuntu/+source/openjdk-lts/tree/src/java.base/share/conf/security/java.security?h=applied/ubuntu/jammy-devel#n520>

```
; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @127.0.0.1 isc.org +dnssec
+multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57669
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
; COOKIE: 71aa6b4e4ca6bb4b01000000643fee81edf0840b48d28d44 (good)
;; QUESTION SECTION:
;isc.org. IN A

;; ANSWER SECTION:
isc.org. 300 IN A 149.20.1.66
isc.org. 300 IN RRSIG A 13 2 300 (
 20230510161711 20230410161439 27566 isc.org.
 EUA5QPEjtVC0scPsvf1c/EIBKHRpS8ektiWiOqk6nb3t
 JhJAt9uCr3e0KNacc3WDU+wJzEvqDyJrLZoELqT/pQ==)

;; Query time: 740 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Wed Apr 19 13:37:05 UTC 2023
;; MSG SIZE rcvd: 183
```

```
$ dig @127.0.0.1 www.dnssec-failed.org +dnssec +multiline

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @127.0.0.1 www.dnssec-failed.org
+dnssec +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 56056
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
; COOKIE: 541f6c66a216acdb01000000643fef9ebb21307fee2ea0e3 (good)
;; QUESTION SECTION:
;www.dnssec-failed.org. IN A

;; Query time: 1156 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Wed Apr 19 13:41:50 UTC 2023
;; MSG SIZE rcvd: 78
```

- 
- 
- 

```
$ journalctl -u named.service -n 10
```

```
Apr 19 13:41:50 j named[3018]: validating dnssec-failed.org/DNSKEY: no valid
signature found (DS)
Apr 19 13:41:50 j named[3018]: no valid RRSIG resolving 'dnssec-failed.org/DNSKEY/IN
': 68.87.76.228#53
Apr 19 13:41:50 j named[3018]: broken trust chain resolving 'www.dnssec-failed.org/
A/IN': 68.87.85.132#53
(...)
```

```
$ dig @127.0.0.1 www.dnssec-failed.org +dnssec +multiline +cd
```

```
; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @127.0.0.1 www.dnssec-failed.org
+dnssec +multiline +cd
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 42703
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
; COOKIE: 3d6a4f4ff0014bdc01000000643ff01c3229ed7d798c5f8d (good)
;; QUESTION SECTION:
;www.dnssec-failed.org. IN A

;; ANSWER SECTION:
www.dnssec-failed.org. 7031 IN A 68.87.109.242
www.dnssec-failed.org. 7031 IN A 69.252.193.191
www.dnssec-failed.org. 7074 IN RRSIG A 5 3 7200 (
 20230505145108 20230418144608 44973 dnssec-failed.
org.
 R6/u+5Gv3rH93g08uNvz3sb9ErQNuVFKu6W5rtUleXF/
 vkqJXbNe8grMuIv6Y+CNEP6jRBu0jOBPncb5cXbfcmfO
 CoV0jpsLySxt4D1EUL4yByWm2ZAdXRrk6A8SaldIdDv8
 9t+FguTdQrZv9Si+afKrLyC7L/mltXMllq3stDI=)

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Wed Apr 19 13:43:56 UTC 2023
;; MSG SIZE rcvd: 287
```

## Restricting DNSSEC algorithms

- 
- 

```
disable-algorithms "." {
 RSAMD5;
 DSA;
};
disable-ds-digest "." {
 GOST;
};
```

DNSSEC Algorithm Numbers<sup>852</sup>

```
isc.org. 300 IN RRSIG A 13 2 300 (
```

```
disable-algorithms "." {
 RSAMD5;
 DSA;
 ECDSAP256SHA256;
};
```

```
$ sudo systemctl restart bind9.service
```

```
$ dig @127.0.0.1 isc.org +dnssec +multiline
```

<sup>852</sup> <https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml>

```
; <<>> DiG 9.18.1-1ubuntu1-Ubuntu <<>> @127.0.0.1 isc.org +dnssec +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43893
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
; COOKIE: 6527ce585598025d01000000643ff8fa02418ce38af13fa7 (good)
;; QUESTION SECTION:
;isc.org. IN A

;; ANSWER SECTION:
isc.org. 300 IN A 149.20.1.66
isc.org. 300 IN RRSIG A 13 2 300 (
 20230510161711 20230410161439 27566 isc.org.
 EUA5QPEjtVC0scPsvf1c/EIBKHRpS8ektiWiOqk6nb3t
 JhJAt9uCr3e0KNAcc3WUDU+wJzEvqDyJrLZoELqT/pQ==)

;; Query time: 292 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Wed Apr 19 14:21:46 UTC 2023
;; MSG SIZE rcvd: 183
```

Apr 19 14:23:01 j-bind9 named[2786]: validating isc.org/A: no valid signature found

### Note:

Remember now to remove the disabling of ECDSAP256SHA256 from `/etc/bind/named.conf` options and restart BIND 9. This change was just a quick test!

### Further reading

- [ISC DNSSEC Guide](#)<sup>853</sup>
- [DNSSEC troubleshooting section of the ISC DNSSEC guide](#)<sup>854</sup>
- [Standard algorithms used for DNSSEC](#)<sup>855</sup>

<sup>853</sup> <https://bind9.readthedocs.io/en/v9.18.14/dnssec-guide.html>

<sup>854</sup> <https://bind9.readthedocs.io/en/v9.18.14/dnssec-guide.html#basic-dnssec-troubleshooting>

<sup>855</sup> <https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml>

## OpenSSH crypto configuration

### Warning:

Be careful when restricting cryptographic algorithms in SSH, specially on the server side. You can inadvertently lock yourself out of a remote system!

### Algorithm configuration general rules

- 

- 

- 

- 

- 

-



**Note:**

The output of the `ssh -Q <name>` command will not take into consideration the configuration changes that may have been made. It cannot therefore be used to test the crypto configuration changes.

## Configuration settings

- 
- 
- 
- via a patch<sup>856</sup> provided  
*Generic Security Services Application Program Interface (GSSAPI)* *Generic Security Services Application Pro-*
- 
- 
- 
- 

```
$ sudo sshd -T | grep ciphers
```

<sup>856</sup> <https://git.launchpad.net/ubuntu/+source/openssh/tree/debian/patches/gssapi.patch>

```
ciphers chacha20-poly1305@openssh.com,aes128-gcm@openssh.com,aes256-gcm@openssh.com,
aes128-ctr,aes192-ctr,aes256-ctr
```

## OpenSSH examples

### Which cipher was used?

```
$ ssh -v <server> exit 2>&1 | grep "cipher:"
debug1: kex: server->client cipher: chacha20-poly1305@openssh.com MAC: <implicit>
compression: none
debug1: kex: client->server cipher: chacha20-poly1305@openssh.com MAC: <implicit>
compression: none
```

```
$ ssh -v -c aes128-ctr <server> exit 2>&1 | grep "cipher:"
debug1: kex: server->client cipher: aes128-ctr MAC: umac-64-etm@openssh.com
compression: none
debug1: kex: client->server cipher: aes128-ctr MAC: umac-64-etm@openssh.com
compression: none
```

### Remove AES 128 from server

*AES*

```
$ sudo sshd -T | grep ciphers
```

```
ciphers chacha20-poly1305@openssh.com,aes128-gcm@openssh.com,aes256-gcm@openssh.com,
aes128-ctr,aes192-ctr,aes256-ctr
```

```
Ciphers -aes128*
```

```
$ sudo sshd -T | grep ciphers
```

```
ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes192-ctr,aes256-ctr
```

```
$ sudo systemctl restart ssh.service
```

```
$ ssh -c aes128-ctr <server>
```

```
Unable to negotiate with 10.0.102.49 port 22: no matching cipher found. Their offer:
chacha20-poly1305@openssh.com,aes192-ctr,aes256-ctr,aes256-gcm@openssh.com
```

## Prioritize AES 256 on the client

```
$ ssh -c ^aes256-ctr -v <server> exit 2>&1 | grep "cipher:"
```

```
debug1: kex: server->client cipher: aes256-ctr MAC: umac-64-etm@openssh.com
compression: none
debug1: kex: client->server cipher: aes256-ctr MAC: umac-64-etm@openssh.com
compression: none
```

```
$ ssh -c ^aes256-ctr -v <server> exit 2>&1 | grep "cipher:"
```

```
debug1: kex: server->client cipher: chacha20-poly1305@openssh.com MAC: <implicit>
compression: none
debug1: kex: client->server cipher: chacha20-poly1305@openssh.com MAC: <implicit>
compression: none
```

## Further reading

- [OpenSSH upstream documentation index](https://www.openssh.org/manual.html)<sup>857</sup>
- [sshd\\_config\(5\)](https://manpages.ubuntu.com/manpages/resolute/man5/sshd_config.5.html)<sup>858</sup>
- [ssh\\_config\(5\)](https://manpages.ubuntu.com/manpages/resolute/man5/ssh_config.5.html)<sup>859</sup>

## Troubleshooting TLS/SSL

### Separate the client and server

## Tools

### OpenSSL server and client apps

```
$ openssl req -new -x509 -nodes -days 30 -out myserver.pem -keyout
myserver.key
```

*hostname*

```
$ openssl s_server -cert myserver.pem -key myserver.key
```

---

<sup>857</sup> <https://www.openssh.org/manual.html>

<sup>858</sup> [https://manpages.ubuntu.com/manpages/resolute/man5/sshd\\_config.5.html](https://manpages.ubuntu.com/manpages/resolute/man5/sshd_config.5.html)

<sup>859</sup> [https://manpages.ubuntu.com/manpages/resolute/man5/ssh\\_config.5.html](https://manpages.ubuntu.com/manpages/resolute/man5/ssh_config.5.html)

- 
- 
- 
- 
- 

```
$ echo | openssl s_client -connect server:port 2>&1 | grep '^New'
```

```
$ echo | openssl s_client -connect j-server.lxd:443 2>&1 | grep '^New'
```

New, TLSv1.3, Cipher is TLS\_AES\_256\_GCM\_SHA384

```
$ echo | openssl s_client -connect j-server.lxd:443 -no_tls1_3 2>&1 | grep '^New'
```

New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384

```
$ echo | openssl s_client -connect j-server.lxd:443 -no_tls1_3 2>&1 -cipher
DEFAULT:-AES256 | grep '^New'
```

New, TLSv1.2, Cipher is ECDHE-RSA-CHACHA20-POLY1305

## The `sslsan` tool

```
$ sslscan j-server.lxd
```

*sslscan(1)*<sup>860</sup>

---

<sup>860</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/sslscan.1.html>

## Further reading

- *OpenSSL s\_server*<sup>861</sup>
- *OpenSSL s\_client*<sup>862</sup>
- *sslscan(1)*<sup>863</sup>
- <https://badssl.com><sup>864</sup>

## About certificates

## Types of certificates

### Note:

Self-signed certificates should not be used in most production environments.

- 
- 

<sup>861</sup> [https://manpages.ubuntu.com/manpages/resolute/man1/openssl-s\\_server.1.html](https://manpages.ubuntu.com/manpages/resolute/man1/openssl-s_server.1.html)

<sup>862</sup> [https://manpages.ubuntu.com/manpages/resolute/man1/openssl-s\\_client.1.html](https://manpages.ubuntu.com/manpages/resolute/man1/openssl-s_client.1.html)

<sup>863</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/sslscan.1.html>

<sup>864</sup> <https://badssl.com/>

- 1.
- 2.
- 3.
- 4.
- 5.

### Generate a Certificate Signing Request (CSR)

#### Warning:

Running your secure service without a passphrase is convenient because you will not need to enter the passphrase every time you start your secure service. But it is insecure – a compromise of the key means a compromise of the server as well.

```
$ openssl genrsa -des3 -out server.key 2048
```

```
Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
```

```
$ openssl rsa -in server.key -out server.key.insecure > mv server.key
server.key.secure > mv server.key.insecure server.key
```

```
$ openssl req -new -key server.key -out server.csr
```

### Creating a self-signed certificate

```
$ openssl x509 -req -days 365 -in server.csr -signkey server.key -out
server.crt
```

#### Warning:

If your secure server is to be used in a production environment, you probably need a CA-signed certificate. It is not recommended to use self-signed certificates in production environments.



## Install the certificate

```
$ sudo cp server.crt /etc/ssl/certs > sudo cp server.key /etc/ssl/private
```

## Certification Authority

*Obtain TLS certificates*

## Further reading

- [HTTPS page](https://en.wikipedia.org/wiki/HTTPS)<sup>865</sup>
- [OpenSSL Home Page](https://www.openssl.org/)<sup>866</sup>
- 

## Virtual Private Network (VPN)

- *Introduction to WireGuard VPN*
- *OpenVPN clients*

## Introduction to WireGuard VPN

---

<sup>865</sup> <https://en.wikipedia.org/wiki/HTTPS>

<sup>866</sup> <https://www.openssl.org/>

`ip-route(8)`<sup>867</sup>      `ip-address(8)`<sup>868</sup>

## WireGuard concepts

`wg(8)`<sup>869</sup>      `wg-quick(8)`<sup>870</sup>

- 
- 
- 
- 
- 
- 

### Note:

Cryptography is not simple. When we say that, for example, a private key is used to decrypt or sign traffic, and a public key is used to encrypt or verify the authenticity of traffic, this is a simplification and is hiding a lot of important details. WireGuard has a detailed explanation of its protocols and cryptography handling [on its website](https://www.wireguard.com/protocol/)<sup>871</sup>.

<sup>871</sup> <https://www.wireguard.com/protocol/>

<sup>867</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/ip-route.8.html>

<sup>868</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/ip-address.8.html>

<sup>869</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/wg.8.html>

<sup>870</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/wg-quick.8.html>

`wg(8)`<sup>872</sup>

`wg-quick(8)`<sup>873</sup>

- 
- 

```
$ sudo wg-quick up wg0
```

#### [Interface]

PrivateKey = eJdSgoS7BZ/uWkuSREN+vhCJPPr3M3U1B3v1Su/amWk=

ListenPort = 51000

Address = 10.10.11.10/24

#### [Peer]

# office

PublicKey = xewmdxiLjgebpcItF1ouRo0ntrgFekquRJJZQ0+vsQVs=

Endpoint = wg.example.com:51000 # fake endpoint, just an example

AllowedIPs = 10.10.11.0/24, 10.10.10.0/24

- 
- 
- 
- 
- 

---

<sup>872</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/wg.8.html>

<sup>873</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/wg-quick.8.html>

- 

```
$ umask 077
```

```
$ wg genkey > wg0.key
```

```
$ wg pubkey < wg0.key > wg0.pub
```

```
$ sudo wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.10.11.10/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
[#] ip -4 route add 10.10.10.0/24 dev wg0
```

- 
- 
- 
- 
- 

```
$ ip route get 10.10.10.201

10.10.10.201 dev wg0 src 10.10.11.10 uid 1000
cache
```

*troubleshooting section*

## Peers

## Putting it all together

- 
- 
- 
- 

*ACL*

## Peer-to-site

- *About peer-to-site*
- *Set up peer-to-site “on router”*
- *Set up peer-to-site on an internal device*

## Site-to-site

- *Set up site-to-site*

## Default gateway

- *Using the VPN as the default gateway*

## Other common tasks, hints and tips

- *Common tasks*
- *Security tips*
- *Troubleshooting*

### Note:

Throughout this guide, we will sometimes mention a VPN “connection”. This is technically false, as WireGuard uses UDP and there is no persistent connection. The term is used just to facilitate understanding, and means that the peers in the examples know each other and have completed a handshake already.

## Further reading

- [WireGuard website](https://www.wireguard.com)<sup>874</sup>
- [WireGuard Quickstart](https://www.wireguard.com/quickstart/)<sup>875</sup>
- [wg\(8\)](https://manpages.ubuntu.com/manpages/resolute/man8/wg.8.html)<sup>876</sup>
- [wg-quick\(8\)](https://manpages.ubuntu.com/manpages/resolute/man8/wg-quick.8.html)<sup>877</sup>
- [Detailed explanation](https://www.wireguard.com/protocol/)<sup>878</sup>

---

<sup>874</sup> <https://www.wireguard.com>

<sup>875</sup> <https://www.wireguard.com/quickstart/>

<sup>876</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/wg.8.html>

<sup>877</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/wg-quick.8.html>

<sup>878</sup> <https://www.wireguard.com/protocol/>

## OpenVPN client implementations

*OpenVPN server*

### Linux Network-Manager GUI for OpenVPN

*GUI*

- 
- 
- 
- 
- 
- 
- 
- 

### OpenVPN with GUI for Mac OS X

Tunnelblick<sup>879</sup>

recommended by upstream<sup>880</sup>

has an alternative<sup>881</sup>

---

<sup>879</sup> <https://tunnelblick.net>

<sup>880</sup> <https://openvpn.net/vpn-server-resources/connecting-to-access-server-with-macos/>

<sup>881</sup> <https://openvpn.net/as-docs/tutorials/tutorial--connect-with-macos.html>

## OpenVPN with GUI for Win

OpenVPN Windows Installer<sup>882</sup>

updated guide by the upstream<sup>883</sup>

### Further reading

- [OpenVPN](#)<sup>884</sup>
- [OpenVPN hardening security guide](#)<sup>885</sup>
- [OpenVPN: Building and Integrating Virtual Private Networks](#)<sup>886</sup>

### See also

- [Security](#)

### *Our networking section*

- 
- 

## Networking

---

<sup>882</sup> <https://openvpn.net/community/>

<sup>883</sup> [https://community.openvpn.net/openvpn/wiki/Easy\\_Windows\\_Guide](https://community.openvpn.net/openvpn/wiki/Easy_Windows_Guide)

<sup>884</sup> <https://openvpn.net/>

<sup>885</sup> <https://openvpn.net/community-docs/hardening-openvpn-security-178571.html>

<sup>886</sup> <https://www.packtpub.com/en-us/product/openvpn-building-and-integrating-virtual-private-networks-9781847190680>



## Introduction

## Introduction to networking

## Networking key concepts

*Networking key concepts*

## Network configuration with Netplan

Netplan<sup>887</sup>  
YAML configuration file<sup>888</sup>

*Netplan*

*more about*

*configuring networks"*

## Network tools and services

## DHCP

*an explanatory guide*

*install isc-kea*

*DHCP server options*

---

<sup>887</sup> <https://netplan.io/>

<sup>888</sup> <https://netplan.readthedocs.io/en/stable/netplan-yaml/>

## Time synchronization

*time synchronization*

*read our how-to guide*

*serve NTP using chrony*

*how to*

## The DPDK library

Data Plane Development Kit (DPDK)<sup>889</sup>  
*about DPDK and its use in Ubuntu*

*with Open vSwitch*

*how to use DPDK*

## Other networking functionality

### Samba

*introduction to Samba*

## Networking key concepts

## The Transmission Control Protocol and Internet Protocol (TCP/IP)

*DARPA*

---

<sup>889</sup> <https://www.dpdk.org/>

## TCP/IP overview

- 

*Datagram*

- 

## TCP/IP configuration

*DHCP*

- 

- 

- 

- 

-

- 

**Tip:**

The IP address (in CIDR notation), gateway (default route), and nameserver addresses are typically specified via Netplan configuration files in `/etc/netplan/`. For more information, view the system manual page for `netplan`, using `man netplan.5` at the terminal prompt.

**IP routing**

*RIP*   *OSPF*

## About TCP and UDP

## Internet Control Messaging Protocol (ICMP)

Request For Comments (RFC) #792<sup>890</sup>

## About daemons

## Further reading

- [\*TCP\*<sup>891</sup> \*IP\*<sup>892</sup>](#)
- [TCP/IP Tutorial and Technical Overview<sup>893</sup>](#)
- [\*netplan\(5\)\*<sup>894</sup>](#)

---

<sup>890</sup> <https://www.rfc-editor.org/rfc/rfc792>

<sup>891</sup> <https://manpages.ubuntu.com/manpages/resolute/man7/tcp.7.html>

<sup>892</sup> <https://manpages.ubuntu.com/manpages/resolute/man7/ip.7.html>

<sup>893</sup> <https://www.redbooks.ibm.com/abstracts/gg243376.html>

<sup>894</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/netplan.5.html>

## Configuration

*network configuration*

*About Netplan*

## About Netplan

Netplan<sup>895</sup>

YAML configuration file<sup>896</sup>

introduction to Netplan<sup>897</sup>

*configuring networks*

list of how-to guides<sup>898</sup>

## Configuring networks

Netplan<sup>899</sup>

YAML configuration file<sup>900</sup>

the Netplan documentation<sup>901</sup>

---

<sup>895</sup> <https://netplan.io/>

<sup>896</sup> <https://netplan.readthedocs.io/en/stable/netplan-yaml/>

<sup>897</sup> <https://netplan.readthedocs.io/en/stable/structure-id/>

<sup>898</sup> <https://netplan.readthedocs.io/en/stable/howto/>

<sup>899</sup> <https://netplan.io/>

<sup>900</sup> <https://netplan.readthedocs.io/en/stable/netplan-yaml/>

<sup>901</sup> <https://netplan.readthedocs.io/en/stable/>

## Ethernet interfaces

### Identify Ethernet interfaces

```
$ ip addr

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
 inet 127.0.0.1/8 scope host lo
 valid_lft forever preferred_lft forever
 inet6 ::1/128 scope host
 valid_lft forever preferred_lft forever
2: enp0s25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
 link/ether 00:16:3e:e2:52:42 brd ff:ff:ff:ff:ff:ff link-netnsid 0
 inet 10.102.66.200/24 brd 10.102.66.255 scope global dynamic eth0
 valid_lft 3257sec preferred_lft 3257sec
 inet6 fe80::216:3eff:fee2:5242/64 scope link
 valid_lft forever preferred_lft forever
```

```
$ sudo netplan status -a
```

```
Online state: online
DNS Addresses: 127.0.0.53 (stub)
DNS Search: internal

[] 1: lo ethernet UNKNOWN/UP (unmanaged)
 MAC Address: 00:00:00:00:00:00
 Addresses: 127.0.0.1/8
 ::1/128

[] 2: enp0s25 ethernet UP (networkd: enp0s25)
 MAC Address: 00:16:3e:e2:52:42 (ASIX Electronics Corp.)
 Addresses: 10.102.66.200/24 (dynamic, dhcp)
 fe80::216:3eff:fee2:5242/64 (link)
 DNS Addresses: 10.102.66.1
 fd00::cece:1eff:fe3d:c737
 DNS Search: internal
 Routes: default via 10.102.66.1 from 10.102.66.200 metric 100 (dhcp)
 10.102.66.0/24 from 10.102.66.200 metric 100 (link)
 2001:16b8:bd99:3d00::/56 via fe80::cece:1eff:fe3d:c737 metric 100
(ra)
 default via fe80::cece:1eff:fe3d:c737 metric 100 (ra)
```

```
$ sudo lshw -class network

*-network
 description: Ethernet interface
 product: MT26448 [ConnectX EN 10GigE, PCIe 2.0 5GT/s]
 vendor: Mellanox Technologies
physical id: 0
bus info: pci@0004:01:00.0
logical name: eth4
version: b0
serial: e4:1d:2d:67:83:56
slot: U78CB.001.WZS09KB-P1-C6-T1
size: 10Gbit/s
capacity: 10Gbit/s
width: 64 bits
clock: 33MHz
capabilities: pm vpd msix pciexpress bus_master cap_list ethernet physical fibre 10000bt-fd
configuration: autonegotiation=off broadcast=yes driver=mlx4_en
driverversion=4.0-0 duplex=full firmware=2.9.1326 ip=192.168.1.1 latency=0 link=yes
multicast=yes port=fibre speed=10Gbit/s
resources: iomemory:24000-23fff irq:481 memory:3fe200000000-3fe2000fffff
memory:240000000000-240007ffffff
```



## Ethernet Interface logical names

```
network:
 version: 2
 renderer: networkd
 ethernets
 eth_lan0:
 dhcp4: true
 match
 macaddress: 00:11:22:33:44:55
 set-name: eth_lan0
```

## Ethernet Interface settings

```
$ sudo ethtool eth4

Settings for eth4:
 Supported ports: [FIBRE]
 Supported link modes: 10000baseT/Full
 Supported pause frame use: No
 Supports auto-negotiation: No
 Supported FEC modes: Not reported
 Advertised link modes: 10000baseT/Full
 Advertised pause frame use: No
 Advertised auto-negotiation: No
 Advertised FEC modes: Not reported
 Speed: 10000Mb/s
 Duplex: Full
 Port: FIBRE
 PHYAD: 0
 Transceiver: internal
 Auto-negotiation: off
 Supports Wake-on: d
 Wake-on: d
 Current message level: 0x00000014 (20)
 link ifdown
 Link detected: yes
```

## IP addressing

### Temporary IP address assignment

*GNU*

```
$ sudo ip addr add 10.102.66.200/24 dev enp0s25
```

```
$ ip link set dev enp0s25 up
```

```
$ ip link set dev enp0s25 down
```

```
$ ip address show dev enp0s25

10: enp0s25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
 link/ether 00:16:3e:e2:52:42 brd ff:ff:ff:ff:ff:ff link-netnsid 0
 inet 10.102.66.200/24 brd 10.102.66.255 scope global dynamic eth0
 valid_lft 2857sec preferred_lft 2857sec
 inet6 fe80::216:3eff:fee2:5242/64 scope link
 valid_lft forever preferred_lft forever6
```

```
$ sudo ip route add default via 10.102.66.1
```

```
$ ip route show

default via 10.102.66.1 dev eth0 proto dhcp src 10.102.66.200 metric 100
10.102.66.0/24 dev eth0 proto kernel scope link src 10.102.66.200
10.102.66.1 dev eth0 proto dhcp scope link src 10.102.66.200 metric 100
```

## DNS

```
$ sudo resolvectl dns enp0s25 8.8.8.8 8.8.4.4
```

```
$ ip addr flush eth0
```

### Note:

Flushing the IP configuration using the `ip` command does not clear the contents of `systemd-resolved`. You must remove or modify those entries manually, e.g by calling `sudo resolvectl revert enp0s25`. Or re-boot, which should also cause the symlink `/etc/resolv.conf -> /run/systemd/resolve/stub-resolv.conf`, to be restored.

