

INDEX

S.No.	LAB	Page
1.	Create the Ubuntu instance by using the VMWare.	2
2.	Configure the KVM virtualization technology.	7
3.	Create the Cirros-6.x instance by using the KVM virtualization technology.	8
4.	Create the OpenV-switch by “sw-1” name.	10
5.	Create the container by using the centos image.	11
6.	Create the 1GB volume on “/BCMP” mount point with ext4 file-system.	12
7.	Create the Ubuntu instance in a Public	15

Q1. Create the Ubuntu instance by using the VMWare.

Step-1: Download VMware Workstation

Visit the [official VMware website](https://www.vmware.com/resources/downloadcenter/download) and download the latest version of VMware Workstation. You may need to create a VMware account and log in to access the download.

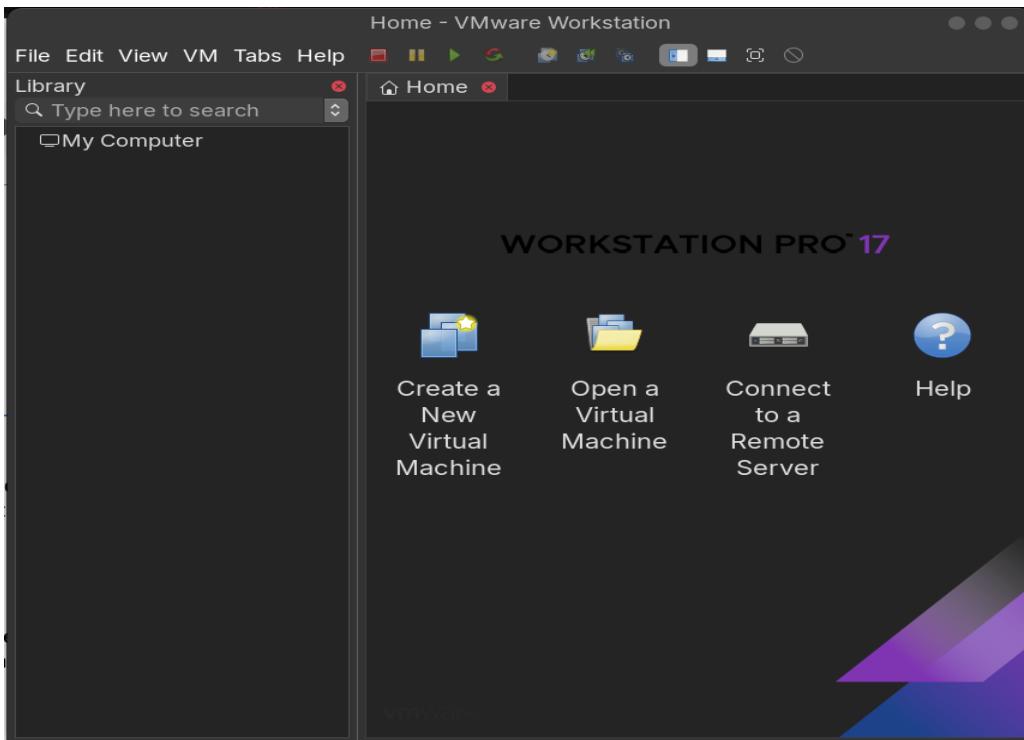
Step-2: Install VMware Workstation

Run the installer you downloaded in Step 1. Follow the on-screen instructions to install VMware Workstation on your computer. You may need to restart your system after the installation is complete.

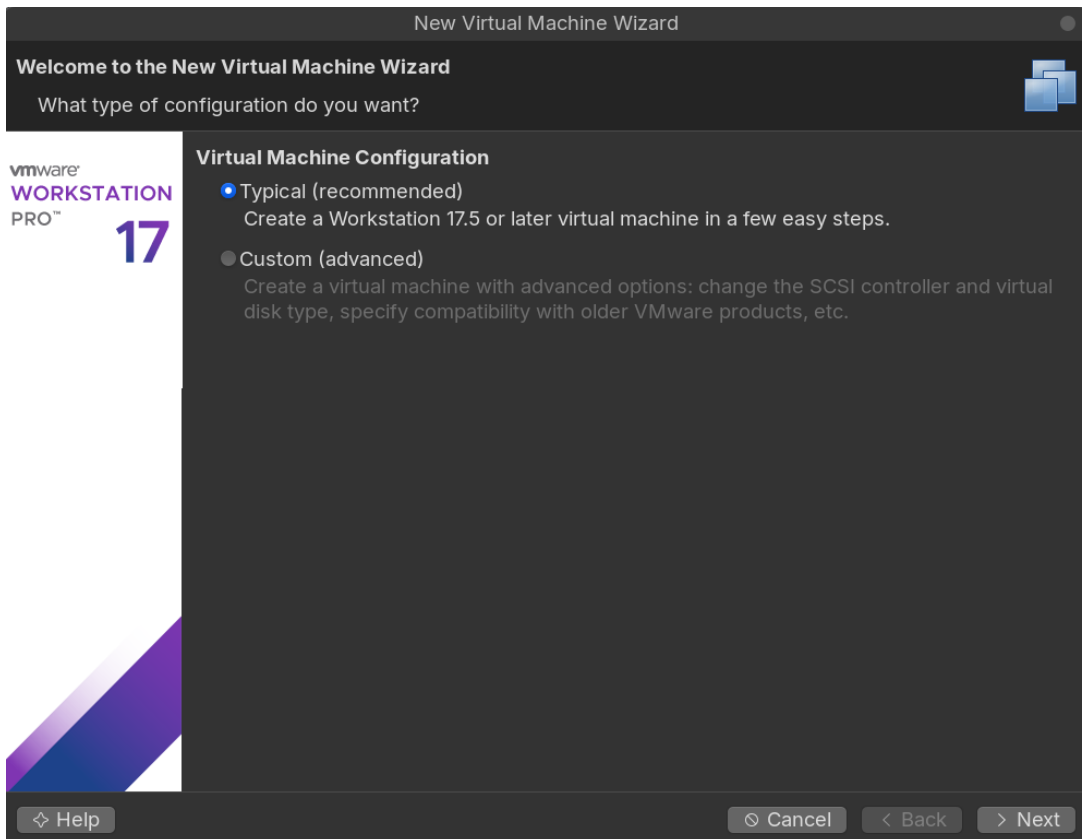
Step 3: Download Ubuntu ISO

Visit the official Ubuntu website <https://ubuntu.com/download/desktop> and download the latest version of the Ubuntu Desktop ISO. Choose the appropriate architecture (32-bit or 64-bit) based on your system.

