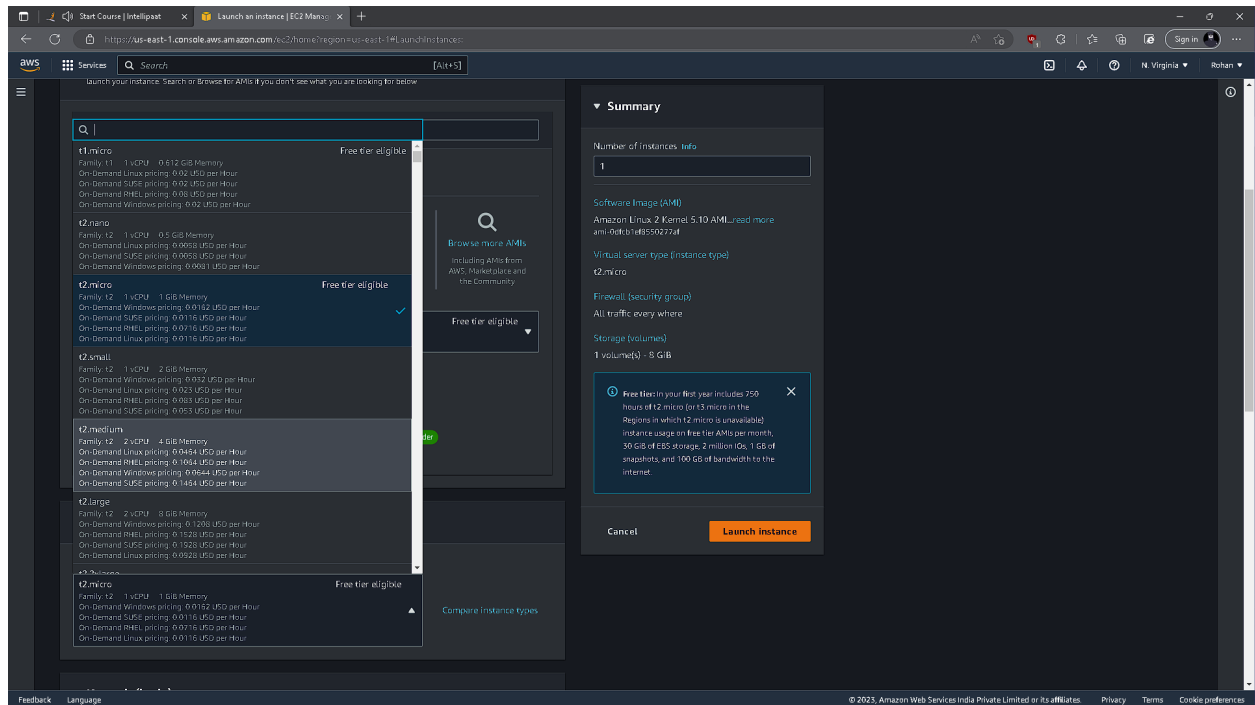


# Kubernetes - 1

You have been asked to:

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

Let's create an instance and install k8s first. Choose a minimum t2.med AWS instance for it.



## 3 instances since 3 node clusters are needed.

**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: K8s [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)** info

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

Recents Quick Start

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type  
ami-0d0c148550277af (64-bit x86) / ami-0c9323ab3463805f (64-bit ARM)  
Virtualization: hvm ENA-enabled: true Root device type: ebs

Description: Amazon Linux 2 Kernel 5.10 AMI 2.0.20230207.0 x86\_64 HVM gp2

**Summary**

Number of instances:  When launching more than 1 instance, consider EC2 Auto Scaling.

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI, read more  
ami-0d0c148550277af

Virtual server type (instance type): t2.medium

Firewall (security group): All traffic every where

Storage (volumes): 1 volumes - 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, 30 GiB of EBS storage, 3 million Ops, 1 GB of snapshots and 100 GB of bandwidth to the internet

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

## Instances created.

**Instances (1 / 3)** info

[Find instance by attribute or tag \(case-sensitive\)](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
K8sMaster	i-0996824dd4ba9eb1	Running	t2.medium	Initializing	No alarms	us-east-1d	ec2-52-202-219-78.com...	52.202.219.78	-
K8sClient1	i-0a681c1a0dc65ca8	Running	t2.medium	Initializing	No alarms	us-east-1d	ec2-44-203-196-82.com...	44.203.196.82	-
K8sClient2	i-0b7a110ec623e3cba	Running	t2.medium	Initializing	No alarms	us-east-1d	ec2-54-85-255-223.com...	54.85.255.223	-

**Instance: i-0b7a110ec623e3cba (K8s)**

**Details** **Security** **Networking** **Storage** **Status checks** **Monitoring** **Tags**

**Instance summary** info

Instance ID: i-0b7a110ec623e3cba (K8s)	Public IPv4 address: 54.85.255.223   <a href="#">open address</a>	Private IPv4 addresses: 172.31.86.163
IPv6 address: -	Instance state: Running	Public IPv4 DNS: ec2-54-85-255-223.compute-1.amazonaws.com   <a href="#">open address</a>
Hostname type: -	Private IP DNS name (IPv4 only): ip-172-31-86-163.ec2.internal	Elastic IP addresses: -
IP name: ip-172-31-86-163.ec2.internal	Instance type: t2.medium	AWS Compute Optimizer finding: No recommendations available for this instance.
Answer private/resource DNS name: IPv4 (A)	VPC ID: vpc-0c0671ad96deea470   <a href="#">open address</a>	Auto Scaling Group name: -
Auto-assigned IP address: 54.85.255.223 (Public IP)	Subnet ID: -	
IAM Role: -		

Use these command to install docker:

**sudo yum update**

**sudo yum install docker**

**sudo systemctl enable docker.service**

**sudo systemctl start docker.service**

**sudo systemctl start docker.service**

**systemctl status docker.service**

```
Start Course | Intellipaat | Instances | EC2 Management | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | My Courses | Anshul Pardehi | Sign in | Rohan
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/?shregion=us-east-1&connType=standard&instanceId=i-08996824dd4ba8eb1&osUser=ec2-user&sshPort=22

aws Services Search [Alt+S]

Installing:
docker x86_64 20.10.17-1.amzn2.0.2 amzn2extra-docker 39 M
Installing for dependencies:
containerd x86_64 1.6.8-1.amzn2.0.1 amzn2extra-docker 27 M
libgroup x86_64 0.41-21.amzn2 amzn2-core 66 k
pigs x86_64 2.3.4-1.amzn2.0.1 amzn2-core 81 k
runc x86_64 1.1.4-1.amzn2.0.1 amzn2extra-docker 2.7 M

Transaction Summary
-----
Install 1 Package (+4 Dependent packages)

Total download size: 69 M
Installed size: 260 M
Downloading packages:
(1/3): libgroup-0.41-21.amzn2.x86_64.rpm | 65 kB 00:00:00
(2/3): pigs-2.3.4-1.amzn2.0.1.x86_64.rpm | 81 kB 00:00:00
(3/3): containerd-1.6.8-1.amzn2.0.1.x86_64.rpm | 27 MB 00:00:00
(4/3): docker-20.10.17-1.amzn2.0.2.x86_64.rpm | 39 MB 00:00:00
(5/3): runc-1.1.4-1.amzn2.0.1.x86_64.rpm | 2.9 MB 00:00:00
-----
Total 99 MB/s | 69 MB 00:00:00

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : runc-1.1.4-1.amzn2.0.1.x86_64 1/5
Installing : containerd-1.6.8-1.amzn2.0.1.x86_64 2/5
Installing : libgroup-0.41-21.amzn2.x86_64 3/5
Installing : pigs-2.3.4-1.amzn2.0.1.x86_64 4/5
Installing : docker-20.10.17-1.amzn2.0.2.x86_64 5/5
Verifying : containerd-1.6.8-1.amzn2.0.1.x86_64 1/5
Verifying : pigs-2.3.4-1.amzn2.0.1.x86_64 2/5
Verifying : libgroup-0.41-21.amzn2.x86_64 3/5
Verifying : docker-20.10.17-1.amzn2.0.2.x86_64 4/5
Verifying : runc-1.1.4-1.amzn2.0.1.x86_64 5/5

Installed:
docker.x86_64 0:20.10.17-1.amzn2.0.2

Dependency Installed:
containerd.x86_64 0:1.6.8-1.amzn2.0.1 libgroup.x86_64 0:0.41-21.amzn2 pigs.x86_64 0:2.3.4-1.amzn2.0.1 runc.x86_64 0:1.1.4-1.amzn2.0.1

Complete!
[ec2-user@ip-172-31-81-112 ~]$

i-08996824dd4ba8eb1 (K8sMaster)
PublicIP: 52.202.219.78 PrivateIP: 172.31.81.112

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
```

## Docker is running

```
Start Course | Intellipast | Instances | EC2 Management Console | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | My Courses | Anshul Pardehi | Sign in
http://us-east-1.console.aws.amazon.com/ec2-instance-connect/home?region=us-east-1&connType=standard&instanceId=i-08996824d4ba8eb118p1&user=ec2-user&port=22&y

AWS Services Search [All+5]

Installing : runc-1.1.4-1.amzn2.0.1.x86_64
Installing : containerd-1.6.8-1.amzn2.0.1.x86_64
Installing : libgroup-0.41-21.amzn2.x86_64
Installing : pigs-2.3.4-1.amzn2.0.1.x86_64
Installing : docker-20.10.17-1.amzn2.0.2.x86_64
Verifying : containerd-1.6.8-1.amzn2.0.1.x86_64
Verifying : pigs-2.3.4-1.amzn2.0.1.x86_64
Verifying : libgroup-0.41-21.amzn2.x86_64
Verifying : docker-20.10.17-1.amzn2.0.2.x86_64
Verifying : runc-1.1.4-1.amzn2.0.1.x86_64

Installed:
docker.x86_64 0:20.10.17-1.amzn2.0.2

Dependency Installed:
containerd.x86_64 0:1.6.8-1.amzn2.0.1 libgroup.x86_64 0:0.41-21.amzn2 pigs.x86_64 0:2.3.4-1.amzn2.0.1 runc.x86_64 0:1.1.4-1.amzn2.0.1

Complete!
[ec2-user@ip-172-31-81-112 ~]$ sudo systemctl enable docker.service
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-81-112 ~]$ sudo systemctl start docker.service
[ec2-user@ip-172-31-81-112 ~]$ systemctl status docker.service
● Docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2023-02-15 06:34:43 UTC; 17s ago
     Docs: https://docs.docker.com
    Process: 3914 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
    Process: 3911 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
   Main PID: 3917 (dockerd)
      Tasks: 8
     Memory: 20.9M
    CGroup: /system.slice/docker.service
           └─3917 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.895530391Z" level=info msg="ClientConn switching balancer to `pick first`" module=grpc
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927437136Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927467782Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927686937Z" level=info msg="Loading containers: start."
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.153054650Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon opt... IP address"
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.217161799Z" level=info msg="Loading containers: done."
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.237971998Z" level=info msg="Docker daemon" commit=a89b642 graphdriver(s)=overlay2 version=20.10.17
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.238787602Z" level=info msg="Daemon has completed initialization"
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal systemd[1]: Started Docker Application Container Engine.
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.270892812Z" level=info msg="API listen on /run/docker.sock"
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-81-112 ~]$

i-08996824d4ba8eb11 (K8sMaster)
PublicIPs: 52.202.219.78 PrivateIPs: 172.31.81.112

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
```

=====installation of kubelet kubeadm kubectl=====

Run on BOTH MASTER and Worker

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
```

```
[kubernetes]
```

```
name=Kubernetes
```

```
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
```

```
exclude=kubelet kubeadm kubectl
```

```
EOF
```

```
# Set SELinux in permissive mode (effectively disabling it)
```

```
sudo setenforce 0
```

```
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
```

```
sudo systemctl enable --now kubelet
```

```
Start Course | Intellipast x Infrance | EC2 Management C x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x Start Course | Intellipast x +
http://aws-east-1.console.aws.amazon.com/ec2-instance-connect/sh/region-us-east-1?connType=standard&srcInstanceId=i-08956824d4ba8eb1&srcUser=ec2-user&srcPort=22#/?
AWS Services Search [All+5] N Virginia Rohan