## Dynamic IP address assignment (DHCP client)

```
network:
 version: 2
 renderer: networkd
 ethernets:
 enp3s0:
 dhcp4: true
```

```
$ sudo netplan apply
```

## Static IP address assignment

```
network:
 version: 2
 renderer: networkd
 ethernets:
```

(continues on next page)

(continued from previous page)

```
eth0:
 addresses:
 - 10.10.10.2/24
 routes:
 - to: default
 via: 10.10.10.1
 nameservers
 search [mydomain, otherdomain]
 addresses: [10.10.10.1, 1.1.1.1]
```

```
$ sudo netplan apply
```

#### Note:

netplan in Ubuntu Bionic 18.04 LTS doesn't understand the "to: default" syntax to specify a default route, and should use the older gateway4: 10.10.10.1 key instead of the whole routes: block.

```
$ ip address show lo

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
 inet 127.0.0.1/8 scope host lo
 valid_lft forever preferred_lft forever
 inet6 ::1/128 scope host
 valid_lft forever preferred_lft forever
```

## Adding a multiple IP addresses to one interface

- 
-

## Multiple static IP addresses

```
network:
 version: 2
 ethernets:
 eno1:
 addresses:
 - 192.168.2.99/24
 - 192.168.2.100/24:
 label: eno1:1
 - 192.168.2.101/24:
 label: eno1:some-label
```

## Dynamic IP addresses

```
network:
 version: 2
 ethernets:
 eno1:
 addresses:
 - 192.168.2.100/24:
 label: eno1:some-label
 dhcp4: true
```

### Note:

A single interface can only have one address assigned by DHCP. To have multiple dynamic addresses, configure multiple interfaces. Do not add DHCP if it is configured on the same interface in another netplan file, such as one written by `cloud-init`.

## Apply and verify the IP addresses

```
$ sudo netplan apply
```

```
$ netplan status #displays the current network state
```

```
$ ip addr #shows all network interfaces
```

```
$ ip addr show <interface_name> #shows single interface
```

```
$ hostname -i #shows only the available IP addresses on the host
```

```
$ ping 192.168.0.100 #configured IP addresses
```

```
$ ping 192.168.0.101
```

## Name resolution

*hostnames*

## DNS client configuration

```
/etc/resolv.conf -> /run/systemd/resolve/stub-resolv.conf
```

```
network:
 version: 2
 renderer: networkd
 ethernets:
 enp0s25:
 addresses:
 - 192.168.0.100/24
 routes:
 - to: default
 via: 192.168.0.1
 nameservers
```

(continues on next page)

(continued from previous page)

```
search [mydomain, otherdomain]
addresses: [1.1.1.1, 8.8.8.8, 4.4.4.4]
```

```
network:
 version: 2
 renderer: networkd
 ethernet:
 enp0s25:
 addresses:
 - 192.168.0.100/24
 routes:
 - to: default
 via: 192.168.0.1
 nameservers:
 search [example.com, sales.example.com, dev.example.com]
 addresses: [1.1.1.1, 8.8.8.8, 4.4.4.4]
```

### *Fully Qualified Domain Name (FQDN)*

- 1.
- 2.
- 3.

## Static hostnames

```
127.0.0.1 localhost
127.0.1.1 ubuntu-server
10.0.0.11 server1 server1.example.com vpn
10.0.0.12 server2 server2.example.com mail
```

(continues on next page)

(continued from previous page)

```
10.0.0.13 server3 server3.example.com www
10.0.0.14 server4 server4.example.com file
```

#### Note:

In this example, notice that each of the servers were given aliases in addition to their proper names and FQDN's. *Server1* has been mapped to the name *vpn*, *server2* is referred to as *mail*, *server3* as *www*, and *server4* as *file*.

## Name Service Switch (NSS) configuration

```
hosts: files mdns4_minimal [NOTFOUND=return] dns mdns4
```

- 
- 
- 
- 
- 

```
hosts: files dns [NOTFOUND=return] mdns4_minimal mdns4
```

## Bridging multiple interfaces

```
network:
 version: 2
```

(continues on next page)



(continued from previous page)

```
renderer: networkd
ethernets
 enp3s0
 dhcp4: no
bridges:
 br0:
 dhcp4: yes
interfaces
- enp3s0
```

```
$ sudo netplan apply
```

## networkd-dispatcher for hook scripts

Netplan configuration<sup>902</sup>

*dispatcher(8)*<sup>903</sup>

*networkd-*

### Note:

If you are on Desktop (not Ubuntu Server) the network is driven by Network Manager - in that case you need [NM Dispatcher scripts](#)<sup>904</sup> instead.

<sup>904</sup> <https://networkmanager.dev/docs/api/latest/>

Netplan FAQ has a great table<sup>905</sup>

Netplan FAQ also has an example<sup>906</sup>

<sup>902</sup> <https://netplan.readthedocs.io/en/stable/netplan-yaml/>

<sup>903</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/networkd-dispatcher.8.html>

<sup>905</sup> <https://netplan.io/faq/>

<sup>906</sup> <https://netplan.io/faq/>

## Further reading

- [Ubuntu Wiki Network page](#)<sup>907</sup>
- [Netplan website](#)<sup>908</sup> [examples](#)<sup>909</sup>
- [manual page](#)<sup>910</sup>
- [systemd-resolved\(8\)](#)<sup>911</sup>
- [netplan configuration reference](#)<sup>912</sup>

## Network tools

- [About DPDK](#)

## About DPDK

*EAL*

## Prerequisites

DPDK 17.08<sup>913</sup>

supported NICs<sup>914</sup>

---

<sup>907</sup> <https://help.ubuntu.com/community/Network>

<sup>908</sup> <https://netplan.io>

<sup>909</sup> <https://netplan.readthedocs.io/en/stable/examples/>

<sup>910</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/netplan.5.html>

<sup>911</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/systemd-resolved.8.html>

<sup>912</sup> <https://netplan.readthedocs.io/en/stable/netplan-yaml/#properties-for-device-type-bridges>

<sup>913</sup> <https://git.dpdk.org/dpdk/commit/?id=f27769f796a0639368117ce22fb124b6030dbf73>

<sup>914</sup> <http://core.dpdk.org/supported/>

many more “in-universe”<sup>915</sup>

## Unassign the default kernel drivers

```
iommu=pt intel_iommu=on
```

*AMD*

```
amd_iommu=pt
```

*firmware*

### Note:

VirtIO is special. DPDK can directly work on these devices without `vfio_pci/uio_pci_generic`. However, to avoid issues that might arise from the kernel and DPDK managing the device, you still need to unassign the kernel driver.

```
dpdk_nic_bind.py --help
```

## Usage

```
dpdk-devbind.py [options] DEVICE1 DEVICE2
```

where DEVICE1, DEVICE2 etc, are specified via PCI `"domain:bus:slot.func"` syntax or `"bus:slot.func"` syntax. For devices bound to Linux kernel drivers, they may also be referred to by Linux interface name e.g. `eth0`, `eth1`, `em0`, `em1`, etc.

Options:  
`--help`, `--usage`:

(continues on next page)

<sup>915</sup> <https://documentation.ubuntu.com/project/how-ubuntu-is-made/concepts/package-archive/#components>

```
Display usage information and quit

-s, --status:
 Print the current status of all known network, crypto, event
 and mempool devices.
 For each device, it displays the PCI domain, bus, slot and function,
 along with a text description of the device. Depending upon whether the
 device is being used by a kernel driver, the igb_uio driver, or no
 driver, other relevant information will be displayed:
 * the Linux interface name e.g. if=eth0
 * the driver being used e.g. drv=igb_uio
 * any suitable drivers not currently using that device
 e.g. unused=igb_uio
 NOTE: if this flag is passed along with a bind/unbind option, the
 status display will always occur after the other operations have taken
 place.

--status-dev:
 Print the status of given device group. Supported device groups are:
 "net", "crypto", "event", "mempool" and "compress"

-b driver, --bind=driver:
 Select the driver to use or "none" to unbind the device

-u, --unbind:
 Unbind a device (Equivalent to "-b none")

--force:
 By default, network devices which are used by Linux - as indicated by
 having routes in the routing table - cannot be modified. Using the
 --force flag overrides this behavior, allowing active links to be
 forcibly unbound.
 WARNING: This can lead to loss of network connection and should be used
 with caution.

Examples:

To display current device status:
 dpdk-devbind.py --status

To display current network device status:
 dpdk-devbind.py --status-dev net

To bind eth1 from the current driver and move to use igb_uio
 dpdk-devbind.py --bind=igb_uio eth1

To unbind 0000:01:00.0 from using any driver
 dpdk-devbind.py -u 0000:01:00.0

To bind 0000:02:00.0 and 0000:02:00.1 to the ixgbe kernel driver
 dpdk-devbind.py -b ixgbe 02:00.0 02:00.1
```

## DPDK device configuration

```
<bus> Currently only "pci" is supported
<id> Device ID on the specified bus
<driver> Driver to bind against (vfio-pci or uio_pci_generic)
#
Be aware that the two DPDK compatible drivers uio_pci_generic and vfio-pci are
part of linux-image-extra-<VERSION> package.
This package is not always installed by default - for example in cloud-images.
So please install it in case you run into missing module issues.
#
<bus> <id> <driver>
pci 0000:04:00.0 uio_pci_generic
pci 0000:04:00.1 vfio-pci
```

```
Network devices using DPDK-compatible driver
=====
0000:04:00.0 'Ethernet Controller 10-Gigabit X540-AT2' drv=uio_pci_generic unused=ixgbe

Network devices using kernel driver
=====
0000:02:00.0 'NetXtreme BCM5719 Gigabit Ethernet PCIe' if=eth0 drv=tg3 unused=uio_pci_
generic *Active*
0000:02:00.1 'NetXtreme BCM5719 Gigabit Ethernet PCIe' if=eth1 drv=tg3 unused=uio_pci_
generic
0000:02:00.2 'NetXtreme BCM5719 Gigabit Ethernet PCIe' if=eth2 drv=tg3 unused=uio_pci_
generic
0000:02:00.3 'NetXtreme BCM5719 Gigabit Ethernet PCIe' if=eth3 drv=tg3 unused=uio_pci_
generic
0000:04:00.1 'Ethernet Controller 10-Gigabit X540-AT2' if=eth5 drv=ixgbe unused=uio_pci_
generic

Other network devices
=====
<none>
```

## DPDK hugepage configuration

```
NR_2M_PAGES=1024
NR_1G_PAGES=4
```

## Compile DPDK applications

valid pkg-config file<sup>916</sup>

```
$ sudo apt-get install dpdk-dev libdpdk-dev
```

```
$ gcc testdpdkprog.c $(pkg-config --libs --cflags libdpdk) -o testdpdkprog
```

Open vSwitch build system<sup>917</sup>

*fallbacks*

<sup>916</sup> <https://people.freedesktop.org/~dbn/pkg-config-guide.html>

<sup>917</sup> <https://github.com/openvswitch/ovs/blob/master/acinclude.m4#L283>

```
$ sudo apt-get install build-dep dpdk
```

## DPDK in KVM guests

```
<cpu mode='host-passthrough'>
```

```
<driver name="vhost" queues="4"/>
```

## Use DPDK

*Open vSwitch*

*DPDK*

## Further reading

- DPDK documentation<sup>918</sup>
- Release Notes matching the version packages in Ubuntu 16.04<sup>919</sup>
- Linux DPDK user getting started<sup>920</sup>
- EAL command-line options<sup>921</sup>
- DPDK API documentation<sup>922</sup>
- Open vSwitch DPDK installation<sup>923</sup>
- Wikipedia definition of DPDK<sup>924</sup>

---

<sup>918</sup> <http://core.dpdk.org/doc/>

<sup>919</sup> [https://doc.dpdk.org/guides/rel\\_notes/release\\_2\\_2.html](https://doc.dpdk.org/guides/rel_notes/release_2_2.html)

<sup>920</sup> [https://doc.dpdk.org/guides/linux\\_gsg/index.html](https://doc.dpdk.org/guides/linux_gsg/index.html)

<sup>921</sup> [https://doc.dpdk.org/guides/testpmd\\_app\\_ug/run\\_app.html](https://doc.dpdk.org/guides/testpmd_app_ug/run_app.html)

<sup>922</sup> <https://doc.dpdk.org/api/>

<sup>923</sup> <https://github.com/openvswitch/ovs/blob/branch-2.5/INSTALL.DPDK.md>

<sup>924</sup> [https://en.wikipedia.org/wiki/Data\\_Plane\\_Development\\_Kit](https://en.wikipedia.org/wiki/Data_Plane_Development_Kit)

## DHCP

- *About DHCP*

### About DHCP

- 
- 
- *DNS*
- *Hostname*
- 
- 
- 

### DHCP configuration

#### Manual allocation (MAC address)



## Dynamic allocation (address pool)

## Automatic allocation

## Available servers

- `Kea`<sup>925</sup>
- *how to install and configure isc-kea*
- `dnsmasq`<sup>926</sup>

### Note:

The `isc-dhcp-server` package is deprecated and unsupported since Ubuntu 24.04 LTS. Please use `isc-kea` or `dnsmasq` instead.

*information on how to configure it*

---

<sup>925</sup> <https://www.isc.org/kea/>

<sup>926</sup> <https://thekelleys.org.uk/dnsmasq/doc.html>

## Further reading

- [isc-dhcp-server Ubuntu Wiki](#)<sup>927</sup>
- [dhcpcd.conf\(5\)](#)<sup>928</sup>
- [ISC dhcp-server](#)<sup>929</sup>
- [ISC Kea Documentation](#)<sup>930</sup>

## Time synchronisation

- [About time synchronisation](#)

## About time synchronization

*synchronize time with chrony*

*use chrony to serve NTP*

*timedatectl and timesyncd*

### Note:

Chrony is replacing systemd-timesyncd as the default in Ubuntu 25.10 and ntpdate/ntpd, which was the default before Ubuntu 18.04 LTS.

On upgraded systems (from Ubuntu 25.04 or below) systemd-timesyncd might still be the active time-daemon and thus render the new chrony service disabled. Similarly, while the use of ntpd is no longer recommended, it might still be installed to retain any previous behavior/config on upgrades (up to Ubuntu 22.04 LTS).

<sup>927</sup> <https://help.ubuntu.com/community/isc-dhcp-server>

<sup>928</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/dhcpd.conf.5.html>

<sup>929</sup> <https://www.isc.org/dhcp/>

<sup>930</sup> <https://kb.isc.org/docs/kea-administrator-reference-manual>

## How time synchronization works

### About chrony

### About `timedatectl`

- 
- 

### Further reading

- [ntp.org](http://www.ntp.org/): home of the Network Time Protocol project<sup>931</sup>
- [pool.ntp.org](https://www.ntppool.org/en/): project of virtual cluster of timeservers<sup>932</sup>
- `timedatectl(1)`<sup>933</sup>
- `systemd-timesyncd.service(8)`<sup>934</sup>
- [chrony FAQ](https://chrony-project.org/faq.html)<sup>935</sup>
- [Feeding chrony from GPSD](https://gpsd.gitlab.io/gpsd/time-service-howto.html#_feeding_chrony_from_gpsd)<sup>936</sup>
- [Ubuntu Time wiki page](https://help.ubuntu.com/community/UbuntuTime)<sup>937</sup>

---

<sup>931</sup> <http://www.ntp.org/>

<sup>932</sup> <https://www.ntppool.org/en/>

<sup>933</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/timedatectl.1.html>

<sup>934</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/systemd-timesyncd.service.8.html>

<sup>935</sup> <https://chrony-project.org/faq.html>

<sup>936</sup> [https://gpsd.gitlab.io/gpsd/time-service-howto.html#\\_feeding\\_chrony\\_from\\_gpsd](https://gpsd.gitlab.io/gpsd/time-service-howto.html#_feeding_chrony_from_gpsd)

<sup>937</sup> <https://help.ubuntu.com/community/UbuntuTime>

## Network shares

- *Introduction to Samba*
- *Active Directory integration*

## Introduction to Samba

Samba<sup>938</sup>

## Samba functionality

## File and printer sharing services

- *configured as a file server*
- *configured as a print server*

---

<sup>938</sup> <https://www.samba.org/>

## Directory services

- [How to set up Ubuntu Server as a member of an Active Directory domain](#)
- [How to set up Ubuntu Server as a Windows NT4-style domain controller](#)
- [How to set up Ubuntu Server as an LDAP in Windows NT4 mode](#)

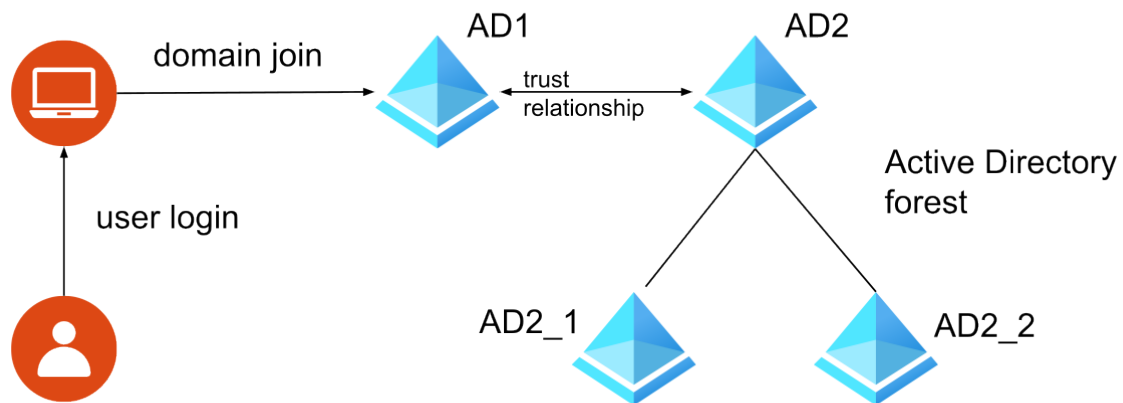
## Authentication and access

- [How to set up Ubuntu Server as a Samba server controlling access to shared directories](#)
- [How to set up Ubuntu Server as a Samba client set up a profile for Samba](#)
- [How to set up Ubuntu Server as a Samba client how to set up Common Internet File System \(CIFS\) shares](#)

## Active Directory integration

## Introduction

### Introduction to Active Directory integration



- 
- 
- 
- 
- 
- 

*group policy objects*

- *Choosing an integration method*
- *Security identifiers*
- *Identity mapping backends*
- *The rid idmap backend*

- *The autorid idmap backend*

## Further reading

- - Security Identifiers (SIDs)<sup>939</sup>
  - Active Directory Domain Services Overview<sup>940</sup>
- - Choosing an identity mapping backend<sup>941</sup>
  - rid identity mapping backend<sup>942</sup>
  - autorid identity mapping backend<sup>943</sup>
- - `idmap_tdb(8)`<sup>944</sup>
  - `idmap_rid(8)`<sup>945</sup>
  - `idmap_autorid(8)`<sup>946</sup>
  - `wbinfo(1)`<sup>947</sup>

## Configuration

### Choosing an integration method

- 

---

<sup>939</sup> <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-identifiers>

<sup>940</sup> <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>

<sup>941</sup> [https://wiki.samba.org/index.php/Setting\\_up\\_Samba\\_as\\_a\\_Domain\\_Member#Choosing\\_an\\_idmap\\_backend](https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Domain_Member#Choosing_an_idmap_backend)

<sup>942</sup> [https://wiki.samba.org/index.php/Idmap\\_config\\_rid](https://wiki.samba.org/index.php/Idmap_config_rid)

<sup>943</sup> [https://wiki.samba.org/index.php/Idmap\\_config\\_autorid](https://wiki.samba.org/index.php/Idmap_config_autorid)

<sup>944</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/idmap\\_tdb.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/idmap_tdb.8.html)

<sup>945</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/idmap\\_rid.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/idmap_rid.8.html)

<sup>946</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/idmap\\_autorid.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/idmap_autorid.8.html)

<sup>947</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/wbinfo.1.html>

- 
- 

| Requirement/Method       | SSSD | idmap_rid | idmap_autorid |
|--------------------------|------|-----------|---------------|
| Deterministic ID         | Y    | Y         | N             |
| Undetermined ID          | Y    | Y         | Y             |
| Member server            | N    | Y         | Y             |
| Not a member server      | Y    | Y         | Y             |
| AD multi-domain (forest) | N    | N(*)      | Y             |
| AD single domain         | Y    | Y         | Y             |

### *Joining a forest with the rid backend*

- *Security identifiers*
- *Identity mapping backends*
- *The rid idmap backend*
- *The autorid idmap backend*

## Security identifiers (SIDs)

### Note:

For a full explanation of what an SID is, and how they are structured and allocated, please refer to [Understand Security Identifiers](https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-identifiers)<sup>948</sup>.