Step 4: Create a New Virtual Machine in VMware Workstation

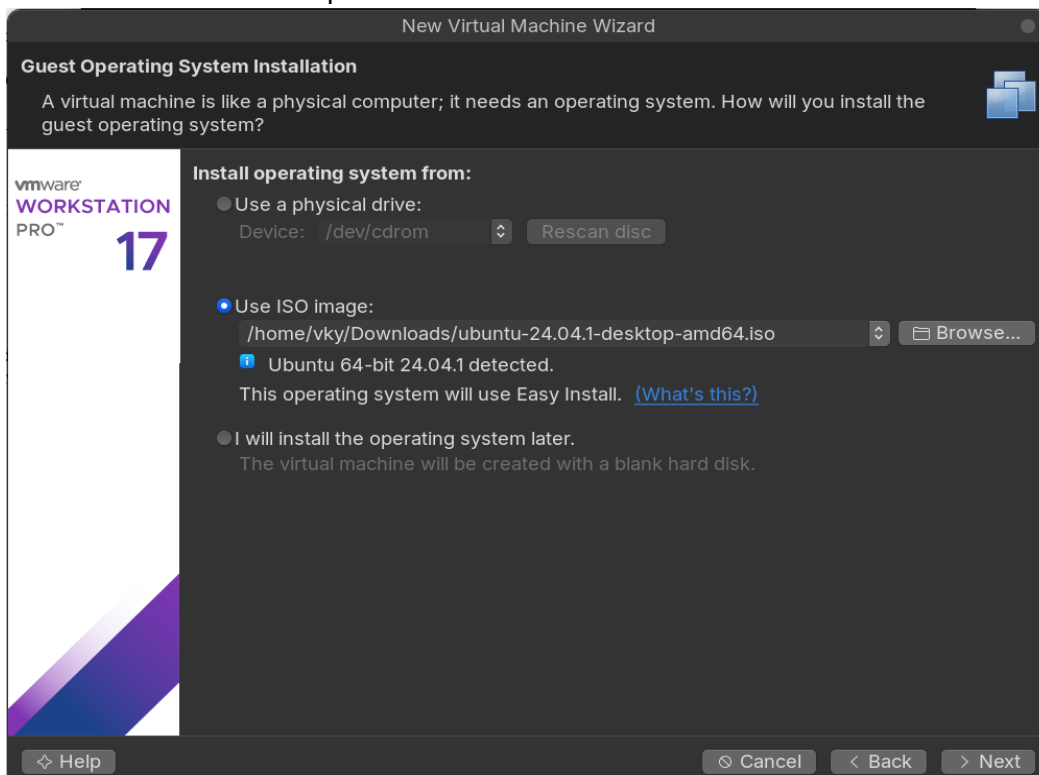


1. Open VMware Workstation.
2. Click on "File" in the menu and select "New Virtual Machine."
3. The New Virtual Machine Wizard will open. Choose "Typical" and click "Next."



Step 5: Specify the Ubuntu ISO

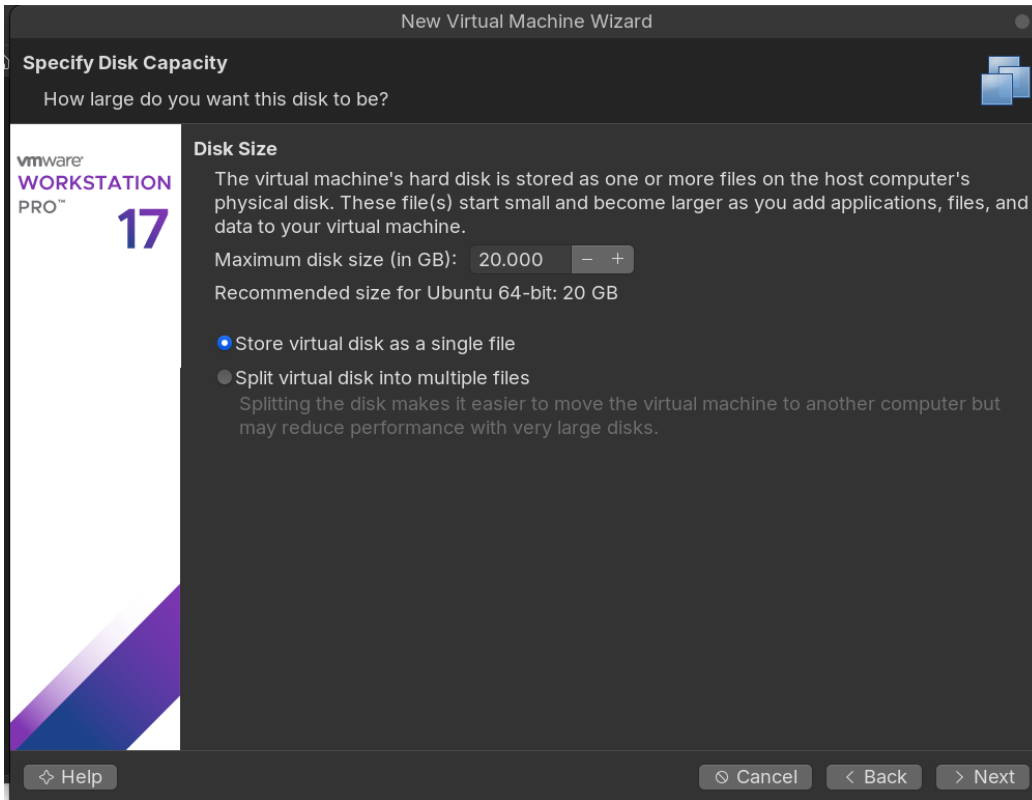
1. Select "Installer disc image file (iso)" and click "Browse."
2. Navigate to the location where you saved the Ubuntu ISO file and select it.
3. Click "Next" to proceed.



