

INDEX

Tasks	Page No
Taks: 1 – Launched Instances	01
Task: 2 – Kubernetes Cluster installation on Master	04
Task: 3 – Kubernetes Cluster installation on Workernode1	05
Task: 4 – Kubernetes Cluster installation on Workernode2	06
Task: 5 – Config Kubernetes on Master	07
Task: 6 – Connecting Workernode1 & 2 from Master	08
Task: 7 – Checking nodes at master for cluster	09
Task: 8 – Creation of Nginx deployment	09

You have been asked to:

- Deploy a Kubernetes Cluster for 3 nodes
 - Create a nginx deployment of 3 replicas
-

Taks: 1 – Launched Instances.

In this case we will launch 3 instances named:

- Master
- Workernode1
- Workernode2

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

[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](#)

Including AMIs from
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-0a1b648e2cd533174 (64-bit (x86)) / ami-08bca0a1996622bee (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Canonical, Ubuntu, 20.04 LTS, amd64 focal image build on 2024-02-28

Architecture

AMI ID

64-bit (x86)

ami-0a1b648e2cd533174

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: 0.0496 USD per Hour

On-Demand Windows base pricing: 0.0676 USD per Hour

On-Demand RHEL base pricing: 0.1096 USD per Hour

On-Demand SUSE base pricing: 0.1496 USD per Hour

☐ All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ 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*

DevOps



[Create new key pair](#)

▼ Network settings [Info](#)

[Edit](#)

Network [Info](#)

vpc-08bde8db1e7da9b9a | Default VPC

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

Common security groups [Info](#)

Select security groups

default sg-04d9bdfee395aade5 ✕
VPC: vpc-08bde8db1e7da9b9a



Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ Summary

Number of instances [Info](#)

3

When launching more than 1 instance, consider [EC2 Auto Scaling](#)

Software Image (AMI)

Canonical, Ubuntu, 20.04 LTS, ...[read more](#)
ami-0a1b648e2cd533174

Virtual server type (instance type)

t2.medium

Firewall (security group)

default

Storage (volumes)

1 volume(s) - 8 GiB



Free tier: In your first year
includes 750 hours of t2.micro for



Cancel

Launch instance

Now we could see that the instances have been launched.

Instances (3) Info						
<div>Find Instance by attribute or tag (case-sensitive)</div> <div>All states</div>						
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	
<input type="checkbox"/>	Master	i-0988b4b9aacdb96d3	Running	t2.medium	2/2 checks passed	
<input type="checkbox"/>	Workernode1	i-028139fa4e43fa81e	Running	t2.medium	2/2 checks passed	
<input type="checkbox"/>	Workernode2	i-02977ce0f3c0912b5	Running	t2.medium	2/2 checks passed	

Task: 2 – Kubernetes Cluster installation on Master.

1. Now we will create a shell script to install the Kubernetes clusters.

```
sudo su –
```

```
vi kubernetes_install.sh
```

```
chmod +x kubernetes_install.sh
```

```
./kubernetes_install.sh
```

```
ubuntu@ip-172-31-7-253:~$ vi kubernetes_install.sh
ubuntu@ip-172-31-7-253:~$ ./kubernetes_install.sh
-bash: ./kubernetes_install.sh: Permission denied
ubuntu@ip-172-31-7-253:~$ chmod +x kubernetes_install.sh
ubuntu@ip-172-31-7-253:~$ ./kubernetes_install.sh
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
```

i-0988b4b9aacdb96d3 (Master)

PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

Command entered in the shell file kubernetes_install.sh :-

```
sudo apt-get update
```

```
sudo apt install docker.io -y
```

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

```
sudo mkdir -p -m 755 /etc/apt/keyrings
```

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
```

```
https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee
```

```
/etc/apt/sources.list.d/kubernetes.list
```

```
sudo apt-get update
```

```
sudo apt-get install -y kubelet kubeadm kubectl
```

```
sudo systemctl enable --now Kubelet
```

```
ubuntu@ip-172-31-7-253:~$ cat kubernetes_install.sh
sudo apt-get update
sudo apt install docker.io -y
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo systemctl enable --now kubelet
ubuntu@ip-172-31-7-253:~$
```

i-0988b4b9aadb96d3 (Master)

PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

Task: 3 – Kubernetes Cluster installation on Workernode1.

1. Now we will create a shell script to install the Kubernetes clusters.