<sup>948</sup> <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-identifiers>



S-1-5-<Domain-Identifier>--<Relative-Identifier>

- 
- 
- 
- 

- *Identity mapping backends*
- *The rid idmap backend*
- *The autorid idmap backend*

## Identity Mapping (idmap) backends

- *idmap\_ad(8)*<sup>949</sup>
- *idmap\_rid(8)*<sup>950</sup>

---

<sup>949</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/idmap\\_ad.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/idmap_ad.8.html)

<sup>950</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/idmap\\_rid.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/idmap_rid.8.html)

- `idmap_autorid(8)`<sup>951</sup>

`tdb(8)`<sup>952</sup>

`idmap_`

user ID ranges on an Ubuntu/Debian system<sup>953</sup>

- 
- 
- 
- 
- 
- 
- 
- 

- *The rid idmap backend*
- *The autorid idmap backend*

## The RID IDMap backend

`idmap_rid(8)`<sup>954</sup>  
GIDs

- 

<sup>951</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/idmap\\_autorid.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/idmap_autorid.8.html)

<sup>952</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/idmap\\_tdb.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/idmap_tdb.8.html)

<sup>953</sup> <https://www.debian.org/doc/debian-policy/ch-opersys.html#uid-and-gid-classes>

<sup>954</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/idmap\\_rid.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/idmap_rid.8.html)

- 

```
[global]
...
security = ads
realm = EXAMPLE.INTERNAL
workgroup = EXAMPLE
...
idmap config * : backend = tdb
idmap config * : range = 100000 - 199999

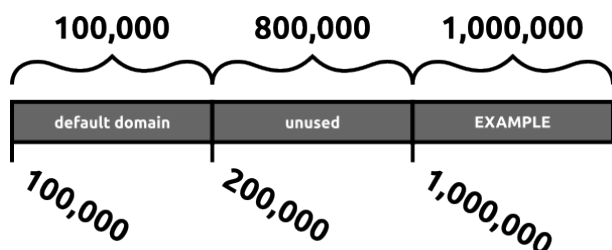
idmap config EXAMPLE : backend = rid
idmap config EXAMPLE : range = 1000000 - 1999999
```

### Note:

This is not yet a complete configuration file, and is just illustrating the IDMap configuration. More is needed to join an Active Directory domain.

- 

- 



### Important:

Planning a range of IDs to be used for the mapping critically important. Such a range can never be reduced, just expanded (carefully!), and it must **NEVER** overlap with another allocated range.

```
[global]
...
security = ads
realm = EXAMPLE.INTERNAL
workgroup = EXAMPLE
...
idmap config * : backend = tdb
idmap config * : range = 100000 - 199999

idmap config EXAMPLE : backend = rid
idmap config EXAMPLE : range = 1000000 - 1999999

idmap config COMPANY : backend = rid
idmap config COMPANY : range = 2000000 - 2999999
```

## Pros and cons of the RID backend

- 
- 
- 
- 
- 
- *The autorid IDMap backend*

## The AutoRID IDMap backend

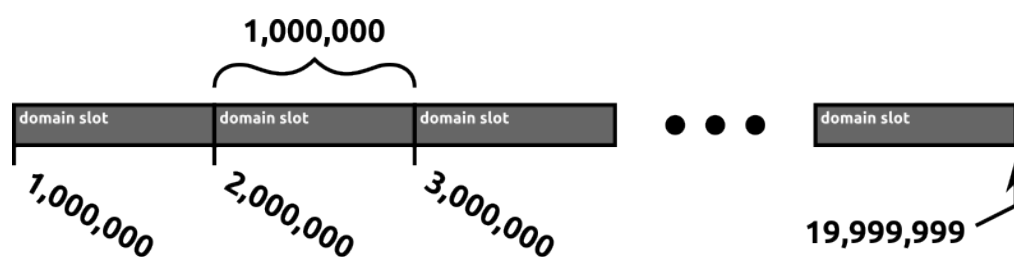
*idmap\_autorid(8)*<sup>955</sup>

- 
- 

### [global]

```
...
security = ads
realm = EXAMPLE.INTERNAL
workgroup = EXAMPLE
...
idmap config * : backend = autorid
1,000,000 - 19,999,999
idmap config * : range = 1000000 - 19999999
1,000,000
idmap config * : rangesize = 1000000
```

- 
- 
- 
- 



<sup>955</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/idmap\\_autorid.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/idmap_autorid.8.html)

**Note:**

The `autorid.tdb` domain mapping database file is kept in `/var/lib/samba/` and should be backed up regularly.

**Pros and cons of the AutoRID backend**

- 
- 
- 
- 
- 
- 
- 

**See also**

- *Active Directory integration*

**See also**

- *Networking section*

*Managing software*

- 
- 
-

## Managing software

### Package sources

- *Third party repository usage*

### Third party repository usage

### Why not use third party software?

### Security risk

*dpkg*

## System integrity

## Lack of official Ubuntu support

## A better solution to third party APT repositories: snaps

snaps<sup>956</sup>

---

<sup>956</sup> <https://snapcraft.io/docs/get-started>



published in the snap store<sup>957</sup>  
*GPG*

## Mitigating the risks

### Security risk mitigation

- 

- 

- 

### System integrity mitigation

- 

- 

---

<sup>957</sup> <https://snapcraft.io/store>

## Dealing with third party APT repositories in Ubuntu

### Add the repository

follow best practices<sup>958</sup>

### Fetch the GPG key

- 
- 

```
$ sudo wget -O /usr/share/keyrings/externalrepo-archive-keyring.asc \ >
https://thirdpartyrepo.com/ubuntu/externalrepo-archive-keyring.asc
```

```
$ curl -fsSL
https://thirdpartyrepo.com/ubuntu/externalrepo-archive-keyring.asc \ > | sudo
gpg --dearmor -o /usr/share/keyrings/externalrepo-archive-keyring.gpg
```

---

<sup>958</sup> <https://wiki.debian.org/DebianRepository/UseThirdParty>

## Sources file entry

- 
- 

```
Types: deb
URIs: https://thirdpartyrepo.com/ubuntu
Suites: resolute
Components: main
Signed-By: /usr/share/keyrings/externalrepo-archive-keyring.asc
```

```
deb [signed-by=/usr/share/keyrings/externalrepo-archive-keyring.gpg] https://
thirdpartyrepo.com/ubuntu/ resolute main
```

## Configure pinning for the repository

```
Package: *
Pin: origin thirdpartyrepo.com
Pin-Priority: 100
```

Debian Reference guide<sup>959</sup>

---

<sup>959</sup> [https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#\\_tweaking\\_candidate\\_version](https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#_tweaking_candidate_version)

## How to remove a repository

- 
- 
- 

```
$ apt remove --purge \ > $(grep "^Package: "
/var/lib/apt/lists/#<SELECT_THE_FILE_FOR_YOUR_REPOSITORY>#*_Packages \ > |
cut -d " " -f2 | sort -u | \ > xargs dpkg-query -W
-f='${binary:Package}\t${db:Status-Abbrev}\n' 2> /dev/null | \ > awk '/\ti
${print $1}')
```

## A special case: Ubuntu PPAs

```
$ add-apt-repository ppa:thirdparty/externalrepo
```

```
$ ppa-purge ppa:thirdparty/externalrepo
```

## Configuration

- *Changing package files*
- *Configuration managers*

## Changing package files

## Configuration files

```
[mysqld]
#
* Basic Settings
#
user = mysql
pid-file = /var/run/mysqld/mysqld.pid
socket = /var/run/mysqld/mysqld.sock
port = 3307
```

### Note:

Some packages do not automatically create files for you to edit in their `.d` directories. In these cases it is often acceptable to just create an additional config file by any name there. When in doubt, check the package's documentation to confirm.

```
$ systemctl restart mysql
```

```
$ netstat -tunpevaW | grep -i 3307
```

```
tcp 0 0 127.0.0.1:3307 0.0.0.0:* LISTEN 106
 416022 1730/mysqlD
```

## Systemd files

Systemd<sup>960</sup>

```
$ sudo systemctl edit apache2
```

```
Editing /etc/systemd/system/apache2.service.d/override.conf
Anything between here and the comment below will become the new contents of the file

Lines below this comment will be discarded

/lib/systemd/system/apache2.service
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
Documentation=https://httpd.apache.org/docs/2.4/
#
[Service]
Type=forking
Environment=APACHE_STARTED_BY_SYSTEMD=true
ExecStart=/usr/sbin/apachectl start
ExecStop=/usr/sbin/apachectl graceful-stop
ExecReload=/usr/sbin/apachectl graceful
KillMode=mixed
PrivateTmp=true
Restart=on-abort
...
```

---

<sup>960</sup> <https://systemd.io/>

```
Editing /etc/systemd/system/apache2.service.d/override.conf
Anything between here and the comment below will become the new contents of the file
```

### [Service]

```
Restart=on-failure
```

```
Lines below this comment will be discarded
```

```
...
```

### Note:

Some options, such as `ExecStart` are additive. If you would like to fully override them add an extra line that clears it (e.g. `ExecStart=`) before providing new options. See [Systemd's man page](https://www.freedesktop.org/software/systemd/man/latest/systemd.service.html)<sup>961</sup> for more information.

<sup>961</sup> <https://www.freedesktop.org/software/systemd/man/latest/systemd.service.html>

```
$ sudo systemctl daemon-reload
```

```
$ systemctl status apache2
```

```
□ apache2.service - The Apache HTTP Server
 Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
 Drop-In: /etc/systemd/system/apache2.service.d
 └─override.conf
 └─run/systemd/system/service.d
 └─zzz-lxc-service.conf
 Active: active (running) since Fri 2023-02-17 16:39:22 UTC; 27min ago
 Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 4735 (apache2)
 Tasks: 55 (limit: 76934)
 Memory: 6.5M
 CPU: 65ms
 CGroup: /system.slice/apache2.service
 └─4735 /usr/sbin/apache2 -k start
 └─4736 /usr/sbin/apache2 -k start
 └─4737 /usr/sbin/apache2 -k start
 ...
```

## AppArmor

### *AppArmor*

```
Configuration file '/etc/apparmor.d/usr.bin.swtpm'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ? Your options are:
 Y or I : install the package maintainer's version
 N or O : keep your currently-installed version
 D : show the differences between the versions
 Z : start a shell to examine the situation
The default action is to keep your current version.
*** usr.bin.swtpm (Y/I/N/O/D/Z) [default=N] ?
```

```
/var/customtpm/** rwk,
```

### AppArmor syntax<sup>962</sup>

#### Note:

Although most local profiles have the same name as the maintainer's, you can often check what file is included based on the main profile's contents. In swtpm's case, `/etc/apparmor.d/usr.bin.swtpm` contains the lines:

```
Site-specific additions and overrides. See local/README for details.
#include <local/usr.bin.swtpm>
```

showing that the local profile is located at `/etc/apparmor.d/local/usr.bin.swtpm`

<sup>962</sup> <https://ubuntu.com/tutorials/beginning-apparmor-profile-development>



## Restoring configuration files

*dpkg*

```
$ dpkg -S /etc/rsyslog.conf
rsyslog: /etc/rsyslog.conf
```

```
$ rm /etc/rsyslog.conf
```

```
$ dpkg -S /etc/rsyslog.conf
rsyslog: /etc/rsyslog.conf
```

```
$ sudo apt install --reinstall -o Dpkg::Options::="--force-confmiss" rsyslog
...
Preparing to unpack .../rsyslog_8.2302.0-1ubuntu3_amd64.deb ...
Unpacking rsyslog (8.2302.0-1ubuntu3) over (8.2302.0-1ubuntu3) ...
Setting up rsyslog (8.2302.0-1ubuntu3) ...

Configuration file '/etc/rsyslog.conf', does not exist on system.
Installing new config file as you requested.
```

*dpkg(1)*<sup>963</sup>

- 
- 
- 
- 

---

<sup>963</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/dpkg.1.html>

```
$ echo badentry >> /etc/rsyslog.conf
```

```
$ sudo apt install --reinstall -o Dpkg::Options::="--force-confask" rsyslog
```

```
...
Preparing to unpack .../rsyslog_8.2302.0-1ubuntu3_amd64.deb ...
Unpacking rsyslog (8.2302.0-1ubuntu3) over (8.2302.0-1ubuntu3) ...
Setting up rsyslog (8.2302.0-1ubuntu3) ...
```

```
Configuration file '/etc/rsyslog.conf'
```

```
==> Modified (by you or by a script) since installation.
```

```
Version in package is the same as at last installation.
```

```
What would you like to do about it ? Your options are:
```

```
Y or I : install the package maintainer's version
```

```
N or O : keep your currently-installed version
```

```
D : show the differences between the versions
```

```
Z : start a shell to examine the situation
```

```
The default action is to keep your current version.
```

```
*** rsyslog.conf (Y/I/N/O/D/Z) [default=N] ? D
```

```
--- /etc/rsyslog.conf 2023-04-18 07:11:50.427040350 +0000
```

```
+++ /etc/rsyslog.conf.dpkg-new 2023-02-23 16:58:03.000000000 +0000
```

```
@@ -51,4 +51,3 @@
```

```
Include all config files in /etc/rsyslog.d/
```

```
#
```

```
$IncludeConfig /etc/rsyslog.d/*.conf
```

```
-badentry
```

```
Configuration file '/etc/rsyslog.conf'
```

```
==> Modified (by you or by a script) since installation.
```

```
Version in package is the same as at last installation.
```

```
What would you like to do about it ? Your options are:
```

```
Y or I : install the package maintainer's version
```

```
N or O : keep your currently-installed version
```

```
D : show the differences between the versions
```

```
Z : start a shell to examine the situation
```

```
The default action is to keep your current version.
```

```
*** rsyslog.conf (Y/I/N/O/D/Z) [default=N] ? y
```

```
Installing new version of config file /etc/rsyslog.conf ...
```

```
$ rm -rf /etc/vim
```

```
$ dpkg -S /etc/vim
```

```
vim-common, vim-tiny: /etc/vim
```

```
$ sudo apt install --reinstall -o Dpkg::Options::="--force-confmiss"
vim-common vim-tiny

...
Configuration file '/etc/vim/vimrc', does not exist on system.
Installing new config file as you requested.
...
Configuration file '/etc/vim/vimrc.tiny', does not exist on system.
Installing new config file as you requested.
```

## Configuration managers

### Ansible

*Ansible*

- 

### Puppet

Puppet<sup>964</sup>

- 

---

<sup>964</sup> <https://www.puppet.com/>

## Further reading

- [Wikipedia table](#)<sup>965</sup> this

## Updates

- [About apt upgrade and phased updates](#)
- [Advance testing of updates in best practice server deployments](#)

## About apt upgrade and phased updates

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
 (Names of <X> held back packages listed here)
0 upgraded, 0 newly installed, 0 to remove and <X> not upgraded.
```

## What are phased updates?

---

<sup>965</sup> [https://en.wikipedia.org/wiki/Comparison\\_of\\_open-source\\_configuration\\_management\\_software](https://en.wikipedia.org/wiki/Comparison_of_open-source_configuration_management_software)

## Why is Ubuntu doing this?

### Note:

It should be mentioned here that security updates are *never* phased.

## Can I turn off phased updates?

## Can I apt upgrade the individual packages? (and should I?)

## How do I turn off phased updates?

```
Update-Manager::Always-Include-Phased-Updates true;
APT::Get::Always-Include-Phased-Updates true;
```

## Why is this suddenly happening now?

implemented in APT in 21.04<sup>966</sup>

## How does it actually work?

wiki page on phased updates<sup>967</sup>

## How can I find out more information about currently phased packages?

```
apt policy <package>
```

---

<sup>966</sup> <https://discourse.ubuntu.com/t/phased-updates-in-apt-in-21-04/20345>

<sup>967</sup> <https://wiki.ubuntu.com/PhasedUpdates>

```
libglapi-mesa:
 Installed: 22.0.5-0ubuntu0.3
 Candidate: 22.2.5-0ubuntu0.1~22.04.1
 Version table:
 22.2.5-0ubuntu0.1~22.04.1 500 (phased 20%)
 500 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages
*** 22.0.5-0ubuntu0.3 100
 100 /var/lib/dpkg/status
 22.0.1-1ubuntu2 500
 500 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages
```

currently being phased<sup>968</sup>

## Further reading

- [excellent post on AskUbuntu](#)<sup>969</sup>  
CC BY-SA 4.0<sup>970</sup>
- [phasing Ubuntu Stable Release Updates](#)<sup>971</sup>
- [Ubuntu wiki](#)<sup>972</sup> [Error Tracker](#)<sup>973</sup> [apt preferences](#)<sup>974</sup>

## Advance testing of updates in best-practice server deployments

<sup>968</sup> <https://ubuntu-archive-team.ubuntu.com/phased-updates.html>

<sup>969</sup> <https://askubuntu.com/questions/1431940/what-are-phased-updates-and-why-does-ubuntu-use-them>

<sup>970</sup> <https://creativecommons.org/licenses/by-sa/4.0/>

<sup>971</sup> <https://ubuntu-archive-team.ubuntu.com/phased-updates.html>

<sup>972</sup> <https://wiki.ubuntu.com/PhasedUpdates>

<sup>973</sup> <https://wiki.ubuntu.com/ErrorTracker/PhasedUpdates>

<sup>974</sup> [https://manpages.ubuntu.com/manpages/resolute/man5/apt\\_preferences.5.html](https://manpages.ubuntu.com/manpages/resolute/man5/apt_preferences.5.html)

## Information about proposed updates

[Pending SRU page<sup>975</sup>](#)

## Testing a single proposed update

[Enable proposed wiki page<sup>976</sup>](#)

## Test all proposed updates together

---

<sup>975</sup> <https://ubuntu-archive-team.ubuntu.com/pending-sru.html>

<sup>976</sup> <https://wiki.ubuntu.com/Testing/EnableProposed>



## How to report a regression

- 
- 
- 
- 

pending SRU report<sup>977</sup>

## General information on reporting bugs<sup>978</sup>

### See also

- *Managing software*

### *In data and storage*

- 
- 

## Data and storage

## Data management

---

<sup>977</sup> <https://ubuntu-archive-team.ubuntu.com/pending-sru.html>

<sup>978</sup> <https://help.ubuntu.com/community/ReportingBugs>

## Introduction to OpenLDAP

Lightweight Directory Access Protocol<sup>979</sup>

RFC 4510<sup>980</sup>

### Key concepts and terms

- *DIT*
- 
- 
- 
- *DN*
- 

#### Note:

The terms **object**, **container**, and **node** have certain connotations but they all essentially mean the same thing as **entry** (the technically correct term).

- 
- 
- 

```
dn: cn=John Doe,dc=example,dc=com
cn: John Doe
givenName: John
```

(continues on next page)

<sup>979</sup> <https://ldap.com/>

<sup>980</sup> <https://datatracker.ietf.org/doc/html/rfc4510>

(continued from previous page)

```
sn: Doe
telephoneNumber: +1 888 555 6789
telephoneNumber: +1 888 555 1232
mail: john@example.com
manager: cn=Larry Smith,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

RFC 2849<sup>981</sup>

## Our OpenLDAP guide

- *Install and configure LDAP*
- *LDAP Access Control*
- *LDAP users and groups*
- *SSL/TLS*
- *Replication*
- *Backup and restore*

## Further reading

- *OpenLDAP administrators guide*<sup>982</sup>
- *RFC 4515: LDAP string representation of search filters*<sup>983</sup>
- *LDAP for Rocket Scientists*<sup>984</sup>
- *LDAP System Administration*<sup>985</sup>
- *Mastering OpenLDAP*<sup>986</sup>

---

<sup>981</sup> <https://datatracker.ietf.org/doc/html/rfc2849>

<sup>982</sup> <https://openldap.org/doc/admin25/>

<sup>983</sup> <https://www.rfc-editor.org/rfc/rfc4515.txt>

<sup>984</sup> <https://www.zytrax.com/books/ldap/>

<sup>985</sup> <https://www.oreilly.com/library/view/ldap-system-administration/1565924916/>

<sup>986</sup> <https://www.packtpub.com/en-us/product/mastering-openldap-configuring-securing-and-integrating-directory-services-9781449122069>

## Introduction to databases

MySQL<sup>987</sup>

PostgreSQL<sup>988</sup>

*section*

*how-to*

- *How to install and configure MySQL*
- *How to install and configure PostgreSQL*

## Storage

- *Storage*
- *Multipath*

## Storage

- *About LVM*

## About Logical Volume Management (LVM)

Logical Volume Management<sup>989</sup>

*refer to this guide*

---

<sup>987</sup> <https://www.mysql.com/>

<sup>988</sup> <https://www.postgresql.org/>

<sup>989</sup> [https://en.wikipedia.org/wiki/Logical\\_volume\\_management](https://en.wikipedia.org/wiki/Logical_volume_management)

## Key concepts

- 
- 
- 

*filesystem*

## Resizing partitions

## Moving Partitions

## Snapshots

## See also

- [\*Storage\*](#)

## Multipath

### Introduction

- [\*Introduction to Device mapper multipathing\*](#)

### Configuration

- [\*Configuration options and overview\*](#)
- [\*Configuration examples\*](#)
- [\*Common tasks and procedures\*](#)

### Introduction to device mapper multipathing (“multipath”)

### How a multipath device works in practice

### What identifier does a multipath device use?

## What does multipath provide?

- *failover*
- 

## Configuring multipath for a storage system

### Multipath components

| Component                            | Description                                                                                                                                                                                                                                                                  |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dm_<br>multipath<br>kernel<br>module | Reroutes I/O and supports <b>failover</b> for paths and path groups.                                                                                                                                                                                                         |
| mul-<br>tipath<br>command            | Lists and configures multipath devices. While this command is normally started up with <code>/etc/rc.sysinit</code> , it can also be started up by a <code>udev</code> program whenever a block device is added, or it can be run by the <code>initramfs</code> file system. |
| multi-<br>pathd<br>daemon            | Monitors paths; as paths fail and come back, it may initiate path group switches. Provides for interactive changes to multipath devices. This daemon must be restarted for any changes to the <code>/etc/multipath.conf</code> file to take effect.                          |
| kpartx<br>com-<br>mand               | Creates device-mapper devices for the partitions on a device. It is necessary to use this command for DOS-based partitions with multipath. The <code>kpartx</code> is provided in its own package, but the <code>multipath-tools</code> package depends on it.               |

## Multipath setup overview

- 1.
- 2.
- 3.
- 4.
- 5.

*DM-Multipath setup*

*DM-Multipath configuration*

## Example of a multipath device in use

*FC*

*HBAs*

- 
- 

### See also

For information on the multipath configuration defaults, including the `user_friendly_names` and `alias` configuration options, see *DM-Multipath configuration* (page 978).



## Consistent multipath device names in a cluster

*use LVM*

1.

2.

```
$ systemctl stop multipath-tools.service & multipath -F
```

3.

4.

```
$ systemctl start multipath-tools.service
```

## Multipath device attributes

DM-

*Multipath configuration*

## Multipath devices in logical volumes

```
$ pvcreate /dev/mapper/mpatha
```

### Note:

If you try to create an LVM physical volume on a whole device on which you have configured partitions, the `pvcreate` command will fail.

*fallbacks*

```
filter = ["r/block/", "r/disk/", "r/sd.*/", "a/.*/"]
```

```
$ update-initramfs -u -k all
```

### Note:

Every time either `/etc/lvm.conf` or `/etc/multipath.conf` is updated, the `initrd` should be rebuilt to reflect these changes. This is imperative when *denylists* and filters are necessary to maintain a stable storage configuration.

## Multipath configuration options and overview

*device mapper multipathing introduction*

### Configuration file overview

```
<section> {
 <attribute> <value>
 ...
 <subsection> {
 <attribute> <value>
 ...
 }
}
```

- 
- 
- 
- 
- 
-

## Configuration file defaults

*in the manual pages*<sup>990</sup>

- 

<sup>990</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/multipath.conf.5.html>

- .....

- 
- 
- 
- 
- 

#### Note:

Previously, the `multipath-tools` project provided a complete configuration file with all the most commonly used options for each of the most-used storage devices. Currently, you can see all those default options by running `sudo multipath -t`. This will dump a used configuration file including all the embedded default options.