Step 6: Name and Specify Storage

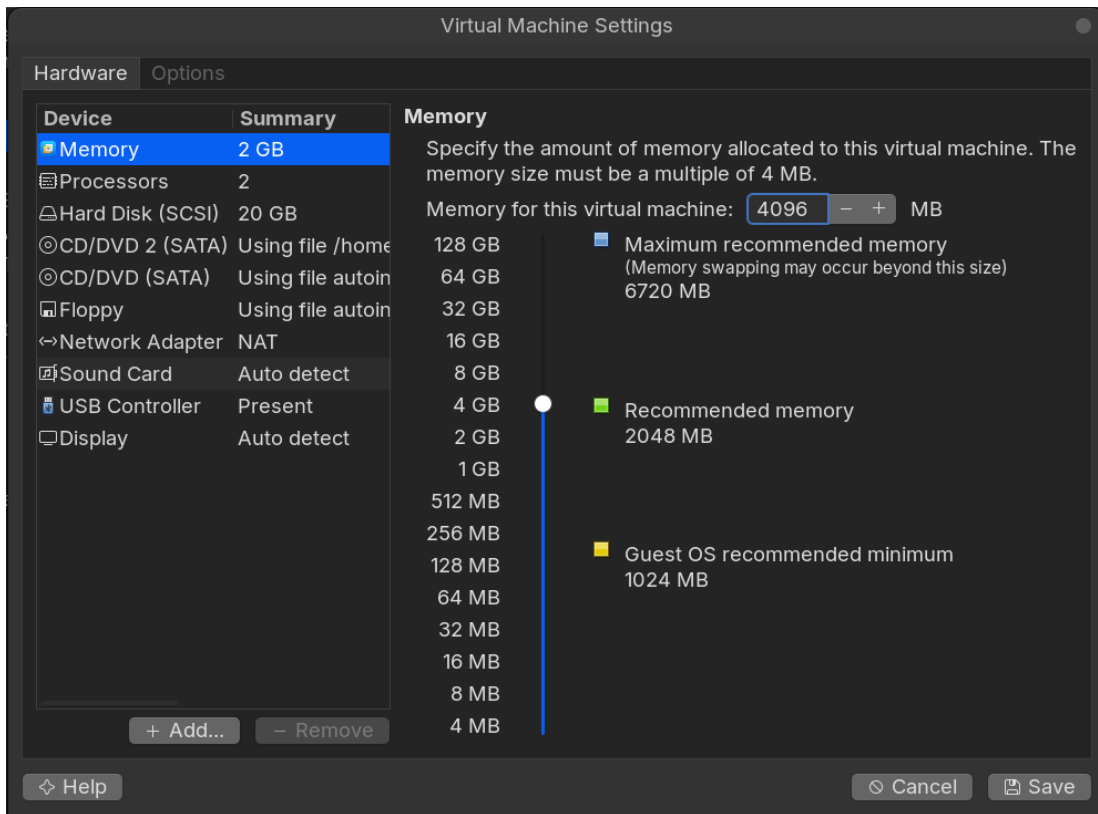
1. Enter a name for your virtual machine.

2. Choose a location to store your virtual machine files.
3. Specify the disk capacity (at least 25 GB is recommended).
4. Select “Store virtual disk as a single file.”
5. Click “Next” to continue.



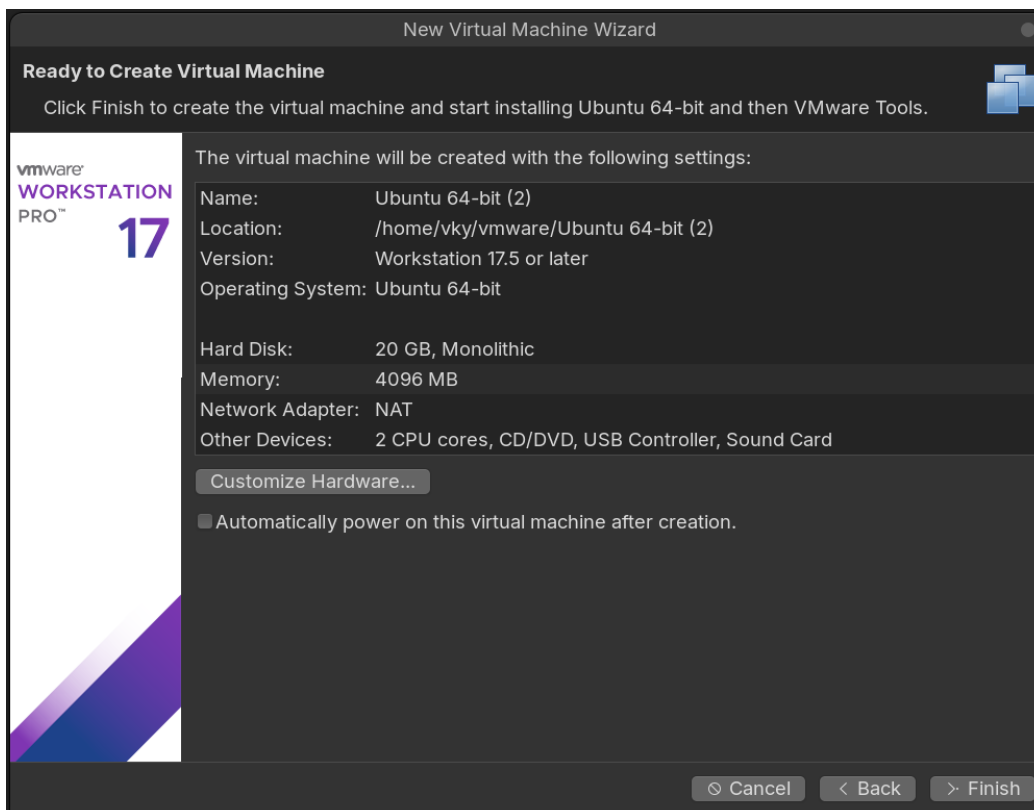
Step 7: Customize Hardware (Optional)

You can customize the hardware settings if needed. Adjust the number of processors, memory, and other settings based on your system's capabilities.



