

# TASK – 19

## Creating EC2 instance:

aws

Services

Search

[Alt+S]

EC2 > Instances > Launch an instance

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

minikube

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

Search our full catalog including 1000s of application and OS images

Recents

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Li

Browse more AMIs

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

Summary

Number of instances

Info

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-0c2af51e265bd5e0e

Virtual server type (instance type)

t2.medium

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, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Review commands

aws

Services

Search

[Alt+S]

Architecture

AMI ID

Verified provider

64-bit (x86)

ami-0c2af51e265bd5e0e

Instance type

Info

Get advice

Instance type

t2.medium

Family: t2 2 vCPU 4 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0496 USD per Hour

On-Demand Windows base pricing: 0.0676 USD per Hour

On-Demand RHEL base pricing: 0.0784 USD per Hour

On-Demand SUSE base pricing: 0.1496 USD per Hour

Additional costs apply for AMIs with pre-installed software

All generations

Compare instance types

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

mypem

Create new key pair

Network settings

Info

Network

Info

vpc-049bb71bf494e967f

Subnet

Info

No preference (Default subnet in any availability zone)

Summary

Number of instances

Info

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-0c2af51e265bd5e0e

Virtual server type (instance type)

t2.medium

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, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Review commands

## Installing docker:

```
etched 32.8 MB in 5s (6477 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-8-1:~$ sudo apt install docker.io
```

i-0991635f9d2675460 (minikube)

PublicIPs: 13.233.153.102 PrivateIPs: 172.31.8.1

```
aws Services Search [Alt+S]
Preparing to unpack .../4-dns-root-data_2023112702~ubuntu0.22.04.1_all.deb ...
Unpacking dns-root-data (2023112702~ubuntu0.22.04.1) ...
Selecting previously unselected package dnsmasq-base.
Preparing to unpack .../5-dnsmasq-base_2.90-0ubuntu0.22.04.1_amd64.deb ...
Unpacking dnsmasq-base (2.90-0ubuntu0.22.04.1) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../6-docker.io_24.0.7-0ubuntu2~22.04.1_amd64.deb ...
Unpacking docker.io (24.0.7-0ubuntu2~22.04.1) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../7-ubuntu-fan_0.12.16_all.deb ...
Unpacking ubuntu-fan (0.12.16) ...
Setting up dnsmasq-base (2.90-0ubuntu0.22.04.1) ...
Setting up runc (1.1.12-0ubuntu2~22.04.1) ...
Setting up dns-root-data (2023112702~ubuntu0.22.04.1) ...
Setting up bridge-utils (1.7-1ubuntu3) ...
Setting up pigz (2.6-1) ...
Setting up containerd (1.7.12-0ubuntu2~22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.

Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.7-0ubuntu2~22.04.1) ...
Adding group `docker' (GID 122) ...
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.socket.
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-8-1:~$
```

i-0991635f9d2675460 (minikube)

PublicIPs: 13.233.153.102 PrivateIPs: 172.31.8.1

## Installing minikube:

```
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.
```

```
ubuntu@ip-172-31-8-1:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```

```
i-0991635f9d2675460 (minikube)
```

```
PublicIPs: 13.233.153.102 PrivateIPs: 172.31.8.1
```

```
ubuntu@ip-172-31-8-1:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
100	91.1M	100	91.1M	0	0	11.3M	0
				0:00:08	0:00:08	--:--:--	16.3M

```
ubuntu@ip-172-31-8-1:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
```

```
i-0991635f9d2675460 (minikube)
```

```
ubuntu@ip-172-31-8-1:~$ minikube start
```

```
i-0991635f9d2675460 (minikube)
```

## Installing kubectl:

```
ubuntu@ip-172-31-8-1:~$ curl -LO "https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl"
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
100	138	100	138	0	0	409	0
				0:00:00	0:00:00	--:--:--	408
100	49.0M	100	49.0M	0	0	64.1M	0
				0:00:00	0:00:00	--:--:--	64.1M

```
ubuntu@ip-172-31-8-1:~$ curl -LO "https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl.sha256"
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
100	138	100	138	0	0	401	0
				0:00:00	0:00:00	--:--:--	402
100	64	100	64	0	0	156	0
				0:00:00	0:00:00	--:--:--	16000

```
ubuntu@ip-172-31-8-1:~$ echo "$(cat kubectl.sha256) kubectl" | sha256sum --check
```

```
kubectl: OK
```

```
ubuntu@ip-172-31-8-1:~$ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

```
ubuntu@ip-172-31-8-1:~$ kubectl version --client
```

```
Client Version: v1.30.3
```

## Creating a new namespace:

```
ubuntu@ip-172-31-8-1:~$ kubectl create namespace my-ns-1
namespace/my-ns-1 created
```

## Viewing created namespace:

```
ubuntu@ip-172-31-8-1:~$ kubectl get namespaces
```

NAME	STATUS	AGE
default	Active	4m42s
kube-node-lease	Active	4m42s
kube-public	Active	4m42s
kube-system	Active	4m42s
my-ns-1	Active	7s

```
ubuntu@ip-172-31-8-1:~$
```

## Creating a pod with the namespace:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  namespace: my-ns-1
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-con
    image: nginx
    ports:
    - containerPort: 80
```

```
ubuntu@ip-172-31-8-1:~$ kubectl apply -f pod.yaml
pod/nginx-pod created
ubuntu@ip-172-31-8-1:~$
```

```
ubuntu@ip-172-31-8-1:~$ kubectl get pods --namespace=my-ns-1
NAME          READY   STATUS    RESTARTS   AGE
nginx-pod     1/1     Running   0           36s
ubuntu@ip-172-31-8-1:~$
```