

Experiment 4

Alm: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

Step 1: Log in to your AWS Academy/personal account and launch a new Ec2 Instance. Select Ubuntu as AMI and t2.medium as Instance Type, create a key of type RSA with .pem extension, and move the downloaded key to the new folder.

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

[Add additional tags](#)

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

[Quick Start](#)

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Li

[Browse more AMIs](#)
Including AMIs from AWS Marketplace and

Summary

Number of instances Info

Firewall (security group)
Master

Storage (volumes)
1 volume(s) - 12 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, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel [Launch instance](#) [Review commands](#)

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096e9a12ea24a797 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

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

Architecture	AMI ID	Username
64-bit (x86)	ami-0e86e20dae9224db8	ubuntu

Verified provider

Instance type Info | Get advice

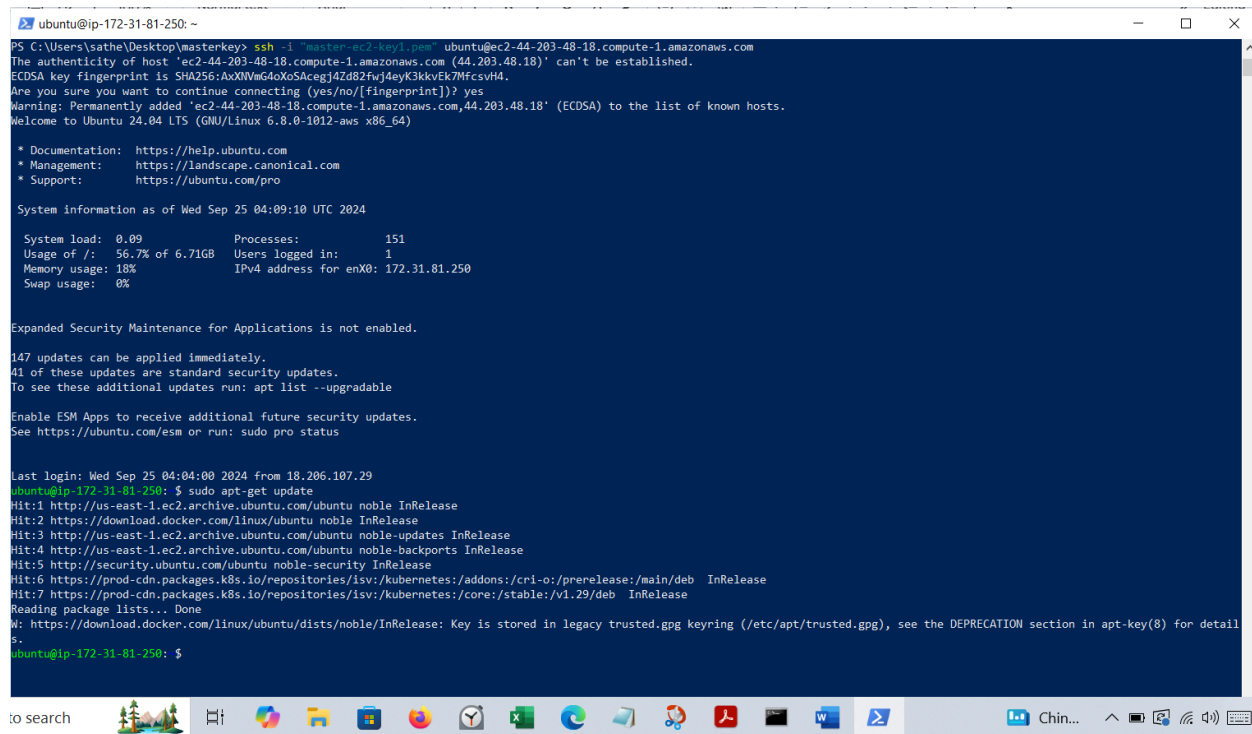
Instance type

t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

☐ All generations
[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

Step 2: After creating the instance click on Connect the instance and navigate to SSH Client. Copy the example command. Open your key Folder in terminal and paste the command there.



```
ubuntu@ip-172-31-81-250: ~
PS C:\Users\sathel\Desktop\masterkey> ssh -i "master-ec2-key1.pem" ubuntu@ec2-44-203-48-18.compute-1.amazonaws.com
The authenticity of host 'ec2-44-203-48-18.compute-1.amazonaws.com (44.203.48.18)' can't be established.
ECDSA key fingerprint is SHA256:AxXNVG4oXoSh4egj4Zd82Fvj9eyK3kkvEk7MfcsvH4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-203-48-18.compute-1.amazonaws.com,44.203.48.18' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Wed Sep 25 04:09:10 UTC 2024

System load: 0.09          Processes:            151
Usage of /:  56.7% of 6.71GB   Users logged in:      1
Memory usage: 18%           IPv4 address for enx0: 172.31.81.250
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

147 updates can be applied immediately.
41 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

Last login: Wed Sep 25 04:04:00 2024 from 18.206.107.29
ubuntu@ip-172-31-81-250:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 https://download.docker.com/linux/ubuntu noble InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/1sv/kubernetes/addons:/cri-o:/prerelease:/main/deb InRelease
Hit:7 https://prod-cdn.packages.k8s.io/repositories/1sv/kubernetes/core:/stable:/v1.29/deb InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for detail
s.
ubuntu@ip-172-31-81-250:~$
```

Step 3: Run the below commands to install and setup Docker.

- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null`
- `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"`

```

ubuntu@ip-172-31-90-179: ~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
> $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [15.3 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [530 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [128 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8548 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [374 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [154 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.6 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [353 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [68.1 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [424 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]

```

- `sudo apt-get install -y apt-transport-https ca-certificates curl`

```

ubuntu@ip-172-31-81-250: ~$ sudo apt-get install -y apt-transport-https ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
The following additional packages will be installed:
  libcurl3t64-gnutls libcurl4t64
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 1 newly installed, 0 to remove and 140 not upgraded.
Need to get 904 kB of archives.
After this operation, 38.9 kB of additional disk space will be used.

```

```
Running kernel seems to be up-to-date.
```

```
Restarting services...
```

```
systemctl restart packagekit.service
```

```
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.
```

Step 4:Download and add the GPG key:

```
sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]
https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee
/etc/apt/sources.list.d/kubernetes.list
```

```
ubuntu@ip-172-31-81-250: $ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
ubuntu@ip-172-31-81-250: $ echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main
```

Step 5: Update package list:

sudo apt-get update

```
ubuntu@ip-172-31-81-250: $ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:6 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Hit:7 https://prod-cdn.packages.k8s.io/repositories/isy/kubernetes/addons/cni-o/prerelease/main/deb InRelease
Err:8 https://packages.cloud.google.com/apt kubernetes-focal Release
 404 Not Found [IP: 172.253.63.100 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
W: Updating from such a repository can't be done securely, and is therefore disabled by default.
W: See apt-secure(8) manpage for repository creation and user configuration details.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
```

sudo apt-get install -y kubectl

```
ubuntu@ip-172-31-81-250: $ sudo apt-get install -y kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kubectl is already the newest version (1.29.0-1.1).
0 upgraded, 0 newly installed, 0 to remove and 137 not upgraded.
ubuntu@ip-172-31-81-250: $ kubectl version --client
Client Version: v1.29.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
```

Step 6: Verify the nodes

kubectl get nodes

```
ubuntu@ip-172-31-81-250: $ kubectl get nodes
NAME                STATUS    ROLES           AGE   VERSION
ip-172-31-81-250    Ready     control-plane   9h    v1.29.0
ip-172-31-81-86     Ready     Node1           9h    v1.29.0
ip-172-31-83-216    Ready     Node2           9h    v1.29.0
ubuntu@ip-172-31-81-250: $ nano nginx-deployment.yaml
ubuntu@ip-172-31-81-250: $ ubuntu@ip-172-31-81-250: $ nano nginx-service.yaml
```

Step 7: Create the Deployment YAML File

1. Create a file named nginx-service.yaml:*
- ```
apiVersion: v1
kind: Service
metadata:
```

```

name: nginx-service
spec:
 selector:
 app: nginx
 ports:
 - protocol: TCP
 port: 80
 targetPort: 80

```

And deploy it

```

ubuntu@ip-172-31-81-250:~$ nano nginx-service.yaml
ubuntu@ip-172-31-81-250:~$ kubectl apply -f nginx-service.yaml
service/nginx-service created

```

**Step 8:**Verify the deployment:

```

kubectl get deployments
kubectl get pods
kubectl get services

```

```

ubuntu@ip-172-31-81-250:~$ kubectl get deployments
kubectl get pods
kubectl get services

```

| NAME             | READY | UP-TO-DATE | AVAILABLE | AGE   |
|------------------|-------|------------|-----------|-------|
| nginx-deployment | 3/3   | 3          | 3         | 4m55s |

```

ubuntu@ip-172-31-81-250:~$ kubectl get pods

```

| NAME                              | READY | STATUS  | RESTARTS | AGE   |
|-----------------------------------|-------|---------|----------|-------|
| nginx-deployment-86dcfdf4c6-219tr | 1/1   | Running | 0        | 4m55s |
| nginx-deployment-86dcfdf4c6-ksdx9 | 1/1   | Running | 0        | 4m55s |
| nginx-deployment-86dcfdf4c6-qdggx | 1/1   | Running | 0        | 4m55s |

```

ubuntu@ip-172-31-81-250:~$ kubectl get services

```

| NAME          | TYPE         | CLUSTER-IP   | EXTERNAL-IP | PORT(S)      | AGE |
|---------------|--------------|--------------|-------------|--------------|-----|
| kubernetes    | ClusterIP    | 10.96.0.1    | <none>      | 443/TCP      | 9h  |
| nginx-service | LoadBalancer | 10.106.1.134 | <pending>   | 80:31668/TCP | 60s |

**Step 9:**Forward the service port:

```

kubectl port-forward service/nginx-service 8080:80

```

```

ubuntu@ip-172-31-81-250:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80

```

**Step 10:**Open your web browser and go to `http://44.203.48.18:31668/`



## 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](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

**Error :** the connection string changes if the ip address is changed

```
PS C:\Users\sathe\Desktop\masterkey> ssh -i "master-ec2-key1.pem" ubuntu@ec2-3-89-117-235.compute-1.amazonaws.com
ssh: connect to host ec2-3-89-117-235.compute-1.amazonaws.com port 22: Connection timed out
PS C:\Users\sathe\Desktop\masterkey> ssh -i "master-ec2-key1.pem" ubuntu@ec2-44-203-48-18.compute-1.amazonaws.com
The authenticity of host 'ec2-44-203-48-18.compute-1.amazonaws.com (44.203.48.18)' can't be established.
ECDSA key fingerprint is SHA256:AxXNVmG4oXoSacegj4Zd82fwj4eyK3kkvEk7MfcsvH4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-203-48-18.compute-1.amazonaws.com,44.203.48.18' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)
```

To correct it copy the connection string again

Error due to incorrect indentation

```
ubuntu@ip-172-31-81-250:~$ kubectl apply -f nginx-deployment.yaml
error: error parsing nginx-deployment.yaml: error converting YAML to JSON: yaml: line 2: mapping values are not allowed in this context
```