Completed!
[ec2-user@ip-172-31-81-112 ~]$ sudo systemctl enable docker.service
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-81-112 ~]$ sudo systemctl start docker.service
[ec2-user@ip-172-31-81-112 ~]$ systemctl status docker.service
* docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2023-02-15 06:34:43 UTC; 17s ago
     Docs: https://docs.docker.com
    Process: 3914 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
    Process: 3911 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
   Main PID: 3917 (dockerd)
      Tasks: 0
     Memory: 20.6M
    CGroup: /system.slice/docker.service
            └─3917 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.895538391Z" level=info msg="ClientConn switching balancer to `pick first`" module=grpc
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927437136Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927476772Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927686597Z" level=info msg="Loading containers: start."
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.153454650Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon opt... IP address"
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.217613795Z" level=info msg="Loading containers: done."
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.235771998Z" level=info msg="Docker daemon" commit=a89b842 graphdriver(s)=overlay2 version=20.10.17
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.235878602Z" level=info msg="Daemon has completed initialization"
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal systemd[1]: Started Docker Application Container Engine.
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.270888312Z" level=info msg="API listen on /run/docker.sock"
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-81-112 ~]$ cat <EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
> [kubernetes]
> name=Kubernetes
> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch
> enabled=1
> gpgcheck=1
> gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
> exclude=kubelet kubeadm kubectl
> EOF
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch
enabled=1
gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kubelet kubeadm kubectl
[ec2-user@ip-172-31-81-112 ~]$
```

```
Start Course | Intellipast x Infrance | EC2 Management C x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x Start Course | Intellipast x +
http://aws-east-1.console.aws.amazon.com/ec2-instance-connect/sh/region-us-east-1?connType=standard&srcInstanceId=i-08956824d4ba8eb1&srcUser=ec2-user&srcPort=22#/?
AWS Services Search [All+5] N Virginia Rohan

Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-81-112 ~]$ sudo systemctl start docker.service
[ec2-user@ip-172-31-81-112 ~]$ systemctl status docker.service
* docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2023-02-15 06:34:43 UTC; 17s ago
     Docs: https://docs.docker.com
    Process: 3914 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
    Process: 3911 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
   Main PID: 3917 (dockerd)
      Tasks: 0
     Memory: 20.6M
    CGroup: /system.slice/docker.service
            └─3917 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.895538391Z" level=info msg="ClientConn switching balancer to `pick first`" module=grpc
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927437136Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927476772Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927686597Z" level=info msg="Loading containers: start."
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.153454650Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon opt... IP address"
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.217613795Z" level=info msg="Loading containers: done."
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.235771998Z" level=info msg="Docker daemon" commit=a89b842 graphdriver(s)=overlay2 version=20.10.17
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal systemd[1]: Started Docker Application Container Engine.
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.270888282Z" level=info msg="API listen on /run/docker.sock"
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-81-112 ~]$ cat <EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
> [kubernetes]
> name=Kubernetes
> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch
> enabled=1
> gpgcheck=1
> gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
> exclude=kubelet kubeadm kubectl
> EOF
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch
enabled=1
gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kubelet kubeadm kubectl
[ec2-user@ip-172-31-81-112 ~]$ sudo setenforce 0
setenforce: SELinux is disabled
[ec2-user@ip-172-31-81-112 ~]$ sudo sed -i 's/"SELINUX=enforcing"/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-81-112 ~]$
```

```
Start Course | Intellipast x Invoices | EC2 Management C... x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x Start Course | Intellipast x +
http://us-east-1.console.aws.amazon.com/ec2-instance-connect/sh?region=us-east-1&connType=standard&instanceId=i-08996824dd4ba8eb1&rootUser=ec2-user&srcPort=22&/

AWS Services Search [AWS] N Virginia Rohan

(11/11): 0f2a2af4740d47ead77c5088d7badi1659afca242816c1f4c6e4432a62df0b9d-kubernetes-cni-1.2.0-0.x86_64.rpm
Exporting gpg key 0x3218112
Retrieving key from https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
50 MB/s | 77 MB 00:00:01
Total
Running transaction test
Transaction test succeeded
Running transaction
Installing : libnetfilter_ghelper-1.0.0-10.amzn2.1.x86_64 1/11
Installing : cri-tools-1.25.0-1.amzn2.0.1.x86_64 2/11
Installing : libnetfilter_queue-1.0.3-1.amzn2.0.2.x86_64 3/11
Installing : ebtables-2.0.10-16.amzn2.0.1.x86_64 4/11
Installing : socat-1.7.3.2-2.amzn2.0.1.x86_64 5/11
Installing : libnetfilter_cttimeout-1.0.0-6.amzn2.1.x86_64 6/11
Installing : conntrack-tools-1.4.4-5.amzn2.2.x86_64 7/11
Installing : kubernetes-cni-1.2.0-0.x86_64 8/11
Installing : kubelet-1.26.1-0.x86_64 9/11
Installing : kubeadm-1.26.1-0.x86_64 10/11
Installing : kubelet-1.26.1-0.x86_64 11/11
Verifying : kubelet-1.26.1-0.x86_64 1/11
Verifying : libnetfilter_cttimeout-1.0.0-6.amzn2.1.x86_64 2/11
Verifying : socat-1.7.3.2-2.amzn2.0.1.x86_64 3/11
Verifying : kubernetes-cni-1.2.0-0.x86_64 4/11
Verifying : ebtables-2.0.10-16.amzn2.0.1.x86_64 5/11
Verifying : kubelet-1.26.1-0.x86_64 6/11
Verifying : libnetfilter_queue-1.0.3-1.amzn2.0.2.x86_64 7/11
Verifying : conntrack-tools-1.4.4-5.amzn2.2.x86_64 8/11
Verifying : kubeadm-1.26.1-0.x86_64 9/11
Verifying : cri-tools-1.25.0-1.amzn2.0.1.x86_64 10/11
Verifying : libnetfilter_ghelper-1.0.0-10.amzn2.1.x86_64 11/11

Installed:
kubeadm.x86_64 0:1.26.1-0 kubelet.x86_64 0:1.26.1-0 kubelet.x86_64 0:1.26.1-0

Dependency Installed:
conntrack-tools.x86_64 0:1.4.4-5.amzn2.2 cri-tools.x86_64 0:1.25.0-1.amzn2.0.1 ebtables.x86_64 0:2.0.10-16.amzn2.0.1 kubernetes-cni.x86_64 0:1.2.0-0
libnetfilter_ghelper.x86_64 0:1.0.0-10.amzn2.1 libnetfilter_cttimeout.x86_64 0:1.0.0-6.amzn2.1 libnetfilter_queue.x86_64 0:1.0.3-1.amzn2.0.2 socat.x86_64 0:1.7.3.2-2.amzn2.0.1

Complete!
[ec2-user@ip-172-31-81-112 ~]$ sudo systemctl enable --now kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /usr/lib/systemd/system/kubelet.service.
[ec2-user@ip-172-31-81-112 ~]$ sudo kubeadm init --apiserver-advertise-address=172.31.81.112 --pod-network-cidr=192.168.0.0/16

i-08996824dd4ba8eb1 (K8sMaster)
PublicIP: 52.202.219.78 PrivateIP: 172.31.81.112

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates Privacy Terms Cookie preferences
```

Note this output.