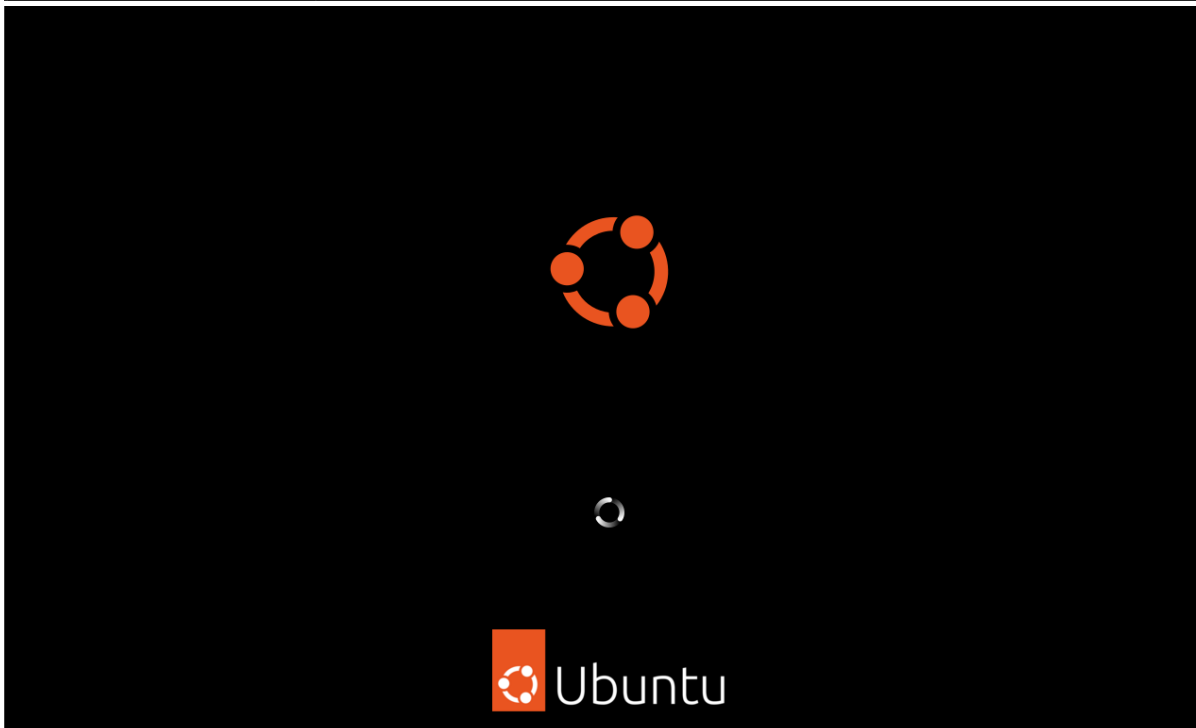
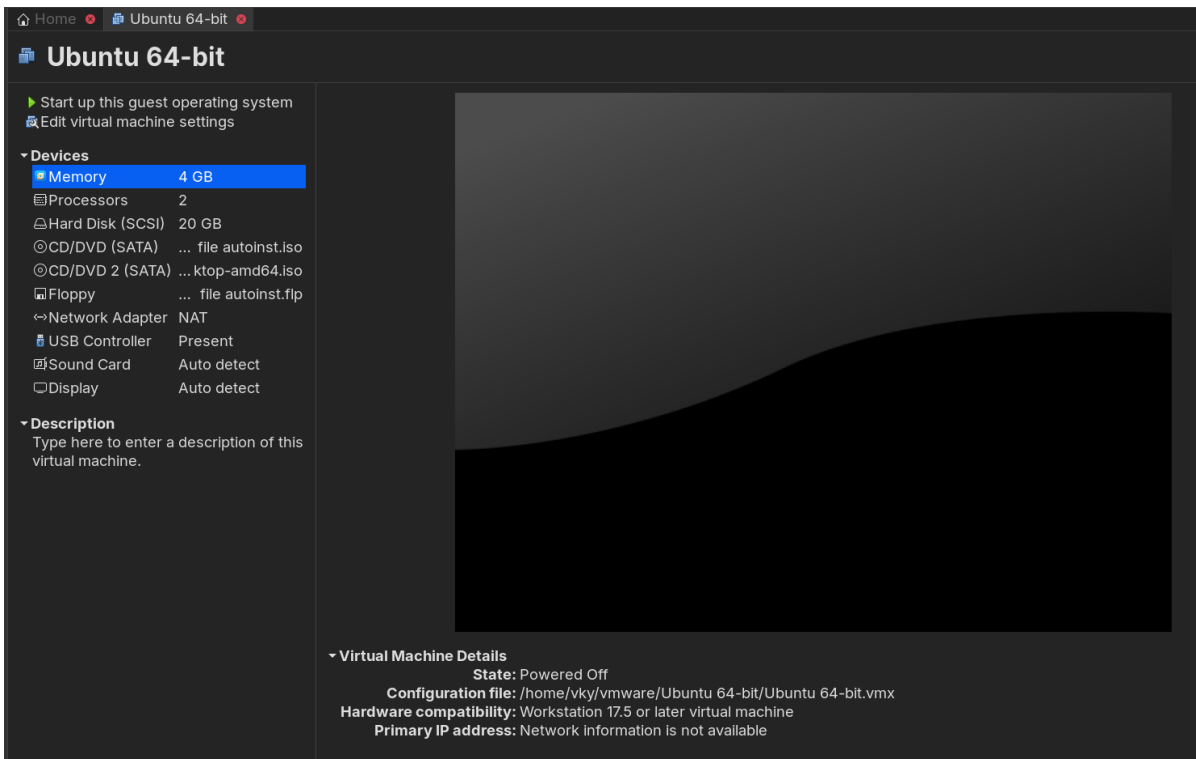
Step 8: Finish and Install

1. Review your settings and click “Finish.”
2. Before starting the virtual machine, you may want to go to the “Options” tab and configure additional settings if necessary.
3. Click “Finish” to create the virtual machine.



