Advanced DevOps Lab

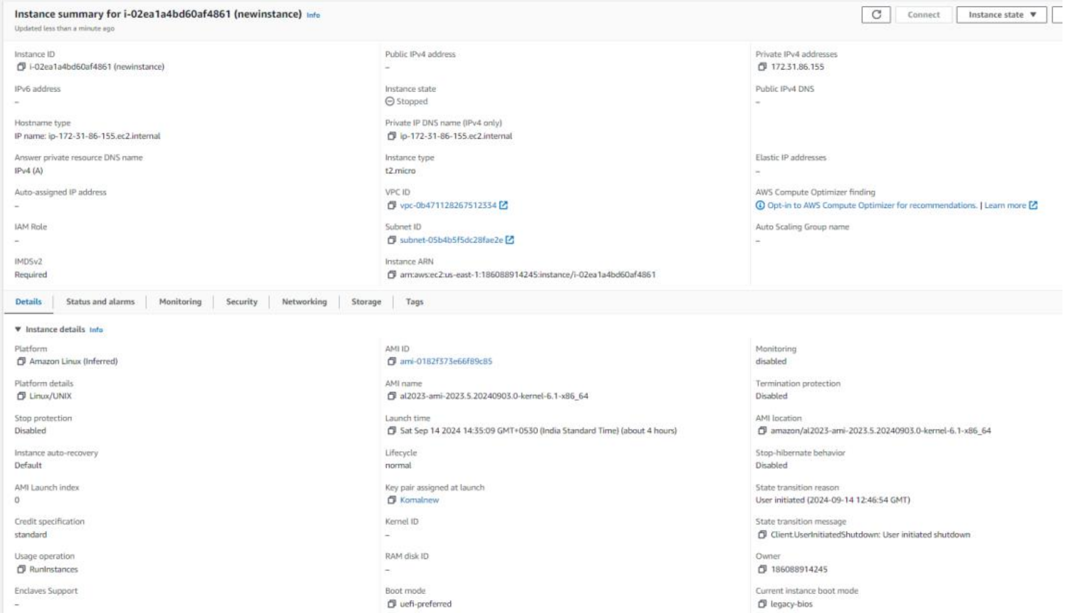
Experiment:4

<u>Aim</u>: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

<u>Execution:</u>

Creating 1 instance, 'newinstance' .

Creating newinstance:



Instance summary for i-02ea1a4bd60af4861 (newinstance) Info

Updated less than a minute ago

Instance ID
i-02ea1a4bd60af4861 (newinstance)

IPv6 address
–

Hostname type
IP name: ip-172-31-86-155.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
–

IAM Role
–

IMDSv2
Required

Public IPv4 address
–

Instance state
⊖ Stopped

Private IP DNS name (IPv4 only)
ip-172-31-86-155.ec2.internal

Instance type
t2.micro

VPC ID
vpc-0b471128267512334

Subnet ID
subnet-05b4b5f5dc28fae2e

Instance ARN
arn:aws:ec2:us-east-1:186088914245:instance/i-02ea1a4bd60af4861

Private IPv4 addresses
172.31.86.155

Public IPv4 DNS
–

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
–

▼ Instance details Info

Platform
Amazon Linux (Inferred)

Platform details
Linux/UNIX

Stop protection
Disabled

Instance auto-recovery
Default

AMI Launch index
0

Credit specification
standard

Usage operation
RunInstances

Enclaves Support
–

AMI ID
ami-0182f373e66f89c85

AMI name
al2023-ami-2023.5.20240903.0-kernel-6.1-x86_64

Launch time
Sat Sep 14 2024 14:35:09 GMT+0530 (India Standard Time) (about 4 hours)

