

Kubernetes

# Install Minikube on Ubuntu 22.04 LTS

7 months ago • by Shahriar Shovon

“Minikube is a program that is used to set up a local Kubernetes cluster very easily. It’s very helpful to learn Kubernetes and develop apps/products for Kubernetes.

In this article, I am going to show you how to install Minikube on Ubuntu 22.04 LTS. I will also show you how to start a local Kubernetes cluster with Minikube and the basics of Kubernetes as well. So, let’s get started.”

## Table of Contents

1. [Things You Will Need](#)
2. [Installing Docker on Ubuntu 22.04 LTS](#)
3. [Installing KVM/QEMU on Ubuntu 22.04 LTS](#)
4. [Installing VirtualBox on Ubuntu 22.04 LTS](#)
5. [Installing Minikube](#)
6. [Installing kubectl – The Kubernetes Command-Line Tool](#)
7. [Starting a Kubernetes Cluster with Minikube](#)
8. [Basics of Kubernetes](#)
9. [Managing Minikube Kubernetes Cluster](#)
10. [Conclusion](#)
11. [References](#)

## Things You Will Need

To set up a Kubernetes cluster with Minikube, you will need either Docker, KVM/QEMU, or VirtualBox installed on your Ubuntu 22.04 LTS machine.

To install Docker on Ubuntu 22.04 LTS, read the [Installing Docker on Ubuntu 22.04 LTS](#) section of this article.

To install KVM/QEMU on Ubuntu 22.04 LTS, read the [Installing KVM/QEMU on Ubuntu 22.04 LTS](#) section of this article.

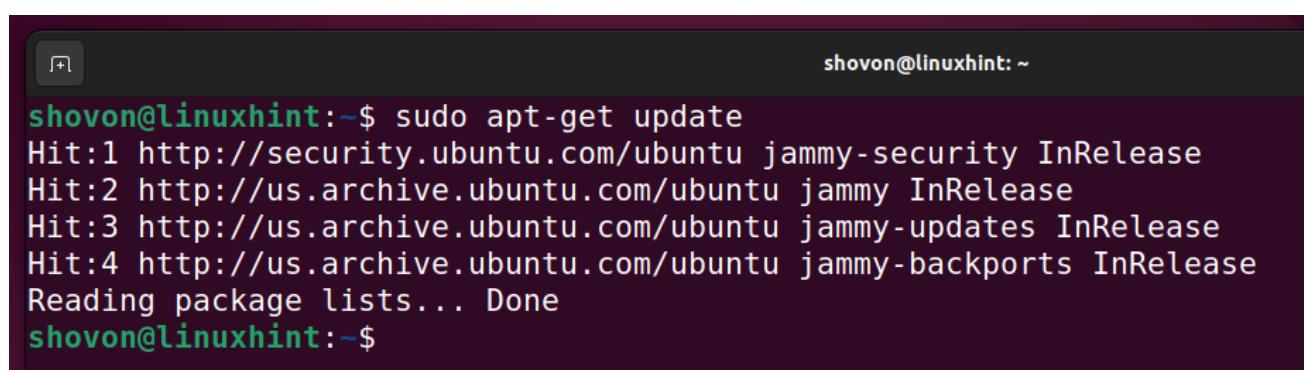
To install VirtualBox on Ubuntu 22.04 LTS, read the [Installing VirtualBox on Ubuntu 22.04 LTS](#) section of this article.

## Installing Docker on Ubuntu 22.04 LTS

In this section, I am going to show you how to install Docker CE on Ubuntu 22.04 LTS so that you can use it as a driver for Minikube.

First, update the APT package repository cache with the following command:

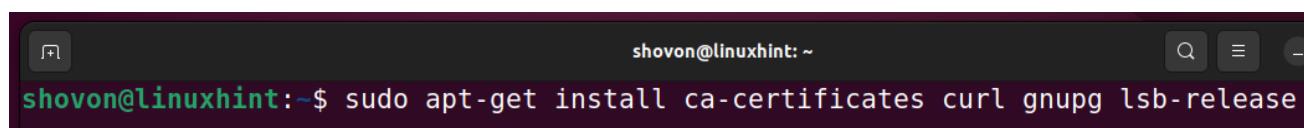
```
$ sudo apt update
```



```
shovon@linuxhint:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
shovon@linuxhint:~$
```

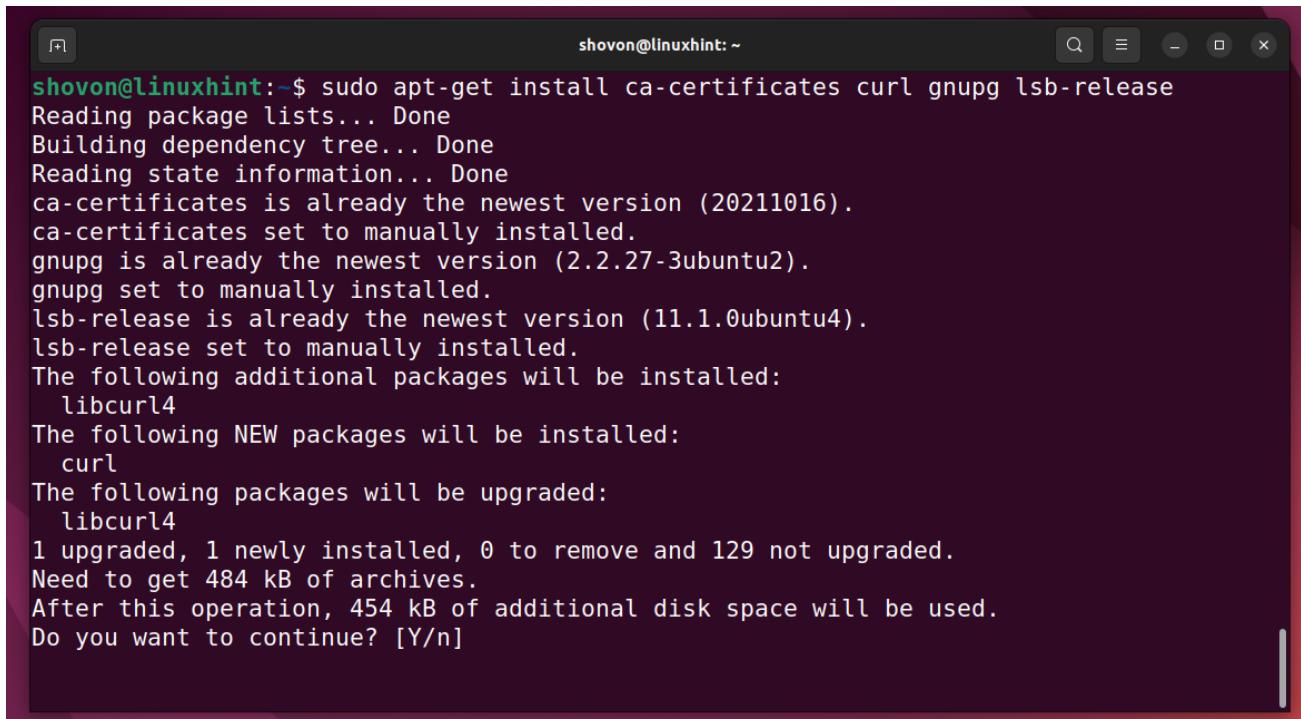
Install the required dependencies with the following command:

```
$ sudo apt install ca-certificates curl gnupg lsb-release
```



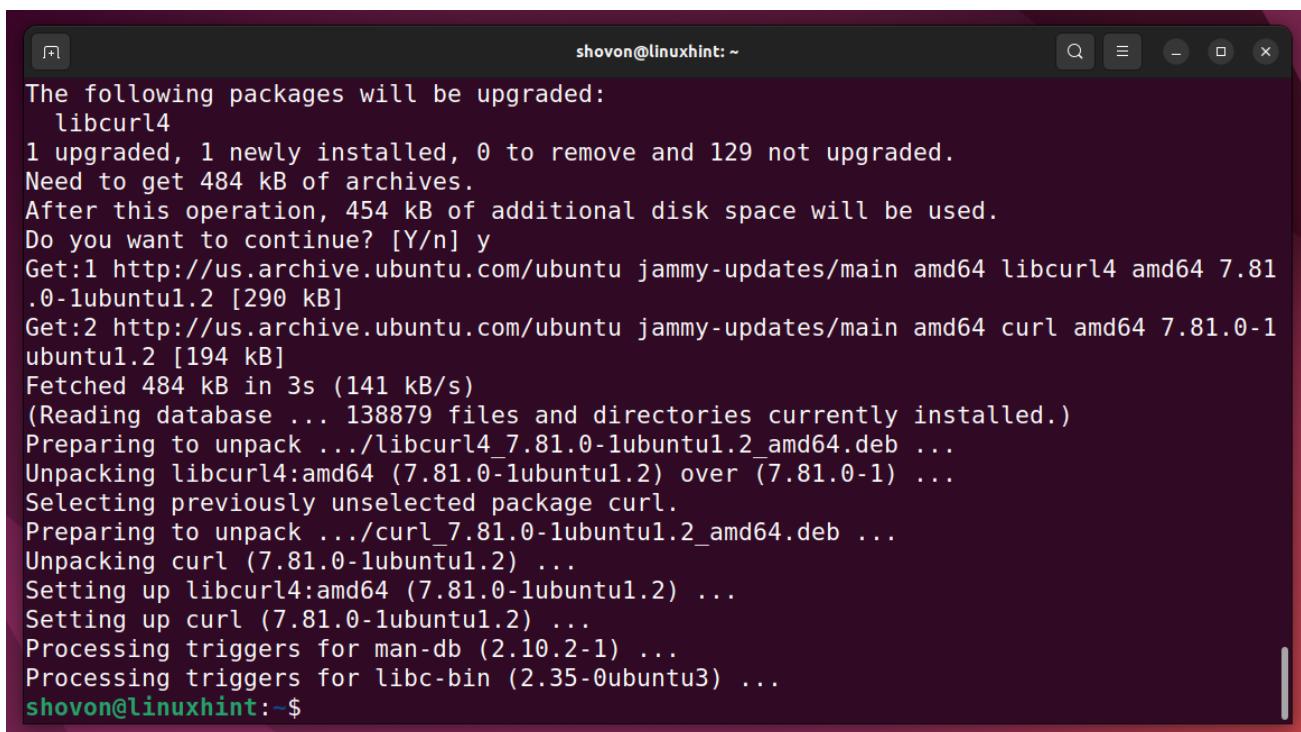
```
shovon@linuxhint:~$ sudo apt-get install ca-certificates curl gnupg lsb-release
```

To confirm the installation, press **Y** and then press **<Enter>**.



```
shovon@linuxhint:~$ sudo apt-get install ca-certificates curl gnupg lsb-release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20211016).
ca-certificates set to manually installed.
gnupg is already the newest version (2.2.27-3ubuntu2).
gnupg set to manually installed.
lsb-release is already the newest version (11.1.0ubuntu4).
lsb-release set to manually installed.
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  libcurl4
1 upgraded, 1 newly installed, 0 to remove and 129 not upgraded.
Need to get 484 kB of archives.
After this operation, 454 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

The required dependencies should be installed.



```
The following packages will be upgraded:
  libcurl4
1 upgraded, 1 newly installed, 0 to remove and 129 not upgraded.
Need to get 484 kB of archives.
After this operation, 454 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcurl4 amd64 7.81.0-1ubuntu1.2 [290 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.2 [194 kB]
Fetched 484 kB in 3s (141 kB/s)
(Reading database ... 138879 files and directories currently installed.)
Preparing to unpack .../libcurl4_7.81.0-1ubuntu1.2_amd64.deb ...
Unpacking libcurl4:amd64 (7.81.0-1ubuntu1.2) over (7.81.0-1) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.81.0-1ubuntu1.2_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.2) ...
Setting up libcurl4:amd64 (7.81.0-1ubuntu1.2) ...
Setting up curl (7.81.0-1ubuntu1.2) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
shovon@linuxhint:~$
```

To install the GPG key of the official Docker CE repository, create a new directory **/etc/apt/keyrings** as follows:

```
$ sudo mkdir -p /etc/apt/keyrings
```

```
shovon@linuxhint:~$ sudo mkdir -p /etc/apt/keyrings
```

Download the GPG key file of the official Docker CE repository with the following command:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

```
shovon@linuxhint:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

Now, add the official Docker CE package repository on your Ubuntu 22.04 LTS machine with the following command:

```
$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
shovon@linuxhint:~$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

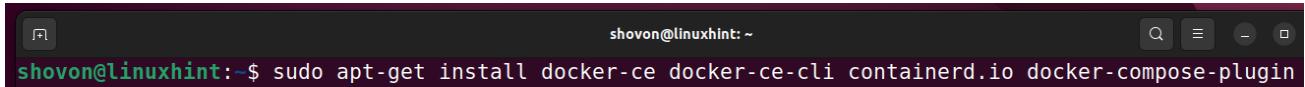
Update the APT package repository cache for the changes to take effect.