Step 9: Start the Virtual Machine

1. The virtual machine will boot from the Ubuntu ISO.
2. Follow the on-screen instructions to install Ubuntu.

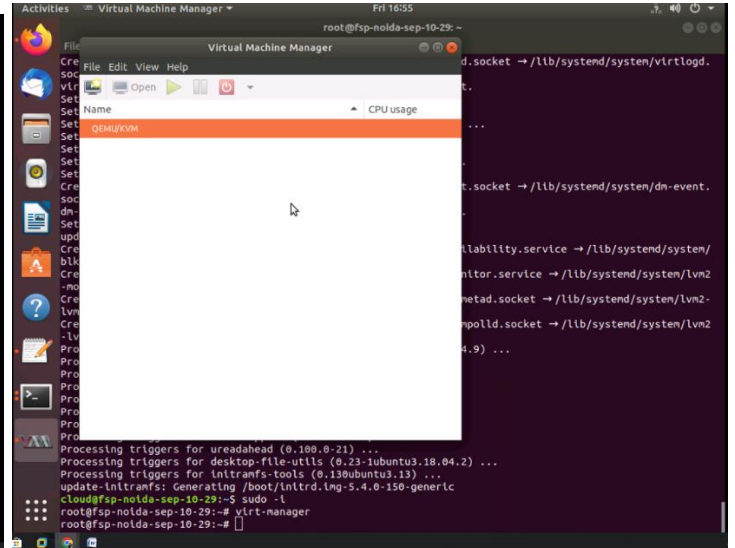


Q2. Create the instance by using the KVM virtualization technology

Install virt-manager

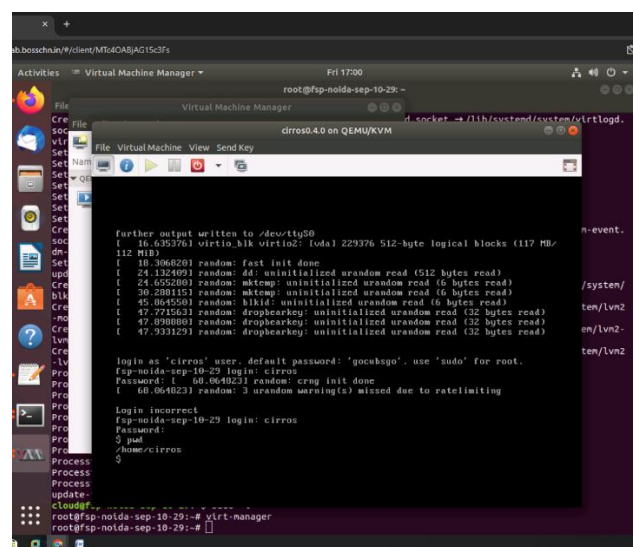
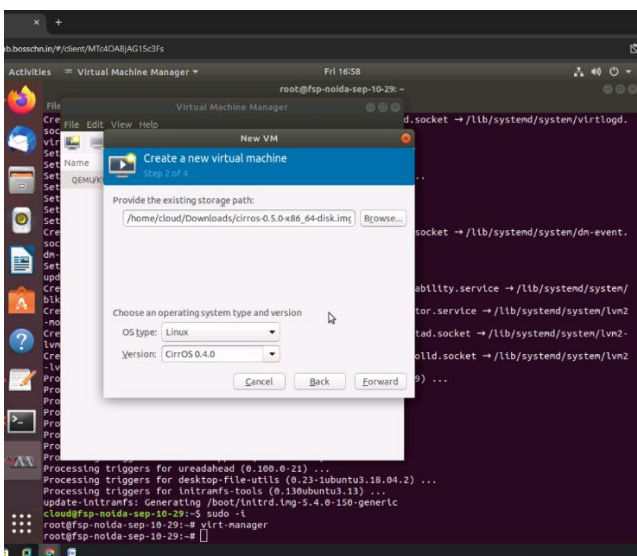
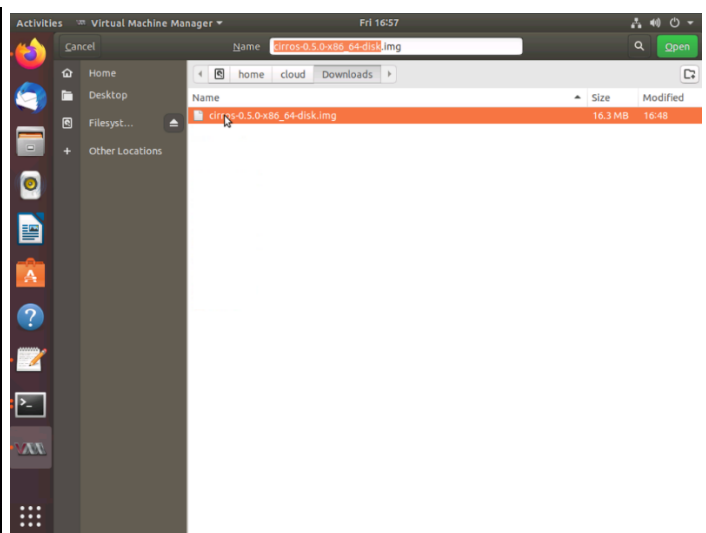
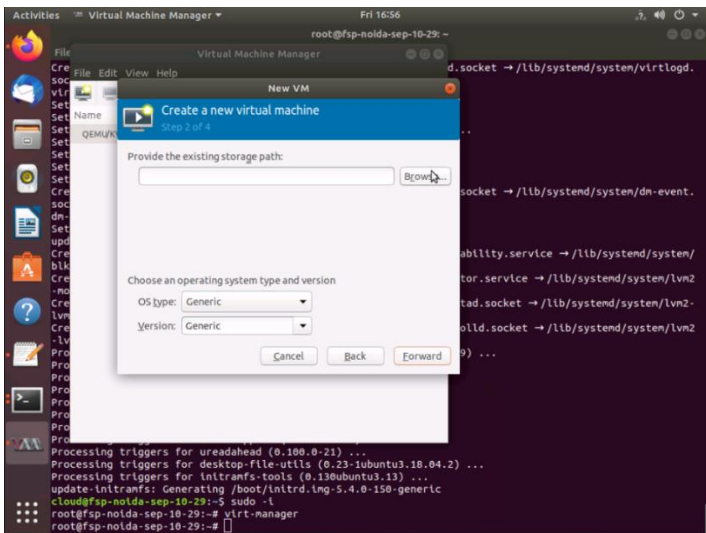
Open virt – manager

```
cloud@fsp-noida-sep-10-29:~$ sudo apt install virt-manager
[sudo] password for cloud:
E: Invalid operation install
cloud@fsp-noida-sep-10-29:~$ sudo apt install virt-manager
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-hwe-5.4-headers-5.4.0-84
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  auger-lenses cpu-checker dswend ebtables glib2.0-appindicator3-0.1 glib2.0-gtk-vnc-2.0
  glib2.0-lbinfo-1.0 glib2.0-lbvirt-glib-1.0 glib2.0-spiceclientglib-2.0 glib2.0-spiceclientgtk-3.0
  ipxe-qemu ipxe-qemu-256k-compat-eft-roms libaugeas0 libcard0 libdevmapper-event1.02.1 libfdt1
  libgiovirt-common libgiovirt2 libgtk-vnc-2.0-0 libgvnc-1.0-0 liblvm2app2.2 liblvm2cmd2.02 libnetcf1
  libosinfo-1.0-0 libphodav-2.0-0 libpython-stdeb librdnss1 libreadlines
  libstd1.2deb1 libspice-client-glib-2.0-8 libspice-client-gtk-3.0-5 libspice-server1
  libusbredirhost1 libusbredirparser1 libvirt-clients libvirt-daemon libvirt-daemon-driver-storage-rbd
  libvirt-daemon-system libvirt-glib-1.0-0 libvirt0 libxen-4.9 libxenstore3.0 libxml2-utils lvm2
  nss-tools osinfo-db python-ansicrypto python-cairo python-certifi python-cffi-backend
  python-charDET python-cryptography python-dbus python-enun3 python-gl python-gi-cairo python-ldna
  python-lpaddr python-lpaddress python-libvirt python-libxml2 python-minimal python-openssl
  python-pkg-resources python-requests python-six python-urllib3 python2.7 python2.7-minimal
  qemu-block-extra qemu-kvm qemu-system-common qemu-system-x86 qemu-utils seabios
  spice-client-glib-usb-acpi-helper virt-viewer virtinst
Suggested packages:
  auger-doc auger-tools libosinfo-1.0m gststreamer1.0-plugins-bad gststreamer1.0-libav
  libvirt-daemon-driver-storage-gluster libvirt-daemon-driver-storage-sheepdog
  libvirt-daemon-driver-storage-zfs nmap radvd audit systemd nfs-common zfsutils pm-utils
  thin-provisioning-tools python-doc python-tk python-cryptography-doc python-cryptography-vectors
  python-dbus-dbg python-dbus-doc python-enun3-doc python-openssl-doc python-openssl-dbg
  python-setuptools python-socks python-ntlm python2.7-doc binfmt-support samba vde2 sgabios ovmf
```