```
Start Course | Intellipast x Invoices | EC2 Management C... x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x Start Course | Intellipast x +
http://us-east-1.console.aws.amazon.com/ec2-instance-connect/sh?region=us-east-1&connType=standard&instanceId=i-08996824dd4ba8eb1&rootUser=ec2-user&srcPort=22&/

AWS Services Search [AWS] N Virginia Rohan

[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
[apiclient] All control plane components are healthy after 0.002400 seconds
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Shipping phase: Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-81-112.ec2.internal as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[bootstrap-token] Using token: 088au1idvvy4uypttjloa
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the certprover controller to automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificates rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

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 172.31.81.112:6443 --token 088au1idvvy4uypttjloa \
--discovery-token-ca-cert-hash sha256:57a5d7a8bc3884e4b4c26f8e87cd42fa32ad7ae2b26c75adb40421386c275

[ec2-user@ip-172-31-81-112 ~]$

i-08996824dd4ba8eb1 (K8sMaster)
PublicIP: 52.202.219.78 PrivateIP: 172.31.81.112

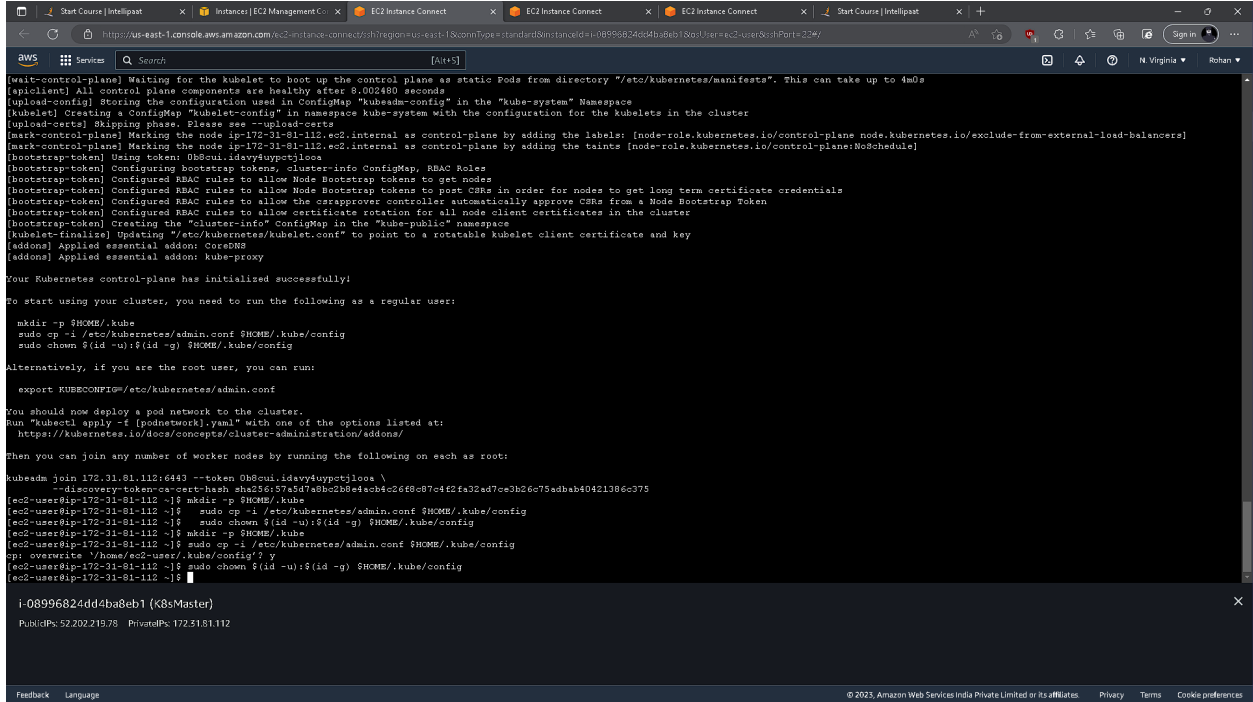
Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates Privacy Terms Cookie preferences
```

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
```



```
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
[apiclient] All control plane components are healthy after 8.002480 seconds
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-81-112.ec2.internal as control-plane by adding the labels: [node-role.kubernetes.io/control-plane:node.kubernetes.io/exclude-from-external-load-balancers]
[bootstrap-token] Using token: 0b8ouidavv4yupotjl0oa
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

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

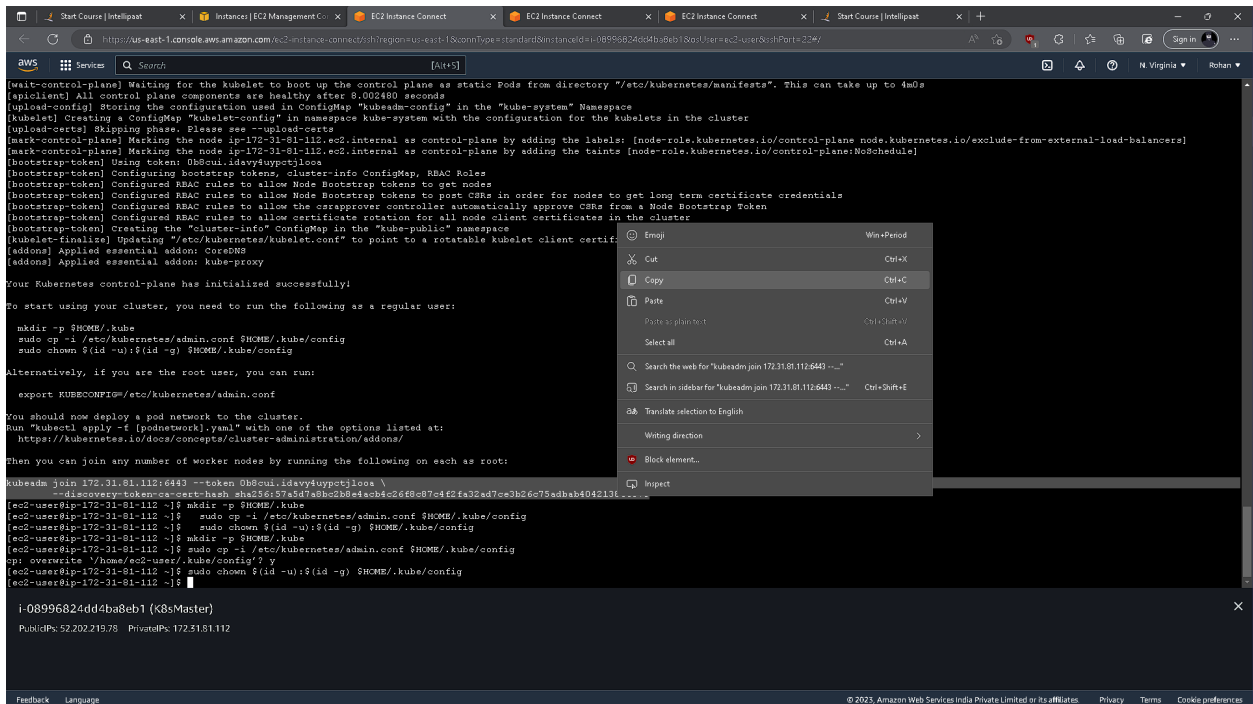
You should now deploy a pod network to the cluster.
Run 'kubectl apply -f [podnetwork].yaml' with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

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

kubeadm join 172.31.81.112:6443 --token 0b8ouidavv4yupotjl0oa \
--discovery-token-unsafe-sha256sum=57a5d1a8b3cb4e4b4c2649c87c4f2fa32ad7ce3b26c75adba40421386c375

[ec2-user@ip-172-31-81-112 ~]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-81-112 ~]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-81-112 ~]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$
```