Lifecycle
normal

Key pair assigned at launch
Komalnew

Kernel ID
–

RAM disk ID
–

Boot mode
uefi-preferred

Monitoring
disabled

Termination protection
Disabled

AMI location
amazon/al2023-ami-2023.5.20240903.0-kernel-6.1-x86_64

Stop-hibernate behavior
Disabled

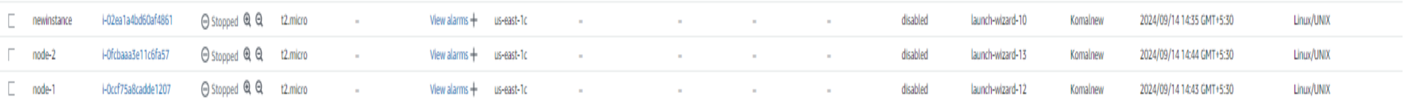
State transition reason
User initiated (2024-09-14 12:46:54 GMT)

State transition message
Client.UserInitiatedShutdown: User initiated shutdown

Owner
186088914245

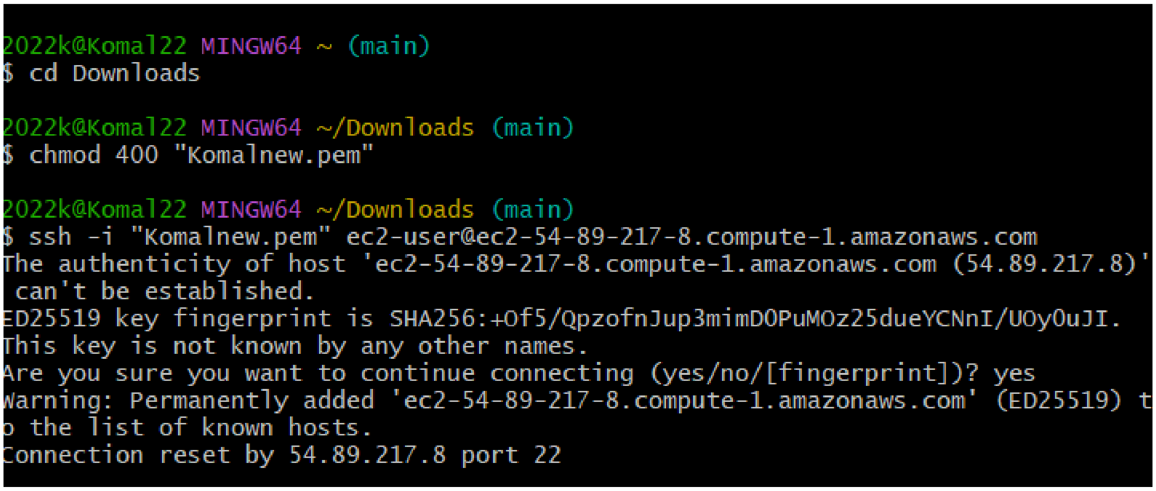
Current instance boot mode
legacy-bios

we created and configured all the instance:

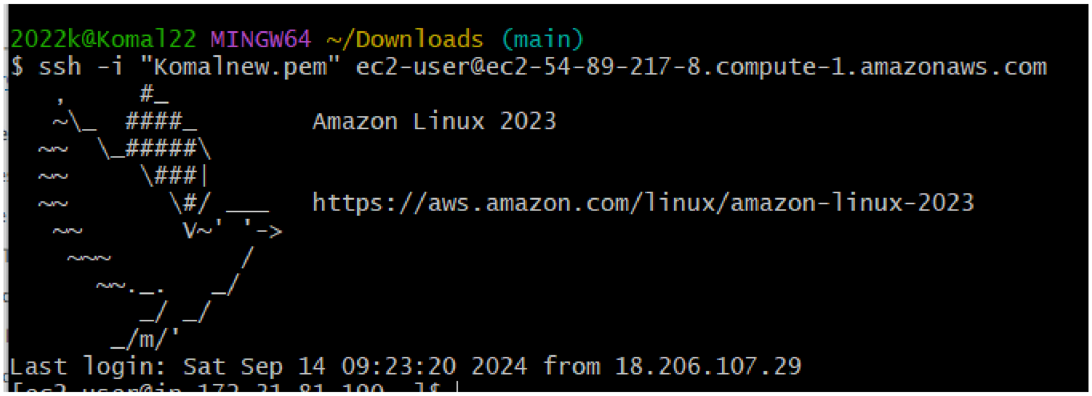


□	newinstance	i-02ea1a4bd60af4861	⊖ Stopped	t2.micro	-	View alarms +	us-east-1c	-	-	-	-	disabled	launch-wizard-10	Komalnew	2024/09/14 14:35 GMT+5:30	Linux/UNIX
□	node-2	i-0fcbaaa3e11c6fa57	⊖ Stopped	t2.micro	-	View alarms +	us-east-1c	-	-	-	-	disabled	launch-wizard-13	Komalnew	2024/09/14 14:44 GMT+5:30	Linux/UNIX
□	node-1	i-0ccf75a8cadde1207	⊖ Stopped	t2.micro	-	View alarms +	us-east-1c	-	-	-	-	disabled	launch-wizard-12	Komalnew	2024/09/14 14:43 GMT+5:30	Linux/UNIX

Following commands are used for making an SSH connection in all 3 machines:

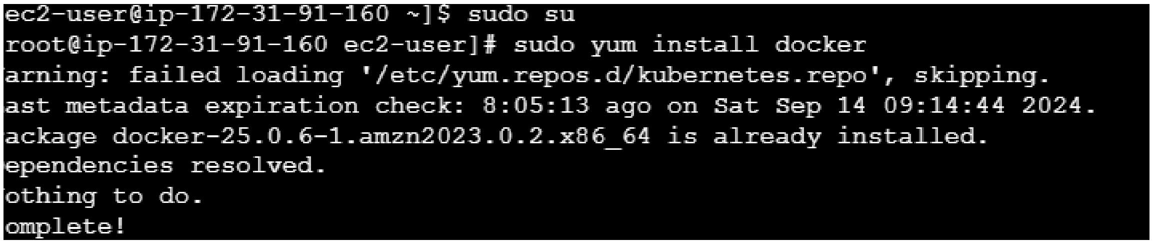


```
2022k@Komal22 MINGW64 ~ (main)
$ cd Downloads

2022k@Komal22 MINGW64 ~/Downloads (main)
$ chmod 400 "Komalnew.pem"

2022k@Komal22 MINGW64 ~/Downloads (main)
$ ssh -i "Komalnew.pem" ec2-user@ec2-54-89-217-8.compute-1.amazonaws.com
The authenticity of host 'ec2-54-89-217-8.compute-1.amazonaws.com (54.89.217.8)'
 can't be established.
ED25519 key fingerprint is SHA256:+Of5/QpzofnJup3mimDOPuMOz25dueYCNnI/UOyOuJI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-89-217-8.compute-1.amazonaws.com' (ED25519) t
o the list of known hosts.
Connection reset by 54.89.217.8 port 22
```

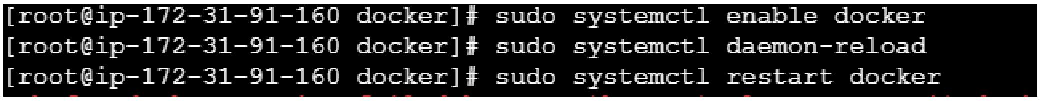


```
2022k@Komal22 MINGW64 ~/Downloads (main)
$ ssh -i "Komalnew.pem" ec2-user@ec2-54-89-217-8.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sat Sep 14 09:23:20 2024 from 18.206.107.29
[ec2-user@ip-172-31-81-190 ~]$ 
```

Docker Installation:

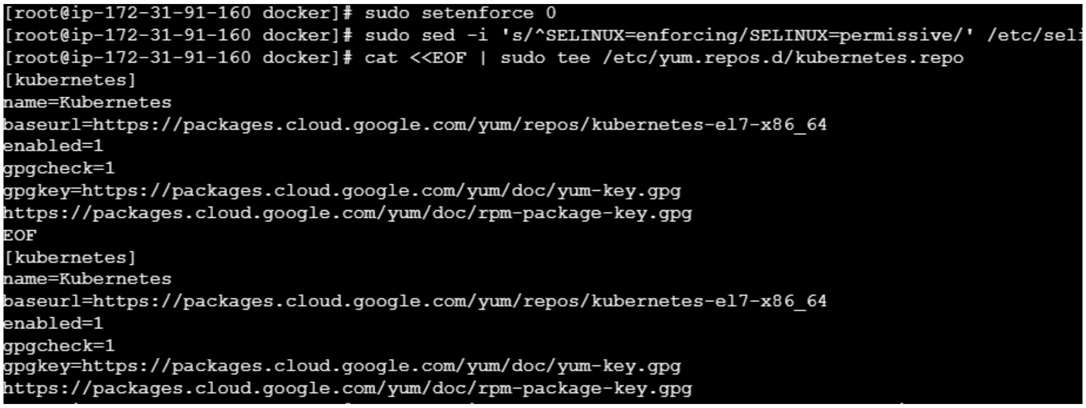


```
ec2-user@ip-172-31-91-160 ~]$ sudo su
root@ip-172-31-91-160 ec2-user]# sudo yum install docker
arning: failed loading '/etc/yum.repos.d/kubernetes.repo', skipping.
ast metadata expiration check: 8:05:13 ago on Sat Sep 14 09:14:44 2024.
ackage docker-25.0.6-1.amzn2023.0.2.x86_64 is already installed.
ependencies resolved.
othing to do.
omplete!
```

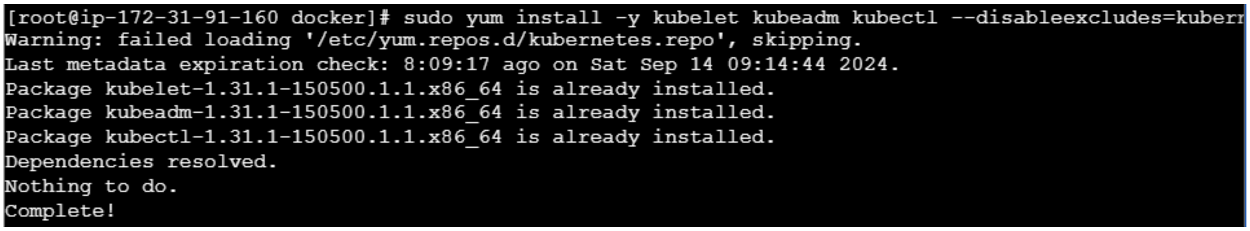


```
[root@ip-172-31-91-160 docker]# sudo systemctl enable docker
[root@ip-172-31-91-160 docker]# sudo systemctl daemon-reload
[root@ip-172-31-91-160 docker]# sudo systemctl restart docker
```

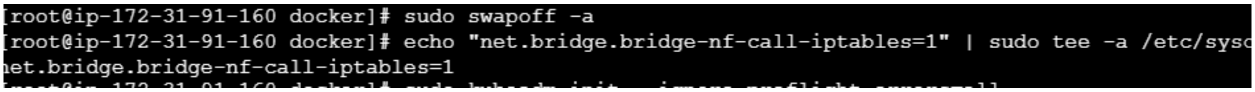
Kubernetes installation:



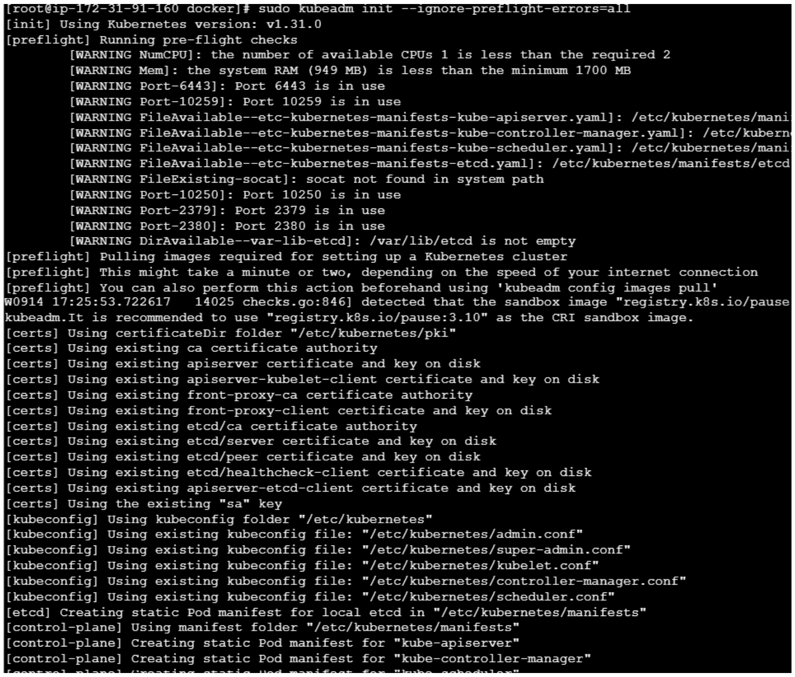
```
[root@ip-172-31-91-160 docker]# sudo setenforce 0
[root@ip-172-31-91-160 docker]# sudo sed -i 's/^SELINUX=enforcing/SELINUX=permissive/' /etc/seli
[root@ip-172-31-91-160 docker]# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
```