Import existing images

Selecting Images from downloads



Ques 3. Create the Cirros-6.x instance by using the KVM virtualization technology.

Ensure that KVM and its dependencies are installed on your system.

Step 1: Download Cirros Image

You can download the Cirros image by visiting the Cirros website.

Step 2: Create a Virtual Machine

You can use virt-install using the command line

1. Open a terminal
2. Run virt-install command

```
virt-install \  
  --name cirros-instance \  
  --ram 128 \  
  --disk path=/var/lib/libvirt/images/cirros-instance.img,size=1 \  
  --vcpus 1 \  
  --os-type Linux \  
  --os-variant generic \  
  --network network=default \  
  --graphics none \  
  --import \  
  --location cirros-0.6.6-x86_64-disk.img
```

--name: Name of the VM.

--ram: Amount of RAM allocated (128 MB is sufficient for Cirros).

--disk: Path and size of the VM's disk image.

--vcpus: Number of virtual CPUs.

--os-type and **--os-variant:** Specify the OS type (Linux) and variant (generic).

--network: Connects the VM to the default network.

--graphics none: No graphical display; useful for minimal VMs.

--import: Indicates that you're importing an existing image.

--location: Path to the Cirros image.

Step 3: Start the VM

```
virsh start cirros-instance
```

Step 4: Access the VM

```
virsh console cirros-instance
```

Step 5: Log In

The default credentials for Cirros are:

- **Username:** cirros
- **Password:** gocubsgo

Q4. Create the OpenV-switch by switch1 name .

Commands:

Step-1: Install Open vSwitch:

```
sudo apt update
```

```
sudo apt install openvswitch-switch
```

Step-2: Use the ovs-vsctl command to create a new bridge named switch1:

```
sudo ovs-vsctl add-br switch1
```

Step-3: To check if the bridge switch1 was created successfully, you can use:

```
sudo ovs-vsctl show
```

```
cloud@fsp-noida-sep-10-9:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://nova.clouds.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
cloud@fsp-noida-sep-10-9:~$ sudo apt install openvswitch-switch
Reading package lists... Done
Building dependency tree
Reading state information... Done
openvswitch-switch is already the newest version (2.9.8-0ubuntu0.18.04.5).
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1 linux-hwe-5.4-headers-5.4.0-84
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
cloud@fsp-noida-sep-10-9:~$ sudo ovs-vsctl add-br switch1
cloud@fsp-noida-sep-10-9:~$ sudo ovs-vsctl show
0e3fb191-1e5f-49d9-b353-e480cf0661ca
    Bridge "switch1"
        Port "switch1"
            Interface "switch1"
                type: internal
            ovs_version: "2.9.8"
cloud@fsp-noida-sep-10-9:~$
```