```
$ sudo apt-get update
```

```
shovon@linuxhint:~$ sudo apt-get update
Get:1 https://download.docker.com/linux/ubuntu jammy InRelease [48.9 kB]
Get:2 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [5,109 B]
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 54.0 kB in 2s (30.7 kB/s)
Reading package lists... Done
shovon@linuxhint:~$
```

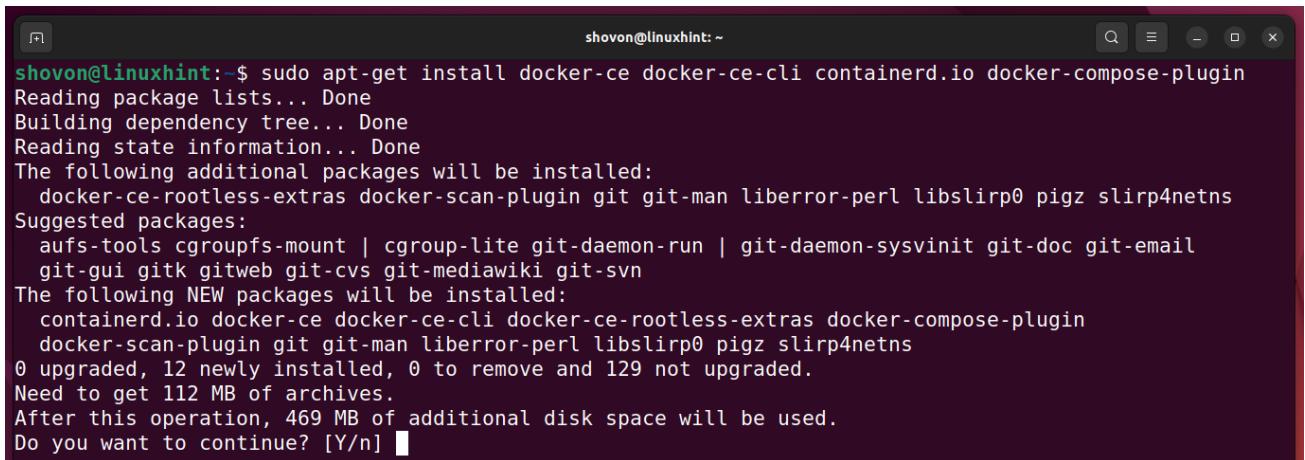
Install Docker CE from the official Docker CE package repository with the following command:

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```



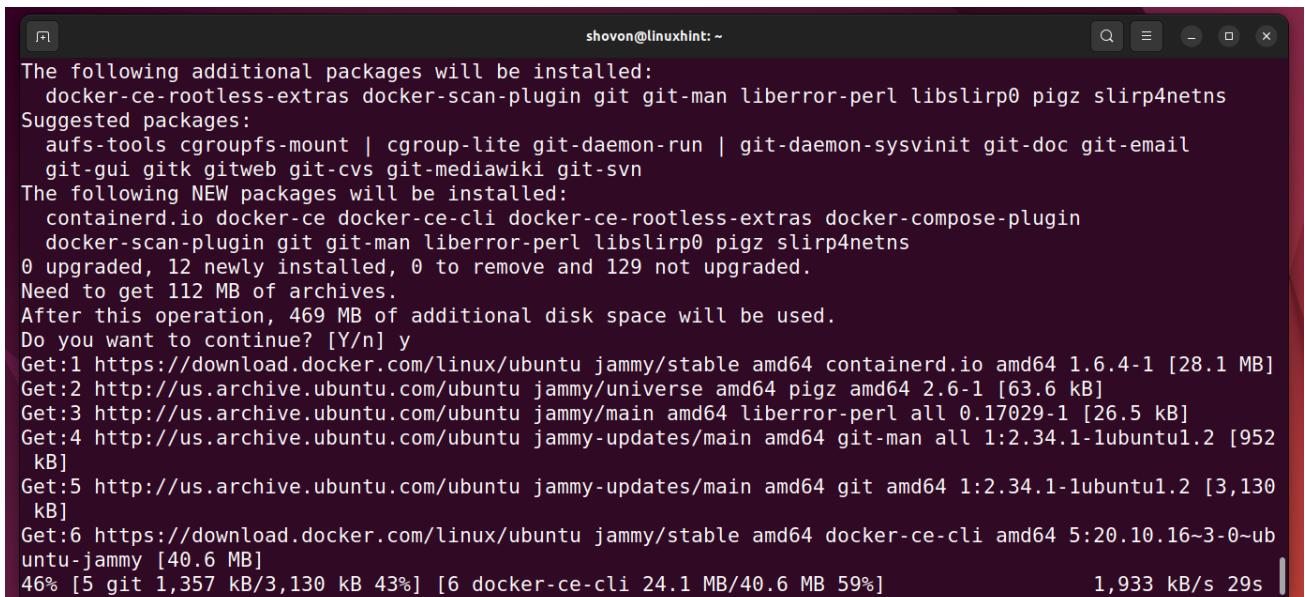
```
shovon@linuxhint:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

To confirm the installation, press **Y** and then press <Enter>.



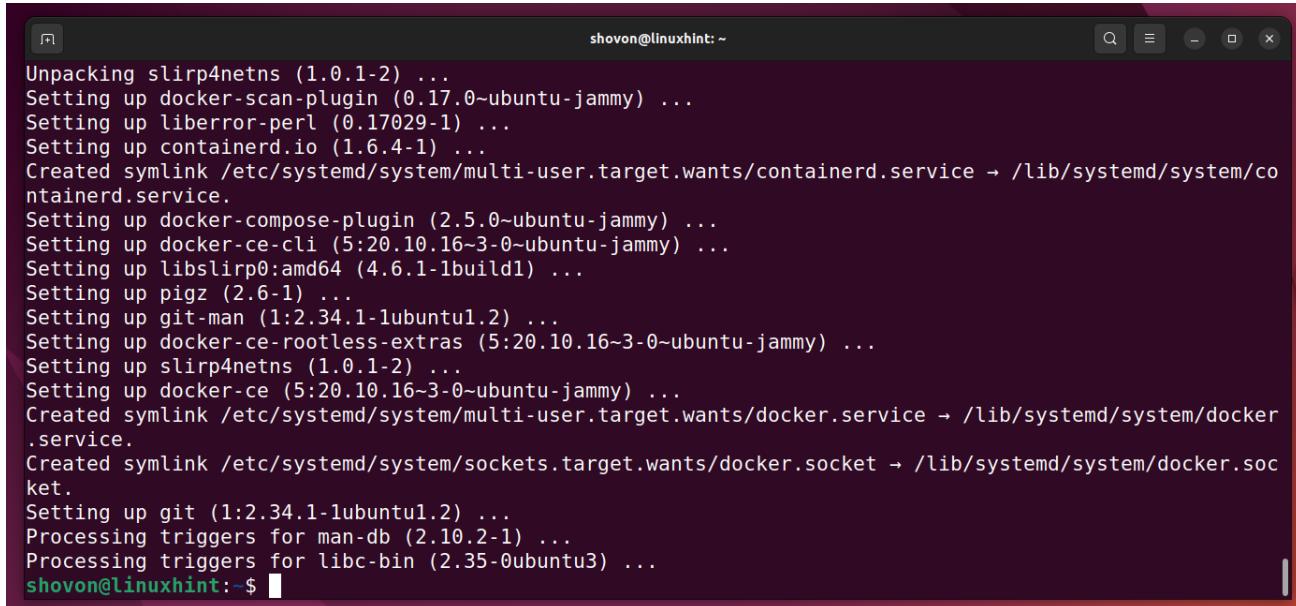
```
shovon@linuxhint:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email
  git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
  docker-scan-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 129 not upgraded.
Need to get 112 MB of archives.
After this operation, 469 MB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

Docker CE and all the required dependency packages are being downloaded and installed. It will take a while to complete.



```
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email
  git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
  docker-scan-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 129 not upgraded.
Need to get 112 MB of archives.
After this operation, 469 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.4-1 [28.1 MB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.2 [952 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.2 [3,130 kB]
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:20.10.16-3-0~ubuntu-jammy [40.6 MB]
46% [5 git 1,357 kB/3,130 kB 43%] [6 docker-ce-cli 24.1 MB/40.6 MB 59%] 1,933 kB/s 29s
```

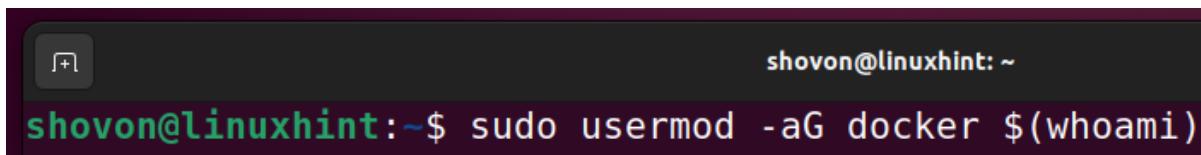
At this point, Docker CE should be installed.



```
shovon@linuxhint: ~
Unpacking slirp4netns (1.0.1-2) ...
Setting up docker-scan-plugin (0.17.0~ubuntu-jammy) ...
Setting up liberror-perl (0.17029-1) ...
Setting up containerd.io (1.6.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.5.0~ubuntu-jammy) ...
Setting up docker-ce-cli (5:20.10.16-3-0~ubuntu-jammy) ...
Setting up libslirp0:amd64 (4.6.1-1build1) ...
Setting up pigz (2.6-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.2) ...
Setting up docker-ce-rootless-extras (5:20.10.16~3-0~ubuntu-jammy) ...
Setting up slirp4netns (1.0.1-2) ...
Setting up docker-ce (5:20.10.16~3-0~ubuntu-jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Setting up git (1:2.34.1-1ubuntu1.2) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
shovon@linuxhint:~$
```

Now, add your login user to the **docker** group with the following command:

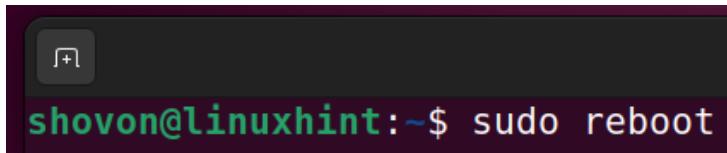
```
$ sudo usermod -aG docker $(whoami)
```



```
shovon@linuxhint: ~
shovon@linuxhint:~$ sudo usermod -aG docker $(whoami)
```

For the changes to take effect, reboot your computer with the following command:

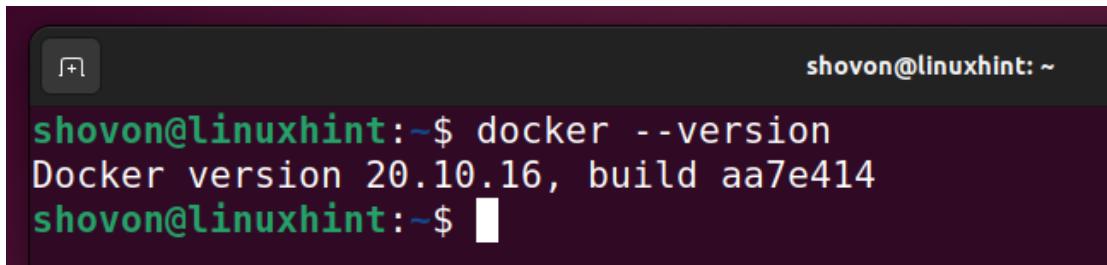
```
$ sudo reboot
```



```
shovon@linuxhint: ~
shovon@linuxhint:~$ sudo reboot
```

Once your computer boots, you can verify whether Docker is working with the following command:

```
$ docker --version
```



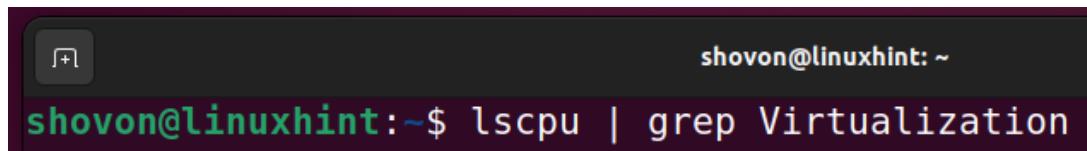
```
shovon@linuxhint: ~
shovon@linuxhint:~$ docker --version
Docker version 20.10.16, build aa7e414
shovon@linuxhint:~$
```

## Installing KVM/QEMU on Ubuntu 22.04 LTS

In this section, I am going to show you how to install KVM/QEMU on Ubuntu 22.04 LTS so that you can use it as a driver for Minikube.

First, make sure that you have hardware virtualization enabled with the following command:

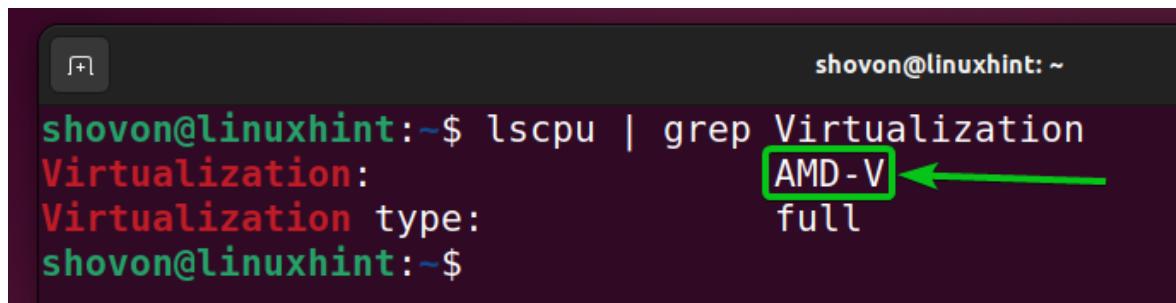
```
$ lscpu | grep Virtualization
```



```
shovon@linuxhint:~$ lscpu | grep Virtualization
```

If hardware virtualization is enabled, you will see the text **AMD-V** (For AMD processors) or **VT-X/VT-D** (For Intel processors) in the **Virtualization** section.

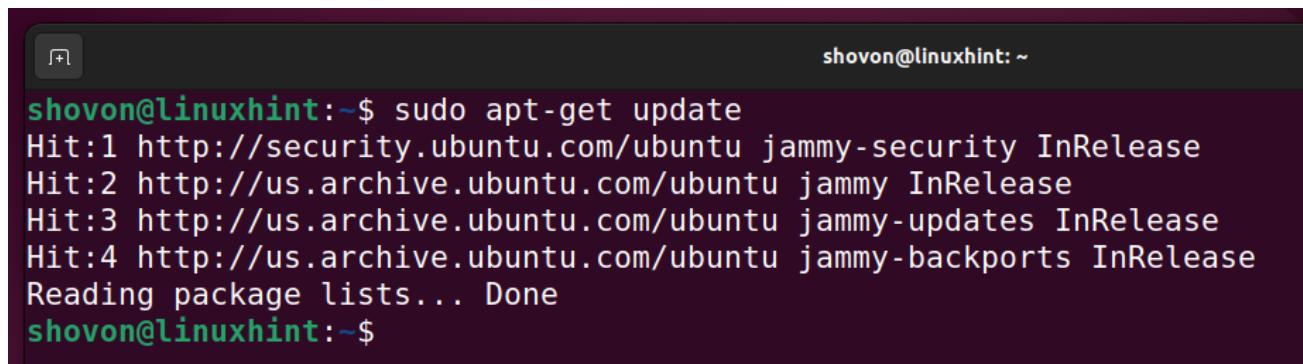
If hardware virtualization is not enabled, you can enable it from the BIOS of your motherboard. Every modern processor support this feature.



```
shovon@linuxhint:~$ lscpu | grep Virtualization
Virtualization: AMD-V
Virtualization type: full
shovon@linuxhint:~$
```

Update the APT package repository cache with the following command:

```
$ sudo apt update
```



```
shovon@linuxhint:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
shovon@linuxhint:~$
```

To install KVM/QEMU on Ubuntu 22.04 LTS, run the following command:

```
$ sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virtinst
virt-manager
```

```
shovon@linuxhint: $ sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virtinst virt-manager
```

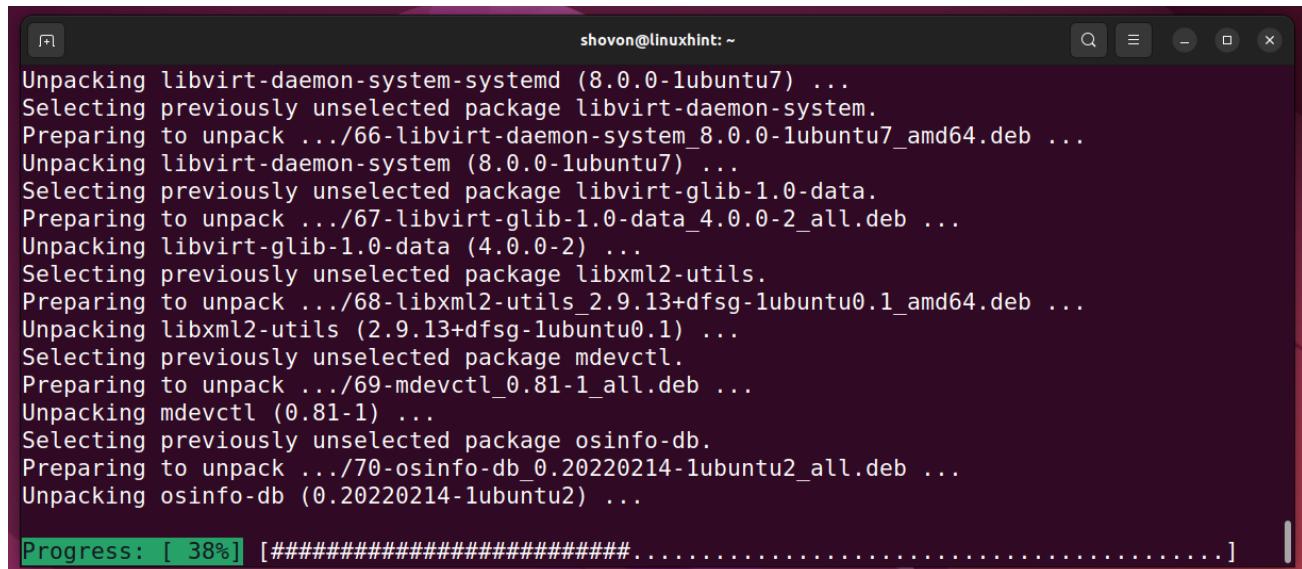
To confirm the installation, press **Y** and then press <Enter>.

```
libndctl6 libnss-mymachines libonig5 libosinfo-1.0-0 libphodav-2.0-0
libphodav-2.0-common libpmem1 libpmemobj1 librados2 librbd1 librdmacm1 libSDL2-2.0-0
libsllrp0 libspice-client-glib-2.0-8 libspice-client-gtk-3.0-5 libspice-server1
libtpms0 liburing2 libusbredirhost1 libusbredirparser1 libvirglrenderer1
libvirt-clients libvirt-daemon libvirt-daemon-config-network
libvirt-daemon-config-nwfilter libvirt-daemon-driver-qemu libvirt-daemon-system
libvirt-daemon-system-systemd libvirt-glib-1.0-0 libvirt-glib-1.0-data libvirt0
libxml2-utils libyajl2 lvm2 mdevctl msr-tools osinfo-db ovmf python3-libvirt
python3-libxml2 qemu-block-extra qemu-system-common qemu-system-data qemu-system-gui
qemu-system-x86 qemu-utils seabios spice-client-glib-usb-acl-helper swtpm swtpm-tools
systemd-container thin-provisioning-tools virt-manager virt-viewer virtinst xorriso
The following packages will be upgraded:
  libnss-systemd libpam-systemd libsystemd0 systemd systemd-oomd systemd-sysv
  systemd-timesyncd
7 upgraded, 94 newly installed, 0 to remove and 123 not upgraded.
Need to get 52.0 MB of archives.
After this operation, 176 MB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

KVM/QEMU and all the required dependency packages are being downloaded from the internet. It will take a few minutes to complete.

```
ll 8.0.0-lubuntu7 [71.5 kB]
Get:74 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libvirt-daemon-system amd64 8.0
.0-lubuntu7 [52.3 kB]
Get:75 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libvirt-glib-1.0-data all 4
.0.0-2 [13.6 kB]
Get:76 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libxml2-utils amd64 2.9
.13+dfsg-1lubuntu0.1 [40.2 kB]
Get:77 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 mdevctl all 0.81-1 [11.7 kB]
Get:78 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 osinfo-db all 0.20220214-1u
buntu2 [140 kB]
Get:79 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 python3-libvirt amd64 8.0.0-1bu
ild1 [148 kB]
Get:80 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-libxml2 amd
64 2.9.13+dfsg-1lubuntu0.1 [148 kB]
Get:81 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libfdt1 amd64 1.6.1-1 [20.0 kB]
Get:82 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 qemu-system-common amd64 1:6.2+
dfsg-2ubuntu6 [2,049 kB]
59% [82 qemu-system-common 1,810 kB/2,049 kB 88%] 208 kB/s 1min 55s ■
```

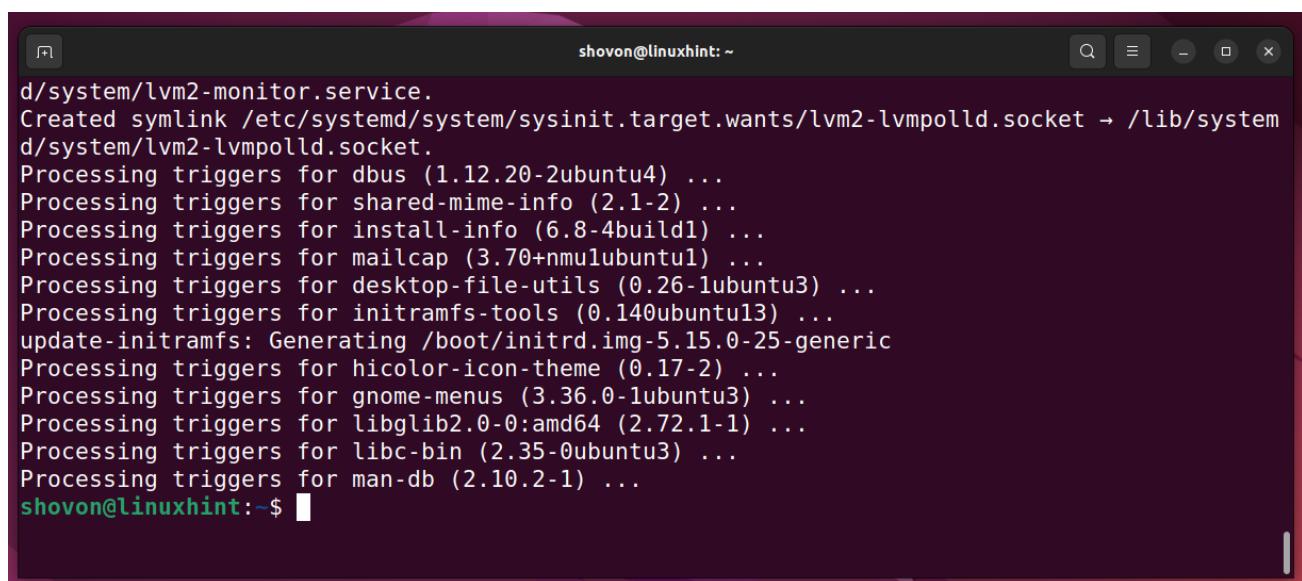
KVM/QEMU and all the required dependency packages are being installed. It will take a few minutes to complete.



```
shovon@linuxhint: ~
Unpacking libvirt-daemon-systemd (8.0.0-1ubuntu7) ...
Selecting previously unselected package libvirt-daemon-system.
Preparing to unpack .../66-libvirt-daemon-system_8.0.0-1ubuntu7_amd64.deb ...
Unpacking libvirt-daemon-system (8.0.0-1ubuntu7) ...
Selecting previously unselected package libvirt-glib-1.0-data.
Preparing to unpack .../67-libvirt-glib-1.0-data_4.0.0-2_all.deb ...
Unpacking libvirt-glib-1.0-data (4.0.0-2) ...
Selecting previously unselected package libxml2-utils.
Preparing to unpack .../68-libxml2-utils_2.9.13+dfsg-1ubuntu0.1_amd64.deb ...
Unpacking libxml2-utils (2.9.13+dfsg-1ubuntu0.1) ...
Selecting previously unselected package mdevctl.
Preparing to unpack .../69-mdevctl_0.81-1_all.deb ...
Unpacking mdevctl (0.81-1) ...
Selecting previously unselected package osinfo-db.
Preparing to unpack .../70-osinfo-db_0.20220214-1ubuntu2_all.deb ...
Unpacking osinfo-db (0.20220214-1ubuntu2) ...

Progress: [ 38%] [#####
.....]
```