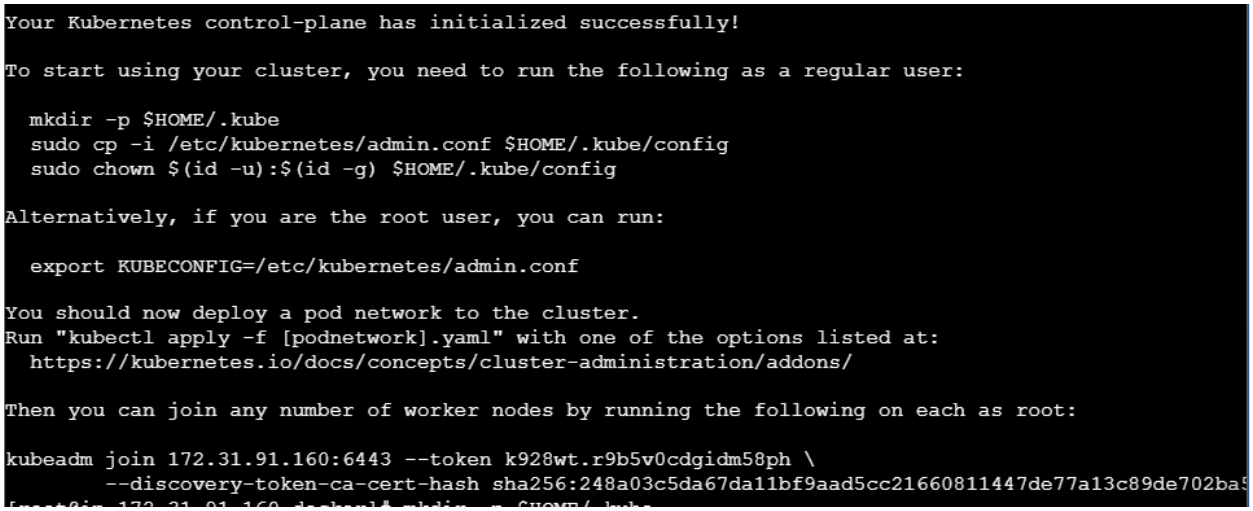
```
[root@ip-172-31-91-160 docker]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubern
Warning: failed loading '/etc/yum.repos.d/kubernetes.repo', skipping.
Last metadata expiration check: 8:09:17 ago on Sat Sep 14 09:14:44 2024.
Package kubelet-1.31.1-150500.1.1.x86_64 is already installed.
Package kubeadm-1.31.1-150500.1.1.x86_64 is already installed.
Package kubectl-1.31.1-150500.1.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```



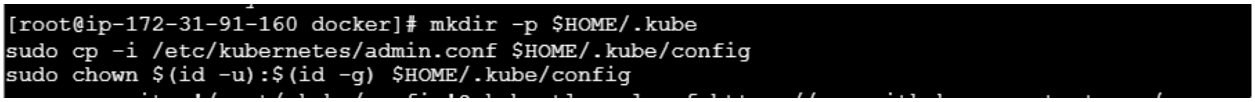
```
[root@ip-172-31-91-160 docker]# sudo swapoff -a
[root@ip-172-31-91-160 docker]# echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysc
net.bridge.bridge-nf-call-iptables=1
```



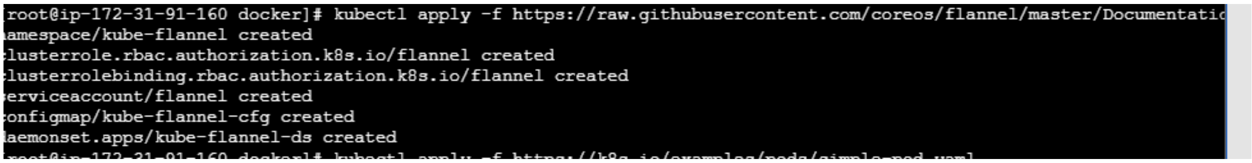
```
[root@ip-172-31-91-160 docker]# sudo kubeadm init --ignore-preflight-errors=all
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
        [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
        [WARNING Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
        [WARNING Port-6443]: Port 6443 is in use
        [WARNING Port-10259]: Port 10259 is in use
        [WARNING FileAvailable--etc-kubernetes-manifests-kube-apiserver.yaml]: /etc/kubernetes/mani
        [WARNING FileAvailable--etc-kubernetes-manifests-kube-controller-manager.yaml]: /etc/kuberne
        [WARNING FileAvailable--etc-kubernetes-manifests-kube-scheduler.yaml]: /etc/kubernetes/mani
        [WARNING FileAvailable--etc-kubernetes-manifests-etcd.yaml]: /etc/kubernetes/manifests/etcd
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING Port-10250]: Port 10250 is in use
        [WARNING Port-2379]: Port 2379 is in use
        [WARNING Port-2380]: Port 2380 is in use
        [WARNING DirAvailable--var-lib-etcd]: /var/lib/etcd is not empty
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0914 17:25:53.722617   14025 checks.go:846] detected that the sandbox image "registry.k8s.io/pause
kubeadm.It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Using existing ca certificate authority
[certs] Using existing apiserver certificate and key on disk
[certs] Using existing apiserver-kubelet-client certificate and key on disk
[certs] Using existing front-proxy-ca certificate authority
[certs] Using existing front-proxy-client certificate and key on disk
[certs] Using existing etcd/ca certificate authority
[certs] Using existing etcd/server certificate and key on disk
[certs] Using existing etcd/peer certificate and key on disk
[certs] Using existing etcd/healthcheck-client certificate and key on disk
[certs] Using existing apiserver-etcd-client certificate and key on disk
[certs] Using the existing "sa" key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Using existing kubeconfig file: "/etc/kubernetes/admin.conf"
[kubeconfig] Using existing kubeconfig file: "/etc/kubernetes/super-admin.conf"
[kubeconfig] Using existing kubeconfig file: "/etc/kubernetes/kubelet.conf"
[kubeconfig] Using existing kubeconfig file: "/etc/kubernetes/controller-manager.conf"
[kubeconfig] Using existing kubeconfig file: "/etc/kubernetes/scheduler.conf"
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[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"
```