Q5. Create the container by using the centos image.

```
cloud@fsp-noida-sep-10-23: ~  
Try 'install --help' for more information.  
cloud@fsp-noida-sep-10-23:~$ sudo apt install docker.io  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  linux-hwe-5.4-headers-5.4.0-84  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan  
Suggested packages:  
  aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-doc  
  rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvinit git-doc  
  git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn  
The following NEW packages will be installed:  
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc  
  ubuntu-fan  
0 upgraded, 9 newly installed, 0 to remove and 58 not upgraded.  
Need to get 70.5 MB of archives.  
After this operation, 326 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://nova.clouds.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]  
Get:2 http://nova.clouds.archive.ubuntu.com/ubuntu bionic/main amd64 bridge-utils amd64 1.5-15ubuntu1 [3  
0.1 kB]  
Get:3 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 runc amd64 1.1.4-0ubunt  
u1-18.04.2 [3,822 kB]  
Get:4 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 containerd amd64 1.6.12  
-0ubuntu1-18.04.1 [31.5 MB]  
Get:5 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 docker.io amd64 20.10.2  
1-0ubuntu1-18.04.3 [30.3 MB]  
Get:6 http://nova.clouds.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0.17025-1 [22.8 k  
B]  
Get:7 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all 1:2.17.1-1ubunt  
u0.18 [804 kB]  
Get:8 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:2.17.1-1ubunt  
u0.18 [3,990 kB]  
Get:9 http://nova.clouds.archive.ubuntu.com/ubuntu bionic/main amd64 ubuntu-fan all 0.12.10 [34.7 kB]  
Fetched 70.5 MB in 13s (5,513 kB/s)
```

```
@1922237ebdf9:/  
File Edit View Search Terminal Help  
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/con  
tainerd.service.  
Setting up bridge-utils (1.5-15ubuntu1) ...  
Setting up ubuntu-fan (0.12.10) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubu  
ntu-fan.service.  
Setting up pigz (2.4-1) ...  
Setting up git (1:2.17.1-1ubuntu0.18) ...  
Setting up docker.io (20.10.21-0ubuntu1-18.04.3) ...  
Adding group 'docker' (GID 128) ...  
Done.  
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.sock  
et.  
Processing triggers for systemd (237-3ubuntu10.57) ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
Processing triggers for ureadahead (0.100.0-21) ...  
cloud@fsp-noida-sep-10-23:~$ sudo docker images  
REPOSITORY TAG IMAGE ID CREATED SIZE  
cloud@fsp-noida-sep-10-23:~$ sudo systemctl start docker  
sudo: systemctl: command not found  
cloud@fsp-noida-sep-10-23:~$ sudo systemctl start docker  
sudo: systemctl: command not found  
cloud@fsp-noida-sep-10-23:~$ sudo service docker start  
cloud@fsp-noida-sep-10-23:~$ sudo service docker enable  
docker: unrecognized service  
cloud@fsp-noida-sep-10-23:~$ docker --version  
Docker version 20.10.21, build 20.10.21-0ubuntu1-18.04.3  
cloud@fsp-noida-sep-10-23:~$ sudo docker pull centos  
Using default tag: latest  
latest: Pulling from library/centos  
a1d0c7532777: Pull complete  
Digest: sha256:a27fd8080b517143cbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177  
Status: Downloaded newer image for centos:latest  
docker.io/library/centos:latest  
cloud@fsp-noida-sep-10-23:~$ sudo docker run -it centos  
[root@1922237ebdf9 /]#
```

Question 6:- Creating a 1GB Volume on “/BCMP” mount point with ext4 Filesystem.

1. Prerequisites

Before proceeding, ensure you have:

- Root or sudo access to the system.
- Necessary tools (`fdisk`, `mkfs`, `mount`, etc.) installed on your Linux system.

2. Create a Partition

1. Open a terminal.
2. Use `fdisk` to create a new partition:

```
```bash
sudo fdisk /dev/sdX
```
```

Replace `/dev/sdX` with the appropriate disk identifier (e.g., `/dev/sda`).

3. Follow these steps in the `fdisk` command line:

- Type `n` to create a new partition.
- Choose `p` for primary partition.
- Select the partition number (e.g., `1` if it's the first partition).
- Accept the default first sector (press Enter).
- Type `+1G` to specify the size as 1GB.
- Type `w` to write the changes to the disk and exit.

4. After exiting `fdisk`, update the partition table with:

```
```bash
sudo partprobe
```
```

3. Format the Partition with ext4 Filesystem

1. Identify the new partition created (e.g., `/dev/sdX1`).
2. Format the partition with the ext4 filesystem:

```
```bash
sudo mkfs.ext4 /dev/sdX1
```
```

Replace `/dev/sdX1` with the correct partition identifier.

4. Mount the Partition

1. Create the mount point directory:

```
```bash
sudo mkdir -p /bootcamp
```
```

2. Mount the newly formatted partition to `/bootcamp`:

```
```bash
sudo mount /dev/sdX1 /bootcamp
```
```

3. Verify the mount:

```
```bash
df -h /bootcamp
```
```

****5. Make the Mount Permanent****

1. Edit the `/etc/fstab` file to make the mount persistent across reboots:

```
```bash
sudo nano /etc/fstab
```
```

2. Add the following line to the end of the file:

```
```plaintext
/dev/sdX1 /bootcamp ext4 defaults 0 2
```
```

Replace `/dev/sdX1` with your partition identifier.

3. Save and exit the editor (in `nano`, press `CTRL+X`, then `Y`, and `Enter`).

****6. Verify Configuration****

1. Test the `/etc/fstab` entry:

```
```bash
sudo mount -a
```
```

2. Verify that the partition is correctly mounted:

```
```bash
df -h /bootcamp
```
```

****Note:**** Be cautious when using disk management tools. Ensure you are working on the correct disk and partition to avoid data loss.

Feel free to copy and paste this guide into a document file as needed.

Q7: Create an Ubuntu Instance in a Public Cloud

Assuming you are using AWS, here's a step-by-step guide.