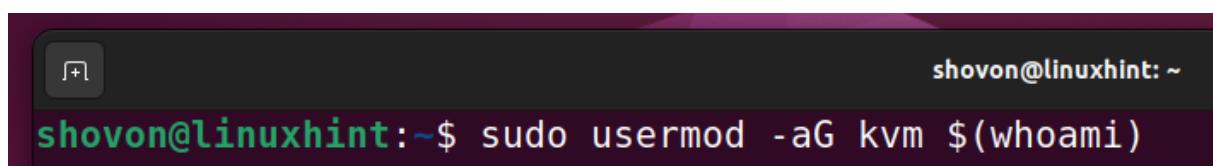
KVM/QEMU and all the required dependency packages are installed at this point.



```
shovon@linuxhint: ~
d/system/lvm2-monitor.service.
Created symlink /etc/systemd/system/sysinit.target.wants/lvm2-lvmpolld.socket → /lib/system
d/system/lvm2-lvmpolld.socket.
Processing triggers for dbus (1.12.20-2ubuntu4) ...
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for install-info (6.8-4build1) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for initramfs-tools (0.140ubuntu13) ...
update-initramfs: Generating /boot/initrd.img-5.15.0-25-generic
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libglib2.0-0:amd64 (2.72.1-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
shovon@linuxhint:~$
```

Add your login user to the **kvm** group with the following command:

```
$ sudo usermod -aG kvm $(whoami)
```



```
shovon@linuxhint: ~
shovon@linuxhint:~$ sudo usermod -aG kvm $(whoami)
```

Add your login user to the **libvirt** group with the following command:

```
$ sudo usermod -aG libvirt $(whoami)
```

```
shovon@linuxhint:~$ sudo usermod -aG libvirt $(whoami)
```

```
$ sudo reboot
```

```
shovon@linuxhint:~$ sudo reboot
```

For the changes to take effect, reboot your computer with the following command:

```
$ sudo reboot
```

```
shovon@linuxhint:~$ virt-host-validate
```

Once your computer boots, run the following command to check if all the KVM/QEMU components are working correctly.

**NOTE:** Some of the **LXC** features might **FAIL**, but it does not matter in this case.

```
$ virt-host-validate
```

```

shovon@linuxhint:~$ virt-host-validate
QEMU: Checking for hardware virtualization : PASS
QEMU: Checking if device /dev/kvm exists : PASS
QEMU: Checking if device /dev/kvm is accessible : PASS
QEMU: Checking if device /dev/vhost-net exists : PASS
QEMU: Checking if device /dev/net/tun exists : PASS
QEMU: Checking for cgroup 'cpu' controller support : PASS
QEMU: Checking for cgroup 'cpuacct' controller support : PASS
QEMU: Checking for cgroup 'cpuset' controller support : PASS
QEMU: Checking for cgroup 'memory' controller support : PASS
QEMU: Checking for cgroup 'devices' controller support : PASS
QEMU: Checking for cgroup controller support in kernel Kconfig file or mount/enable cgroup controller in your system) : WARN (Enable 'devices' in kernel Kconfig file or mount/enable cgroup controller in your system)
QEMU: Checking for cgroup 'blkio' controller support : PASS
QEMU: Checking for device assignment IOMMU support : WARN (No ACPI IVRS table found, IOMMU either disabled in BIOS or not supported by this hardware platform)
QEMU: Checking for secure guest support : WARN (Unknown if this platform has Secure Guest support)
LXC: Checking for Linux >= 2.6.26 : PASS
LXC: Checking for namespace ipc : PASS
LXC: Checking for namespace mnt : PASS
LXC: Checking for namespace pid : PASS
LXC: Checking for namespace uts : PASS
LXC: Checking for namespace net : PASS
LXC: Checking for namespace user : PASS
LXC: Checking for cgroup 'cpu' controller support : PASS
LXC: Checking for cgroup 'cpuacct' controller support : PASS
LXC: Checking for cgroup 'cpuset' controller support : PASS
LXC: Checking for cgroup 'memory' controller support : PASS
LXC: Checking for cgroup 'devices' controller support : FAIL (Enable 'devices' in kernel Kconfig file or mount/enable cgroup controller in your system)
LXC: Checking for cgroup 'freezer' controller support : FAIL (Enable 'freezer' in kernel Kconfig file or mount/enable cgroup controller in your system)
LXC: Checking for cgroup 'blkio' controller support : PASS
LXC: Checking if device /sys/fs/fuse/connections exists : PASS
shovon@linuxhint:~$
```

## Installing VirtualBox on Ubuntu 22.04 LTS

In this section, I am going to show you how to install VirtualBox on Ubuntu 22.04 LTS so that you can use it as a driver for Minikube.

First, make sure that you have hardware virtualization enabled with the following command:

```
$ lscpu | grep Virtualization
```

```

shovon@linuxhint:~$ lscpu | grep Virtualization
```

If hardware virtualization is enabled, you will see the text **AMD-V** (For AMD processors) or **VT-X/VT-D** (For Intel processors) in the **Virtualization** section.

If hardware virtualization is not enabled, you can enable it from the BIOS of your motherboard. Every modern processor supports this feature.

```
shovon@linuxhint:~$ lscpu | grep Virtualization
Virtualization:
Virtualization type: full
shovon@linuxhint:~$
```

Update the APT package repository cache with the following command:

```
$ sudo apt update
```

```
shovon@linuxhint:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
shovon@linuxhint:~$
```

To install VirtualBox on Ubuntu 22.04 LTS, run the following command:

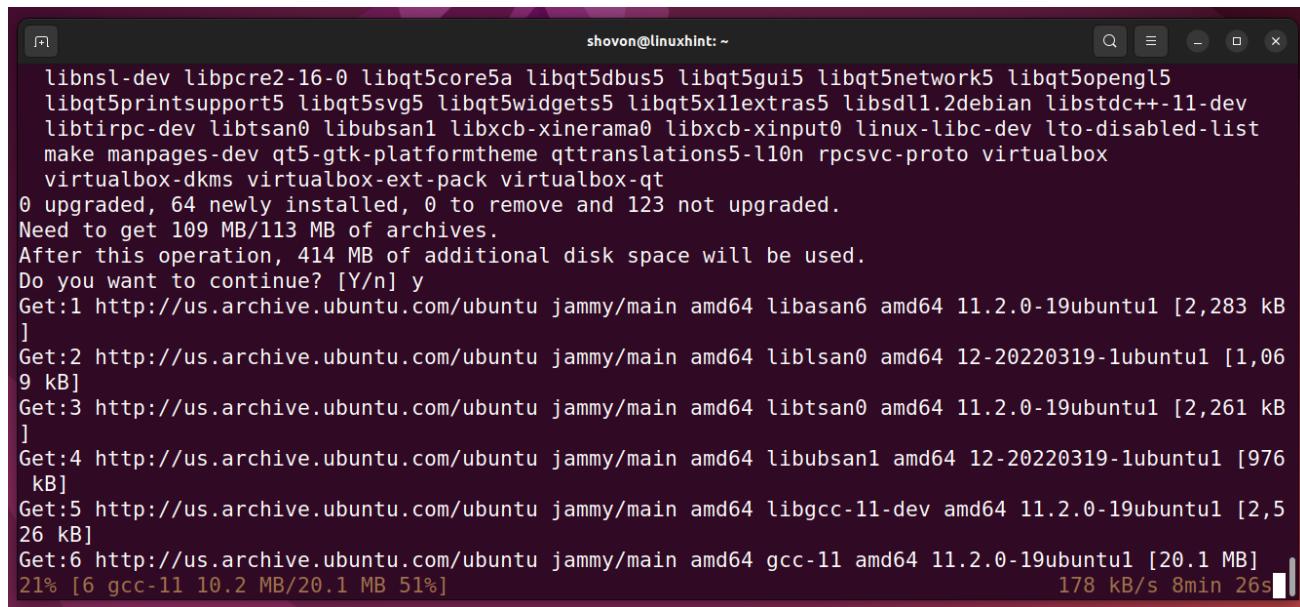
```
$ sudo apt install virtualbox virtualbox-dkms virtualbox-qt virtualbox-ext-p
```

```
shovon@linuxhint:~$ sudo apt install virtualbox virtualbox-dkms virtualbox-qt virtualbox-ext-p
```

To confirm the installation, press **Y** and then press <Enter>.

```
shovon@linuxhint:~$ sudo apt install virtualbox virtualbox-dkms virtualbox-qt virtualbox-ext-p
make manpages-dev qt5-gtk-platformtheme qttranslations5-l10n rpcsvc-proto
Suggested packages:
binutils-doc debtags menu debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib
autoconf automake libtool flex bison gcc-doc gcc-11-multilib gcc-11-locales glibc-doc git bzr
qt5-image-formats-plugins qtwayland5 libstdc++-11-doc make-doc vde2
virtualbox-guest-additions-iso
The following NEW packages will be installed:
binutils binutils-common binutils-x86_64-linux-gnu build-essential dkctrl-tools dkms dpkg-dev
fakeroot g++ g++-11 gcc gcc-11 libalgorithm-diff-perl libalgorithm-diff-xs-perl
libalgorithm-merge-perl libasan6 libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev
libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdouble-conversion3 libdpkg-perl libfakeroot
libfile-fcntllock-perl libgcc-11-dev libgsoap-2.8.11 libitm1 liblsan0 liblzfl libmd4c0
libnsl-dev libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5 libqt5network5 libqt5opengl5
libqt5printsupport5 libqt5svg5 libqt5widgets5 libqt5x11extras5 libssl1.2debian libstdc++-11-dev
libtirpc-dev libtsan0 libubsan1 libxcb-xinerama0 libxcb-xinput0 linux-libc-dev lto-disabled-list
make manpages-dev qt5-gtk-platformtheme qttranslations5-l10n rpcsvc-proto virtualbox
virtualbox-dkms virtualbox-ext-p virtualbox-qt
0 upgraded, 64 newly installed, 0 to remove and 123 not upgraded.
Need to get 113 MB of archives.
After this operation, 414 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

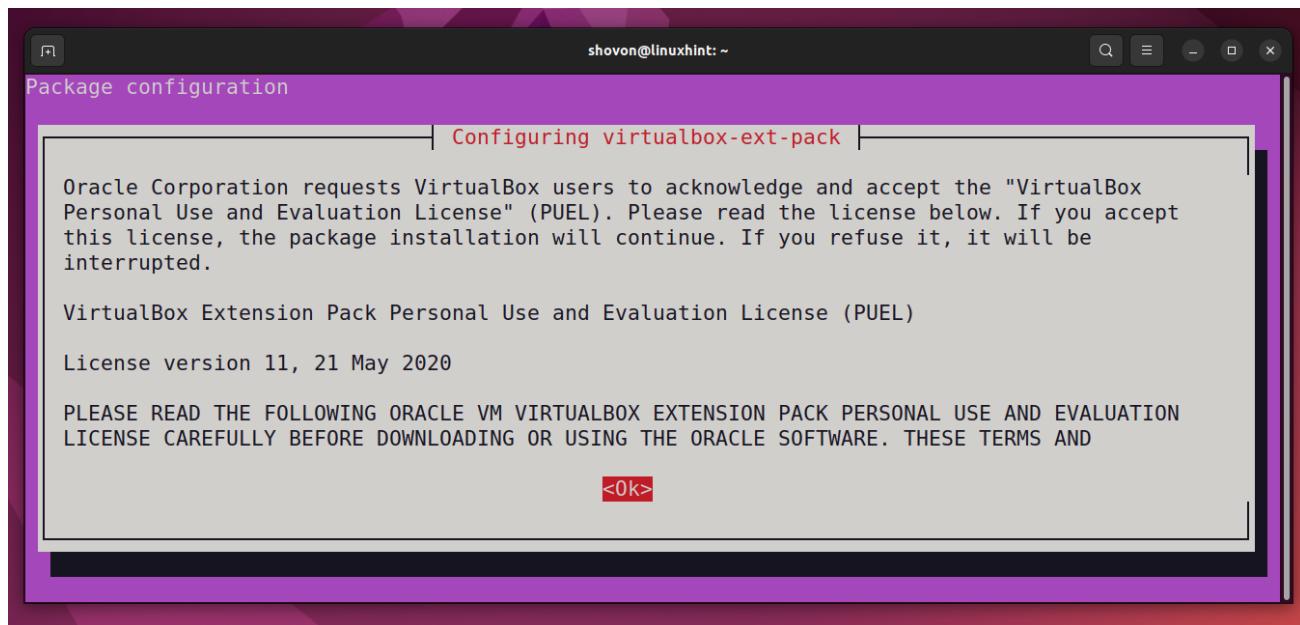
VirtualBox and all the required dependency packages are being downloaded from the internet. It will take a few minutes to complete.



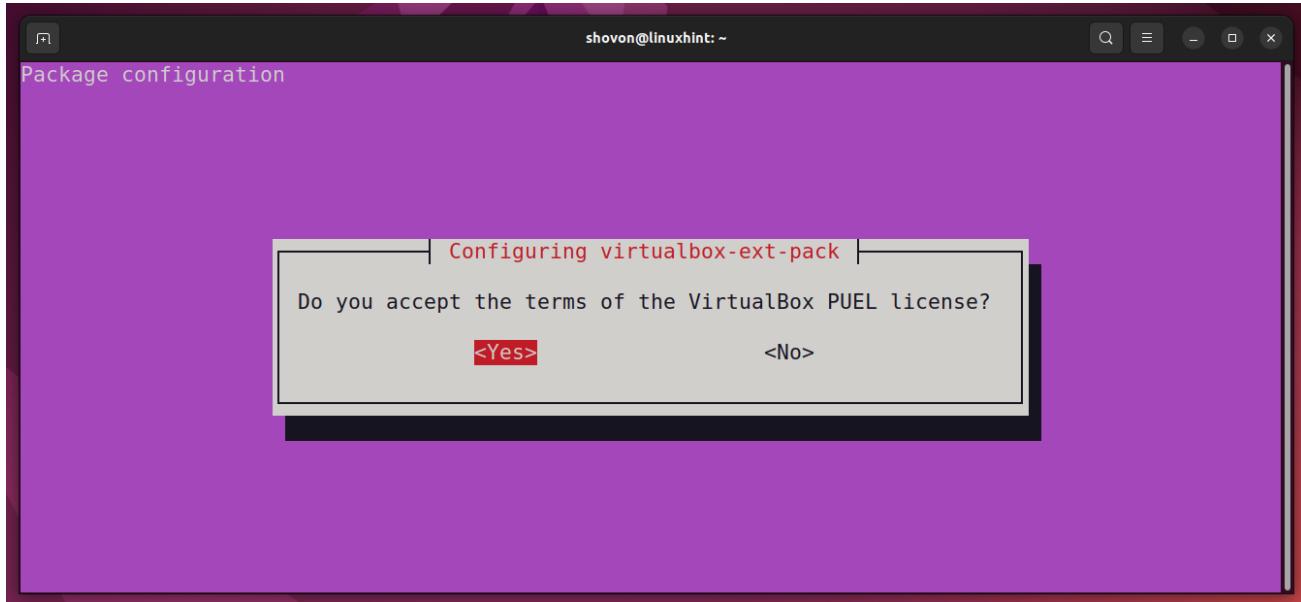
A terminal window titled 'shovon@linuxhint: ~' showing the output of a package manager command. The output includes a list of packages being upgraded, the number of newly installed packages (0), and the total disk space required (109 MB). It also shows the download progress for several packages from the 'http://us.archive.ubuntu.com/ubuntu' repository, including libasan6, liblsan0, libubsan1, libgcc-11-dev, and gcc-11, with speeds ranging from 178 kB/s to 20.1 MB/s and estimated times of 8min 26s.