```
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 172.31.91.160:6443 --token k928wt.r9b5v0cdgidm58ph \
        --discovery-token-ca-cert-hash sha256:248a03c5da67da11bf9aad5cc21660811447de77a13c89de702ba5
[root@ip-172-31-91-160 docker]# mkdir -p $HOME/.kube
```

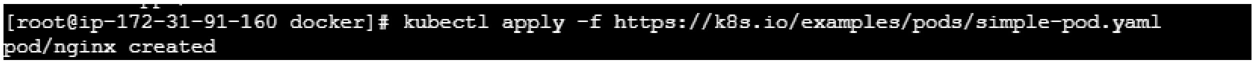


```
[root@ip-172-31-91-160 docker]# mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

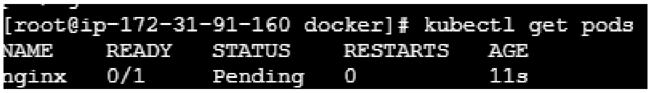


```
root@ip-172-31-91-160 docker]# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentatic
amespace/kube-flannel created
lusterrole.rbac.authorization.k8s.io/flannel created
lusterrolebinding.rbac.authorization.k8s.io/flannel created
erviceaccount/flannel created
onfigmap/kube-flannel-cfg created
aemonset.apps/kube-flannel-ds created
root@ip-172-31-91-160 docker]# kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml
```

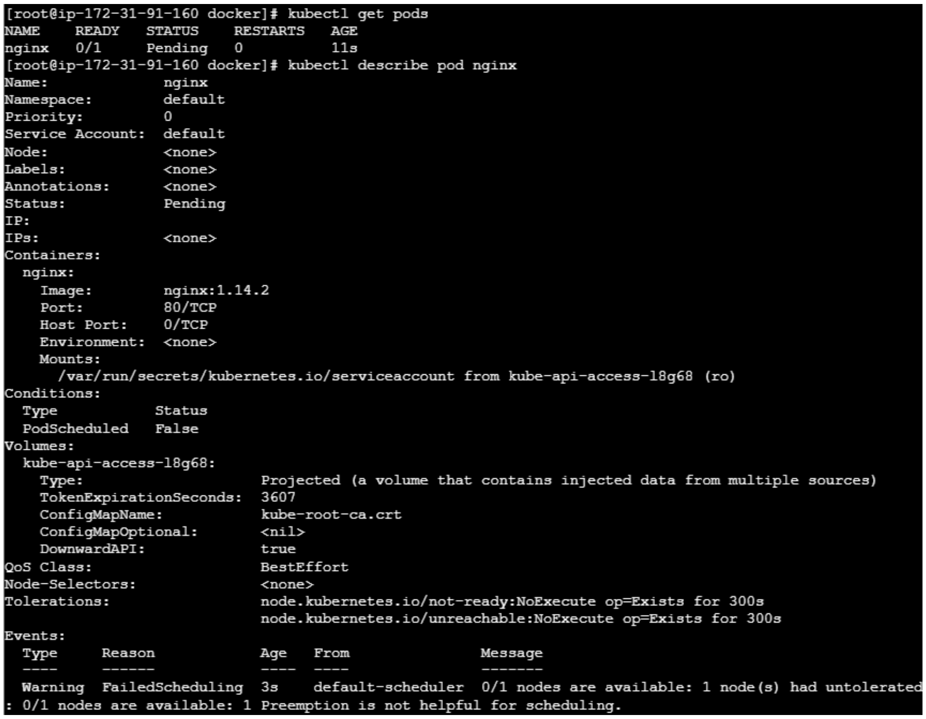
Deploying nginx server on the cluster:



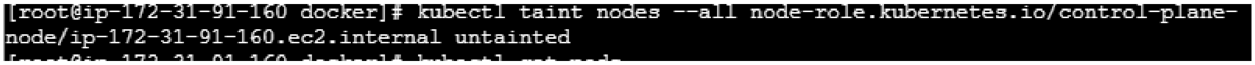
```
[root@ip-172-31-91-160 docker]# kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml
pod/nginx created
```



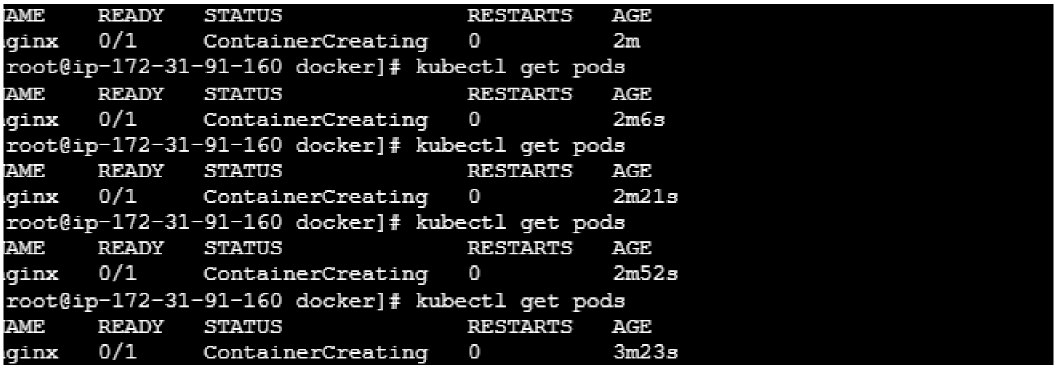
```
[root@ip-172-31-91-160 docker]# kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   0/1     Pending   0          11s
```



```
[root@ip-172-31-91-160 docker]# kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   0/1     Pending   0          11s
[root@ip-172-31-91-160 docker]# kubectl describe pod nginx
Name:             nginx
Namespace:        default
Priority:         0
Service Account:  default
Node:             <none>
Labels:           <none>
Annotations:      <none>
Status:           Pending
IP:
IPs:              <none>
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-l8g68 (ro)
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  kube-api-access-l8g68:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason            Age   From               Message
  ----     ------            ----  ----               -------
  Warning  FailedScheduling  3s    default-scheduler  0/1 nodes are available: 1 node(s) had untolerated
: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```



```
[root@ip-172-31-91-160 docker]# kubectl taint nodes --all node-role.kubernetes.io/control-plane-
node/ip-172-31-91-160.ec2.internal untainted
```



```
AME    READY   STATUS              RESTARTS   AGE
ginx   0/1     ContainerCreating   0          2m
root@ip-172-31-91-160 docker]# kubectl get pods
AME    READY   STATUS              RESTARTS   AGE
ginx   0/1     ContainerCreating   0          2m6s
root@ip-172-31-91-160 docker]# kubectl get pods
AME    READY   STATUS              RESTARTS   AGE
ginx   0/1     ContainerCreating   0          2m21s
root@ip-172-31-91-160 docker]# kubectl get pods
AME    READY   STATUS              RESTARTS   AGE
ginx   0/1     ContainerCreating   0          2m52s
root@ip-172-31-91-160 docker]# kubectl get pods
AME    READY   STATUS              RESTARTS   AGE
ginx   0/1     ContainerCreating   0          3m23s
```

Conclusion: Here, we successfully created an EC2 instance on AWS Linux, installed Docker and Kubernetes on the same. Then I tried to deployed nginx which initially showed status pending for the deployment however, it later became ContainerCreating, which finally has to be in running status which could not be achieved despite of waiting for several minutes. Hence, it couldn't get hosted on port 8081. However, rest of the procedure is completed.