You need to copy paste this in client.



```
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
[apiclient] All control plane components are healthy after 8.002480 seconds
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-81-112.ec2.internal as control-plane by adding the labels: [node-role.kubernetes.io/control-plane:node.kubernetes.io/exclude-from-external-load-balancers]
[bootstrap-token] Using token: 0b8ouidavv4yupotjl0oa
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

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 172.31.81.112:6443 --token 0b8ouidavv4yupotjl0oa \
--discovery-token-unsafe-sha256sum=57a5d1a8b3cb4e4b4c2649c87c4f2fa32ad7ce3b26c75adba40421386c375

[ec2-user@ip-172-31-81-112 ~]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-81-112 ~]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-81-112 ~]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$
```

## Pasted in client 1

```
Start Course | Intellipast x EC2 Management C... x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x Start Course | Intellipast x +
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/home?region=us-east-1&scanType=standard&srcInstanceId=i-0e861c1ae0cde5ca8&srcUser=ec2-user&srcPort=22#v
AWS Services Search [Alt+] N Virginia Rohan
Dependency Installed:
containerd.x86_64 0:1.6.8-1.amzn2.0.1 libgroup.x86_64 0:0.41-21.amzn2 pigz.x86_64 0:2.3.4-1.amzn2.0.1 runc.x86_64 0:1.1.4-1.amzn2.0.1
Completed
[ec2-user@ip-172-31-92-211 ~]$ sudo systemctl enable docker.service
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-92-211 ~]$ sudo systemctl start docker.service
[ec2-user@ip-172-31-92-211 ~]$ systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2023-02-15 06:34:45 UTC; 19s ago
     Docs: https://docs.docker.com
    Process: 3920 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
    Process: 3928 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
   Main PID: 3935 (dockerd)
      Tasks: 7
     Memory: 22.1M
    CGroup: /system.slice/docker.service
            └─3935 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Feb 15 06:34:45 ip-172-31-92-211.ec2.internal dockerd[3935]: time="2023-02-15T06:34:45.190597740Z" level=info msg="ClientConn switching balances to \"pick first\"" module=grpc
Feb 15 06:34:45 ip-172-31-92-211.ec2.internal dockerd[3935]: time="2023-02-15T06:34:45.224078274Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Feb 15 06:34:45 ip-172-31-92-211.ec2.internal dockerd[3935]: time="2023-02-15T06:34:45.224114252Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Feb 15 06:34:45 ip-172-31-92-211.ec2.internal dockerd[3935]: time="2023-02-15T06:34:45.224292801Z" level=info msg="Loading containers: start."
Feb 15 06:34:45 ip-172-31-92-211.ec2.internal dockerd[3935]: time="2023-02-15T06:34:45.415040932Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon opt... IP address"
Feb 15 06:34:45 ip-172-31-92-211.ec2.internal dockerd[3935]: time="2023-02-15T06:34:45.461726517Z" level=info msg="Loading containers: done."
Feb 15 06:34:45 ip-172-31-92-211.ec2.internal dockerd[3935]: time="2023-02-15T06:34:45.477210782Z" level=info msg="Docker daemon" commit=a99b42 graphdriver(s)=overlay2 version=20.10.17
Feb 15 06:34:45 ip-172-31-92-211.ec2.internal dockerd[3935]: time="2023-02-15T06:34:45.477319242Z" level=info msg="Daemon has completed initialization"
Feb 15 06:34:45 ip-172-31-92-211.ec2.internal systemd[1]: Started Docker Application Container Engine.
Feb 15 06:34:45 ip-172-31-92-211.ec2.internal dockerd[3935]: time="2023-02-15T06:34:45.500771108Z" level=info msg="API listen on /run/docker.sock"
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-92-211 ~]$ sudo systemctl enable --now kubelet
[ec2-user@ip-172-31-92-211 ~]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-92-211 ~]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
kubeadm.x86_64 0:1.26.1-0 kubelet.x86_64 0:1.26.1-0
No package kubelet available.
No package kubeadm available.
No package kubectl available.
Error: Nothing to do
[ec2-user@ip-172-31-92-211 ~]$ sudo systemctl enable --now kubelet
Failed to execute operation: No such file or directory
[ec2-user@ip-172-31-92-211 ~]$ kubeadm join 172.31.81.112:6443 --token 0b8cui.idavy4uyppctjloa \
--discovery-token-ca-cert-hash sha256:57a5d7a8bc2b8e4ac426f8c7c4f2fa32a47ce3b26c75adbab40421386c375
i-0e861c1ae0cde5ca8 (K8sClient)
PublicIPs: 44.205.196.82 PrivateIPs: 172.31.92.211
Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
```