```
libns1-dev libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5 libqt5network5 libqt5opengl5
libqt5printsupport5 libqt5svg5 libqt5widgets5 libqt5x11extras5 libssl1.2debian libstdc++-11-dev
libtirpc-dev libtsan0 libubsan1 libxcb-xinerama0 libxcb-xinput0 linux-libc-dev lto-disabled-list
make manpages-dev qt5-gtk-platformtheme qttranslations5-l10n rpcsvc-proto virtualbox
virtualbox-dkms virtualbox-ext-pack virtualbox-qt
0 upgraded, 64 newly installed, 0 to remove and 123 not upgraded.
Need to get 109 MB/113 MB of archives.
After this operation, 414 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libasan6 amd64 11.2.0-19ubuntu1 [2,283 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 liblsan0 amd64 12-20220319-1ubuntu1 [1,069 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libubsan1 amd64 11.2.0-19ubuntu1 [2,261 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libubsan1 amd64 12-20220319-1ubuntu1 [976 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libgcc-11-dev amd64 11.2.0-19ubuntu1 [2,526 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 gcc-11 amd64 11.2.0-19ubuntu1 [20.1 MB]
21% [6 gcc-11 10.2 MB/20.1 MB 51%] 178 kB/s 8min 26s
```

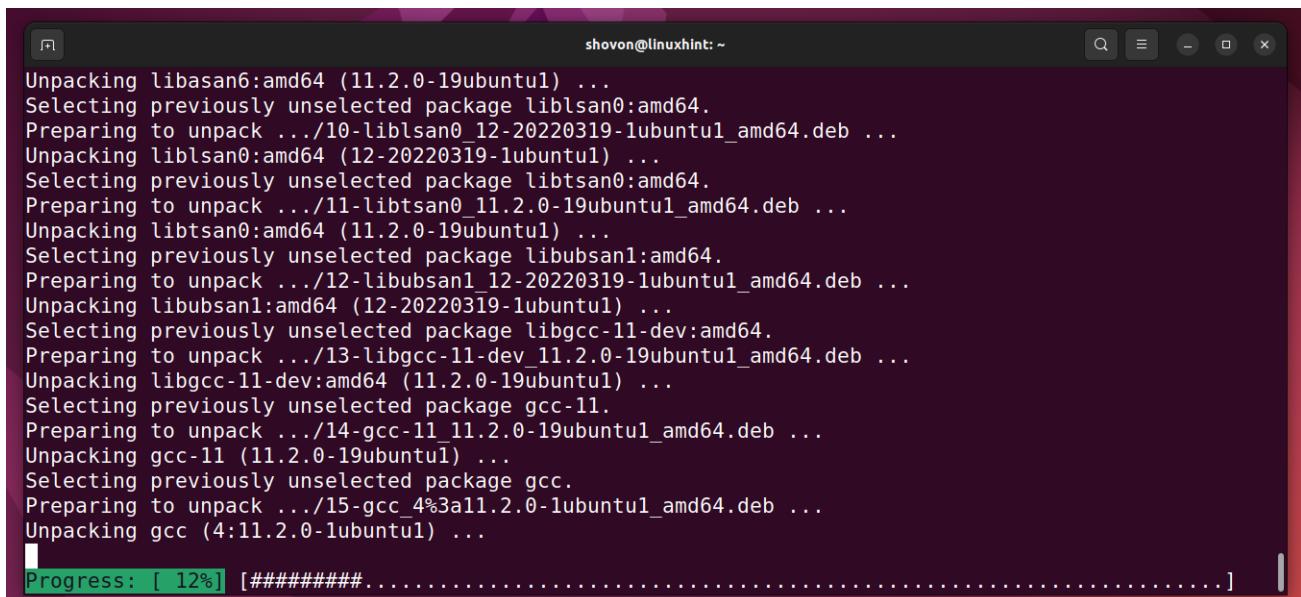
Select <Ok> (by pressing <Tab>) and press <Enter>.



To accept the VirtualBox Extension Pack license, select <Yes> and press <Enter>.



VirtualBox installation should continue. It will take a few minutes to complete the installation.



At this point, VirtualBox should be installed.

```
depmod...
Setting up virtualbox (6.1.32-dfsg-1build1) ...
Setting up virtualbox-ext-pack (6.1.32-1) ...
virtualbox-ext-pack: downloading: https://download.virtualbox.org/virtualbox/6.1.32/Oracle_VM_VirtualBox_Extension_Pack-6.1.32.vbox-extpack
The file will be downloaded into /usr/share/virtualbox-ext-pack
License accepted.
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Successfully installed "Oracle VM VirtualBox Extension Pack".
Setting up virtualbox-qt (6.1.32-dfsg-1build1) ...
Processing triggers for mailcap (3.70+nmulubuntu1) ...
Processing triggers for desktop-file-utils (0.26-lubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for shared-mime-info (2.1-2) ...
shovon@linuxhint:~$ █
```

To confirm whether VirtualBox is working, run the following command:

```
$ vboxheadless --version
```

```
shovon@linuxhint:~$ vboxheadless --version
Oracle VM VirtualBox Headless Interface 6.1.32_Ubuntu
(C) 2008-2022 Oracle Corporation
All rights reserved.

6.1.32_Ubuntur149290
shovon@linuxhint:~$ █
```

## Installing Minikube

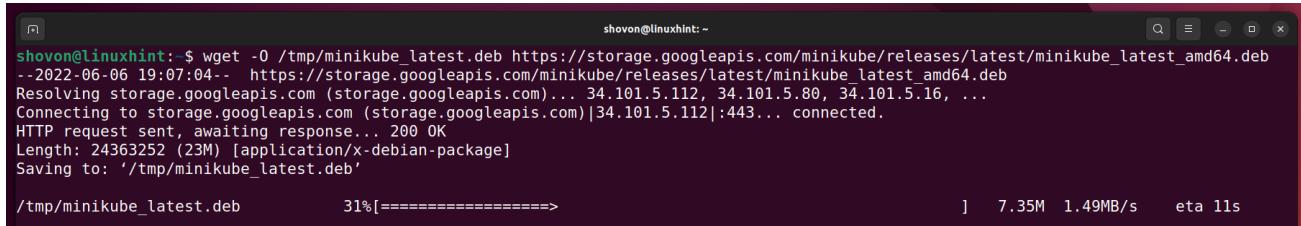
In this section, I am going to show you how to install the latest version of Minikube on Ubuntu 22.04 LTS.

To download the Minikube DEB package from the official website of Minikube as follows:

```
$ wget -O /tmp/minikube_latest.deb
https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
```

```
shovon@linuxhint:~$ wget -O /tmp/minikube_latest.deb https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
```

Minikube DEB package file **minikube\_latest\_amd64.deb** file is being downloaded. It will take a few seconds to complete.



```
shovon@linuxhint: $ wget -O /tmp/minikube_latest.deb https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
--2022-06-06 19:07:04-- https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
Resolving storage.googleapis.com (storage.googleapis.com)... 34.101.5.112, 34.101.5.80, 34.101.5.16, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|34.101.5.112|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24363252 (23M) [application/x-debian-package]
Saving to: '/tmp/minikube_latest.deb'

/tmp/minikube_latest.deb      31%[=====] 7.35M 1.49MB/s    eta 11s
```

Minikube DEB package file should be downloaded and saved in the **/tmp/** directory as **minikube\_latest.deb**.



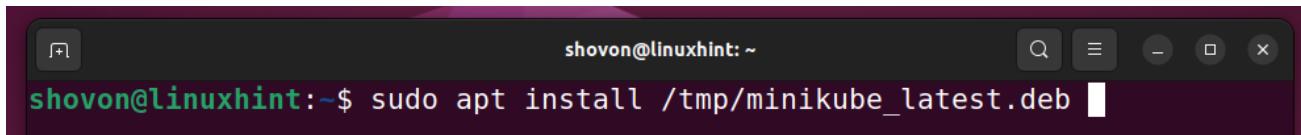
```
shovon@linuxhint: $ wget -O /tmp/minikube_latest.deb https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
--2022-06-06 19:07:04-- https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
Resolving storage.googleapis.com (storage.googleapis.com)... 34.101.5.112, 34.101.5.80, 34.101.5.16, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|34.101.5.112|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24363252 (23M) [application/x-debian-package]
Saving to: '/tmp/minikube_latest.deb'