## Configuration file blacklist and exceptions

### By devnode

```
blacklist {
 devnode "^(ram|zram|raw|loop|fd|md|dm-|sr|scd|st|dcssblk)[0-9]"
 devnode "^(td|hd|vd)[a-z]"
 devnode "^cciss!c[0-9]d[0-9]*"
}
```

### By wwid

### By device

```
device {
 vendor "LENOVO"
 product "Universal Xport"
}
```

## By property

### Blocklist by protocol

```
$ multipathd show paths format "%d %P"
```

### Blocklist exceptions

```
blacklist_exceptions {
 property "(SCSI_IDENT_|ID_WWN)"
}
```

#### Note:

A common use is to blocklist “everything” using a catch-all regular expression, and create specific `blacklist_exceptions` entries for those devices that should be handled by `multipath-tools`.

### Configuration file multipath section

-

- 

## Optional attributes

*multipath.conf*(5)<sup>991</sup>

- • • • •

<sup>991</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/multipath.conf.5.html>



## Example

```
multipaths {
 multipath {
 wwid 3600508b4000156d700012000000b0000
 alias yellow
 path_grouping_policy multibus
 path_selector "round-robin 0"
 failback manual
 rr_weight priorities
 no_path_retry 5
 }
 multipath {
 wwid 1DEC_____321816758474
 alias red
 }
}
```

## Configuration file devices section

1.

2.

3.

4.

5.

6.

- *DGC*

- *Engenio*

•

## Optional attributes

•

- 
- 
- 
- 
- 
- 
- 

## Example

```
devices {
 device {
 vendor "3PARdata"
 product "VV"
 path_grouping_policy "group_by_prio"
 hardware_handler "1 alua"
 prio "alua"
 failback "immediate"
 no_path_retry 18
 fast_io_fail_tmo 10
 dev_loss_tmo "infinity"
 }
 device {
 vendor "DEC"
 product "HSG80"
 path_grouping_policy "group_by_prio"
 path_checker "hp_sw"
 hardware_handler "1 hp_sw"
 prio "hp_sw"
 no_path_retry "queue"
 }
}
```

## Multipath configuration examples

1. *Introduction to device mapper multipathing*
2. *Configuration options and overview*

## Basic setup

*the configuration overview*

```
$ sudo multipath -t
```

## Example of a defaults section

```
defaults {
 #
 # name : polling_interval
 # scope : multipathd
 # desc : interval between two path checks in seconds. For
 # properly functioning paths, the interval between checks
 # will gradually increase to (4 * polling_interval).
 # values : n > 0
 # default : 5
 #
 polling_interval 10

 #
 # name : path_selector
 # scope : multipath & multipathd
 # desc : the default path selector algorithm to use
 # these algorithms are offered by the kernel multipath target
 # values : "round-robin 0" = Loop through every path in the path group,
 # sending the same amount of IO to each.
 # "queue-length 0" = Send the next bunch of IO down the path
 # with the least amount of outstanding IO.
 # "service-time 0" = Choose the path for the next bunch of IO
 # based on the amount of outstanding IO to
 # the path and its relative throughput.
 # default : "service-time 0"
 #
 path_selector "round-robin 0"
```

(continues on next page)

(continued from previous page)

```
#
name : path_grouping_policy
scope : multipath & multipathd
desc : the default path grouping policy to apply to unspecified
multipaths
values : failover = 1 path per priority group
multibus = all valid paths in 1 priority group
group_by_serial = 1 priority group per detected serial
number
group_by_prio = 1 priority group per path priority
value
group_by_node_name = 1 priority group per target node name
default : failover
#
path_grouping_policy multibus

#
name : uid_attribute
scope : multipath & multipathd
desc : the default udev attribute from which the path
identifier should be generated.
default : ID_SERIAL
#
uid_attribute "ID_SERIAL"

#
name : getuid_callout
scope : multipath & multipathd
desc : the default program and args to callout to obtain a unique
path identifier. This parameter is deprecated.
This parameter is deprecated, superseded by uid_attribute
default : /lib/udev/scsi_id --whitelisted --device=/dev/%n
#
getuid_callout "/lib/udev/scsi_id --whitelisted --device=/dev/%n"

#
name : prio
scope : multipath & multipathd
desc : the default function to call to obtain a path
priority value. The ALUA bits in SPC-3 provide an
exploitable prio value for example.
default : const
#
prio "alua"

#
name : prio_args
scope : multipath & multipathd
desc : The arguments string passed to the prio function
Most prio functions do not need arguments. The
datacore prioritizer need one.
default : (null)
#
prio_args "timeout=1000 preferredsds=foo"

#
```

(continues on next page)

(continued from previous page)

```
name : features
scope : multipath & multipathd
desc : The default extra features of multipath devices.
Syntax is "num[feature_0 feature_1 ...]", where 'num' is the
number of features in the following (possibly empty) list of
features.
values : queue_if_no_path = Queue IO if no path is active; consider
using the 'no_path_retry' keyword instead.
no_partitions = Disable automatic partitions generation via
kpartx.
default : "0"
#
features "0"
#features "1 queue_if_no_path"
#features "1 no_partitions"
#features "2 queue_if_no_path no_partitions"

#
name : path_checker, checker
scope : multipath & multipathd
desc : the default method used to determine the paths' state
values : readsector0|tur|emc_clariion|hp_sw|directio|rdac|cciss_tur
default : directio
#
path_checker directio

#
name : rr_min_io
scope : multipath & multipathd
desc : the number of IO to route to a path before switching
to the next in the same path group for the bio-based
multipath implementation. This parameter is used for
kernels version up to 2.6.31; newer kernel version
use the parameter rr_min_io_rq
default : 1000
#
rr_min_io 100

#
name : rr_min_io_rq
scope : multipath & multipathd
desc : the number of IO to route to a path before switching
to the next in the same path group for the request-based
multipath implementation. This parameter is used for
kernels versions later than 2.6.31.
default : 1
#
rr_min_io_rq 1

#
name : flush_on_last_del
scope : multipathd
desc : If set to "yes", multipathd will disable queueing when the
last path to a device has been deleted.
values : yes|no
default : no
```

(continues on next page)

(continued from previous page)

```
#
flush_on_last_del yes

#
name : max_fds
scope : multipathd
desc : Sets the maximum number of open file descriptors for the
multipathd process.
values : max|n > 0
default : None
#
max_fds 8192

#
name : rr_weight
scope : multipath & multipathd
desc : if set to priorities the multipath configurator will assign
path weights as "path prio * rr_min_io"
values : priorities|uniform
default : uniform
#
rr_weight priorities

#
name : failback
scope : multipathd
desc : tell the daemon to manage path group failback, or not to.
0 means immediate failback, values >0 means deferred
failback expressed in seconds.
values : manual|immediate|n > 0
default : manual
#
failback immediate

#
name : no_path_retry
scope : multipath & multipathd
desc : tell the number of retries until disable queueing, or
"fail" means immediate failure (no queueing),
"queue" means never stop queueing
values : queue|fail|n (>0)
default : (null)
#
no_path_retry fail

#
name : queue_without_daemon
scope : multipathd
desc : If set to "no", multipathd will disable queueing for all
devices when it is shut down.
values : yes|no
default : yes
queue_without_daemon no

#
name : user_friendly_names
```

(continues on next page)

(continued from previous page)

```
scope : multipath & multipathd
desc : If set to "yes", using the bindings file
/etc/multipath/bindings to assign a persistent and
unique alias to the multipath, in the form of mpath<n>.
If set to "no" use the WWID as the alias. In either case
this be will be overridden by any specific aliases in this
file.
values : yes|no
default : no
user_friendly_names yes

#
name : mode
scope : multipath & multipathd
desc : The mode to use for the multipath device nodes, in octal.
values : 0000 - 0777
default : determined by the process
mode 0644

#
name : uid
scope : multipath & multipathd
desc : The user id to use for the multipath device nodes. You
may use either the numeric or symbolic uid
values : <user_id>
default : determined by the process
uid 0

#
name : gid
scope : multipath & multipathd
desc : The group id to user for the multipath device nodes. You
may use either the numeric or symbolic gid
values : <group_id>
default : determined by the process
gid disk

#
name : checker_timeout
scope : multipath & multipathd
desc : The timeout to use for path checkers and prioritizers
that issue scsi commands with an explicit timeout, in
seconds.
values : n > 0
default : taken from /sys/block/sd<x>/device/timeout
checker_timeout 60

#
name : fast_io_fail_tmo
scope : multipath & multipathd
desc : The number of seconds the scsi layer will wait after a
problem has been detected on a FC remote port before failing
IO to devices on that remote port.
values : off | n >= 0 (smaller than dev_loss_tmo)
default : determined by the OS
fast_io_fail_tmo 5
```

(continues on next page)



(continued from previous page)

```
#
name : dev_loss_tmo
scope : multipath & multipathd
desc : The number of seconds the scsi layer will wait after a
problem has been detected on a FC remote port before
removing it from the system.
values : infinity | n > 0
default : determined by the OS
dev_loss_tmo 120

#
name : bindings_file
scope : multipath
desc : The location of the bindings file that is used with
the user_friendly_names option.
values : <full_pathname>
default : "/var/lib/multipath/bindings"
bindings_file "/etc/multipath/bindings"

#
name : wwids_file
scope : multipath
desc : The location of the wwids file multipath uses to
keep track of the created multipath devices.
values : <full_pathname>
default : "/var/lib/multipath/wwids"
wwids_file "/etc/multipath/wwids"

#
name : reservation_key
scope : multipath
desc : Service action reservation key used by mpathpersist.
values : <key>
default : (null)
reservation_key "mpathkey"

#
name : force_sync
scope : multipathd
desc : If set to yes, multipath will run all of the checkers in
sync mode, even if the checker has an async mode.
values : yes|no
default : no
force_sync yes

#
name : config_dir
scope : multipath & multipathd
desc : If not set to an empty string, multipath will search
this directory alphabetically for files ending in ".conf"
and it will read configuration information from these
files, just as if it was in /etc/multipath.conf
values : "" or a fully qualified pathname
default : "/etc/multipath/conf.d"
```

(continues on next page)

(continued from previous page)

```
#
name : delay_watch_checks
scope : multipathd
desc : If set to a value greater than 0, multipathd will watch
paths that have recently become valid for this many
checks. If they fail again while they are being watched,
when they next become valid, they will not be used until
they have stayed up for delay_wait_checks checks.
values : no|<n> > 0
default : no
delay_watch_checks 12

#
name : delay_wait_checks
scope : multipathd
desc : If set to a value greater than 0, when a device that has
recently come back online fails again within
delay_watch_checks checks, the next time it comes back
online, it will marked and delayed, and not used until
it has passed delay_wait_checks checks.
values : no|<n> > 0
default : no
delay_wait_checks 12
}
```

## Example of a multipaths section

### Note:

You can obtain the WWIDs for your LUNs by running: `multipath -ll` after the service `multipath-tools.service` has been restarted.

```
multipaths {
 multipath {
 wwid 36000000000000000e00000000030001
 alias yellow
 }
 multipath {
 wwid 36000000000000000e00000000020001
 alias blue
 }
 multipath {
 wwid 36000000000000000e00000000010001
 alias red
 }
 multipath {
 wwid 36000000000000000e00000000040001
 alias green
 }
 multipath {
 wwid 36000000000000000e00000000050001
 alias purple
 }
}
```

(continues on next page)

(continued from previous page)

```
}
}
```

### Example of a devices section

```
devices {
device {
vendor "IBM"
product "2107900"
path_grouping_policy group_by_serial
}
}
#
```

### Example of a blacklist section

```
name : blacklist
scope : multipath & multipathd
desc : list of device names to discard as not multipath candidates

Devices can be identified by their device node name "devnode",
their WWID "wwid", or their vendor and product strings "device"
default : fd, hd, md, dm, sr, scd, st, ram, raw, loop, dcscblk

blacklist {
wwid 26353900f02796769
devnode "^ (ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
devnode "^hd[a-z]"
devnode "^dcscblk[0-9]*"
device {
vendor DEC.*
product MSA[15]00
}
}
```

### Example of a blacklist exception section

```
name : blacklist_exceptions
scope : multipath & multipathd
desc : list of device names to be treated as multipath candidates
even if they are on the blacklist.

Note: blacklist exceptions are only valid in the same class.
It is not possible to blacklist devices using the devnode keyword
and to exclude some devices of them using the wwid keyword.
default : -

blacklist_exceptions {
devnode "^dasd[c-d]+[0-9]*"
```

(continues on next page)

(continued from previous page)

```
wwid "IBM.75000000092461.4d00.34"
wwid "IBM.75000000092461.4d00.35"
wwid "IBM.75000000092461.4d00.36"
}
```

## Common tasks and procedures

1. *Introduction to device mapper multipathing*
2. *Configuration options and overview*
3. *Configuration examples*

## Resizing online multipath devices

```
$ sudo multipath -ll

mpathb (360014056eee8ec6e1164fcb959086482) dm-0 LIO-ORG,lun01
size=1.0G features='0' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| '- 7:0:0:1 sde 8:64 active ready running
`+- policy='service-time 0' prio=50 status=enabled
 '- 8:0:0:1 sdf 8:80 active ready running
mpatha (36001405e3c2841430ee4bf3871b1998b) dm-1 LIO-ORG,lun02
size=1.0G features='0' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| '- 7:0:0:2 sdc 8:32 active ready running
`+- policy='service-time 0' prio=50 status=enabled
 '- 8:0:0:2 sdd 8:48 active ready running
```

```
$ echo 1 | sudo tee /sys/block/sde/device/rescan
```

```
1
```

```
$ echo 1 | sudo tee /sys/block/sdf/device/rescan
```

```
1
```

```
$ sudo multipath -ll

mpathb (360014056eee8ec6e1164fcb959086482) dm-0 LIO-ORG,lun01
size=1.0G features='0' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| '- 7:0:0:1 sde 8:64 active ready running
`+- policy='service-time 0' prio=50 status=enabled
 '- 8:0:0:1 sdf 8:80 active ready running
mpatha (36001405e3c2841430ee4bf3871b1998b) dm-1 LIO-ORG,lun02
size=1.0G features='0' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| '- 7:0:0:2 sdc 8:32 active ready running
`+- policy='service-time 0' prio=50 status=enabled
 '- 8:0:0:2 sdd 8:48 active ready running
```

```
$ sudo multipathd resize map mpathb
```

```
ok
```

```
$ sudo multipath -ll

mpathb (360014056eee8ec6e1164fcb959086482) dm-0 LIO-ORG,lun01
size=2.0G features='0' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| '- 7:0:0:1 sde 8:64 active ready running
.-+- policy='service-time 0' prio=50 status=enabled
 '- 8:0:0:1 sdf 8:80 active ready running
mpatha (36001405e3c2841430ee4bf3871b1998b) dm-1 LIO-ORG,lun02
size=1.0G features='0' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| '- 7:0:0:2 sdc 8:32 active ready running
.-+- policy='service-time 0' prio=50 status=enabled
 '- 8:0:0:2 sdd 8:48 active ready running
```

*filesystem*

## Move root file system from a single path device to a multipath device

### Note:

Whenever `multipath.conf` is updated, `initrd` should be updated as well by running: `update-initramfs -u -k all`. The reason for this is that `multipath.conf` is copied to the RAM disk, and is integral to determining the available devices to map via its `denylist` and `devices` sections.

### The multipathd daemon

*the example configuration page*

### Multipath command output

```
action_if_any: alias (wwid_if_different_from_alias) dm_device_name_if_known vendor,product
size=size features='features' hwhandler='hardware_handler' wp=write_permission_if_known
```

```
-+- policy='scheduling_policy' prio=prio_if_known
status=path_group_status_if_known
```

```
`- host:channel:id:lun devnode major:minor dm_status_if_known path_status
online_status
```

```
mpathb (360014056eee8ec6e1164fcb959086482) dm-0 LIO-ORG,lun01
size=2.0G features='0' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| `-- 7:0:0:1 sde 8:64 active ready running
`-+- policy='service-time 0' prio=50 status=enabled
 `-- 8:0:0:1 sdf 8:80 active ready running
```

## Multipath queries with the multipath command

- 
- 
- 
- 
- 

### Note:

The default verbosity level of multipath is 2 and can be globally modified by defining the verbosity attribute in the defaults section of `multipath.conf`

```
mpathb (360014056eee8ec6e1164fcb959086482) dm-0 LIO-ORG,lun01
size=2.0G features='0' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=0 status=active
| `-- 7:0:0:1 sde 8:64 active undef running
`+- policy='service-time 0' prio=0 status=enabled
 `-- 8:0:0:1 sdf 8:80 active undef running
mpatha (36001405e3c2841430ee4bf3871b1998b) dm-1 LIO-ORG,lun02
size=1.0G features='0' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=0 status=active
| `-- 7:0:0:2 sdc 8:32 active undef running
`+- policy='service-time 0' prio=0 status=enabled
 `-- 8:0:0:2 sdd 8:48 active undef running
```

## Determining device mapper entries with dmsetup

```
$ sudo dmsetup ls
```

```
mpathb (253:0)
mpatha (253:1)
```

```
$ ls -lahd /dev/dm*
```

```
brw-rw---- 1 root disk 253, 0 Apr 27 14:49 /dev/dm-0
brw-rw---- 1 root disk 253, 1 Apr 27 14:47 /dev/dm-1
```

## Troubleshooting with the multipathd interactive console

```
$ sudo multipathd -k
```

```
> show config
> CTRL-D
```

```
$ sudo multipathd -k
```

```
> reconfigure
> CTRL-D
```

```
$ sudo multipathd -k
```

```
> show paths
> CTRL-D
```



```
$ echo 'show config' | sudo multipathd -k
```

## See also

- [Data and storage](#)
- [Backup section](#)

[Web services](#)

## Web services

### Introduction

#### Introduction to web services

[on the key concepts](#)

## Squid proxy server

[about Squid proxy servers](#)  
[install a Squid server](#)

[how to](#)

## Web servers

- [How to install and configure Apache2](#)
- [How to configure Apache2 for your needs](#)
- [How to extend Apache2's functionality with modules](#)

- *How to install Nginx*
- *How to configure Nginx*
- *How to use Nginx modules*

## Database server

*how to install MySQL*

*PostgreSQL*

## Scripting languages

*install Ruby on Rails*

*how to install PHP*

*how to*

## Key concepts

### About web servers

## URLs

*Qualified Domain Name (FQDN)*  
*Ubuntu Web site*<sup>992</sup>

*Fully*

`www.ubuntu.com`

`community`<sup>993</sup>

`www.ubuntu.com/community`

---

<sup>992</sup> <https://ubuntu.com/>

<sup>993</sup> <https://ubuntu.com/community>

## Transfer protocols

## HTTP status codes

- 
- 
- 
- 
- 

check the RFC 2616<sup>994</sup>

## Implementation

- 
- 

## About Squid proxy servers

Squid<sup>995</sup>

---

<sup>994</sup> <https://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html#sec6.1.1>

<sup>995</sup> <http://www.squid-cache.org/>

## Caching

*DNS*

## Filtering

## Installing and configuring

*installation guide*

*our*

## Further reading

- The Squid website<sup>996</sup>
- Ubuntu Wiki page on Squid<sup>997</sup>

---

<sup>996</sup> <http://www.squid-cache.org/>

<sup>997</sup> <https://help.ubuntu.com/community/Squid>

## See also

- *Web services*

*Virtualisation and containers*

## Virtualisation and containers

- *Introduction to virtualization*

## Introduction to virtualization

## Virtual machines

*and technologies*

*the different VM tools*

## Containers

## System containers

## Application containers

*overview*

*take a look at this*

## Virtualization

- *VM tools overview*
- *QEMU microvm*
- *Upgrading the machine type of your VM*

## VM tools in the Ubuntu space

## KVM

Kernel-Based Virtual Machine (KVM)<sup>998</sup>

*AMD*

---

<sup>998</sup> [https://www.linux-kvm.org/page/Main\\_Page](https://www.linux-kvm.org/page/Main_Page)

## QEMU

Quick Emulator (QEMU)<sup>999</sup>

command line interface<sup>1000</sup>      a monitor<sup>1001</sup>

*how to set up QEMU*

## libvirt

libvirt<sup>1002</sup>

*to use libvirt*

*how*

## Multipass and UVtool

Multipass<sup>1003</sup>      UVtool<sup>1004</sup>

*how to use*

*Multipass*

*how to use UVtool*

---

<sup>999</sup> <https://www.qemu.org/>

<sup>1000</sup> <https://qemu-project.gitlab.io/qemu/system/invoke.html>

<sup>1001</sup> <https://qemu-project.gitlab.io/qemu/system/monitor.html>

<sup>1002</sup> <https://libvirt.org/>

<sup>1003</sup> <https://canonical.com/multipass>

<sup>1004</sup> <https://launchpad.net/uvtool>

## virt-manager

Virt-manager<sup>1005</sup>

*graphical user interface (GUI)*

*this how-to guide*

## QEMU microvm

1. `microvm` machine type<sup>1006</sup>
2. *firmware (FW)* called `qboot`<sup>1007</sup>
- 3.