1. Log into AWS Console.



Sign in

☒ **Root user**

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ **IAM user**

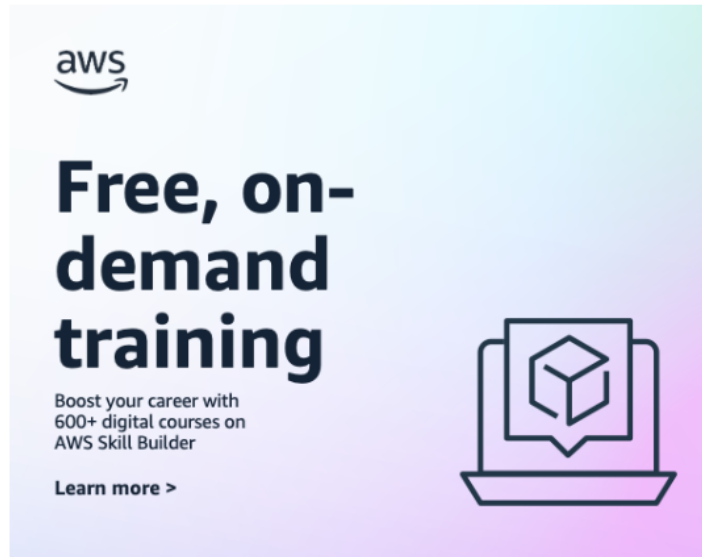
User within an account that performs daily tasks. [Learn more](#)

Root user email address

username@example.com

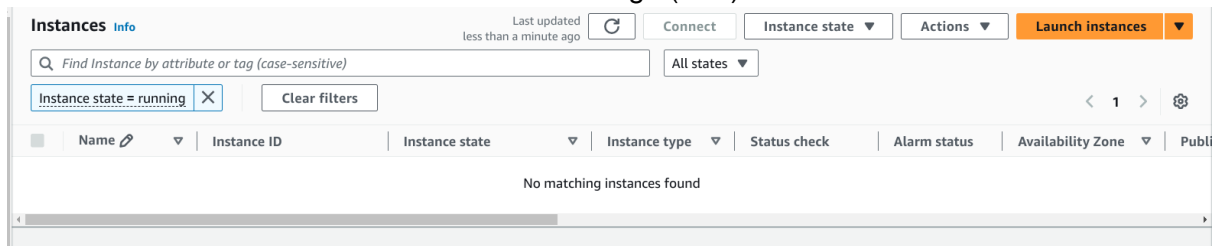
Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.



2. Navigate to EC2 Dashboard:

- Go to “Launch Instance.”
- Select the Ubuntu Amazon Machine Image (AMI).



Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-0dee22c13ea7a9a67 (64-bit (x86)) / ami-0c8eea98010057bd0 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (x86)

AMI ID

ami-0dee22c13ea7a9a67

Username

ubuntu



Verified provider

3. Configure Instance Details:

- Choose instance type (e.g., t2.micro for free tier).

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

☐ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

- Configure network and storage options.

▼ Network settings [Info](#)

[Edit](#)

Network [Info](#)

vpc-0450babe1dcd77390

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

☐ Enable

[Additional charges apply](#) when outside of [free tier allowance](#)

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called **'launch-wizard-3'** with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0

☐ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

4. Add Storage and Tags (if necessary).

▼ Configure storage [Info](#)

[Advanced](#)

1x GiB

[Free tier eligible](#)

[Add new volume](#)

The selected AMI contains
volumes from the AMI will

[Click refresh to view](#)

The tags that you assign to
Data Lifecycle Manager p

0 x File systems

Q |

Select

General purpose SSD (gp3)

General purpose SSD (gp2)

Provisioned IOPS SSD (io1)

Provisioned IOPS SSD (io2)

Cold HDD (sc1)

This volume type is not compatible with root volumes.

Throughput Optimized HDD (st1)

This volume type is not compatible with root volumes.

Magnetic (standard)

Choose (SSD) or Magnetic storage

allows. Only the first 0 instance store

by any

[Edit](#)

5. Configure Security Group:

- Set rules for SSH access (e.g., open port 22 for SSH).

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

- ☒ Allow SSH traffic from

Anywhere
0.0.0.0/0

Helps you connect to your instance
- ☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
- ☐ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

6. Launch the Instance:

- Choose or create a key pair.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

youtube-key

[Create new key pair](#)

Q |

Proceed without a key pair (Not recommended)

Default value

youtube-key

Type: rsa



Edit

- Click "Launch."

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0dee22c13ea7a9a67

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB



Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month



Cancel

Launch instance



Preview code

7. Connect to the Instance:

- Use SSH to connect:
- `ssh -i your-key.pem ubuntu@your-public-ip`

Instances (1) [Info](#)

Last updated
less than a minute ago



Connect

Instance state ▼

Find Instance by attribute or tag (case-sensitive)

All states ▼

Instance ID = i-046362bed93475598



Clear filters



Name



Instance ID

Instance state



Instance type



Status check



Ubuntu-Server

i-046362bed93475598

Running



t2.micro

Initializing

Connect to instance Info


Connect to your instance i-046362bed93475598 (Ubuntu-Server) using any of these options

EC2 Instance Connect

Session Manager


SSH client

EC2 serial console



Port 22 (SSH) is open to all IPv4 addresses
Port 22 (SSH) is currently open to all IPv4 addresses, indicated by **0.0.0.0/0** in the inbound rule in [your security group](#). For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 13.233.177.0/29. [Learn more](#).

Instance ID




 i-046362bed93475598 (Ubuntu-Server)

Connection Type

☒ **Connect using EC2 Instance Connect**
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

☐ **Connect using EC2 Instance Connect Endpoint**
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

☒ **Public IPv4 address**




 13.235.69.162


☐ **IPv6 address**
—


Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.



 ubuntu





Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Connect

19

```
< > C ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?addressFamily=ipv4&connType=standard&i...
aws Services Search [Alt+S]
System information as of Tue Nov 12 12:19:50 UTC 2024
System load: 0.14          Processes:           106
Usage of /:  22.8% of 6.71GB Users logged in:       0
Memory usage: 21%         IPv4 address for enx0: 172.31.13.243
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-13-243:~$

i-046362bed93475598 (Ubuntu-Server)
PublicIPs: 13.235.69.162 PrivateIPs: 172.31.13.243
```