/tmp/minikube_latest.deb      100%[=====] 23.23M 1.86MB/s    in 13s
2022-06-06 19:07:18 (1.74 MB/s) - '/tmp/minikube_latest.deb' saved [24363252/24363252]

shovon@linuxhint: $
```

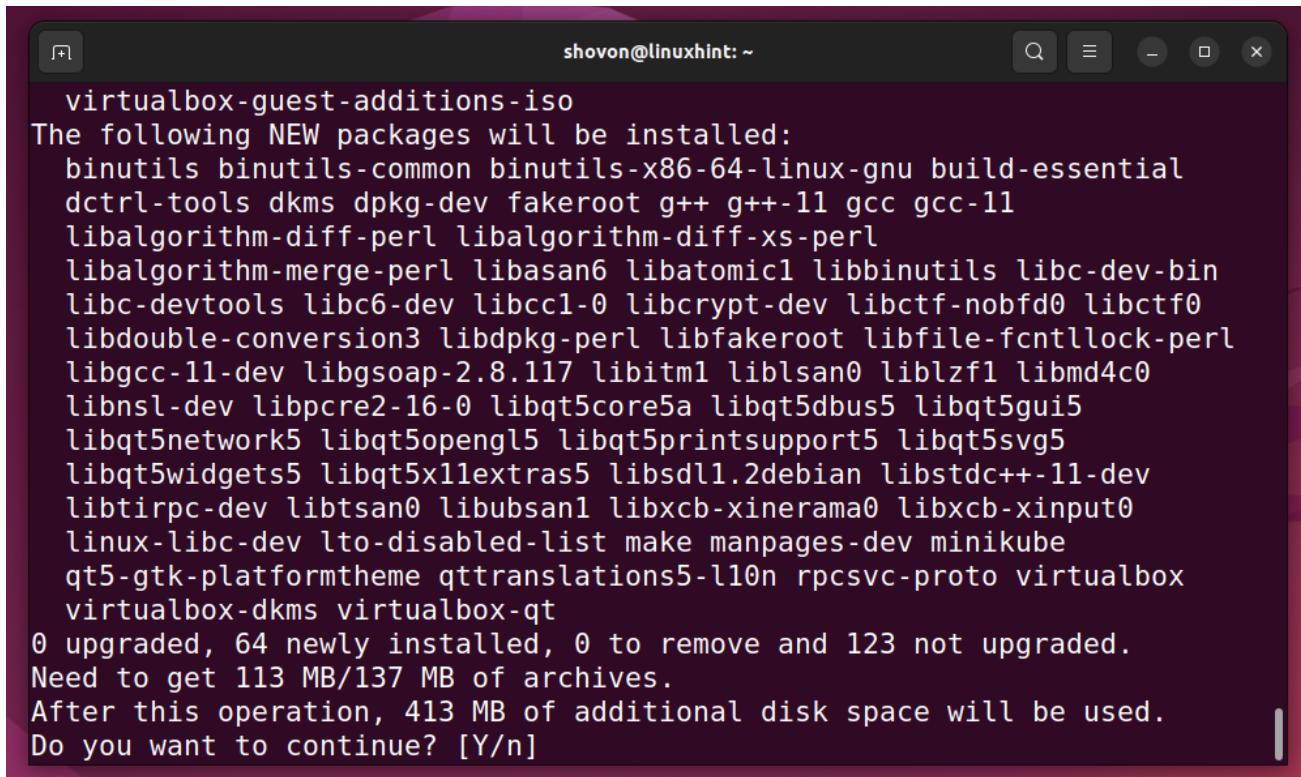
To install Minikube from the DEB package file **minikube\_latest.deb**, run the following command:

```
$ sudo apt install /tmp/minikube_latest.deb
```



```
shovon@linuxhint: ~$ sudo apt install /tmp/minikube_latest.deb
```

To confirm the installation, press **Y** and then press **<Enter>**.

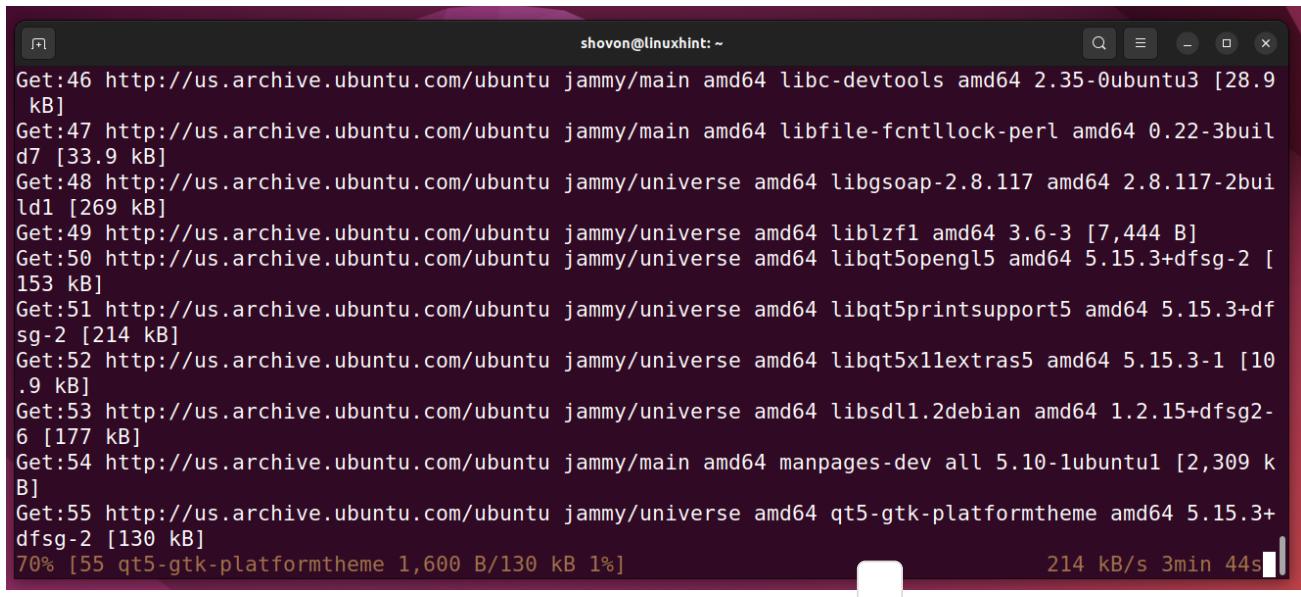


```

virtualbox-guest-additions-iso
The following NEW packages will be installed:
binutils binutils-common binutils-x86_64-linux-gnu build-essential
dctrl-tools dkms dpkg-dev fakeroot g++ g++-11 gcc gcc-11
libalgorithm-diff-perl libalgorithm-diff-xs-perl
libalgorithm-merge-perl libasan6 libatomic1 libbinutils libc-dev-bin
libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0
libdouble-conversion3 libdpkg-perl libfakeroot libfile-fcntllock-perl
libgcc-11-dev libgsoap-2.8.117 libhtml liblsan0 liblzf1 libmd4c0
libnsl-dev libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5
libqt5network5 libqt5opengl5 libqt5printsupport5 libqt5svg5
libqt5widgets5 libqt5x11extras5 libSDL1.2debian libstdc++-11-dev
libtirpc-dev libtsan0 libubsan1 libxcb-xinerama0 libxcb-xinput0
linux-libc-dev lto-disabled-list make manpages-dev minikube
qt5-gtk-platformtheme qttranslations5-l10n rpcsvc-proto virtualbox
virtualbox-dkms virtualbox-qt
0 upgraded, 64 newly installed, 0 to remove and 123 not upgraded.
Need to get 113 MB/137 MB of archives.
After this operation, 413 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

All the required dependency packages are being downloaded from the internet. It will take a few minutes to complete.



```

shovon@linuxhint: ~
Get:46 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libc-devtools amd64 2.35-0ubuntu3 [28.9 kB]
Get:47 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libfile-fcntllock-perl amd64 0.22-3buil
d7 [33.9 kB]
Get:48 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libgsoap-2.8.117 amd64 2.8.117-2bui
ld1 [269 kB]
Get:49 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 liblzf1 amd64 3.6-3 [7,444 B]
Get:50 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5opengl5 amd64 5.15.3+dfsg-2 [153 kB]
Get:51 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5printsupport5 amd64 5.15.3+df
sg-2 [214 kB]
Get:52 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5x11extras5 amd64 5.15.3-1 [10
.9 kB]
Get:53 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libSDL1.2debian amd64 1.2.15+dfsg2-
6 [177 kB]
Get:54 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 manpages-dev all 5.10-1ubuntul [2,309 k
B]
Get:55 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 qt5-gtk-platformtheme amd64 5.15.3+
dfsg-2 [130 kB]
70% [55 qt5-gtk-platformtheme 1,600 B/130 kB 1%]
214 kB/s 3min 44s

```

Minikube and the required dependency packages are being installed. It will take a few minutes to complete.

```
shovon@linuxhint: ~
Selecting previously unselected package libasan6:amd64.
Preparing to unpack .../09-libasan6_11.2.0-19ubuntu1_amd64.deb ...
Unpacking libasan6:amd64 (11.2.0-19ubuntu1) ...
Selecting previously unselected package liblsan0:amd64.
Preparing to unpack .../10-liblsan0_12-20220319-1ubuntu1_amd64.deb ...
Unpacking liblsan0:amd64 (12-20220319-1ubuntu1) ...
Selecting previously unselected package libtsan0:amd64.
Preparing to unpack .../11-libtsan0_11.2.0-19ubuntu1_amd64.deb ...
Unpacking libtsan0:amd64 (11.2.0-19ubuntu1) ...
Selecting previously unselected package libubsan1:amd64.
Preparing to unpack .../12-libubsan1_12-20220319-1ubuntu1_amd64.deb ...
Unpacking libubsan1:amd64 (12-20220319-1ubuntu1) ...
Selecting previously unselected package libgcc-11-dev:amd64.
Preparing to unpack .../13-libgcc-11-dev_11.2.0-19ubuntu1_amd64.deb ...
Unpacking libgcc-11-dev:amd64 (11.2.0-19ubuntu1) ...
Selecting previously unselected package gcc-11.
Preparing to unpack .../14-gcc-11_11.2.0-19ubuntu1_amd64.deb ...
Unpacking gcc-11 (11.2.0-19ubuntu1) ...

Progress: [ 11%] [#####.....]
```

Minikube should be installed at this point.

```
vboxnetflt.ko:
Running module version sanity check.
- Original module
  - No original module exists within this kernel
- Installation
  - Installing to /lib/modules/5.15.0-25-generic/updates/dkms/
depmod.....
Setting up virtualbox (6.1.32-dfsg-1build1) ...
Setting up virtualbox-qt (6.1.32-dfsg-1build1) ...
Processing triggers for mailcap (3.70+nmulubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for shared-mime-info (2.1-2) ...
shovon@linuxhint:~$
```

To check whether Minikube is installed, run the following command:

```
$ minikube version
```

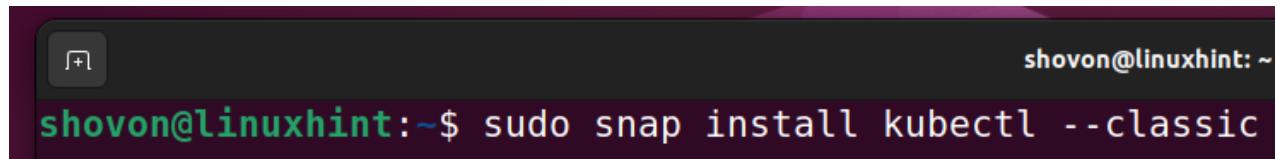
```
shovon@linuxhint: ~
shovon@linuxhint:~$ minikube version
minikube version: v1.25.2
commit: 362d5fdc0a3dbe389b3d3f1034e8023e72bd3a7
shovon@linuxhint:~$
```

## Installing kubectl – The Kubernetes Command-Line Tool

In this section, I am going to show you how to install the Kubernetes command-line tool **kubectl** on Ubuntu 22.04 LTS. `kubectl` is used to inspect and manage the Kubernetes cluster resources and view cluster logs. It's also used to deploy, configure, and manage applications on the Kubernetes cluster.

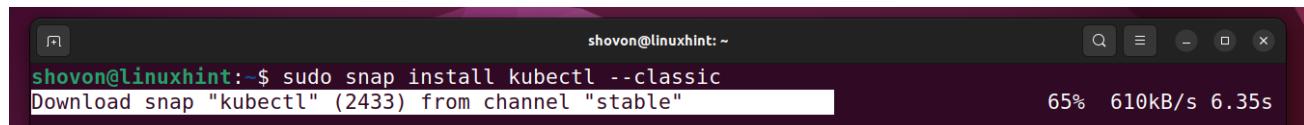
On Ubuntu 22.04 LTS, you can install the latest version of Kubernetes command-line tool **kubectl** from the snap store as follows:

```
$ sudo snap install kubectl --classic
```



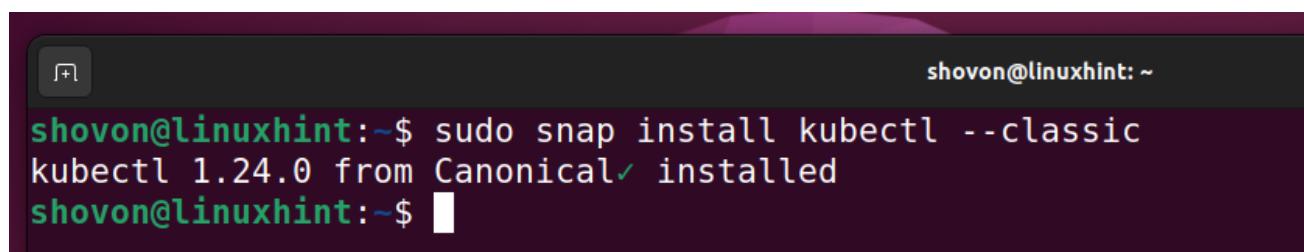
```
shovon@linuxhint:~$ sudo snap install kubectl --classic
```

`kubectl` is being downloaded and installed. It will take a few seconds to complete.



```
shovon@linuxhint:~$ sudo snap install kubectl --classic
Download snap "kubectl" (2433) from channel "stable"
65% 610kB/s 6.35s
```

At this point, `kubectl` should be installed.



```
shovon@linuxhint:~$ sudo snap install kubectl --classic
kubectl 1.24.0 from Canonical✓ installed
shovon@linuxhint:~$
```

## Starting a Kubernetes Cluster with Minikube

In this section, I am going to show you how to start a local Kubernetes cluster with Minikube using the Docker, KVM/QEMU, and VirtualBox drivers.