## Why a special workload?

- *bootloaders*

---

<sup>1005</sup> <https://virt-manager.org/>

<sup>1006</sup> <https://www.qemu.org/docs/master/system/i386/microvm.html>

<sup>1007</sup> <https://github.com/bonzini/qboot>



- *filesystem*
- BusyBox<sup>1008</sup>
- 
- kata containers<sup>1009</sup>
- Firecracker<sup>1010</sup>
- *LXD*
- *LXD*
- slim from dockerfiles<sup>1011</sup>
- krunvm from OCI images<sup>1012</sup>

## Create the example workload artifact

sliced rock<sup>1013</sup>

OCI-compatible<sup>1014</sup>

---

<sup>1008</sup> <https://github.com/SecureVirtualization/busybox>

<sup>1009</sup> <https://katacontainers.io/>

<sup>1010</sup> <https://firecracker-microvm.github.io/>

<sup>1011</sup> <https://github.com/ottomatica/slim>

<sup>1012</sup> <https://github.com/containers/krunvm>

<sup>1013</sup> <https://documentation.ubuntu.com/rockcraft/stable/tutorial/chisel/>

<sup>1014</sup> <https://github.com/opencontainers/image-spec/blob/main/spec.md>

```
$ sudo rockcraft.skopeo --insecure-policy copy
oci-archive:chiselled-hello_latest_amd64.rock oci:chiselled-hello.oci:latest
```

```
$ sudo apt install oci-image-tool
```

```
$ oci-image-tool unpack --ref name=latest chiselled-hello.oci
/tmp/chiselled-hello
```

```
$ mkdir /tmp/chiselled-hello/{dev,proc,sys,run,var}
```

```
$ sudo apt install guestfs-tools
```

```
$ sudo virt-make-fs --format=qcow2 --size=50M /tmp/chiselled-hello
chiselled-hello.qcow2
```

## Run the stripped workload in QEMU

```
$ sudo qemu-system-x86_64 -m 128M -machine accel=kvm \ > -kernel
/boot/vmlinuz-$(uname -r) \ > -append 'console=ttyS0 root=/dev/vda
fsck.mode=skip init=/usr/bin/hello' \ > -nodefaults -no-user-config \ >
-display none -serial mon:stdio \ > -drive
file=chiselled-hello.qcow2,index=0,format=qcow2,media=disk,if=none,id=virtio1
\ > -device virtio-blk-pci,drive=virtio1

...
[2.116207] Run /usr/bin/hello as init process
Hello, world!
```

| command-line element                            | Explanation                                                                                                                                                                                                                                    |
|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>sudo</code>                               | <code>sudo</code> is a simple way for this example to work, but not recommended. Scenarios outside of an example should use separate kernel images and a user that is a member of the <code>kvm</code> group to access <code>/dev/kvm</code> . |
| <code>qemu-system-x86_64</code>                 | Call the usual binary of QEMU used for system virtualization.                                                                                                                                                                                  |
| <code>-m 128M</code>                            | Allocate 128 megabytes of RAM for the guest.                                                                                                                                                                                                   |
| <code>-machine accel=kvm</code>                 | Enable KVM.                                                                                                                                                                                                                                    |
| <code>-kernel /boot/vmlinuz-\$(uname -r)</code> | Load the currently running kernel for the guest.                                                                                                                                                                                               |
| <code>-append '...'</code>                      | This passes four arguments to the kernel explained one by one in the following rows.                                                                                                                                                           |
| <code>console=ttyS0</code>                      | Tells the kernel which serial console it should send its output to.                                                                                                                                                                            |
| <code>root=/dev/vda</code>                      | Informs it where to expect the root device matching the <code>virtio-block</code> device we provide.                                                                                                                                           |
| <code>fsck.mode=skip</code>                     | Instructs the kernel to skip filesystem checks, which saves time.                                                                                                                                                                              |
| <code>init=/usr/bin/hello</code>                | Tell the kernel to directly start our test workload.                                                                                                                                                                                           |
| <code>-nodefaults</code>                        | Do not create the default set of devices.                                                                                                                                                                                                      |
| <code>-no-user-config</code>                    | Do not load any user provided config files.                                                                                                                                                                                                    |
| <code>-display none</code>                      | Disable video output (due to <code>-nodefaults</code> and <code>-display none</code> we do not also need <code>-nographic</code> ).                                                                                                            |
| <code>-serial mon:stdio</code>                  | Map the virtual serial port and the monitor (for debugging) to <code>stdio</code> .                                                                                                                                                            |
| <code>-drive ... -device ...</code>             | Provide our test image as <code>virtio</code> -based block device.                                                                                                                                                                             |

## Using qboot and microvm

SeaBIOS<sup>1016</sup>   UEFI<sup>1017</sup>   qboot<sup>1015</sup>

QEMU microvm<sup>1018</sup>

Firecracker<sup>1019</sup>

microvm<sup>1020</sup>

```
$ sudo qemu-system-x86_64-microvm -m 128M -machine accel=kvm,rtc=on \ > -bios
/usr/share/qemu/qboot.rom \ > -kernel /boot/vmlinuz-$(uname -r) \ > -append
'console=ttyS0 root=/dev/vda fsck.mode=skip init=/usr/bin/hello' \ >
-nodefaults -no-user-config \ > -display none -serial mon:stdio \ > -drive
file=chiselled-hello.qcow2,index=0,format=qcow2,media=disk,if=none,id=virtio1
-device virtio-blk-device,drive=virtio1
```

---

<sup>1015</sup> <https://github.com/bonzini/qboot>

<sup>1016</sup> <https://www.seabios.org>

<sup>1017</sup> <https://github.com/tianocore/edk2>

<sup>1018</sup> <https://github.com/qemu/qemu/blob/master/docs/system/i386/microvm.rst>

<sup>1019</sup> <https://firecracker-microvm.github.io/>

<sup>1020</sup> <https://github.com/qemu/qemu/blob/master/docs/system/i386/microvm.rst#running-a-microvm-based-vm>

| command-<br>line ele-<br>ment                      | Explanation                                                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| qemu-<br>system-<br>x86_64-<br>microvm             | Call the lighter, feature-reduced QEMU binary.                                                                                                                                                                                                                                                                               |
| -bios<br>/usr/<br>share/<br>qemu/<br>qboot.<br>rom | Running QEMU as <code>qemu-system-x86_64-microvm</code> will auto-select <code>/usr/share/seabios/bios-microvm.bin</code> which is a simplified SeaBIOS for this purpose. But, for the example shown here we want the even simpler <code>qboot</code> , so in addition we set <code>-bios /usr/share/qemu/qboot.rom</code> . |
| <i>info</i>                                        | QEMU will auto-select the <code>microvm</code> machine type, equivalent to <code>-M microvm</code> which therefore doesn't need to be explicitly included here.                                                                                                                                                              |
| ...<br>virtio-<br>blk-<br>device<br>...            | This feature-reduced QEMU only supports <code>virtio-bus</code> , so we need to switch the type <code>virtio-blk-pci</code> to <code>virtio-blk-device</code> .                                                                                                                                                              |

Sadly, polluting this nice showcase there is currently an issue with the RTC initialisation not working in this mode - which makes the guest kernel wait  $\sim 1.3 + \sim 1.4$  seconds. See this [QEMU bug<sup>1021</sup>](https://bugs.launchpad.net/qemu/bugs/2074073) if you are curious about that.

But these changes were not about making the guest faster once it runs, instead it mostly is about the initialisation time (and kernel init by having less virtual hardware). And that we can check despite this issue.

<sup>1021</sup> <https://bugs.launchpad.net/qemu/bugs/2074073>

## Alternative - using virtiofs

```
$ sudo rockcraft.skopeo --insecure-policy copy
oci-archive:chiselled-hello_latest_amd64.rock oci:chiselled-hello.oci:latest
```

```
$ sudo apt install oci-image-tool
```

```
$ oci-image-tool unpack --ref name=latest chiselled-hello.oci
/tmp/chiselled-hello
```

```
$ sudo apt install virtiofsd > /usr/libexec/virtiofsd
--socket-path=/tmp/vfsd.sock --shared-dir /tmp/chiselled-hello
...
[INFO virtiofsd] Waiting for vhost-user socket connection...
```

```
...
-object memory-backend-memfd,id=mem,share=on,size=128M \
-numa node,memdev=mem -chardev socket,id=char0,path=/tmp/vfsd.sock \
-device vhost-user-fs-pci,queue-size=1024,chardev=char0,tag=myfs
...
```

## Upgrading the machine type of your VM

## Why should you do this for a VM?

- 
- *GPU*
- 
- 
- 
- 

## How does this work?

### Backing up guest definitions

### Upgrade the machine type

## Shut down the VM

```
virsh shutdown <your_machine>
```

```
$ virsh list --inactive
```

## Edit the guest definition

```
virsh edit <your_machine>
<type arch='x86_64' machine='pc-i440fx-bionic'>hvm</type>
```

```
$ kvm -M ?

pc-i440fx-xenial Ubuntu 16.04 PC (i440FX + PIIX, 1996) (default)
...
pc-i440fx-bionic Ubuntu 18.04 PC (i440FX + PIIX, 1996) (default)
...
```

## Restart the guest

```
virsh start <your_machine>
check from host, via dumping the active xml definition
virsh dumpxml <your_machine> | xmllint --xpath "string(//domain/os/type/@machine)" -
or from the guest via dmidecode (if supported)
sudo dmidecode | grep Product -A 1
 Product Name: Standard PC (i440FX + PIIX, 1996)
 Version: pc-i440fx-bionic
```



## Further reading

- [Ubuntu Wiki](#)<sup>1022</sup>

## Containers

- [Container tools overview](#)
- [About rock images](#)
- [Docker storage, networking, and logging](#)

## Container tools in the Ubuntu space

## LXC

Linux Containers, or LXC<sup>1023</sup>

started<sup>1025</sup>

[LXC Introduction](#)<sup>1024</sup>

[LXC Getting](#)

## LXD

Linux Containers Daemon, or LXD<sup>1026</sup>

---

<sup>1022</sup> [https://wiki.ubuntu.com/QemuKVMMigration#Upgrade\\_machine\\_type](https://wiki.ubuntu.com/QemuKVMMigration#Upgrade_machine_type)

<sup>1023</sup> <https://linuxcontainers.org/>

<sup>1024</sup> <https://linuxcontainers.org/lxc/introduction/>

<sup>1025</sup> <https://linuxcontainers.org/lxc/getting-started/>

<sup>1026</sup> <https://canonical.com/lxd>

*how to get started with LXD*  
recommend this tutorial<sup>1027</sup>

chines<sup>1028</sup>

used to create virtual ma-

## Docker

*guide for sysadmins*

*Docker*

## About rock images

Ubuntu-based container image<sup>1029</sup>

OCI-compliant<sup>1030</sup>  
*Docker*

created using Rockcraft<sup>1031</sup>

uses Chisel<sup>1032</sup>

*our how-to guide*  
official

## Rockcraft documentation<sup>1033</sup>

<sup>1027</sup> [https://canonical.com/lxd/docs/latest/tutorial/first\\_steps/](https://canonical.com/lxd/docs/latest/tutorial/first_steps/)

<sup>1028</sup> [https://canonical.com/lxd/docs/latest/howto/instances\\_create/#create-a-virtual-machine](https://canonical.com/lxd/docs/latest/howto/instances_create/#create-a-virtual-machine)

<sup>1029</sup> <https://documentation.ubuntu.com/rockcraft/latest/explanation/rocks/>

<sup>1030</sup> <https://opencontainers.org/>

<sup>1031</sup> <https://documentation.ubuntu.com/rockcraft/latest/explanation/rockcraft/>

<sup>1032</sup> <https://documentation.ubuntu.com/rockcraft/latest/explanation/chisel/>

<sup>1033</sup> <https://documentation.ubuntu.com/rockcraft/latest/>

## Docker storage, networking, and logging

### Storage

- 
- 

### Volumes

- 
- 
- 
-

- 

- 

- 

- 

- 

- 

- 

- 

- 

- 

## Bind mounts

- 

- 

- 

- 

- 

-

- 
- 
- 
- 
- 

## Tmpfs

- 
- 
- 
- 
- mentation<sup>1034</sup>
- 

Docker docu-

## Storage drivers

*filesystem*

how layers, images and containers work<sup>1035</sup>

- 

<sup>1034</sup> <https://docs.docker.com/engine/storage/tmpfs/#options-for---tmpfs>

<sup>1035</sup> <https://docs.docker.com/engine/storage/drivers/#images-and-layers>

- 

*plication*

- 

- 

- 

mentation<sup>1036</sup>

the storage driver docu-  
*JSON*

## Networking

- 

- 

- 

- 

- 

- 

## Logging

- 

---

<sup>1036</sup> <https://docs.docker.com/engine/storage/drivers/>

- 
- **journal**
- 
- Logentries<sup>1037</sup>
- **gelf** *Graylog* *Logstash*<sup>1039</sup> *Fluentd*<sup>1040</sup>
- Amazon CloudWatch Logs<sup>1041</sup>
- 
- **fluentd** *Fluentd*<sup>1042</sup>
- **gcplogs** *Google Cloud Logging*<sup>1043</sup>
- *HTTP Event Collector*<sup>1044</sup>

## Resources

- *Docker for system admins*

## Other virtualisation tools

- *About OpenStack*

## About OpenStack

*QEMU/KVM*

*libvirt*

<sup>1037</sup> <https://www.rapid7.com/products/siem/>

<sup>1038</sup> <https://graylog.org/>

<sup>1039</sup> <https://www.elastic.co/logstash>

<sup>1040</sup> <https://www.fluentd.org>

<sup>1041</sup> <https://aws.amazon.com/cloudwatch/features/>

<sup>1042</sup> <https://www.fluentd.org>

<sup>1043</sup> <https://docs.cloud.google.com/logging/docs>

<sup>1044</sup> <https://dev.splunk.com/enterprise/docs/devtools/httpeventcollector/>

- 
- 
- 
- 
- 
- 

OpenStack components<sup>1045</sup>

## OpenStack installation

Canonical

OpenStack<sup>1046</sup>

- OpenStack Charms<sup>1047</sup>
- OpenStack Ansible<sup>1048</sup>
- Manual Installation<sup>1049</sup>
- DevStack<sup>1050</sup>

<sup>1045</sup> <https://www.openstack.org/software/project-navigator/openstack-components#openstack-services>

<sup>1046</sup> <https://canonical-openstack.readthedocs-hosted.com/en/latest/tutorial/get-started-with-openstack/>

<sup>1047</sup> <https://docs.openstack.org/project-deploy-guide/charm-deployment-guide/latest/>

<sup>1048</sup> <https://docs.openstack.org/project-deploy-guide/openstack-ansible/latest/>

<sup>1049</sup> <https://docs.openstack.org/install-guide/>

<sup>1050</sup> <https://docs.openstack.org/devstack/latest/>



## See also

- *Virtualisation*
- *Containers*

*clouds section*

## Clouds

## Cloud-init

## Introduction to cloud-init

Cloud-init<sup>1051</sup>

## How does it work?

*hostname*

---

<sup>1051</sup> <https://docs.cloud-init.io/en/latest/>

## Related products

the data-

source<sup>1052</sup>

- 
- 
- 
- 

check out their tutorials<sup>1053</sup>

## Cloud images

- *Cloud images overview*

Public Cloud documentation<sup>1054</sup>

- Amazon EC2<sup>1055</sup> Google Compute Engine (GCE)<sup>1056</sup> Microsoft Azure<sup>1057</sup>

## Cloud images

<https://cloud-images.ubuntu.com>

public-cloud documentation.<sup>1058</sup>

---

<sup>1052</sup> <https://docs.cloud-init.io/en/latest/reference/datasources.html>

<sup>1053</sup> <https://docs.cloud-init.io/en/latest/tutorial/index.html>

<sup>1054</sup> <https://documentation.ubuntu.com/public-cloud/>

<sup>1055</sup> <https://documentation.ubuntu.com/aws/aws-how-to/instances/find-ubuntu-images/>

<sup>1056</sup> <https://documentation.ubuntu.com/gcp/google-how-to/gce/find-ubuntu-images/>

<sup>1057</sup> <https://documentation.ubuntu.com/azure/azure-how-to/instances/find-ubuntu-images/>

<sup>1058</sup> <https://documentation.ubuntu.com/public-cloud/>

## Public clouds

### Compute offerings

- Amazon Elastic Compute Cloud (EC2)<sup>1059</sup>
- Google Compute Engine (GCE)<sup>1060</sup>
- IBM Cloud<sup>1061</sup>
- Microsoft Azure<sup>1062</sup>
- Oracle Cloud<sup>1063</sup>

### Container offerings

- Amazon Elastic Kubernetes Service (EKS)<sup>1064</sup>
- <https://cloud-images.ubuntu.com/docs/aws/eks/>  
release of GKE<sup>1065</sup>

## Private clouds

[cloud-images.ubuntu.com](https://cloud-images.ubuntu.com)<sup>1066</sup>

- 
- 
- 
- 
- 

---

<sup>1059</sup> <https://documentation.ubuntu.com/aws/aws-how-to/instances/find-ubuntu-images/>

<sup>1060</sup> <https://documentation.ubuntu.com/gcp/google-how-to/gce/find-ubuntu-images/>

<sup>1061</sup> <https://canonical-ibm.readthedocs-hosted.com/ibm-how-to/find-ubuntu-images/>

<sup>1062</sup> <https://documentation.ubuntu.com/azure/azure-how-to/instances/find-ubuntu-images/>

<sup>1063</sup> <https://documentation.ubuntu.com/oracle/oracle-how-to/find-ubuntu-images/>

<sup>1064</sup> <https://cloud-images.ubuntu.com/docs/aws/eks/>

<sup>1065</sup> <https://docs.cloud.google.com/kubernetes-engine/docs/release-schedule#schedule-for-release-channels>

<sup>1066</sup> <https://cloud-images.ubuntu.com>

## Release support

lifecycle of an Ubuntu release<sup>1067</sup>

*High Availability*

## High Availability

### Introduction

- *Introduction to High Availability*

### Key concepts

- *Pacemaker resource agents*
- *Pacemaker fence agents*

### Introduction to High Availability

from Wikipedia:<sup>1068</sup>

---

<sup>1067</sup> <https://ubuntu.com/about/release-cycle>

<sup>1068</sup> [https://en.wikipedia.org/wiki/High-availability\\_cluster](https://en.wikipedia.org/wiki/High-availability_cluster)

## High Availability clusters

**High-availability clusters** (also known as **HA clusters**, **failover clusters** or **Metroclusters Active/Active**) are groups of **computers**<sup>1069</sup> that support **server**<sup>1070</sup> **applications**<sup>1071</sup> that can be reliably utilised with a **minimum amount of down-time**<sup>1072</sup>. They operate by using **high availability software**<sup>1073</sup> to harness **redundant**<sup>1074</sup> computers in groups or **clusters**<sup>1075</sup> that provide continued service when system components fail. Without clustering, if a server running a particular application crashes, the application will be unavailable until the crashed server is fixed. HA clustering remedies this situation by detecting hardware/software faults, and immediately restarting the application on another system without requiring administrative intervention, a process known as **failover**<sup>1076</sup>. As part of this process, clustering software may configure the node before starting the application on it. For example, appropriate file systems may need to be imported and mounted, network hardware may have to be configured, and some supporting applications may need to be running as well.

HA clusters are often used for critical **databases**<sup>1077</sup>, file sharing on a network, business applications, and customer services such as **electronic commerce**<sup>1078</sup> **websites**<sup>1079</sup>.

<sup>1069</sup> <https://en.wikipedia.org/wiki/Computer>

<sup>1070</sup> [https://en.wikipedia.org/wiki/Server\\_\(computing\)](https://en.wikipedia.org/wiki/Server_(computing))

<sup>1071</sup> [https://en.wikipedia.org/wiki/Application\\_software](https://en.wikipedia.org/wiki/Application_software)

<sup>1072</sup> [https://en.wikipedia.org/wiki/High\\_availability](https://en.wikipedia.org/wiki/High_availability)

<sup>1073</sup> [https://en.wikipedia.org/wiki/High\\_availability\\_software](https://en.wikipedia.org/wiki/High_availability_software)

<sup>1074</sup> [https://en.wikipedia.org/wiki/Redundancy\\_\(engineering\)](https://en.wikipedia.org/wiki/Redundancy_(engineering))

<sup>1075</sup> [https://en.wikipedia.org/wiki/Computer\\_cluster](https://en.wikipedia.org/wiki/Computer_cluster)

<sup>1076</sup> <https://en.wikipedia.org/wiki/Failover>

<sup>1077</sup> [https://en.wikipedia.org/wiki/Database\\_management\\_system](https://en.wikipedia.org/wiki/Database_management_system)

<sup>1078</sup> [https://en.wikipedia.org/wiki/Electronic\\_commerce](https://en.wikipedia.org/wiki/Electronic_commerce)

<sup>1079</sup> <https://en.wikipedia.org/wiki/Websites>

## High Availability cluster heartbeat

HA cluster implementations attempt to build redundancy into a cluster to eliminate single points of failure, including multiple network connections and data storage which is redundantly connected via **storage area networks**<sup>1080</sup>.

HA clusters usually use a **heartbeat**<sup>1081</sup> private network connection which is used to monitor the health and status of each node in the cluster. One subtle but serious condition all clustering software must be able to handle is **split-brain**<sup>1082</sup>, which occurs when all of the private links go down simultaneously, but the cluster nodes are still running. If that happens, each node in the cluster may mistakenly decide that every other node has gone down and attempt to start services that other nodes are still running. Having duplicate instances of services may cause data corruption on the shared storage.

<sup>1080</sup> [https://en.wikipedia.org/wiki/Storage\\_area\\_network](https://en.wikipedia.org/wiki/Storage_area_network)

<sup>1081</sup> [https://en.wikipedia.org/wiki/Heartbeat\\_\(computing\)](https://en.wikipedia.org/wiki/Heartbeat_(computing))

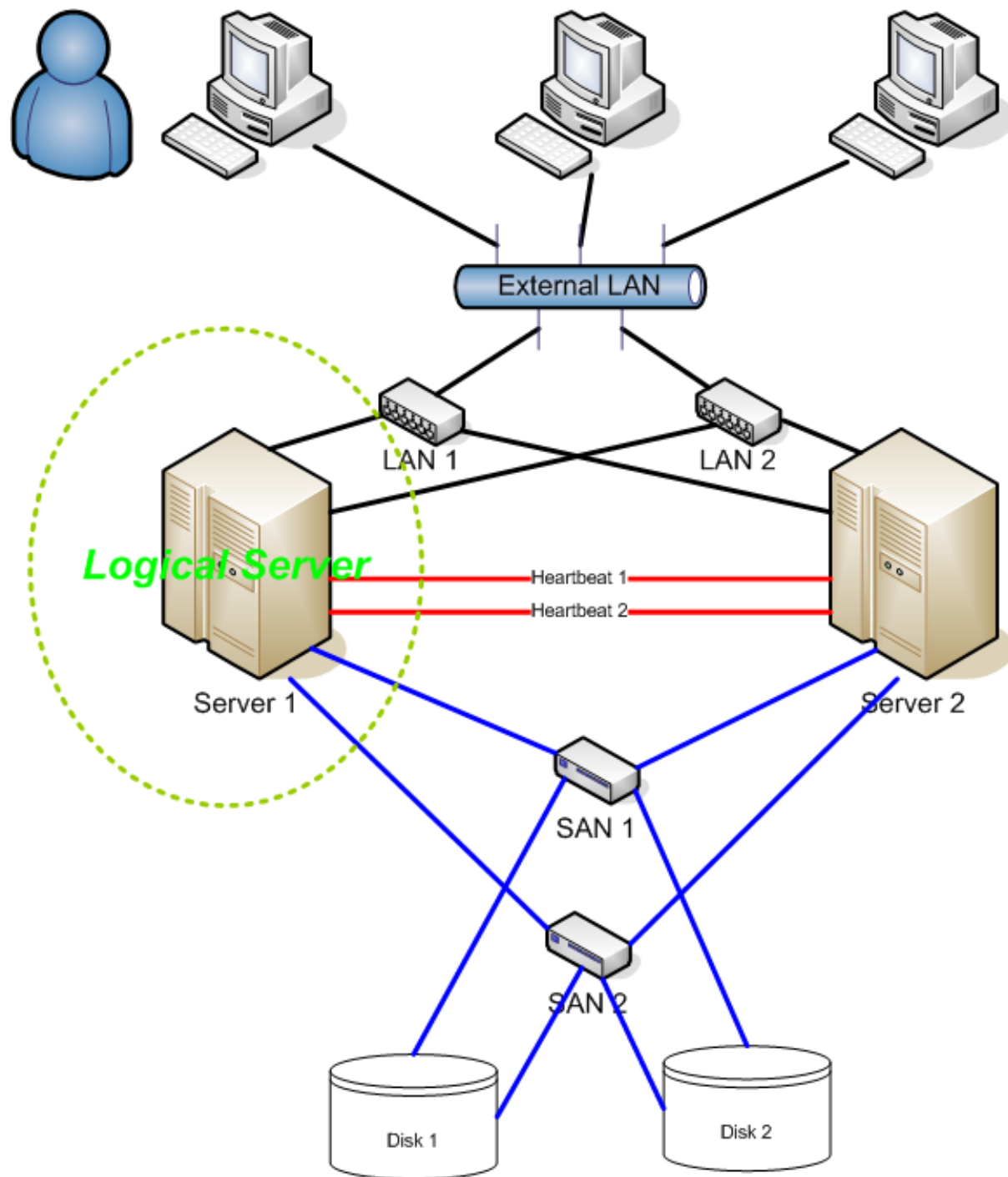
<sup>1082</sup> [https://en.wikipedia.org/wiki/Split-brain\\_\(computing\)](https://en.wikipedia.org/wiki/Split-brain_(computing))

## High Availability cluster quorum

HA clusters often also use [quorum](https://en.wikipedia.org/wiki/Quorum_(distributed_computing))<sup>1083</sup> witness storage (local or cloud) to avoid this scenario. A witness device cannot be shared between two halves of a split cluster, so in the event that all cluster members cannot communicate with each other (e.g., failed heartbeat), if a member cannot access the witness, it cannot become active.