```
sudo su –
```

```
vi kubernetes_install.sh
```

```
chmod +x kubernetes_install.sh
```

```
./kubernetes_install.sh
```

```
ubuntu@ip-172-31-3-134:~$ vi kubernetes_install.sh
ubuntu@ip-172-31-3-134:~$ chmod +x kubernetes_install.sh
ubuntu@ip-172-31-3-134:~$ ./kubernetes_install.sh
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done

i-028139fa4e43fa81e (Workernode1)

PublicIPs: 13.201.193.18 PrivateIPs: 172.31.3.134
```

Command entered in the shell file kubernetes_install.sh :-

```
sudo apt-get update
```

```
sudo apt install docker.io -y
```

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

```
sudo mkdir -p -m 755 /etc/apt/keyrings
```

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee
/etc/apt/sources.list.d/kubernetes.list
```

```
sudo apt-get update
```

```
sudo apt-get install -y kubelet kubeadm kubectl
```

```
sudo systemctl enable --now Kubelet
```

```
ubuntu@ip-172-31-3-134:~$ cat kubernetes_install.sh
sudo apt-get update
sudo apt install docker.io -y
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.g
pg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/ap
t/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo systemctl enable --now kubelet
ubuntu@ip-172-31-3-134:~$

i-028139fa4e43fa81e (Workernode1)
PublicIPs: 13.201.193.18 PrivateIPs: 172.31.3.134
```

Task: 4 – Kubernetes Cluster installation on Workernode2.

1. Now we will create a shell script to install the Kubernetes clusters.

```
sudo su –
```

```
vi kubernetes_install.sh
```

```
chmod +x kubernetes_install.sh
```

```
./kubernetes_install.sh
```

```
ubuntu@ip-172-31-4-178:~$ vi kubernetes_install.sh
ubuntu@ip-172-31-4-178:~$ chmod +x kubernetes_install.sh
ubuntu@ip-172-31-4-178:~$ ./kubernetes_install.sh
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease

i-02977ce0f3c0912b5 (Workernode2)
PublicIPs: 13.232.14.40 PrivateIPs: 172.31.4.178
```

Command entered in the shell file kubernetes_install.sh :-

```
sudo apt-get update
```

```
sudo apt install docker.io -y
```

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

```
sudo mkdir -p -m 755 /etc/apt/keyrings
```

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ ' | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
sudo apt-get update
```

```
sudo apt-get install -y kubelet kubeadm kubectl
```

```
sudo systemctl enable --now Kubelet
```

```
ubuntu@ip-172-31-4-178:~$ cat kubernetes_install.sh
sudo apt-get update
sudo apt install docker.io -y
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ ' | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo systemctl enable --now kubelet
ubuntu@ip-172-31-4-178:~$
```

i-02977ce0f3c0912b5 (Workernode2)
PublicIPs: 13.232.14.40 PrivateIPs: 172.31.4.178

Task: 5 – Config Kubernetes on Master.

1. sudo kubeadm init --apiserver-advertise-address=privateipofmaster

```
ubuntu@ip-172-31-7-253:~$ sudo kubeadm init --apiserver-advertise-address=172.31.7.253
I0422 09:49:29.277642 5912 version.go:256] remote version is much newer: v1.30.0; falling back to: stable-1.28
[init] Using Kubernetes version: v1.28.9
[preflight] Running pre-flight checks
```

i-0988b4b9aacdb96d3 (Master)
PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.7.253:6443 --token 0hmgaa.nzbp0de4yyw2ifhu \
--discovery-token-ca-cert-hash sha256:e2409ec1fc471692efb36ba9809a2dd3650d49912dc6a9c96646a730103dab90
ubuntu@ip-172-31-7-253:~$
```

i-0988b4b9aacdb96d3 (Master)
PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

```
ubuntu@ip-172-31-7-253:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-7-253:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-7-253:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-7-253:~$
```

i-0988b4b9aacdb96d3 (Master)

PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

Installing Calico:

curl

<https://raw.githubusercontent.com/projectcalico/calico/v3.27.2/manifests/calico.yaml> -O

kubectl apply -f calico.yaml

```
ubuntu@ip-172-31-7-253:~$ curl https://raw.githubusercontent.com/projectcalico/calico/v3.27.2/manifests/calico.yaml -O
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 246k 100 246k    0     0  721k    0 --:--:-- --:--:-- --:--:--  723k
ubuntu@ip-172-31-7-253:~$ kubectl apply -f calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
```

i-0988b4b9aacdb96d3 (Master)

PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

```
clusterrole.rbac.authorization.k8s.io/calico-cni-plugin created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
ubuntu@ip-172-31-7-253:~$
```

i-0988b4b9aacdb96d3 (Master)

PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

Task: 6 – Connecting Workernode1 & 2 from Master.

Command:

sudo su -

kubeadm join 172.31.7.253:6443 --token 0hmgaa.nzbp0de4yyw2ifhu \

--discovery-token-ca-cert-hash

sha256:e2409ec1fc471692efb36ba9809a2dd3650d49912dc6a9c96646a730103dab90


```

ubuntu@ip-172-31-3-134:~$ sudo su -
root@ip-172-31-3-134:~# kubeadm join 172.31.7.253:6443 --token 0hmgaa.nzbp0de4yyw2ifhu \
> --discovery-token-ca-cert-hash sha256:e2409ec1fc471692efb36ba9809a2dd3650d49912dc6a9c96646a730103dab90
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

root@ip-172-31-3-134:~# █

i-028139fa4e43fa81e (Workernode1)
PublicIPs: 13.201.193.18 PrivateIPs: 172.31.3.134

```

```

ubuntu@ip-172-31-4-178:~$ sudo su -
root@ip-172-31-4-178:~# kubeadm join 172.31.7.253:6443 --token 0hmgaa.nzbp0de4yyw2ifhu \
> --discovery-token-ca-cert-hash sha256:e2409ec1fc471692efb36ba9809a2dd3650d49912dc6a9c96646a730103dab90
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

root@ip-172-31-4-178:~# █

i-02977ce0f3c0912b5 (Workernode2)
PublicIPs: 13.232.14.40 PrivateIPs: 172.31.4.178

```

Task: 7 – Checking nodes at master for cluster.

```

ubuntu@ip-172-31-7-253:~$ kubectl get nodes
NAME                STATUS    ROLES          AGE      VERSION
ip-172-31-3-134     Ready    <none>         2m23s    v1.28.9
ip-172-31-4-178     Ready    <none>         105s     v1.28.9
ip-172-31-7-253     Ready    control-plane  10m      v1.28.9
ubuntu@ip-172-31-7-253:~$ █

i-0988b4b9aacdb96d3 (Master)
PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

```

Task: 8 – Creation of Nginx deployment.

sudo nano assignment1.yaml

cat assignment1.yaml

apiVersion: apps/v1

kind: Deployment

```
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

```

ubuntu@ip-172-31-7-253:~$ sudo nano assignment1.yaml
ubuntu@ip-172-31-7-253:~$ cat assignment1.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
ubuntu@ip-172-31-7-253:~$ █

```

i-0988b4b9aacdb96d3 (Master)

PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

kubectl apply -f assignment1.yaml

kubectl get deploy

```

ubuntu@ip-172-31-7-253:~$ kubectl apply -f assignment1.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-7-253:~$ kubectl get deploy
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment    3/3     3             3           23s
ubuntu@ip-172-31-7-253:~$ █

```

i-0988b4b9aacdb96d3 (Master)

PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

kubectl get pods

```

ubuntu@ip-172-31-7-253:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-86dcfdf4c6-kwd88  1/1     Running   0           5m54s
nginx-deployment-86dcfdf4c6-qwgsr   1/1     Running   0           5m54s
nginx-deployment-86dcfdf4c6-tpr9f   1/1     Running   0           5m54s
ubuntu@ip-172-31-7-253:~$ █

```

i-0988b4b9aacdb96d3 (Master)

PublicIPs: 43.204.236.148 PrivateIPs: 172.31.7.253

******* THE END *******