If you want to start a local Kubernetes cluster with Minikube using the Docker driver, run the following command:

**NOTE:** You must have Docker installed for this to work. If you need any assistance, read the [Installing Docker on Ubuntu 22.04 LTS](#) section of this article.

```
$ minikube start --driver=docker
```

```
shovon@linuxhint:~$ minikube start --driver=docker
```

If you want to start a local Kubernetes cluster with Minikube using the KVM/QEMU driver, run the following command:

**NOTE:** You must have KVM/QEMU installed for this to work. If you need any assistance, read the [Installing KVM/QEMU on Ubuntu 22.04 LTS](#) section of this article.

```
$ minikube start --driver=kvm2
```

```
shovon@linuxhint:~$ minikube start --driver=kvm2
```

If you want to start a local Kubernetes cluster with Minikube using the VirtualBox driver, run the following command:

**NOTE:** You must have VirtualBox installed for this to work. If you need any assistance, read the [Installing VirtualBox on Ubuntu 22.04 LTS](#) section of this article.

```
$ minikube start --driver=virtualbox
```

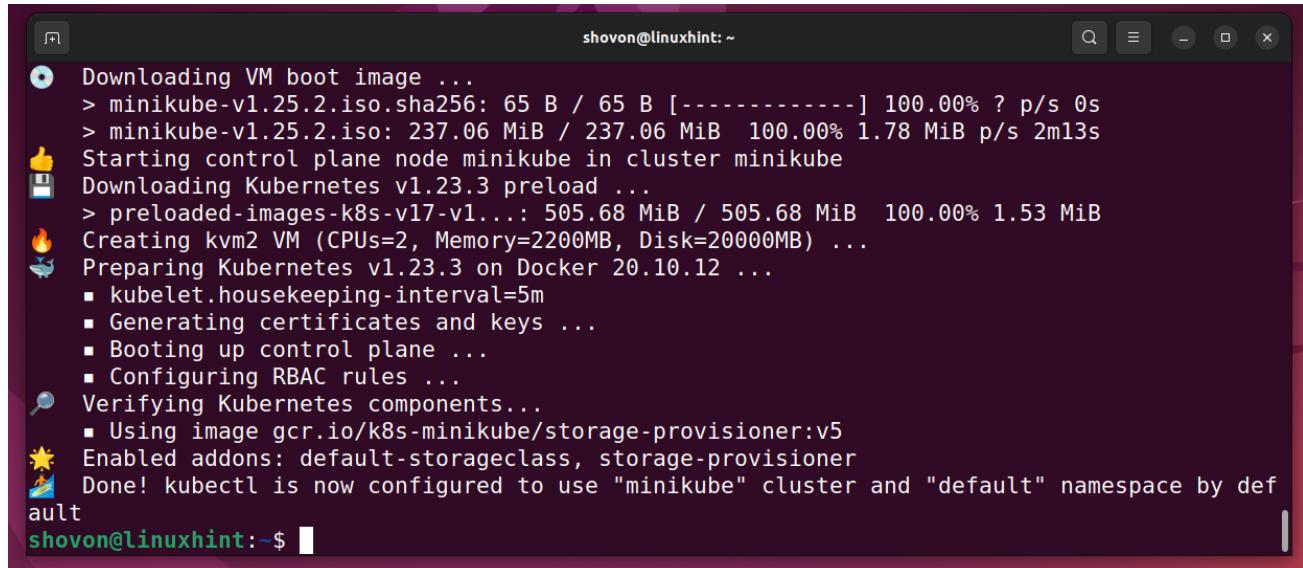
```
shovon@linuxhint:~$ minikube start --driver=virtualbox
```

**NOTE:** I will use the KVM/QEMU driver in this article for the demonstration. The other drivers will work the same way.

Minikube is downloading all the required files to set up a local Kubernetes cluster. It will take a while to complete.

```
shovon@linuxhint:~$ minikube start --driver=kvm2
minikube v1.25.2 on Ubuntu 22.04
Using the kvm2 driver based on user configuration
Downloading driver docker-machine-driver-kvm2:
> docker-machine-driver-kvm2-....: 65 B / 65 B [-----] 100.00% ? p/s 0s
> docker-machine-driver-kvm2-....: 11.62 MiB / 11.62 MiB 100.00% 2.00 MiB p
Downloading VM boot image ...
> minikube-v1.25.2.iso.sha256: 65 B / 65 B [-----] 100.00% ? p/s 0s
> minikube-v1.25.2.iso: 10.11 MiB / 237.06 MiB 4.26% 2.11 MiB p/s ETA 1m47s
```

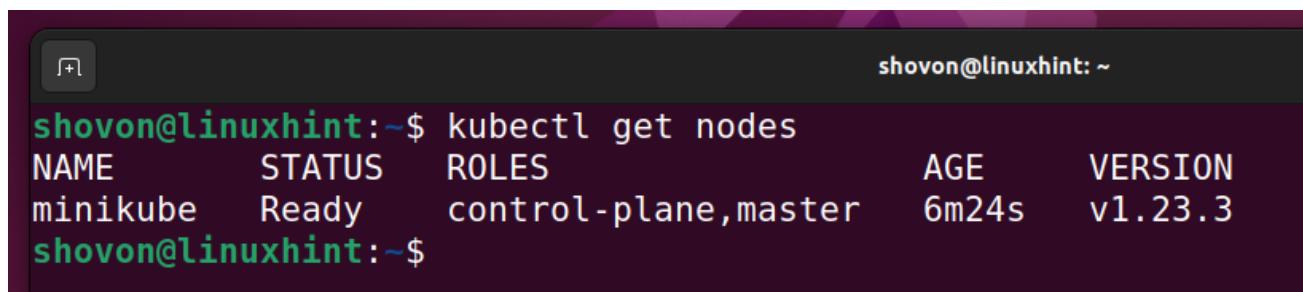
At this point, Minikube should create and configure a local Kubernetes cluster.



```
shovon@linuxhint:~$ minikube start
● Downloading VM boot image ...
> minikube-v1.25.2.iso.sha256: 65 B / 65 B [-----] 100.00% ? p/s 0s
> minikube-v1.25.2.iso: 237.06 MiB / 237.06 MiB 100.00% 1.78 MiB p/s 2m13s
Starting control plane node minikube in cluster minikube
● Downloading Kubernetes v1.23.3 preload ...
> preloaded-images-k8s-v17-v1...: 505.68 MiB / 505.68 MiB 100.00% 1.53 MiB
Creating kvm2 VM (CPUs=2, Memory=2200MB, Disk=20000MB) ...
● Preparing Kubernetes v1.23.3 on Docker 20.10.12 ...
  ■ kubelet.housekeeping-interval=5m
  ■ Generating certificates and keys ...
  ■ Booting up control plane ...
  ■ Configuring RBAC rules ...
● Verifying Kubernetes components...
  ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
● Enabled addons: default-storageclass, storage-provisioner
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
shovon@linuxhint:~$
```

As you can see, Minikube has created a new Kubernetes cluster.

```
$ kubectl get nodes
```



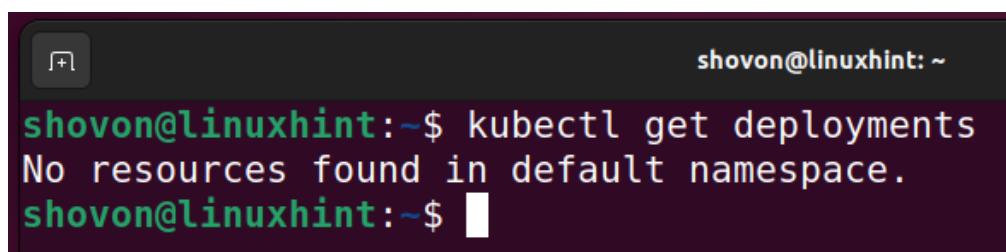
```
shovon@linuxhint:~$ kubectl get nodes
NAME      STATUS   ROLES      AGE      VERSION
minikube  Ready    control-plane,master  6m24s   v1.23.3
shovon@linuxhint:~$
```

## Basics of Kubernetes

In this section, I am going to show you the basics of Kubernetes. It should help you get started with Kubernetes.

As you have just created a new Kubernetes cluster with Minikube, you won't see any deployments as of yet.

```
$ kubectl get deployments
```



```
shovon@linuxhint:~$ kubectl get deployments
No resources found in default namespace.
shovon@linuxhint:~$
```

To create a new **echoserver** deployment **hello-minikube**, run the following command:

```
$ kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4
```

```
shovon@linuxhint:~$ kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4
deployment.apps/hello-minikube created
shovon@linuxhint:~$
```

A new deployment **hello-minikube** should be created, as you can see in the screenshot below.

```
$ kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
hello-minikube	1/1	1	1	7s

```
shovon@linuxhint:~$ kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
hello-minikube   1/1     1           1          7s
shovon@linuxhint:~$
```

To expose the deployment **hello-minikube** on the port **8080**, run the following command:

```
$ kubectl expose deployment hello-minikube --type=NodePort --port=8080
```

```
shovon@linuxhint:~$ kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
shovon@linuxhint:~$
```

A new service should be created for the **hello-minikube** deployment, as you can see in the screenshot below. This service will expose the port **8080** of the **hello-minikube** deployment.

```
$ kubectl get services hello-minikube
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
hello-minikube	NodePort	10.108.149.81	<none>	8080:30653/TCP	16m

```
shovon@linuxhint:~$ kubectl get services hello-minikube
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
hello-minikube   NodePort    10.108.149.81    <none>        8080:30653/TCP   16m
shovon@linuxhint:~$
```

To find the service URL of the **hello-minikube** deployment, run the following command:

```
$ minikube service hello-minikube --url
```

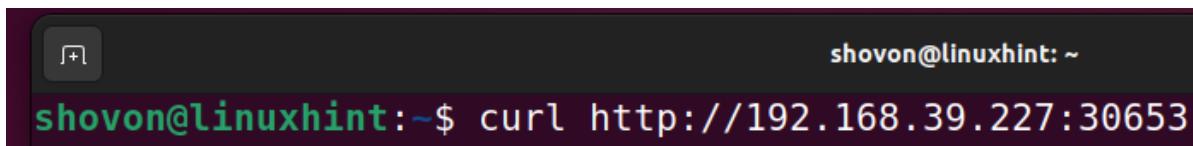


```
shovon@linuxhint:~$ minikube service hello-minikube --url
http://192.168.39.227:30653
shovon@linuxhint:~$ █
```

A green arrow points from the URL output of the previous command to the terminal prompt.

Once you know the service URL of the **hello-minikube** deployment, you can access the service with **curl** as follows:

```
$ curl http://192.168.39.227:30653
```



```
shovon@linuxhint:~$ curl http://192.168.39.227:30653
```

If everything works, you will see the following output. It means the service of the **hello-minikube** deployment is accessible.