<sup>1083</sup> [https://en.wikipedia.org/wiki/Quorum\\_\(distributed\\_computing\)](https://en.wikipedia.org/wiki/Quorum_(distributed_computing))

### Example of HA cluster quorum



## Fencing

- 
- 
- *GFS2*

## Linux High Availability projects

## Main Ubuntu HA packages



| Package         | URL                                                                                      |
|-----------------|------------------------------------------------------------------------------------------|
| libqb           | <a href="#">libqb<sup>1084</sup></a>   <a href="#">Upstream<sup>1085</sup></a>           |
| kronosnet       | <a href="#">kronosnet<sup>1086</sup></a>   <a href="#">Upstream<sup>1087</sup></a>       |
| corosync        | <a href="#">corosync<sup>1088</sup></a>   <a href="#">Upstream<sup>1089</sup></a>        |
| pacemaker       | <a href="#">pacemaker<sup>1090</sup></a>   <a href="#">Upstream<sup>1091</sup></a>       |
| resource-agents | <a href="#">resource-agents<sup>1092</sup></a>   <a href="#">Upstream<sup>1093</sup></a> |
| fence-agents    | <a href="#">fence-agents<sup>1094</sup></a>   <a href="#">Upstream<sup>1095</sup></a>    |
| crmsh           | <a href="#">crmsh<sup>1096</sup></a>   <a href="#">Upstream<sup>1097</sup></a>           |
| pcs*            | <a href="#">pcs<sup>1098</sup></a>   <a href="#">Upstream<sup>1099</sup></a>             |
| cluster-glue    | <a href="#">cluster-glue<sup>1100</sup></a>   <a href="#">Upstream<sup>1101</sup></a>    |
| drbd-utils      | <a href="#">drbd-utils<sup>1102</sup></a>   <a href="#">Upstream<sup>1103</sup></a>      |
| dlm             | <a href="#">dlm<sup>1104</sup></a>   <a href="#">Upstream<sup>1105</sup></a>             |
| gfs2-utils      | <a href="#">gfs2-utils<sup>1106</sup></a>   <a href="#">Upstream<sup>1107</sup></a>      |
| keepalived      | <a href="#">keepalived<sup>1108</sup></a>   <a href="#">Upstream<sup>1109</sup></a>      |

•

•

<sup>1084</sup> <https://launchpad.net/ubuntu/+source/libqb>  
<sup>1085</sup> <http://clusterlabs.github.io/libqb/>  
<sup>1086</sup> <https://launchpad.net/ubuntu/+source/kronosnet>  
<sup>1087</sup> <https://kronosnet.org/>  
<sup>1088</sup> <https://launchpad.net/ubuntu/+source/corosync>  
<sup>1089</sup> <http://corosync.github.io/corosync/>  
<sup>1090</sup> <https://launchpad.net/ubuntu/+source/pacemaker>  
<sup>1091</sup> <https://clusterlabs.org/projects/pacemaker/>  
<sup>1092</sup> <https://launchpad.net/ubuntu/+source/resource-agents>  
<sup>1093</sup> <https://github.com/ClusterLabs/resource-agents>  
<sup>1094</sup> <https://launchpad.net/ubuntu/+source/fence-agents>  
<sup>1095</sup> <https://github.com/ClusterLabs/fence-agents>  
<sup>1096</sup> <https://launchpad.net/ubuntu/+source/crmsh>  
<sup>1097</sup> <https://github.com/ClusterLabs/crmsh>  
<sup>1098</sup> <https://launchpad.net/ubuntu/+source/pcs>  
<sup>1099</sup> <https://github.com/ClusterLabs/pcs/>  
<sup>1100</sup> <https://launchpad.net/ubuntu/+source/cluster-glue>  
<sup>1101</sup> <https://github.com/ClusterLabs/cluster-glue>  
<sup>1102</sup> <https://launchpad.net/ubuntu/+source/drbd-utils>  
<sup>1103</sup> <https://linbit.com/drbd/>  
<sup>1104</sup> <https://launchpad.net/ubuntu/+source/dlm>  
<sup>1105</sup> <https://pagure.io/dlm>  
<sup>1106</sup> <https://launchpad.net/ubuntu/+source/gfs2-utils>  
<sup>1107</sup> <https://pagure.io/gfs2-utils>  
<sup>1108</sup> <https://launchpad.net/ubuntu/+source/keepalived>  
<sup>1109</sup> <https://www.keepalived.org/>

## Totem Redundant Ring Protocol<sup>1110</sup>

- 

- 

- 

- 

- 

- *GNU*

- 

- 

- 

- age system<sup>1111</sup>

distributed replicated stor-

- 

- 

- *filesystem*

- 

## Linux Virtual Server (IPVS)<sup>1112</sup>

### VRRP<sup>1113</sup>

---

<sup>1110</sup> <https://discourse.ubuntu.com/t/corosync-and-redundant-rings/11627>

<sup>1111</sup> [https://en.wikipedia.org/wiki/Distributed\\_Replicated\\_Block\\_Device](https://en.wikipedia.org/wiki/Distributed_Replicated_Block_Device)

<sup>1112</sup> <http://www.linux-vs.org/>

<sup>1113</sup> <https://datatracker.ietf.org/wg/vrrp/documents/>

## Ubuntu HA community packages

| Package          | URL                                               |
|------------------|---------------------------------------------------|
| pcs*             | Ubuntu <sup>1114</sup>   Upstream <sup>1115</sup> |
| csync2           | Ubuntu <sup>1116</sup>   Upstream <sup>1117</sup> |
| corosync-qdevice | Ubuntu <sup>1118</sup>   Upstream <sup>1119</sup> |
| fence-virt       | Ubuntu <sup>1120</sup>   Upstream <sup>1121</sup> |
| sbd              | Ubuntu <sup>1122</sup>   Upstream <sup>1123</sup> |
| booth            | Ubuntu <sup>1124</sup>   Upstream <sup>1125</sup> |

- 
- 

### Note:

**pcs** was added to the main repository in Ubuntu Lunar Lobster (23.04).

- <sup>1114</sup> <https://launchpad.net/ubuntu/+source/libqb>
- <sup>1115</sup> <https://github.com/ClusterLabs/pcs>
- <sup>1116</sup> <https://launchpad.net/ubuntu/+source/csync2>
- <sup>1117</sup> <https://github.com/LINBIT/csync2>
- <sup>1118</sup> <https://launchpad.net/ubuntu/+source/corosync-qdevice>
- <sup>1119</sup> <https://github.com/corosync/corosync-qdevice>
- <sup>1120</sup> <https://launchpad.net/ubuntu/+source/fence-virt>
- <sup>1121</sup> <https://github.com/ClusterLabs/fence-virt>
- <sup>1122</sup> <https://launchpad.net/ubuntu/+source/sbd>
- <sup>1123</sup> <https://github.com/ClusterLabs/sbd>
- <sup>1124</sup> <https://launchpad.net/ubuntu/+source/booth>
- <sup>1125</sup> <https://github.com/ClusterLabs/booth>

## Ubuntu HA deprecated packages

| Package     | URL                                                                             |
|-------------|---------------------------------------------------------------------------------|
| ocfs2-tools | <a href="#">Ubuntu<sup>1126</sup></a>   <a href="#">Upstream<sup>1127</sup></a> |

## Ubuntu HA related packages

| Package              | URL                                                                             |
|----------------------|---------------------------------------------------------------------------------|
| multipath-tools      | <a href="#">Ubuntu<sup>1128</sup></a>   <a href="#">Upstream<sup>1129</sup></a> |
| open-iscsi           | <a href="#">Ubuntu<sup>1130</sup></a>   <a href="#">Upstream<sup>1131</sup></a> |
| sg3-utils            | <a href="#">Ubuntu<sup>1132</sup></a>   <a href="#">Upstream<sup>1133</sup></a> |
| tgt OR targetcli-fb* | <a href="#">Ubuntu<sup>1134</sup></a>   <a href="#">Upstream<sup>1135</sup></a> |
| lvm2                 | <a href="#">Ubuntu<sup>1136</sup></a>   <a href="#">Upstream<sup>1137</sup></a> |

•

*LVM*

**lvmlockd(8)**<sup>1138</sup>

<sup>1126</sup> <https://launchpad.net/ubuntu/+source/ocfs2-tools>

<sup>1127</sup> <https://github.com/markfasheh/ocfs2-tools>

<sup>1128</sup> <https://launchpad.net/ubuntu/+source/multipath-tools>

<sup>1129</sup> <https://github.com/opensvc/multipath-tools>

<sup>1130</sup> <https://launchpad.net/ubuntu/+source/open-iscsi>

<sup>1131</sup> <https://github.com/open-iscsi/open-iscsi>

<sup>1132</sup> <https://launchpad.net/ubuntu/+source/sg3-utils>

<sup>1133</sup> [http://sg.danny.cz/sg/sg3\\_utils.html](http://sg.danny.cz/sg/sg3_utils.html)

<sup>1134</sup> <https://launchpad.net/ubuntu/+source/tgt>

<sup>1135</sup> <https://github.com/fujita/tgt>

<sup>1136</sup> <https://launchpad.net/ubuntu/+source/lvm2>

<sup>1137</sup> <https://sourceware.org/lvm2/>

<sup>1138</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/lvmlockd.8.html>

- 
- 
- 

**Note:**

`targetcli - fb` (Linux LIO) will likely replace `tgt` in future Ubuntu versions.

## Upstream documentation

- - [Clusters From Scratch](#)<sup>1139</sup>
  - [Managing Pacemaker Clusters](#)<sup>1140</sup>
  - [Pacemaker Configuration Explained](#)<sup>1141</sup>
- - [Ubuntu Bionic HA in Shared Disk Environments \(Azure\)](#)<sup>1142</sup>

A very special thanks and all the credit to [ClusterLabs Project](#)<sup>1143</sup> for their detailed documentation.

---

<sup>1143</sup> <https://clusterlabs.org/>

## Pacemaker resource agents

Resource agents are the abstraction that allows Pacemaker to manage services it knows nothing about. They contain the logic for what to do when the cluster wishes to start, stop or check the health of a service. This particular set of agents conform to the Open Cluster Framework (OCF) specification.

---

<sup>1139</sup> [https://clusterlabs.org/projects/pacemaker/doc/3.0/Clusters\\_from\\_Scratch/html/](https://clusterlabs.org/projects/pacemaker/doc/3.0/Clusters_from_Scratch/html/)

<sup>1140</sup> [https://clusterlabs.org/projects/pacemaker/doc/3.0/Pacemaker\\_Administration/html/](https://clusterlabs.org/projects/pacemaker/doc/3.0/Pacemaker_Administration/html/)

<sup>1141</sup> [https://clusterlabs.org/projects/pacemaker/doc/3.0/Pacemaker\\_Explained/html/](https://clusterlabs.org/projects/pacemaker/doc/3.0/Pacemaker_Explained/html/)

<sup>1142</sup> <https://discourse.ubuntu.com/t/ubuntu-high-availability-corosync-pacemaker-shared-disk-environments/14874>

- 
- 
- 
- 
- 
- 

#### Note:

There are two well known tools used to manage fence agents, they are `crmsh` and `pcs`. Here we will present examples with both, since `crmsh` is the recommended and supported tool until Ubuntu 22.10 Kinetic Kudu, and `pcs` is the recommended and supported tool from Ubuntu 23.04 Lunar Lobster onwards. For more information on how to migrate from `crmsh` to `pcs` [refer to this migration guide](#) (page 827).

## IPaddr2

This Linux-specific resource manages IP alias IP addresses. It can add an IP alias, or remove one. In addition, it can implement Cluster Alias IP functionality if invoked as a clone resource.

```
$ crm configure primitive $RESOURCE_NAME ocf:heartbeat:IPaddr2 \ >
ip=$IP_ADDRESS \ > cidr_netmask=$NET_MASK \ > op monitor interval=30s
```

```
$ pcs resource create $RESOURCE_NAME ocf:heartbeat:IPaddr2 \ > ip=$IP_ADDRESS
\ > cidr_netmask=$NET_MASK \ > op monitor interval=30s
```

## iscsi

Manages a local iSCSI initiator and its connections to iSCSI targets.

```
$ crm configure primitive $RESOURCE_NAME ocf:heartbeat:iscsi \ >
target=$TARGET \ > portal=$PORTAL
```

```
$ pcs resource create $RESOURCE_NAME ocf:heartbeat:iscsi \ > target=$TARGET \
> portal=$PORTAL
```

*to its manual page*<sup>1145</sup>

## iSCSILogicalUnit

Manages iSCSI Logical Unit. An iSCSI Logical unit is a subdivision of an SCSI Target, exported via a daemon that speaks the iSCSI protocol.

```
$ crm configure primitive $RESOURCE_NAME ocf:heartbeat:iSCSILogicalUnit \ >
implementation=lio-t \ > target_iqn=$IQN_TARGET \ > path=$DEVICE \ > lun=$LUN
```

```
$ pcs resource create $RESOURCE_NAME ocf:heartbeat:iSCSILogicalUnit \ >
implementation=lio-t \ > target_iqn=$IQN_TARGET \ > path=$DEVICE \ > lun=$LUN
```

<sup>1144</sup> [https://manpages.ubuntu.com/manpages/resolute/man7/ocf\\_heartbeat\\_IPaddr2.7.html](https://manpages.ubuntu.com/manpages/resolute/man7/ocf_heartbeat_IPaddr2.7.html)

<sup>1145</sup> [https://manpages.ubuntu.com/manpages/resolute/man7/ocf\\_heartbeat\\_iscsi.7.html](https://manpages.ubuntu.com/manpages/resolute/man7/ocf_heartbeat_iscsi.7.html)

page<sup>1146</sup>

*to its manual*

## iSCSITarget

Manages iSCSI targets. An iSCSI target is a collection of SCSI Logical Units (LUs) exported via a daemon that speaks the iSCSI protocol.

```
$ crm configure primitive $RESOURCE_NAME ocf:heartbeat:iSCSITarget \ >
implementation=lio-t \ > iqn=$IQN_TARGET
```

```
$ pcs resource create $RESOURCE_NAME ocf:heartbeat:iSCSITarget \ >
implementation=lio-t \ > iqn=$IQN_TARGET
```

*to its manual page<sup>1147</sup>*

## LVM-activate

This agent manages LVM activation/deactivation work for a given volume group.

```
$ crm configure primitive $RESOURCE_NAME ocf:heartbeat:LVM-activate \ >
vgname=$VOLUME_GROUP \ > vg_access_mode=system_id
```

<sup>1146</sup> [https://manpages.ubuntu.com/manpages/resolute/man7/ocf\\_heartbeat\\_iSCSILogicalUnit.7.html](https://manpages.ubuntu.com/manpages/resolute/man7/ocf_heartbeat_iSCSILogicalUnit.7.html)

<sup>1147</sup> [https://manpages.ubuntu.com/manpages/resolute/man7/ocf\\_heartbeat\\_iSCSITarget.7.html](https://manpages.ubuntu.com/manpages/resolute/man7/ocf_heartbeat_iSCSITarget.7.html)



```
$ pcs resource create $RESOURCE_NAME ocf:heartbeat:LVM-activate \ >
vgname=$VOLUME_GROUP \ > vg_access_mode=system_id
```

*to its manual page*<sup>1148</sup>

## Systemd

```
$ crm configure primitive $RESOURCE_NAME systemd:$SERVICE_NAME
```

```
$ pcs resource create $RESOURCE_NAME systemd:$SERVICE_NAME
```

## Further reading

- [ClusterLabs website](#)<sup>1149</sup>
- [The OCF resource-agent developer's guide](#)<sup>1150</sup>

## Pacemaker fence agents

A **fence agent** (or **fencing agent**) is a **stonith**-class resource agent.

The fence agent standard provides commands (such as off and reboot) that the cluster can use to fence nodes. As with other resource agent classes, this allows a layer of abstraction so that Pacemaker doesn't need any knowledge about specific fencing technologies — that knowledge is isolated in the agent.

<sup>1148</sup> [https://manpages.ubuntu.com/manpages/resolute/man7/ocf\\_heartbeat\\_LVM-activate.7.html](https://manpages.ubuntu.com/manpages/resolute/man7/ocf_heartbeat_LVM-activate.7.html)

<sup>1149</sup> <https://clusterlabs.org/>

<sup>1150</sup> <https://github.com/ClusterLabs/resource-agents/blob/master/doc/dev-guides/ra-dev-guide.asc>

- - 
  - 
  - 
  - 
  - 
  -
- 
- 
- 
- 

#### Note:

There are two well known tools used to manage fence agents, they are `crmsh` and `pcs`. Here we present examples with both, since `crmsh` is the recommended and supported tool until Ubuntu 22.10 Kinetic Kudu, and `pcs` is the recommended and supported tool from Ubuntu 23.04 Lunar Lobster onwards. For more information on how to migrate from `crmsh` to `pcs` [refer to our migration guide](#) (page 827).

### fence\_ipmilan

`fence_ipmilan` is an I/O Fencing agent which can be used with machines controlled by IPMI. This agent calls support software `ipmitool`. WARNING! This fence agent might report success before the node is powered off. You should use `-m/method onoff` if your fence device works correctly with that option.

```
$ crm configure primitive $RESOURCE_NAME stonith:fence_ipmilan \ > ip=$IP \ >
ipport=$PORT \ > username=$USER \ > password=$PASSWD \ > lanplus=1 \ >
action=$ACTION
```

```
$ pcs stonith create $RESOURCE_NAME fence_ipmilan \ > ip=$IP \ > ipport=$PORT \ > username=$USER \ > password=$PASSWD \ > lanplus=1 \ > action=$ACTION
```

*hostname*

*to its manual page<sup>1151</sup>*

## fence\_mpath

fence\_mpath is an I/O fencing agent that uses SCSI-3 persistent reservations to control access multipath devices. Underlying devices must support SCSI-3 persistent reservations (SPC-3 or greater) as well as the “preempt-and-abort” subcommand. The fence\_mpath agent works by having a unique key for each node that has to be set in /etc/multipath.conf. Once registered, a single node will become the reservation holder by creating a “write exclusive, registrants only” reservation on the device(s). The result is that only registered nodes may write to the device(s). When a node failure occurs, the fence\_mpath agent will remove the key belonging to the failed node from the device(s). The failed node will no longer be able to write to the device(s). A manual reboot is required.

```
$ crm configure primitive $RESOURCE_NAME stonith:fence_mpath \ > pcmk_host_map="$NODE1:$NODE1_RES_KEY;$NODE2:$NODE2_RES_KEY;$NODE3:$NODE3_RES_KEY" \ > pcmk_host_argument=key \ > pcmk_monitor_action=metadata \ > pcmk_reboot_action=off \ > devices=$MPATH_DEVICE \ > meta provides=unfencing
```

```
$ pcs stonith create $RESOURCE_NAME fence_mpath \ > pcmk_host_map="$NODE1:$NODE1_RES_KEY;$NODE2:$NODE2_RES_KEY;$NODE3:$NODE3_RES_KEY" \ > pcmk_host_argument=key \ > pcmk_monitor_action=metadata \ > pcmk_reboot_action=off \ > devices=$MPATH_DEVICE \ > meta provides=unfencing
```

*its manual page<sup>1152</sup>*

<sup>1151</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/fence\\_ipmilan.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/fence_ipmilan.8.html)

<sup>1152</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/fence\\_mpath.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/fence_mpath.8.html)

## fence\_sbd

fence\_sbd is I/O Fencing agent which can be used in environments where SBD can be used (shared storage).

```
$ crm configure primitive $RESOURCE_NAME stonith:fence_sbd devices=$DEVICE
```

```
$ pcs stonith create $RESOURCE_NAME fence_sbd devices=$DEVICE
```

*to its manual page<sup>1153</sup>*

## fence\_scsi

fence\_scsi is an I/O fencing agent that uses SCSI-3 persistent reservations to control access to shared storage devices. These devices must support SCSI-3 persistent reservations (SPC-3 or greater) as well as the “preempt-and-abort” subcommand. The fence\_scsi agent works by having each node in the cluster register a unique key with the SCSI device(s). Reservation key is generated from “node id” (default) or from “node name hash” (RECOMMENDED) by adjusting “key\_value” option. Using hash is recommended to prevent issues when removing nodes from cluster without full cluster restart. Once registered, a single node will become the reservation holder by creating a “write exclusive, registrants only” reservation on the device(s). The result is that only registered nodes may write to the device(s). When a node failure occurs, the fence\_scsi agent will remove the key belonging to the failed node from the device(s). The failed node will no longer be able to write to the device(s). A manual reboot is required.

```
$ crm configure primitive $RESOURCE_NAME stonith:fence_scsi \ >
pcmk_host_list="$NODE1 $NODE2 $NODE3" \ > devices=$SCSI_DEVICE \ > meta
provides=unfencing
```

```
$ pcs stonith create $RESOURCE_NAME fence_scsi \ > pcmk_host_list="$NODE1
$NODE2 $NODE3" \ > devices=$SCSI_DEVICE \ > meta provides=unfencing
```

<sup>1153</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/fence\\_sbd.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/fence_sbd.8.html)

*to its manual page<sup>1154</sup>*

## fence\_virsh

fence\_virsh is an I/O Fencing agent which can be used with the virtual machines managed by libvirt. It logs via SSH to a dom0 and there run virsh command, which does all work. By default, virsh needs root account to do properly work. So you must allow SSH login in your sshd\_config.

```
$ crm configure primitive $RESOURCE_NAME stonith:fence_virsh \ >
ip=$HOST_IP_ADDRESS \ > login=$HOST_USER \ > identity_file=$SSH_KEY \ >
plug=$NODE \ > ssh=true \ > use_sudo=true
```

```
$ pcs stonith create $RESOURCE_NAME fence_virsh \ > ip=$HOST_IP_ADDRESS \ >
login=$HOST_USER \ > identity_file=$SSH_KEY \ > plug=$NODE \ > ssh=true \ >
use_sudo=true
```

*to its manual page<sup>1155</sup>*

```
$ crm configure location fence-$NODE-location $RESOURCE_NAME -inf: $NODE
```

```
$ pcs constraint location $RESOURCE_NAME avoids $NODE
```

## Further reading

- [ClusterLabs website<sup>1156</sup>](#)
- [Fence agents API documentation<sup>1157</sup>](#)

<sup>1154</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/fence\\_scsi.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/fence_scsi.8.html)

<sup>1155</sup> [https://manpages.ubuntu.com/manpages/resolute/man8/fence\\_virsh.8.html](https://manpages.ubuntu.com/manpages/resolute/man8/fence_virsh.8.html)

<sup>1156</sup> <https://clusterlabs.org/>

<sup>1157</sup> <https://github.com/ClusterLabs/fence-agents/blob/master/doc/FenceAgentAPI.md>

## See also

- [Set up a Distributed Replicated Block Device \(DRBD\)](#)
- [High availability](#)

## Our system tuning

## Performance

### Note:

Disclaimer - To tune you need to know your system and workload

Almost all tunable parameters can be good for one and bad for another type of workload or environment. If the system could do it for you, it probably would be the default setting already.

Not even the goal of tuning is the same for everyone; do you want to improve latency, throughput, thermal or work-unit-per-power? But those default settings generally have to aim for a good compromise on all of these aspects, no matter what you will do with your system.