## Worker node has been joined.Similarly do it for another client/worker server as well.

```
Start Course | Intellipast x EC2 Management C... x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x Start Course | Intellipast x +
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/home?region=us-east-1&scanType=standard&srcInstanceId=i-0e861c1ae0cde5ca8&srcUser=ec2-user&srcPort=22#v
AWS Services Search [Alt+] N Virginia Rohan
kubeadm.x86_64 0:1.26.1-0 kubelet.x86_64 0:1.26.1-0 kubelet.x86_64 0:1.26.1-0
Dependency Installed:
containerd-tools.x86_64 0:1.4.4-5.amzn2.2 cri-tools.x86_64 0:1.25.0-1.amzn2.0.1 ebtables.x86_64 0:2.0.10-16.amzn2.0.1 kubernetes-cni.x86_64 0:1.2.0-0
libnetfilter_cthnatpr.x86_64 0:1.0.0-10.amzn2.1 libnetfilter_cttimeout.x86_64 0:1.0.0-6.amzn2.1 libnetfilter_queue.x86_64 0:1.0.2-2.amzn2.0.1 socat.x86_64 0:1.7.3.2-2.amzn2.0.1
Completed
[ec2-user@ip-172-31-92-211 ~]$ sudo systemctl enable --now kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /usr/lib/systemd/system/kubelet.service.
[ec2-user@ip-172-31-92-211 ~]$ kubeadm join 172.31.81.112:6443 --token 0b8cui.idavy4uyppctjloa \
--discovery-token-ca-cert-hash sha256:57a5d7a8bc2b8e4ac426f8c7c4f2fa32a47ce3b26c75adbab40421386c375
[preflight] Running pre-flight checks
error execution phase preflight: [preflight] Some fatal errors occurred:
[WARNING IsPrivilegedUser] user is not running as root
[preflight] If you know what you are doing, you can make a check non-fatal with '--ignore-preflight-errors=...'
To see the stack trace of this error execute with '--vv5 or higher
[ec2-user@ip-172-31-92-211 ~]$ kubeadm join 172.31.81.112:6443 --token 0b8cui.idavy4uyppctjloa \
--discovery-token-ca-cert-hash sha256:57a5d7a8bc2b8e4ac426f8c7c4f2fa32a47ce3b26c75adbab40421386c375
[preflight] Running pre-flight checks
error execution phase preflight: [preflight] Some fatal errors occurred:
[WARNING IsPrivilegedUser] user is not running as root
[preflight] If you know what you are doing, you can make a check non-fatal with '--ignore-preflight-errors=...'
To see the stack trace of this error execute with '--vv5 or higher
[ec2-user@ip-172-31-92-211 ~]$ sudo kubeadm join 172.31.81.112:6443 --token 0b8cui.idavy4uyppctjloa \
--discovery-token-ca-cert-hash sha256:57a5d7a8bc2b8e4ac426f8c7c4f2fa32a47ce3b26c75adbab40421386c375
[preflight] Running pre-flight checks
[WARNING FileExisting-tls-crt: tc not found in system path
[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 apiserer and a response was received.
* The Kubelet was informed of the new secure connection details.

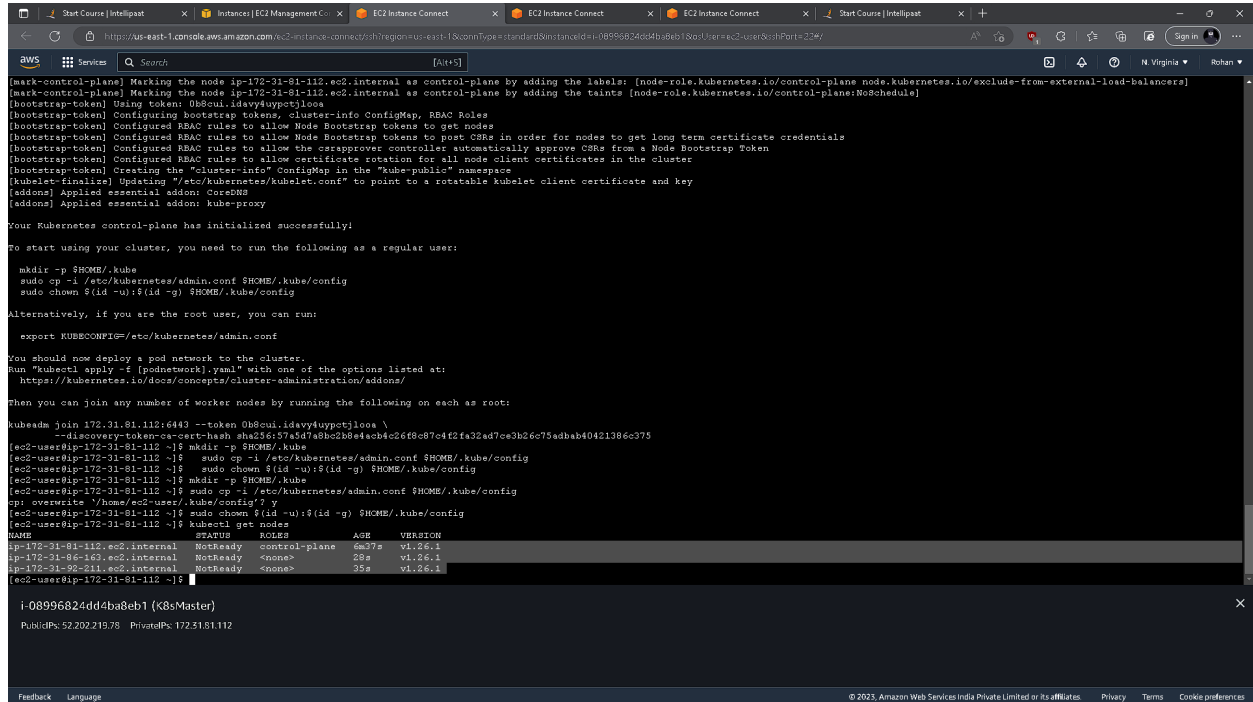
Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

[ec2-user@ip-172-31-92-211 ~]$
i-0e861c1ae0cde5ca8 (K8sClient)
PublicIPs: 44.205.196.82 PrivateIPs: 172.31.92.211
Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
```