**NOTE:** If you don't have **curl** installed on your computer or want to learn how to use curl, read [this article](#).

```
shovon@linuxhint:~$ curl http://192.168.39.227:30653
CLIENT VALUES:
client_address=172.17.0.1
command=GET
real path=/
query=nil
request_version=1.1
request_uri=http://192.168.39.227:8080/

SERVER VALUES:
server_version=nginx: 1.10.0 - lua: 10001

HEADERS RECEIVED:
accept=*/
host=192.168.39.227:30653
user-agent=curl/7.81.0
BODY:
shovon@linuxhint:~$
```

You can also forward the port **8080** of the **hello-minikube** deployment to the port **7080** (let's say) of your computer as follows:

```
$ kubectl port-forward service/hello-minikube 7080:8080
```

```
shovon@linuxhint:~$ kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::1]:7080 -> 8080
[
```

Now, you should be able to access the service of the deployment **hello-minikube** using the URL <http://localhost:7080>.

```
$ curl http://localhost:7080
```

```
shovon@linuxhint:~$ curl http://localhost:7080
CLIENT VALUES:
client_address=127.0.0.1
command=GET
real_path=/
query=nil
request_version=1.1
request_uri=http://localhost:8080/

SERVER VALUES:
server_version=nginx: 1.10.0 - lua: 10001

HEADERS RECEIVED:
accept=*/
host=localhost:7080
user-agent=curl/7.81.0
BODY:
-no body in request-shovon@linuxhint:~$
```

You can find a lot of information on each of the services of your Kubernetes deployments with the following command:

```
$ kubectl get services
```

```
shovon@linuxhint:~$ kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
hello-minikube   NodePort    10.108.149.81    <none>        8080:30653/TCP   19m
kubernetes   ClusterIP  10.96.0.1       <none>        443/TCP     11h
shovon@linuxhint:~$
```

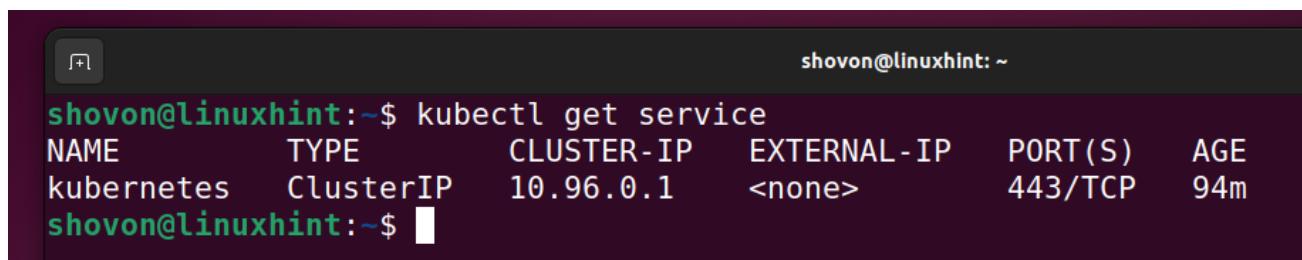
To delete a service **hello-minikube** from the **hello-minikube** deployment, run the following command:

```
$ kubectl delete service hello-minikube
```

```
shovon@linuxhint:~$ kubectl delete service hello-minikube
service "hello-minikube" deleted
shovon@linuxhint:~$
```

The **hello-minikube** service should be removed, as you can see in the screenshot below.

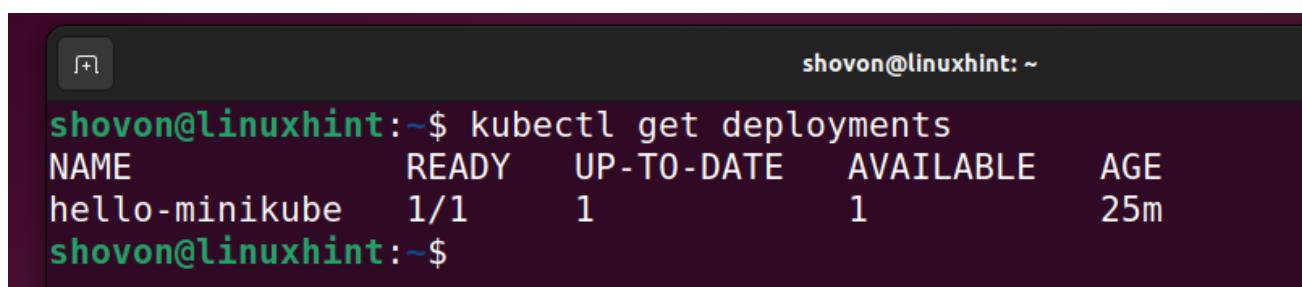
```
$ kubectl delete service hello-minikube
```



```
shovon@linuxhint:~$ kubectl get service
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP  94m
shovon@linuxhint:~$
```

Right now, I have only the **hello-minikube** deployment. Let's delete it.

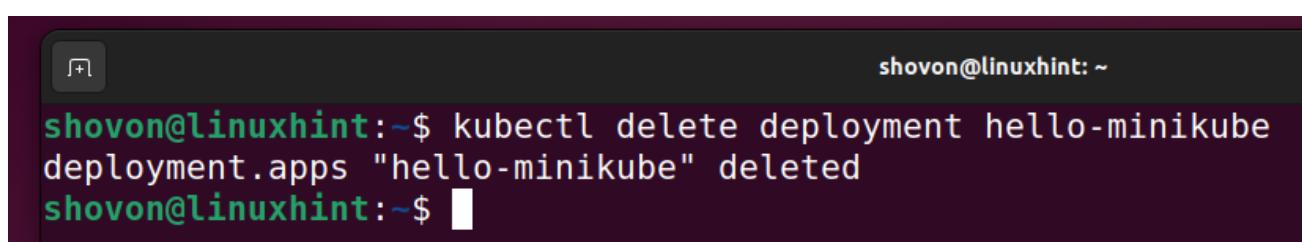
```
$ kubectl get deployments
```



```
shovon@linuxhint:~$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
hello-minikube 1/1     1           1           25m
shovon@linuxhint:~$
```

To delete the **hello-minikube** deployment, run the following command:

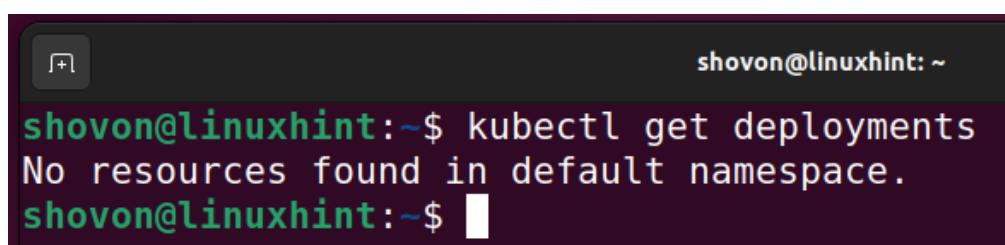
```
$ kubectl delete deployment hello-minikube
```



```
shovon@linuxhint:~$ kubectl delete deployment hello-minikube
deployment.apps "hello-minikube" deleted
shovon@linuxhint:~$
```

The **hello-minikube** deployment should be removed from the Kubernetes deployment list, as you can see in the screenshot below.

```
$ kubectl get deployments
```



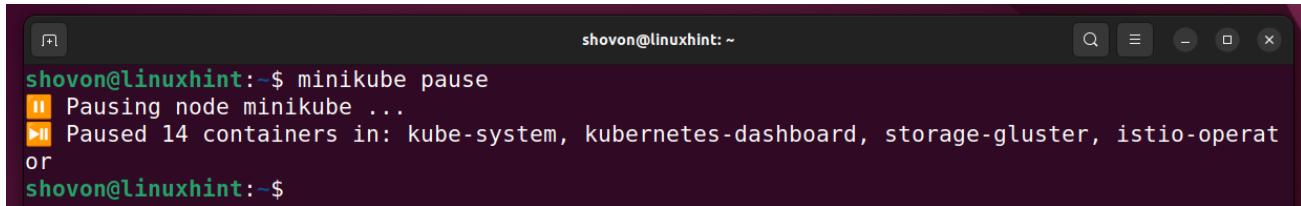
```
shovon@linuxhint:~$ kubectl get deployments
No resources found in default namespace.
shovon@linuxhint:~$
```

## Managing Minikube Kubernetes Cluster

You can start, stop, pause, and resume/unpause the Kubernetes cluster that you've created with Minikube.

To pause the Kubernetes cluster with Minikube, run the following command:

```
$ minikube pause
```

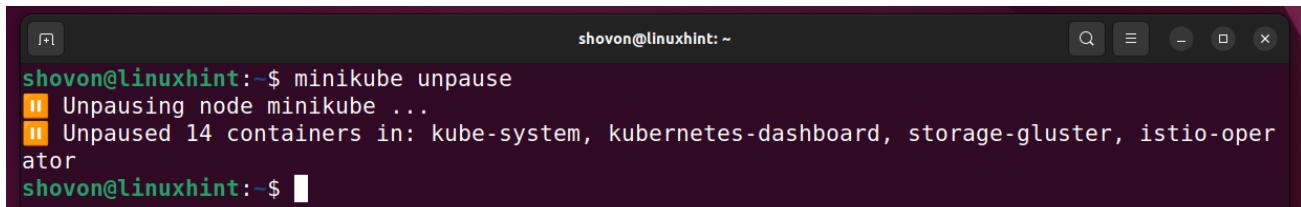


```
shovon@linuxhint:~$ minikube pause
[!] Pausing node minikube ...
[!] Paused 14 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
or
shovon@linuxhint:~$
```

A screenshot of a terminal window titled "shovon@linuxhint: ~". It shows the command "minikube pause" being run. The output indicates that the node "minikube" is being paused and that 14 containers are being paused across various namespaces: kube-system, kubernetes-dashboard, storage-gluster, and istio-operator.

To resume the Kubernetes cluster with Minikube, run the following command:

```
$ minikube unpause
```

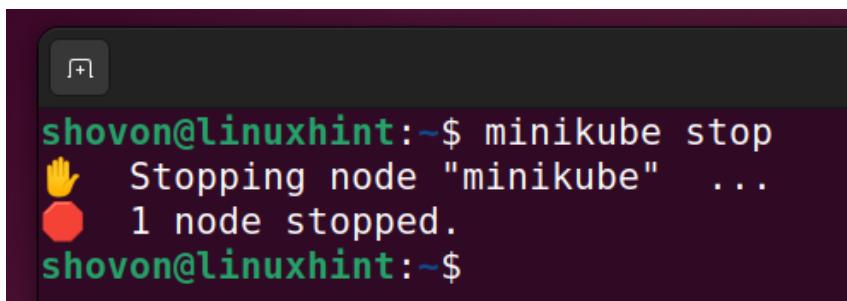


```
shovon@linuxhint:~$ minikube unpause
[!] Unpausing node minikube ...
[!] Unpaused 14 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
shovon@linuxhint:~$
```

A screenshot of a terminal window titled "shovon@linuxhint: ~". It shows the command "minikube unpause" being run. The output indicates that the node "minikube" is being unpause and that 14 containers are being unpause across various namespaces: kube-system, kubernetes-dashboard, storage-gluster, and istio-operator.

To stop the Kubernetes cluster with Minikube, run the following command:

```
$ minikube stop
```



```
shovon@linuxhint:~$ minikube stop
✋ Stopping node "minikube" ...
🛑 1 node stopped.
shovon@linuxhint:~$
```

A screenshot of a terminal window titled "shovon@linuxhint: ~". It shows the command "minikube stop" being run. The output indicates that the node "minikube" is being stopped, and one node has been stopped.

Once the Kubernetes cluster is stopped, you can start it again with Minikube. If you don't remember how to start a Kubernetes cluster with Minikube, read the [Starting a Kubernetes Cluster with Minikube](#) section of this article.

## Conclusion

In this article, I have shown you how to install **Minikube** and Kubernetes command-line tool **kubectl** on Ubuntu 22.04 LTS and how to create a local Kubernetes cluster with Minikube using the Docker, KVM/QEMU, and VirtualBox drivers. I have also shown you the basics of

using Kubernetes on your computer. This should help you get started with Kubernetes and Minikube.

## References

1. [minikube start | minikube](#)
2. [Drivers | minikube](#)
3. [docker | minikube](#)
4. [kvm2 | minikube](#)
5. [virtualbox | minikube](#)

## Explore More

search text



[SHOW ALL](#)

#kubernetes

## ABOUT THE AUTHOR



### Shahriar Shovon

Freelancer & Linux System Administrator. Also loves Web API development with Node.js and JavaScript. I was born in Bangladesh. I am currently studying Electronics and Communication Engineering at Khulna University of Engineering & Technology (KUET), one of the demanding public engineering universities of Bangladesh.

[View all posts](#)

## RELATED LINUX HINT POSTS

## Kubernetes HostPath Volumes

[How to Use Linux sysctls in Kubernetes](#)

[How to Set Different Kubernetes Restart Policies](#)

[How to Create Local Persistent Volume in Kubernetes](#)

[How to Create Kubernetes Network Policies](#)

[How to Configure Event Rate Limit in Kubernetes](#)

[How to Configure Deny Service External IPs in Kubernetes](#)

Linux Hint LLC, editor@linuxhint.com  
1309 S Mary Ave Suite 210, Sunnyvale, CA  
94087  
[Privacy Policy](#) and [Terms of Use](#)