Therefore you have to know your workload, your system and the important metrics of what you want to achieve. The more you know, the more you'll be able to improve your system it to suit your needs by:

- Identifying bottlenecks as unblocking them usually has the biggest impact for your workload
- Identifying where your needs are different from the generic assumptions to change related tunables
- Identifying architectural mismatches between the solution and your needs to allow adapting

- [PGO](#)
- [hwloc and lstopo](#)
- [CPU Power states](#)
- [cpupower](#)
- [with TuneD](#)

## Profile-Guided Optimization

Profile-guided optimization<sup>1158</sup>

### Caveats

- 

- 

- 

### perf and AutoFDO

GCC<sup>1159</sup>

autofdo<sup>1160</sup>

---

<sup>1158</sup> [https://en.wikipedia.org/wiki/Profile-guided\\_optimization](https://en.wikipedia.org/wiki/Profile-guided_optimization)

<sup>1159</sup> <https://gcc.gnu.org>

<sup>1160</sup> <https://github.com/google/autofdo>

- 
- 
- 

*AMD*

for Ubuntu Noble<sup>1161</sup>

upstream repository<sup>1162</sup>

**A simple PGO scenario: openssl speed**

**Running OpenSSL tests**

---

<sup>1161</sup> <https://launchpad.net/~sergiodj/+archive/ubuntu/autofdo>

<sup>1162</sup> <https://github.com/google/autofdo>



```
$ openssl speed -seconds 60 -evp md5 sha512 rsa2048 aes-256-cbc
```

```
The 'numbers' are in 1000s of bytes per second processed.
type 16 bytes 64 bytes 256 bytes 1024 bytes 8192 bytes 16384
bytes
sha512 91713.74k 366632.96k 684349.30k 1060512.22k 1260359.00k
1273277.92k
aes-256-cbc 1266018.36k 1437870.33k 1442743.14k 1449933.84k 1453336.84k
1453484.99k
md5 140978.58k 367562.53k 714094.14k 955267.21k 1060969.81k
1068296.87k
 sign verify sign/s verify/s
rsa 2048 bits 0.000173s 0.000012s 5776.8 86791.9
```

## Building OpenSSL for profiling

```
$ file /usr/bin/openssl
```

```
/usr/bin/openssl: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV),
dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=42a5ac797b6981bd45a9ece3f67646edded56bdb, for GNU/Linux 3.2.0, with
debug_info, not stripped
```

```
$ eu-readelf -S /usr/bin/openssl
```

```
[...]
Section Headers:
[Nr] Name Type Addr Off Size ES Flags
Lk Inf Al
[...]
[31] .debug_abbrev PROGBITS 0000000000000000 001bcbf3 000110c4 0
0 0 1
[...]
```

## Using perf to obtain profile data

```
$ echo -1 > /proc/sys/kernel/perf_event_paranoid
```

```
$ sudo perf record \> -e br_inst_retired.near_taken \> --branch-any \> --all-cpus \> --output openssl-nonpgo.perfdata \> -- openssl speed -mr -seconds 60 -evp md5 sha512 rsa2048 aes-256-cbc
```

```
$ sudo perf record \> -e 'cpu/event=0xc4,umask=0x0,name=ex_ret_brn_tkn/' \> --branch-any \> --all-cpus \> --output openssl-nonpgo.perfdata \> -- openssl speed -mr -seconds 60 -evp md5 sha512 rsa2048 aes-256-cbc
```

```
$ sudo chown ubuntu:ubuntu openssl-nonpgo.perfdata
```

## Converting perfdata to gcov with autotfdo

```
$ create_gcov \> --binary /usr/bin/openssl \> --gcov openssl-nonpgo.gcov \> --profile openssl-nonpgo.perfdata \> --gcov_version 2 \> -use_lbr false
```

```
[WARNING:[...]/perf_reader.cc:1322] Skipping 200 bytes of metadata: HEADER_CPU_
TOPOLOGY
[WARNING:[...]/perf_reader.cc:1069] Skipping unsupported event PERF_RECORD_ID_INDEX
[WARNING:[...]/perf_reader.cc:1069] Skipping unsupported event PERF_RECORD_EVENT_
UPDATE
[WARNING:[...]/perf_reader.cc:1069] Skipping unsupported event PERF_RECORD_CPU_MAP
[WARNING:[...]/perf_reader.cc:1069] Skipping unsupported event UNKNOWN_EVENT_17
[WARNING:[...]/perf_reader.cc:1069] Skipping unsupported event UNKNOWN_EVENT_18
[...] many more lines [...]
```

```
$ dump_gcov openssl-nonpgo.gcov

cpu_get_tb_cpu_state total:320843603 head:8557544
2: 8442523
7: 8442518
7.1: 8442518
7.4: 9357851
8: 9357851
10: 9357854
19: 0
21: 0
[...] many more lines [...]
```

## Rebuilding OpenSSL with PGO

```
-fauto-profile=/path/to/openssl-nonpgo.gcov
```

## Our results

- 

- 

- 

- [blog post](#)<sup>1163</sup>

## Alternative scenario: profiling the program using other approaches

[profiled QEMU internally](#)<sup>1164</sup>

---

<sup>1163</sup> <https://ubuntu.com/blog/profile-guided-optimization-a-case-study>

<sup>1164</sup> <https://ubuntu.com/blog/profile-guided-optimization-a-case-study>

```
$ sudo perf record \ > -e br_inst_retired.near_taken \ > --branch-any \ >
--all-cpus \ > -p 1234 \ > --output myprogram.perfdata

-- sleep 2m
```

## Conclusion

### hwloc

#### Note:

System tuning tools are either about better understanding the system's performance, or applying such knowledge to improve it. See our common [system tuning thoughts](#) (page 1045) for some generally applicable considerations about that.

*GUI*

## hwloc-info - hierarchy and details

```
$ hwloc-info
```

```
depth 0: 1 Machine (type #0)
depth 1: 1 Package (type #1)
 depth 2: 1 L3Cache (type #6)
 depth 3: 4 L2Cache (type #5)
 depth 4: 4 L1dCache (type #4)
 depth 5: 4 L1iCache (type #9)
 depth 6: 4 Core (type #2)
 depth 7: 4 PU (type #3)
Special depth -3: 1 NUMANode (type #13)
Special depth -4: 3 Bridge (type #14)
Special depth -5: 4 PCIDev (type #15)
Special depth -6: 3 OSDev (type #16)
```

```
$ hwloc-info -p core:0
```

```
Core L#0
type = Core
full type = Core
logical index = 0
os index = 0
gp index = 2
depth = 6
sibling rank = 0
children = 1
memory children = 0
i/o children = 0
misc children = 0
cpuset = 0x00000001
complete cpuset = 0x00000001
allowed cpuset = 0x00000001
nodeset = 0x00000001
complete nodeset = 0x00000001
allowed nodeset = 0x00000001
symmetric subtree = 1
cpukind = 0
cpukind efficiency = 0
cpukind info FrequencyMaxMHz = 4000
cpukind info FrequencyBaseMHz = 1800
```

*hwloc-info(1)*<sup>1165</sup>

<sup>1165</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/hwloc-info.1.html>

## hwloc-ls - system topology in CLI

```
$ hwloc-ls

Machine (31GB total)
 Package L#0
 NUMANode L#0 (P#0 31GB)
 L3 L#0 (8192KB)
 L2 L#0 (256KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0 + PU L#0 (P#0)
 L2 L#1 (256KB) + L1d L#1 (32KB) + L1i L#1 (32KB) + Core L#1 + PU L#1 (P#1)
 L2 L#2 (256KB) + L1d L#2 (32KB) + L1i L#2 (32KB) + Core L#2 + PU L#2 (P#2)
 L2 L#3 (256KB) + L1d L#3 (32KB) + L1i L#3 (32KB) + Core L#3 + PU L#3 (P#3)
 HostBridge
 PCI 00:02.0 (VGA)
 PCIBridge
 PCI 04:00.0 (Network)
 Net "wlp4s0"
 PCIBridge
 PCI 40:00.0 (NVMExp)
 Block(Disk) "nvme0n1"
 PCI 00:1f.6 (Ethernet)
 Net "enp0s31f6"
```

```
Package L#0
 NUMANode L#0 (P#0 378GB)
 L3 L#0 (32MB)
 L2 L#0 (1024KB) + L1d L#0 (48KB) + L1i L#0 (32KB) + Core L#0
 PU L#0 (P#0)
 PU L#1 (P#128)
 ...
 L2 L#63 (1024KB) + L1d L#63 (48KB) + L1i L#63 (32KB) + Core L#63
 PU L#126 (P#63)
 PU L#127 (P#191)
 HostBridge
 PCIBridge
 PCI 12:00.0 (NVMEExp)
 Block(Disk) "nvme1n1"
 HostBridge
 PCIBridge
 PCI 51:00.0 (NVMEExp)
 Block(Disk) "nvme0n1"
 PCIBridge
 PCI 52:00.0 (Ethernet)
 Net "enp82s0"
 PCIBridge
 PCI 54:00.0 (VGA)
Package L#1
 NUMANode L#1 (P#1 378GB)
 L3 L#8 (32MB)
 L2 L#64 (1024KB) + L1d L#64 (48KB) + L1i L#64 (32KB) + Core L#64
 PU L#128 (P#64)
 PU L#129 (P#192)
 ...
 L2 L#127 (1024KB) + L1d L#127 (48KB) + L1i L#127 (32KB) + Core L#127
 PU L#254 (P#127)
 PU L#255 (P#255)
 HostBridge
 PCIBridge
 2 x { PCI f2:00.0-1 (SATA) }
 Misc(MemoryModule)
```

## hwloc-ls - system topology in GUI

```
$ hwloc-ls
```



#### Keyboard shortcuts:

##### Zooming, scrolling and closing:

```

Zoom-in or out + -
Reset scale to default 1
Try to fit scale to window F
Resize window to the drawing r
Toggle auto-resizing of the window .. R
Scroll vertically Up Down PageUp PageDown
Scroll horizontally Left Right Ctrl+PageUp/Down
Scroll to the top-left corner Home
Scroll to the bottom-right corner ... End
Refresh the topology F5
Show this help h H ?
Exit q Q Esc

```

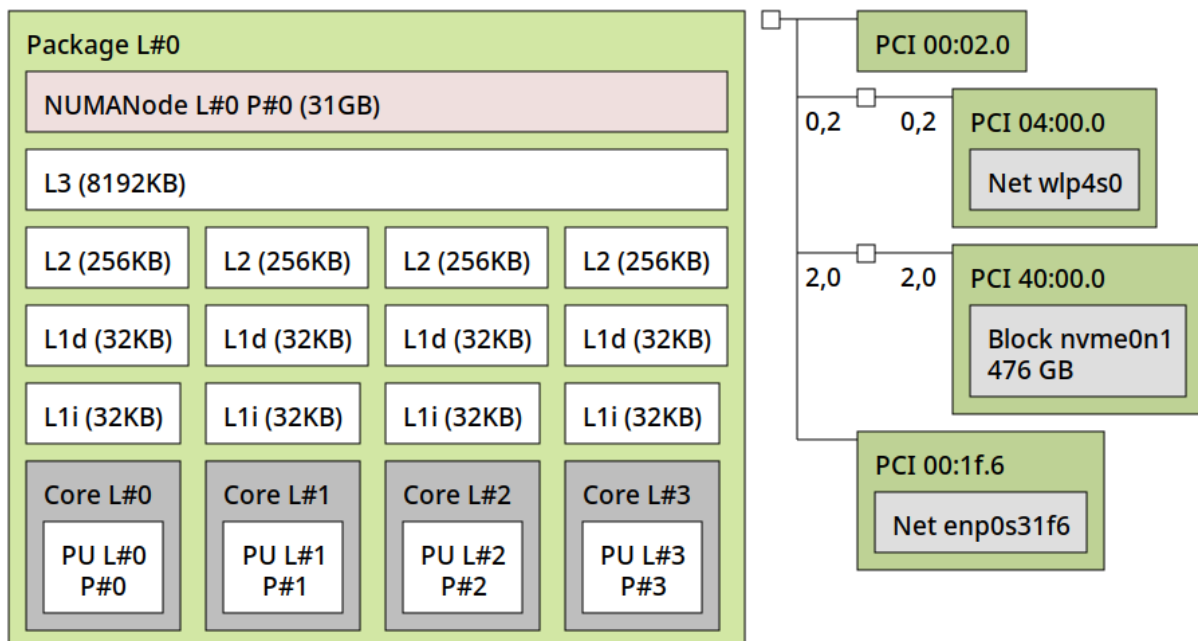
##### Configuration tweaks:

```

Toggle factorizing or collapsing f
Switch display mode for indexes i
Toggle displaying of object text t
Toggle displaying of obj attributes . a
Toggle displaying of CPU kinds k
Toggle color for disallowed objects . d
Toggle color for binding objects b
Toggle displaying of legend lines ... l
Export to file with current config .. E

```

#### Machine (31GB total)



## Further reading

- `lstopo(1)`<sup>1166</sup>
- `hwloc(7)`<sup>1167</sup>
- `hwloc-diff(1)`<sup>1168</sup>

## CPU Power states

### Note:

System tuning tools are either about better understanding the system's performance, or applying such knowledge to improve it. See our common *system tuning thoughts* (page 1045) for some generally applicable considerations about that.

Power Management<sup>1169</sup>

*The*

*cpupower tool*

---

<sup>1166</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/lstopo.1.html>

<sup>1167</sup> <https://manpages.ubuntu.com/manpages/resolute/man7/hwloc.7.html>

<sup>1168</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/hwloc-diff.1.html>

<sup>1169</sup> <https://docs.kernel.org/admin-guide/pm/working-state.html>

## CPU governors

| Governor     | Design philosophy                                                                                                                                                                                      |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ondemand     | This sets the CPU frequency depending on the current system load. This behavior is usually a good balance between the more extreme options.                                                            |
| conservative | Similar to ondemand, but adapting CPU speed more gracefully rather than jumping to max speed the moment there is any load on the CPU. This behavior is more suitable in a battery-powered environment. |
| performance  | This sets the CPU statically to the highest frequency. This behavior is best to optimize for speed and latency, but might waste power if being under-used.                                             |
| powersave    | Sets the CPU statically to the lowest frequency, essentially locking it to P2. This behavior is suitable to save power without compromises.                                                            |
| userspace    | Allows a user-space program to control the CPU frequency.                                                                                                                                              |

Linux Governors Documentation<sup>1170</sup>

## Setting CPU Governors

```
$ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

- 
- 

```
$ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors
```

<sup>1170</sup> <https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt>

```
$ echo performance | sudo tee
/sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

## Energy Performance Preference

### Caveats

- 
- 
- 
- *BIOS*

### EPP values on Ubuntu

```
$ cat /sys/devices/system/cpu/cpu0/cpufreq/energy_performance_preference
```

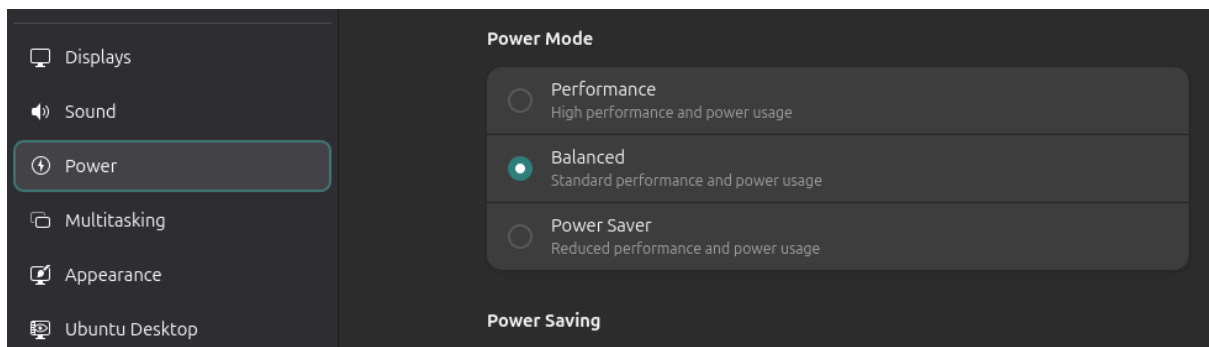
- 
- 
- 
- 
-

```
$ cat
/sys/devices/system/cpu/cpu0/cpufreq/energy_performance_available_preferences

default performance balance_performance balance_power power
```

## Setting EPP values

```
$ echo performance | sudo tee
/sys/devices/system/cpu/cpu0/cpufreq/energy_performance_preference
```



| Power Mode setting | EPP profile         |
|--------------------|---------------------|
| Performance        | performance         |
| Balanced           | balance_performance |
| Power Saver        | power               |

## The cpupower tool

### Note:

System tuning tools are either about better understanding the system's performance, or applying such knowledge to improve it. See our common [system tuning thoughts](#) (page 1045) for some generally applicable considerations about that.

### *CPU Power states*

## Monitor CPU frequency

```
$ sudo cpupower monitor -l

Monitor "Nehalem" (4 states) - Might overflow after 922000000 s
C3 [C] -> Processor Core C3
C6 [C] -> Processor Core C6
PC3 [P] -> Processor Package C3
PC6 [P] -> Processor Package C6
Monitor "Mperf" (3 states) - Might overflow after 922000000 s
C0 [T] -> Processor Core not idle
Cx [T] -> Processor Core in an idle state
Freq [T] -> Average Frequency (including boost) in MHz
Monitor "RAPL" (4 states) - Might overflow after 8640000 s
pack [M] ->
dram [M] ->
core [M] ->
unco [M] ->
Monitor "Idle_Stats" (9 states) - Might overflow after 4294967295 s
POLL [T] -> CPUIDLE CORE POLL IDLE
C1 [T] -> MWAIT 0x00
C1E [T] -> MWAIT 0x01
C3 [T] -> MWAIT 0x10
C6 [T] -> MWAIT 0x20
C7s [T] -> MWAIT 0x33
C8 [T] -> MWAIT 0x40
C9 [T] -> MWAIT 0x50
C10 [T] -> MWAIT 0x60
```

- 
-

- 
- 
- 
- 
- 
- 

```
$ sudo cpupower monitor -i 60 -m Mperf,RAPL
```

|     | Mperf |       |      | RAPL      |           |           |          |
|-----|-------|-------|------|-----------|-----------|-----------|----------|
| CPU | C0    | Cx    | Freq | pack      | dram      | core      | unco     |
| 0   | 61,83 | 38,17 | 1850 | 616950936 | 145911797 | 375373063 | 71556823 |
| 1   | 62,03 | 37,97 | 1848 | 616950936 | 145911797 | 375373063 | 71556823 |
| 2   | 65,51 | 34,49 | 1852 | 616950936 | 145911797 | 375373063 | 71556823 |
| 3   | 62,04 | 37,96 | 1852 | 616950936 | 145911797 | 375373063 | 71556823 |

Get details about the boundaries for the CPU frequency

```
$ cpupower frequency-info
```

```
analyzing CPU 8:
 driver: amd-pstate-epp
 CPUs which run at the same hardware frequency: 8
 CPUs which need to have their frequency coordinated by software: 8
 energy performance preference: balance_performance
 hardware limits: 400 MHz - 5.14 GHz
 available cpufreq governors: performance powersave
 current policy: frequency should be within 1.10 GHz and 5.14 GHz.
 The governor "powersave" may decide which speed to use
 within this range.
 current CPU frequency: 2.63 GHz (asserted by call to kernel)
 boost state support:
 Supported: yes
 Active: yes
 amd-pstate limits:
 Highest Performance: 196. Maximum Frequency: 5.14 GHz.
 Nominal Performance: 145. Nominal Frequency: 3.80 GHz.
 Lowest Non-linear Performance: 42. Lowest Non-linear Frequency: 1.10 GHz.
 Lowest Performance: 16. Lowest Frequency: 400 MHz.
 Preferred Core Support: 1. Preferred Core Ranking: 220.
```

## Get details about the idle states

Idle states<sup>1171</sup>

```
$ cpupower idle-info
```

<sup>1171</sup> <https://docs.kernel.org/admin-guide/pm/cpuidle.html>



```
CPUidle driver: intel_idle
CPUidle governor: menu
analyzing CPU 0:

Number of idle states: 9
Available idle states: POLL C1 C1E C3 C6 C7s C8 C9 C10
POLL:
Flags/Description: CPUIDLE CORE POLL IDLE
Latency: 0
Usage: 26053898
Duration: 695768311
C1:
Flags/Description: MWAIT 0x00
Latency: 2
Usage: 263751626
Duration: 21296361635
C1E:
Flags/Description: MWAIT 0x01
Latency: 10
Usage: 1071864698
Duration: 122465703132
C3:
Flags/Description: MWAIT 0x10
Latency: 70
Usage: 941753727
Duration: 117177626397
C6:
Flags/Description: MWAIT 0x20
Latency: 85
Usage: 2580936435
Duration: 1258804567087
C7s:
Flags/Description: MWAIT 0x33
Latency: 124
Usage: 2946723
Duration: 1783856599
C8:
Flags/Description: MWAIT 0x40
Latency: 200
Usage: 1580297534
Duration: 1234136981613
C9:
Flags/Description: MWAIT 0x50
Latency: 480
Usage: 2015405
Duration: 3198208930
C10:
Flags/Description: MWAIT 0x60
Latency: 890
Usage: 511786893
Duration: 1546264384800
```

```
$ sudo cpupower monitor
```

```

 | Nehalem || Mperf || RAPL
 || Idle_Stats
CPU| C3 | C6 | PC3 | PC6 || C0 | Cx | Freq || pack | dram | core | unco
 || POLL | C1 | C1E | C3 | C6 | C7s | C8 | C9 | C10
 0| 2,99| 11,92| 0,00| 0,00|| 70,98| 29,02|
1991||13733058|2706597|7438396|3080986|| 0,05| 1,84| 5,01| 3,87| 14,05| 0,06|
3,81| 0,00| 0,04
 1| 3,58| 14,84| 0,00| 0,00|| 67,65| 32,35|
1991||13733058|2706597|7438396|3080986|| 0,07| 1,87| 5,42| 4,46| 17,21| 0,36|
2,73| 0,00| 0,00
 2| 3,99| 7,15| 0,00| 0,00|| 73,25| 26,75|
1990||13733058|2706597|7438396|3080986|| 0,09| 1,95| 8,76| 5,20| 9,44| 0,01|
1,12| 0,04| 0,00
 3| 3,86| 13,68| 0,00| 0,00|| 68,40| 31,60|
1990||13733058|2706597|7438396|3080986|| 0,03| 2,52| 6,35| 4,92| 15,97| 0,00|
1,52| 0,00| 0,00

```

What should I do with all of this?

- 
- 

Does it match what you expect?

- 
- 
-

## Control the CPU governors and CPU frequency

```
$ cpupower frequency-set -g performance
```

```
$ cpupower -c 0-7 frequency-set -g conservative
```

## Powertop

```
$ sudo powertop
```

- 
- 
- 
- 
- 
- 

## Further reading

- [Intel Frequency scaling](#)<sup>1172</sup>
- [Idle states](#)<sup>1173</sup>
- [C-states](#)<sup>1174</sup>

<sup>1172</sup> [https://docs.kernel.org/admin-guide/pm/intel\\_uncore\\_frequency\\_scaling.html](https://docs.kernel.org/admin-guide/pm/intel_uncore_frequency_scaling.html)

<sup>1173</sup> <https://docs.kernel.org/admin-guide/pm/cpuidle.html>

<sup>1174</sup> [https://docs.kernel.org/admin-guide/pm/intel\\_idle.html](https://docs.kernel.org/admin-guide/pm/intel_idle.html)

## TuneD

### Note:

System tuning tools are either about better understanding the system's performance, or applying such knowledge to improve it. See our common *system tuning thoughts* (page 1045) for some generally applicable considerations about that.

## TuneD<sup>1175</sup>

- 
- 
- 

## Static vs. dynamic tuning

- 
- 

---

<sup>1175</sup> <https://tuned-project.org/>

## Profiles

| Environment      | Default profile        |
|------------------|------------------------|
| Compute nodes    | throughput-performance |
| Virtual Machines | virtual-guest          |
| Containers       | default                |
| Other cases      | balanced               |

## Anatomy of a profile

### Note:

In TuneD version 2.23 and thereby Ubuntu 24.04 Noble (and earlier), the location of the profiles was different. An upgrade would have migrated any custom profiles, however for reference the paths changes were:

- Predefined profiles:  
/usr/lib/tuned/<profile-name> -> /usr/lib/tuned/profiles/<profile-name>
- User defined profiles:  
/etc/tuned/<profile-name> -> /etc/tuned/profiles/<profile-name>

```
[main]
include=PROFILE # if inheriting from another profile
summary=A short summary
description=A short description

[plugin instance]
type=<TYPE like cpu, scheduler, ...>
replace=<REPLACE = true | false>
enabled=<ENABLED = true | false>
devices=<DEVICES like /dev/sda, /dev/sdb>

[another plugin instance]
type=<TYPE like cpu, scheduler, ...>
replace=<REPLACE = true | false>
enabled=<ENABLED = true | false>
```

(continues on next page)

(continued from previous page)

```
devices=<DEVICES like /dev/sda, /dev/sdb>
... other plugin-specific options ...

...
```

- 
- 
- 
- 
- 

*tuned.conf(5)*<sup>1176</sup>

```
[fast_storage]
type=disk
devices=nvme0n*
... options for these devices

[slow_storage]
type=disk
devices=sda, sdb
... options for these devices
```

## Available profiles and plugins

```
$ tuned-adm list profiles
```

<sup>1176</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/tuned.conf.5.html>

Available profiles:

- accelerator-performance      - Throughput performance based tuning with disabled higher latency STOP states
- atomic-guest                      - Optimize virtual guests based on the Atomic variant
- atomic-host                       - Optimize bare metal systems running the Atomic variant
- (...)