## kubectl get node

You can see 3 node cluster is created. They are not ready yet. For it install calico on master.



```
[mark-control-plane] Marking the node ip-172-31-81-112.ec2.internal as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-81-112.ec2.internal as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 0b8cui.idavy4uyptjloos
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kubernetes" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

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 172.31.81.112:6443 --token 0b8cui.idavy4uyptjloos \
--discovery-token-ca-cert-hash sha256:57a5d7a8b0c2b8e4ac4c26f8c87c4f2fe32ad7ce3b26c75adbab40421386c375
[ec2-user@ip-172-31-81-112 ~]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-81-112 ~]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-81-112 ~]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE     VERSION
ip-172-31-81-112.ec2.internal        NotReady  control-plane  6m37s   v1.26.1
ip-172-31-81-102.ec2.internal        NotReady  <none>        28s     v1.26.1
ip-172-31-92-211.ec2.internal        NotReady  <none>        35s     v1.26.1
[ec2-user@ip-172-31-81-112 ~]$
```

i 08996824dd4ba8eb1 (K8sMaster)

PublicIPs: 52.202.219.78 PrivateIPs: 172.31.81.112

## Install Calico on MASTER ONLY:

**kubectl create -f**

**[https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/tigera-operator.](https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/tigera-operator.yaml)**

**yaml**

**curl**

**<https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/custom-resources.yaml> -O**

**kubectl create -f custom-resources.yaml**

## Nodes are ready.

```
Start Course | Intellipast x Instances | EC2 Management C... x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x Start Course | Intellipast x +
http://us-east-1.console.aws.amazon.com/ec2-instance-connect/sh?region=us-east-1&connType=standard&instanceId=i-08996824dd4ba8eb1&srcUser=ec2-user&sshPort=22&w/

AWS Services Search [All+5]
customresourcedefinition.apis.k8s.io/apiservers.operator.tigera.io created
customresourcedefinition.apis.k8s.io/imagesets.operator.tigera.io created
customresourcedefinition.apis.k8s.io/installations.operator.tigera.io created
customresourcedefinition.apis.k8s.io/tigerastatus.operator.tigera.io created
servicesaccount/tigera-operator created
clusterrole.rbac.authorization.k8s.io/tigera-operator created
clusterrolebinding.rbac.authorization.k8s.io/tigera-operator created
deployment.apps/tigera-operator created
[ec2-user@ip-172-31-81-112 ~]$ curl https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/custom-resources.yaml -O
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 827 100 827 0 0 2947 0 --:--:-- --:--:-- --:--:-- 2943
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f custom-resources.yaml
installation.operator.tigera.io/default created
apiserver.operator.tigera.io/default created
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal NotReady control-plane 7m25s v1.26.1
ip-172-31-86-163.ec2.internal NotReady <none> 76s v1.26.1
ip-172-31-92-211.ec2.internal NotReady <none> 83s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal NotReady control-plane 7m27s v1.26.1
ip-172-31-86-163.ec2.internal NotReady <none> 78s v1.26.1
ip-172-31-92-211.ec2.internal NotReady <none> 85s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal NotReady control-plane 7m30s v1.26.1
ip-172-31-86-163.ec2.internal NotReady <none> 81s v1.26.1
ip-172-31-92-211.ec2.internal NotReady <none> 88s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal NotReady control-plane 7m36s v1.26.1
ip-172-31-86-163.ec2.internal Ready <none> 87s v1.26.1
ip-172-31-92-211.ec2.internal Ready <none> 94s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal NotReady control-plane 7m40s v1.26.1
ip-172-31-86-163.ec2.internal Ready <none> 91s v1.26.1
ip-172-31-92-211.ec2.internal Ready <none> 98s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal Ready control-plane 7m46s v1.26.1
ip-172-31-86-163.ec2.internal Ready <none> 97s v1.26.1
ip-172-31-92-211.ec2.internal Ready <none> 104s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$

i-08996824dd4ba8eb1 (K8sMaster)
PublicIPs: 52.202.219.78 PrivateIPs: 172.31.81.112

Feedback Language
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
```

To create deployment:

nano nginx-deployment.yaml

#paste this:

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

labels:

app: nginx

spec:

replicas: 3

strategy:

type: RollingUpdate

rollingUpdate:

maxUnavailable: 1

maxSurge: 1

selector:

matchLabels:

app: nginx

environment: production

template:

metadata:

labels:

app: nginx

environment: production

spec:

containers:

- name: nginx-container

image: nginx

kubectl create -f nginx-deployment.yaml

kubectl get deploy #will get you deployment pods

kubectl get pods

kubectl get rs

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-replicaset
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      environment: production
  template:
    metadata:
      labels:
        environment: production
    spec:
      containers:
        - name: nginx-container
          image: nginx
```

08996824dd4ba8eb1 (K8sMaster)  
PublicIP: 52.202.219.78 PrivateIP: 172.31.81.112

```
clusterrolebinding.rbac.authorization.k8s.io/tigers-operator created
deployment.apps/tigers-operator created
[ec2-user@ip-172-31-81-112 ~]$ curl https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/custom-resources.yaml -O
Total    Received    Merged    Average Speed    Elapsed Time    Current Time    Upload Total    Spent    Left    Speed
100    827    100    827    0    0    2947    0    -----    2943
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f custom-resources.yaml
Installation.operator.tigers.io/default created
apiserver.operator.tigers.io/default created
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
ip-172-31-81-112.ec2.internal        NotReady    control-plane    7m25s    v1.26.1
ip-172-31-86-163.ec2.internal        NotReady    <none>          76s    v1.26.1
ip-172-31-92-211.ec2.internal        NotReady    <none>          82s    v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
ip-172-31-81-112.ec2.internal        NotReady    control-plane    7m27s    v1.26.1
ip-172-31-86-163.ec2.internal        NotReady    <none>          78s    v1.26.1
ip-172-31-92-211.ec2.internal        NotReady    <none>          85s    v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
ip-172-31-81-112.ec2.internal        NotReady    control-plane    7m30s    v1.26.1
ip-172-31-86-163.ec2.internal        NotReady    <none>          81s    v1.26.1
ip-172-31-92-211.ec2.internal        NotReady    <none>          88s    v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
ip-172-31-81-112.ec2.internal        NotReady    control-plane    7m32s    v1.26.1
ip-172-31-86-163.ec2.internal        NotReady    <none>          83s    v1.26.1
ip-172-31-92-211.ec2.internal        NotReady    <none>          89s    v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
ip-172-31-81-112.ec2.internal        NotReady    control-plane    7m34s    v1.26.1
ip-172-31-86-163.ec2.internal        NotReady    <none>          85s    v1.26.1
ip-172-31-92-211.ec2.internal        NotReady    <none>          91s    v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
ip-172-31-81-112.ec2.internal        NotReady    control-plane    7m36s    v1.26.1
ip-172-31-86-163.ec2.internal        NotReady    <none>          87s    v1.26.1
ip-172-31-92-211.ec2.internal        NotReady    <none>          97s    v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ nano nginx-deployment.yaml
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f nginx-deployment.yaml
error: the path "nginx-deployment.yaml" does not exist
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f nginx-deployment.yaml
replicaset.apps/nginx-replicaset created
[ec2-user@ip-172-31-81-112 ~]$
```

08996824dd4ba8eb1 (K8sMaster)  
PublicIP: 52.202.219.78 PrivateIP: 172.31.81.112

```
Start Course | Intellipast x Infa... | EC2 Management C... x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x Start Course | Intellipast x +
http://us-east-1.console.aws.amazon.com/ec2-instance-connect/home?region=us-east-1&connType=standard&instanceId=i-08996824dd4ba8eb1&srcUser=ec2-user&srcPort=22w/

AWS Services Search [Alt+]

NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal NotReady control-plane 7m30s v1.26.1
ip-172-31-86-163.ec2.internal NotReady <none> 81s v1.26.1
ip-172-31-92-211.ec2.internal NotReady <none> 88s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal NotReady control-plane 7m36s v1.26.1
ip-172-31-86-163.ec2.internal Ready <none> 87s v1.26.1
ip-172-31-92-211.ec2.internal Ready <none> 94s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal NotReady control-plane 7m40s v1.26.1
ip-172-31-86-163.ec2.internal Ready <none> 91s v1.26.1
ip-172-31-92-211.ec2.internal Ready <none> 98s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal Ready control-plane 7m46s v1.26.1
ip-172-31-86-163.ec2.internal Ready <none> 97s v1.26.1
ip-172-31-92-211.ec2.internal Ready <none> 104s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ nano nginx-deployment.yaml
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f nginx-deployment.yaml
error: the path "nginxdeployment.yaml" does not exist
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f nginx-deployment.yaml
replicaset.apps/nginx-replicaset created
[ec2-user@ip-172-31-81-112 ~]$ kubectl get deploy
No resources found in default namespace.
[ec2-user@ip-172-31-81-112 ~]$ kubectl get deploy
No resources found in default namespace.
[ec2-user@ip-172-31-81-112 ~]$ kubectl get deployments
No resources found in default namespace.
[ec2-user@ip-172-31-81-112 ~]$ kubectl get po
NAME READY STATUS RESTARTS AGE
nginx-replicaset-2nc45 1/1 Running 0 38s
nginx-replicaset-t25m9 1/1 Running 0 38s
[ec2-user@ip-172-31-81-112 ~]$ kubectl get deploy nginx-deployment.yaml
Error from server (NotFound): deployments.apps "nginx-deployment.yaml" not found
[ec2-user@ip-172-31-81-112 ~]$ kubectl get deploy nginx-replicaset
Error from server (NotFound): deployments.apps "nginx-replicaset" not found
[ec2-user@ip-172-31-81-112 ~]$ kubectl get po
NAME READY STATUS RESTARTS AGE
nginx-replicaset-2nc45 1/1 Running 0 75s
nginx-replicaset-t25m9 1/1 Running 0 75s
nginx-replicaset-zm6jg 1/1 Running 0 75s
[ec2-user@ip-172-31-81-112 ~]$

i-08996824dd4ba8eb1 (K8sMaster)
PublicIPs: 52.202.219.78 PrivateIPs: 172.31.81.112

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
```

kubectl get rs.  
You can see three rs created.

```
Start Course | Intellipast x Infa... | EC2 Management C... x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x Start Course | Intellipast x +
http://us-east-1.console.aws.amazon.com/ec2-instance-connect/home?region=us-east-1&connType=standard&instanceId=i-08996824dd4ba8eb1&srcUser=ec2-user&srcPort=22w/

AWS Services Search [Alt+]

NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal NotReady control-plane 7m36s v1.26.1
ip-172-31-86-163.ec2.internal Ready <none> 87s v1.26.1
ip-172-31-92-211.ec2.internal Ready <none> 94s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal NotReady control-plane 7m40s v1.26.1
ip-172-31-86-163.ec2.internal Ready <none> 91s v1.26.1
ip-172-31-92-211.ec2.internal Ready <none> 98s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-81-112.ec2.internal Ready control-plane 7m46s v1.26.1
ip-172-31-86-163.ec2.internal Ready <none> 97s v1.26.1
ip-172-31-92-211.ec2.internal Ready <none> 104s v1.26.1
[ec2-user@ip-172-31-81-112 ~]$ nano nginx-deployment.yaml
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f nginx-deployment.yaml
error: the path "nginxdeployment.yaml" does not exist
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f nginx-deployment.yaml
replicaset.apps/nginx-replicaset created
[ec2-user@ip-172-31-81-112 ~]$ kubectl get deploy
No resources found in default namespace.
[ec2-user@ip-172-31-81-112 ~]$ kubectl get deploy
No resources found in default namespace.
[ec2-user@ip-172-31-81-112 ~]$ kubectl get deployments
No resources found in default namespace.
[ec2-user@ip-172-31-81-112 ~]$ kubectl get po
NAME READY STATUS RESTARTS AGE
nginx-replicaset-2nc45 1/1 Running 0 38s
nginx-replicaset-t25m9 1/1 Running 0 38s
nginx-replicaset-zm6jg 1/1 Running 0 38s
[ec2-user@ip-172-31-81-112 ~]$ kubectl get deploy nginx-deployment.yaml
Error from server (NotFound): deployments.apps "nginx-deployment.yaml" not found
[ec2-user@ip-172-31-81-112 ~]$ kubectl get deploy nginx-replicaset
Error from server (NotFound): deployments.apps "nginx-replicaset" not found
[ec2-user@ip-172-31-81-112 ~]$ kubectl get po
NAME READY STATUS RESTARTS AGE
nginx-replicaset-2nc45 1/1 Running 0 75s
nginx-replicaset-t25m9 1/1 Running 0 75s
nginx-replicaset-zm6jg 1/1 Running 0 75s
[ec2-user@ip-172-31-81-112 ~]$ kubectl get rs
NAME DESIRED CURRENT READY AGE
nginx-replicaset 3 3 97s
[ec2-user@ip-172-31-81-112 ~]$

i-08996824dd4ba8eb1 (K8sMaster)
PublicIPs: 52.202.219.78 PrivateIPs: 172.31.81.112

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
```