- 
- 
- 

```
cpu
 load_threshold
 latency_low
 latency_high
 force_latency
 governor
 sampling_down_factor
 energy_perf_bias
 min_perf_pct
 max_perf_pct
 no_turbo
 pm_qos_resume_latency_us
 energy_performance_preference
 boost
```

## Customizing a profile

-

- 

- 

```
[main]
include=postgresql

[cpu]
load_threshold=0.5
latency_low=10
latency_high=10000
```

## Merging profiles

```
$ sudo tuned-adm profile cpu-partitioning intel-sst
```

```
$ sudo tuned-adm profile virtual-guest powersave
```



### Warning:

Just because tuned-adm accepted to merge two profiles doesn't mean it makes sense. There is no checking done on the resulting merged parameters, and the second profile could completely revert what the first profile adjusted.

### An example profile: hpc-compute

```
#
tuned configuration
#

[main]
summary=Optimize for HPC compute workloads
description=Configures virtual memory, CPU governors, and network settings for HPC compute
workloads.
include=latency-performance

[vm]
Most HPC application can take advantage of hugepages. Force them to on.
transparent_hugepages=always

[disk]
Increase the readahead value to support large, contiguous, files.
readahead=>4096

[sysctl]
Keep a reasonable amount of memory free to support large mem requests
vm.min_free_kbytes=135168

Most HPC applications are NUMA aware. Enabling zone reclaim ensures
memory is reclaimed and reallocated from local pages. Disabling
automatic NUMA balancing prevents unwanted memory unmapping.
vm.zone_reclaim_mode=1
kernel.numa_balancing=0

Busy polling helps reduce latency in the network receive path
by allowing socket layer code to poll the receive queue of a
network device, and disabling network interrupts.
busy_read value greater than 0 enables busy polling. Recommended
net.core.busy_read value is 50.
busy_poll value greater than 0 enables polling globally.
Recommended net.core.busy_poll value is 50
net.core.busy_read=50
net.core.busy_poll=50

TCP fast open reduces network latency by enabling data exchange
during the sender's initial TCP SYN. The value 3 enables fast open
on client and server connections.
net.ipv4.tcp_fastopen=3
```

- *HPC*
- 
- 

\*1: This is a universe package Ubuntu ships this software as part of its [universe repository](#)<sup>1177</sup>, which is maintained by volunteers from the Ubuntu community. Canonical also offers [Ubuntu Pro](#)<sup>1178</sup> – a free-for-personal-use subscription that provides a 10 year [security maintenance commitment](#)<sup>1179</sup>.

<sup>1177</sup> [https://documentation.ubuntu.com/pro-client/en/latest/explanations/about\\_esm/](https://documentation.ubuntu.com/pro-client/en/latest/explanations/about_esm/#what-are-main-and-universe)

#what-are-main-and-universe

<sup>1178</sup> <https://ubuntu.com/pro>

<sup>1179</sup> <https://ubuntu.com/security/esm>

## Further reading

- [TuneD website](#)<sup>1180</sup>
- *tuned-adm(8)*<sup>1181</sup>
- *TuneD profiles*<sup>1182</sup>
- *TuneD daemon*<sup>1183</sup>
- *TuneD configuration*<sup>1184</sup>

Docs<sup>1186</sup>

hosted in [GitHub](#)<sup>1185</sup>

rendered on [Read the](#)

*options*

*that don't require any coding*

*please let us know*

<sup>1180</sup> <https://tuned-project.org/>

<sup>1181</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/tuned-adm.8.html>

<sup>1182</sup> <https://manpages.ubuntu.com/manpages/resolute/man7/tuned-profiles.7.html>

<sup>1183</sup> <https://manpages.ubuntu.com/manpages/resolute/man8/tuned.8.html>

<sup>1184</sup> <https://manpages.ubuntu.com/manpages/resolute/man5/tuned.conf.5.html>

<sup>1185</sup> <https://github.com/canonical/ubuntu-server-documentation>

<sup>1186</sup> <https://ubuntu.com/server/docs/>

*directly in the GitHub web interface*

- 
- 
- 
- 
- 
- 
- 
- 
- 

*Local development*

**The outline of proposed work has not already been agreed by a maintainer.**

*follow the outlined procedures*

**The pull request far exceeds the scope of what was agreed.**

**The contribution makes changes without value.**

---

<sup>1187</sup> <https://github.com/canonical/open-documentation-academy>

**We deem the contribution to be AI generated.**

*Acceptable use of AI*

**The issue the pull request is addressing was assigned to someone else.**

#### Acceptable use of AI

- 
- 
- 
- 
- 

1.

2.

3.

4.

*our contributors*

## Types of contributions

issues list<sup>1188</sup>

### Technical

- 
- 

Find technical issues to work on<sup>1189</sup>

### Coding options

- 
- 
- 
- 

using Mermaid<sup>1190</sup>

Find coding issues to work on<sup>1191</sup>

---

<sup>1188</sup> <https://github.com/canonical/ubuntu-server-documentation/issues>

<sup>1189</sup> <https://github.com/canonical/ubuntu-server-documentation/issues?q=is%3Aissue+is%3Aopen+label%3A%22code%3A+technical%22+no%3Aassignee>

<sup>1190</sup> <https://mermaid.live/>

<sup>1191</sup> <https://github.com/canonical/ubuntu-server-documentation/issues?q=is%3Aissue+is%3Aopen+label%3A%22code%3A+coding%22+no%3Aassignee>

## Low-code options

- 
- 
- 
- 
- 

online generator<sup>1192</sup>

Find low-code issues to work on<sup>1193</sup>

## No-code options

## Creating issues

## User-experience testing

---

<sup>1192</sup> <https://www.mermaidchart.com/app/dashboard>

<sup>1193</sup> <https://github.com/canonical/ubuntu-server-documentation/issues?q=is%3Aissue+is%3Aopen+label%3A%22code%3A+low-code%22+no%3Aassignee>

Find no-code issues to work on<sup>1194</sup>

## Local development

## Prerequisites

## Code of Conduct

Code of Conduct<sup>1195</sup>

## GitHub account

GitHub account<sup>1196</sup>

## Workflow overview

- *Find an issue to work on*
- *Pick your workflow*
- *get help*
- *Test your contribution*
- *Submit it for review*

## Find issues to work on

checking our issues list<sup>1197</sup>

use this link<sup>1198</sup>

<sup>1194</sup> <https://github.com/canonical/ubuntu-server-documentation/issues?q=is%3Aissue+is%3Aopen+label%3A%22code%3A+non-code%22+no%3Aassignee>

<sup>1195</sup> <https://ubuntu.com/community/docs/ethos/code-of-conduct>

<sup>1196</sup> <https://github.com/>

<sup>1197</sup> <https://github.com/canonical/ubuntu-server-documentation/issues>

<sup>1198</sup> <https://github.com/canonical/ubuntu-server-documentation/issues?q=is%3Aissue+is%3Aopen+no%3Aassignee>

## Claim an issue

## Create a new issue

- 
- 
- 

## Choose your workflow

- 
- 

*quickstart guide*

*Local setup*

## Small changes directly in GitHub

GitHub's web interface<sup>1199</sup>

1. in the documentation<sup>1200</sup>
- 2.
- 3.
- 4.
- 5.
- 6.

---

<sup>1199</sup> <https://docs.github.com/en/repositories/working-with-files/managing-files/editing-files>

<sup>1200</sup> <https://ubuntu.com/server/docs>



## Local setup

### Create a fork

Ubuntu Server GitHub repository<sup>1201</sup>

with git<sup>1202</sup>

working

### Get the documentation

```
$ git clone git@github.com:canonical/ubuntu-server-documentation.git
```

```
$ cd ubuntu-server-documentation
```

### Create a new branch

```
$ git pull
```

#### Note:

This won't have any effect if you have only just cloned the repository (since you're already up-to-date).

```
$ git checkout -b my-new-branch
```

<sup>1201</sup> <https://github.com/canonical/ubuntu-server-documentation>

<sup>1202</sup> [https://documentation.academy/docs/howto/get-started/using\\_git/](https://documentation.academy/docs/howto/get-started/using_git/)

## Build and preview

```
$ make run
```

<http://127.0.0.1:8000/>

```
$ export SPHINX_HOST=0.0.0.0
```

### Note:

If you have problems getting the documentation to run on your machine, reach out to the team or leave a comment on your issue to get additional support.

## Other build commands

- 
- 
- 

## Writing guidance

*guidance for writing*

## Test your changes

## Automatic testing

```
$ make spelling > make linkcheck
```

## Manual testing

## Creating redirects

## Internal redirects

```
redirect/path/from/ redirect/path/to/
```

Sphinx Rediraffe docs<sup>1203</sup>

---

<sup>1203</sup> <https://sphinxext-rediraffe.readthedocs.io/en/latest/>

## External redirects

```
redirects = {
 "example/source": "https://exampletarget.org",
 "how-to/containers/lxc-containers": "https://linuxcontainers.org/lxc/documentation/",
}
```

<https://ubuntu.com/server/docs>

## Submitting your pull request

### Commit your changes

- 
- 
- 

#### Note:

Try to group your changes logically. For example, if you have one set of changes that modifies spelling in 10 files, and another set of changes that modifies formatting in 10 different files, you can group them into two commits (one for spelling, and one for formatting). You don't need a separate commit for every file.

## Push and create a pull request

```
$ git push <your username> <branch name>
```

[create a pull request](#)<sup>1204</sup>

<sup>1204</sup> <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-a-pull-request>

## Read the Docs preview

## Reviews

### Note:

The team has adopted the [Conventional Comments](https://conventionalcomments.org/)<sup>1205</sup> approach with the intention of making feedback easier to parse.

<sup>1205</sup> <https://conventionalcomments.org/>

## Guidance for writing

## Files and structure

support<sup>1206</sup>

MyST

Diátaxis<sup>1207</sup>

<sup>1206</sup> <https://myst-parser.readthedocs.io/en/latest/intro.html>

<sup>1207</sup> <https://diataxis.fr/>

**Tip:**

For example, if a page appears in the How-to section about Virtualisation, the landing page you need will be `how-to/virtualisation.md`.

**Versioning policy****Versioning by admonition**

```
 :::{note}
For Ubuntu 24.04 LTS (Noble) onwards, the recommended method for ...
 :::
```

**Versioning by tabs**

helpful demonstration<sup>1208</sup>

*the QEMU page*

---

<sup>1208</sup> <https://mystmd.org/guide/dropdowns-cards-and-tabs#tabs>

## Style guide

Canonical documentation style guide<sup>1209</sup>

## Language

- 
- 

## Acronyms

## Links

---

<sup>1209</sup> <https://docs.ubuntu.com/styleguide/en>

## Writing style

### Markdown elements

### Sections and headings

```
Heading level 1
Heading level 2
Heading level 3
Heading level 4
```

### Semantic markup

MyST

Roles<sup>1210</sup>

```
Keyboard keys: {kbd}`Ctrl`, {kbd}`C`
```

```
Manpages: {manpage}`package-name(section)` (e.g. {manpage}`dpkg(1)`)
```

*dpkg(1)*<sup>1211</sup>

```
Matrix: {matrix}`channel-name` (e.g. {matrix}`server`)
```

**#server:ubuntu.com**<sup>1212</sup>

```
Launchpad source: {lpsrc}`package-name` (e.g. {lpsrc}`bash`)
```

**bash**<sup>1213</sup>

<sup>1210</sup> <https://myst-parser.readthedocs.io/en/latest/syntax/roles-and-directives.html>

<sup>1211</sup> <https://manpages.ubuntu.com/manpages/resolute/man1/dpkg.1.html>

<sup>1212</sup> <https://matrix.to/#/#server:ubuntu.com>

<sup>1213</sup> <https://launchpad.net/ubuntu/+source/bash>



Launchpad bug: {lpbug}`bug-number` (e.g. {lpbug}`1852389`)

LP: #1852389<sup>1214</sup>

## Lists

```
1. This is the first item
1. This is the second
1. This is the third
```

## The {terminal} directive

```
```{terminal}
:copy:
:user:
:host:
:dir:
command

output
...`
```

```
$ command
```

```
output
```

-
-
-
-

¹²¹⁴ <https://bugs.launchpad.net/bugs/1852389>

```
'''{terminal}
:copy:
:user: username
:host: ubuntu
:dir: ~/src
command
```

```
output
'''
```

```
username@ubuntu:~/src$ command
```

```
output
```

```
'''{terminal}
:copy:
:user: username
:host: ubuntu
:dir: ~/src
command
'''
```

Some additional info for the reader.

```
'''{terminal}
:output-only:
The output
'''
```

```
username@ubuntu:~/src$ command
```

```
The output
```

Code blocks

```
```yaml
Some code block here
```
```

Admonition blocks

allowed by the MyST syntax¹²¹⁵

Getting help

Directly on GitHub issues

¹²¹⁵ <https://myst-parser.readthedocs.io/en/stable/syntax/admonitions.html>

Help with your submission

Community forums

[create a post](#)¹²¹⁶

[also has a forum](#)¹²¹⁷

Chat to us

- [To chat to the Server team](#)¹²¹⁸
- [To talk to technical authors about documentation](#)¹²¹⁹

Our contributors

¹²¹⁶ <https://discourse.ubuntu.com/c/project/server/17>

¹²¹⁷ <https://discourse.ubuntu.com/c/community/open-documentation-academy/166>

¹²¹⁸ <https://matrix.to/#/#server:ubuntu.com>

¹²¹⁹ <https://matrix.to/#/#documentation:ubuntu.com>

Content updates

PR number	Contributed by	Contribution made
PR #202¹²²⁰	network-charles	Updated instructions for building docs locally
PR #200¹²²¹	Zetway-Technologies	Clarified instructions in <code>install-lxc-dhcp-server.md</code>
PR #192¹²²²	minaelee	Updated CODA links in our contributing guide
PR #190¹²²³	kenyon	Removed no-longer-needed reference to a recently closed bug
PR #181¹²²⁴	davidekete	Deprecated the <code>backup-with-shell-scripts.md</code> page
PR #130¹²²⁵	Sophie-Pages	Deprecated the <code>lxc-containers.md</code> page
PR #119¹²²⁶	Sophie-Pages	Added introduction to <code>install-openldap.md</code> page
PR #115¹²²⁷	minaelee	Updated and refreshed content on <code>lxd-containers.md</code> page
PR #113¹²²⁸	teward	Updated <code>apparmor.md</code> to explain how to totally enable/disable AppArmor
PR #54¹²²⁹	khbecker	Added details about <code>kdump</code> enablement in 24.10 to <code>kernel-crash-dump.md</code>

¹²²⁰ <https://github.com/canonical/ubuntu-server-documentation/pull/202>

¹²²¹ <https://github.com/canonical/ubuntu-server-documentation/pull/200>

¹²²² <https://github.com/canonical/ubuntu-server-documentation/pull/192>

¹²²³ <https://github.com/canonical/ubuntu-server-documentation/pull/190>

¹²²⁴ <https://github.com/canonical/ubuntu-server-documentation/pull/181>

¹²²⁵ <https://github.com/canonical/ubuntu-server-documentation/pull/130>

¹²²⁶ <https://github.com/canonical/ubuntu-server-documentation/pull/119>

¹²²⁷ <https://github.com/canonical/ubuntu-server-documentation/pull/115>

¹²²⁸ <https://github.com/canonical/ubuntu-server-documentation/pull/130>

¹²²⁹ <https://github.com/canonical/ubuntu-server-documentation/pull/54>

Copyediting and proofreading

PR number	Contributed by	Contribution made
PR #184 ¹²³	thorn-shadow99	Improved the sssd.md page
PR #183 ¹²³	thorn-shadow99	Improved the security.md page
PR #167 ¹²³	POS-ToakDEV	Improved the libvirt.md page
PR #155 ¹²³	thorn-shadow99	Improved the multipath.md page
PR #149 ¹²³	thorn-shadow99	Improved the high-availability.md page
PR #148 ¹²³	thorn-shadow99	Improved the ebpf.md page
PR #147 ¹²³	bird-stare	Improved the introduction-to-backups.md, introduction-to-mail-services.md and introduction-to-virtualisation.md pages
PR #146 ¹²³	thorn-shadow99	Improved the crypto-libraries.md page
PR #141 ¹²³	gregbell	Improved the qemu-microvm.md page
PR #132 ¹²³	bird-stare	Substantial copyediting on the networking.md page
PR #131 ¹²⁴	network-charles	Improved the ebpf.md page
PR #123 ¹²⁴	davideket	Converted "note" blocks into MyST admonitions (explanation section)
PR #122 ¹²⁴	YanisaHS	Improved the install-dns.md page
PR #120 ¹²⁴	davideket	Converted "note" blocks into MyST admonitions (reference section)
PR #83 ¹²⁴⁴	daveedoo	Improved the serve-ntp-with-chrony.md page
PR #81 ¹²⁴⁵	peat-psuwit	Improved the report-a-bug.md page
PR #67 ¹²⁴⁶	pnhuy	Improved the configur-nginx.md page

Error fixes

PR number	Contributed by	Contribution made
PR #97 ¹²⁴⁷	yukihane	Corrected an error in <code>install-openvpn.md</code>
PR #66 ¹²⁴⁸	Pexers	Removed duplicate text block in <code>peer-to-site-on-router.md</code>
PR #52 ¹²⁴⁹	sbworth	Corrected an error in <code>WireGuard VPN on-an-internal-system.md</code>
PR #48 ¹²⁵⁰	nikolas	Corrected an error in <code>install-postfix.md</code>

Functionality and enhancements

PR number	Contributed by	Contribution made
PR #193 ¹²⁵¹	network-charles	Added hover-over referencing functionality
PR #116 ¹²⁵²	activus-d	Created an issue template to help readers provide better feedback
PR #112 ¹²⁵³	goodylili	Created a pull request template

¹²³⁰ <https://github.com/canonical/ubuntu-server-documentation/pull/184>

¹²³¹ <https://github.com/canonical/ubuntu-server-documentation/pull/183>

¹²³² <https://github.com/canonical/ubuntu-server-documentation/pull/167>

¹²³³ <https://github.com/canonical/ubuntu-server-documentation/pull/155>

¹²³⁴ <https://github.com/canonical/ubuntu-server-documentation/pull/149>

¹²³⁵ <https://github.com/canonical/ubuntu-server-documentation/pull/148>

¹²³⁶ <https://github.com/canonical/ubuntu-server-documentation/pull/147>

¹²³⁷ <https://github.com/canonical/ubuntu-server-documentation/pull/146>

¹²³⁸ <https://github.com/canonical/ubuntu-server-documentation/pull/141>

¹²³⁹ <https://github.com/canonical/ubuntu-server-documentation/pull/132>

¹²⁴⁰ <https://github.com/canonical/ubuntu-server-documentation/pull/131>

¹²⁴¹ <https://github.com/canonical/ubuntu-server-documentation/pull/123>

¹²⁴² <https://github.com/canonical/ubuntu-server-documentation/pull/122>

¹²⁴³ <https://github.com/canonical/ubuntu-server-documentation/pull/120>

¹²⁴⁴ <https://github.com/canonical/ubuntu-server-documentation/pull/83>

¹²⁴⁵ <https://github.com/canonical/ubuntu-server-documentation/pull/81>

¹²⁴⁶ <https://github.com/canonical/ubuntu-server-documentation/pull/67>

¹²⁴⁷ <https://github.com/canonical/ubuntu-server-documentation/pull/97>

¹²⁴⁸ <https://github.com/canonical/ubuntu-server-documentation/pull/66>

¹²⁴⁹ <https://github.com/canonical/ubuntu-server-documentation/pull/52>

¹²⁵⁰ <https://github.com/canonical/ubuntu-server-documentation/pull/48>

Glossary

PR number	Contributed by	Contribution made
PR #194 ¹²⁵⁴	network-charles	Contributed terms to the glossary
PR #187 ¹²⁵⁵	network-charles	Contributed terms to the glossary
PR #186 ¹²⁵⁶	network-charles	Contributed terms to the glossary
PR #185 ¹²⁵⁷	network-charles	Contributed terms to the glossary
PR #182 ¹²⁵⁸	network-charles	Contributed terms to the glossary
PR #174 ¹²⁵⁹	network-charles	Contributed terms to the glossary
PR #162 ¹²⁶⁰	network-charles	Contributed terms to the glossary
PR #161 ¹²⁶¹	kaushikjayant	Contributed terms to the glossary
PR #160 ¹²⁶²	network-charles	Contributed terms to the glossary
PR #159 ¹²⁶³	network-charles	Contributed terms to the glossary
PR #158 ¹²⁶⁴	network-charles	Contributed terms to the glossary
PR #156 ¹²⁶⁵	network-charles	Contributed terms to the glossary
PR #152 ¹²⁶⁶	Ibom99	Contributed "J" terms to the glossary
PR #134 ¹²⁶⁷	RomainDust	Contributed terms to the glossary
PR #124 ¹²⁶⁸	activus-d	Added the glossary page and functionality

¹²⁵¹ <https://github.com/canonical/ubuntu-server-documentation/pull/193>

¹²⁵² <https://github.com/canonical/ubuntu-server-documentation/pull/116>

¹²⁵³ <https://github.com/canonical/ubuntu-server-documentation/pull/112>

PDF improvements

PR number	Contributed by	Contribution made
PR #10¹²⁶⁹	SecondSkoll	Added the ability to build the PDF locally
PR #9¹²⁷⁰	SecondSkoll	Fixed our PDF build issues

¹²⁵⁴ <https://github.com/canonical/ubuntu-server-documentation/pull/194>
¹²⁵⁵ <https://github.com/canonical/ubuntu-server-documentation/pull/187>
¹²⁵⁶ <https://github.com/canonical/ubuntu-server-documentation/pull/186>
¹²⁵⁷ <https://github.com/canonical/ubuntu-server-documentation/pull/185>
¹²⁵⁸ <https://github.com/canonical/ubuntu-server-documentation/pull/182>
¹²⁵⁹ <https://github.com/canonical/ubuntu-server-documentation/pull/174>
¹²⁶⁰ <https://github.com/canonical/ubuntu-server-documentation/pull/162>
¹²⁶¹ <https://github.com/canonical/ubuntu-server-documentation/pull/161>
¹²⁶² <https://github.com/canonical/ubuntu-server-documentation/pull/160>
¹²⁶³ <https://github.com/canonical/ubuntu-server-documentation/pull/159>
¹²⁶⁴ <https://github.com/canonical/ubuntu-server-documentation/pull/158>
¹²⁶⁵ <https://github.com/canonical/ubuntu-server-documentation/pull/156>
¹²⁶⁶ <https://github.com/canonical/ubuntu-server-documentation/pull/152>
¹²⁶⁷ <https://github.com/canonical/ubuntu-server-documentation/pull/134>
¹²⁶⁸ <https://github.com/canonical/ubuntu-server-documentation/pull/124>
¹²⁶⁹ <https://github.com/canonical/ubuntu-server-documentation/pull/10>
¹²⁷⁰ <https://github.com/canonical/ubuntu-server-documentation/pull/9>