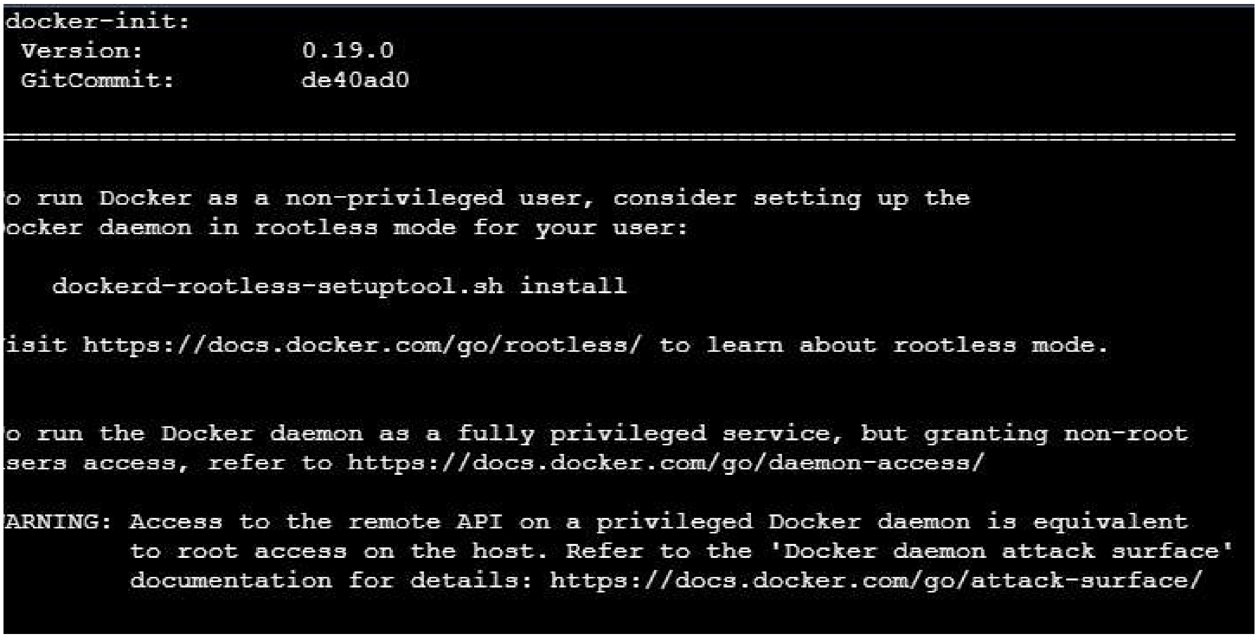
Kubernet kops create

Sudo su -

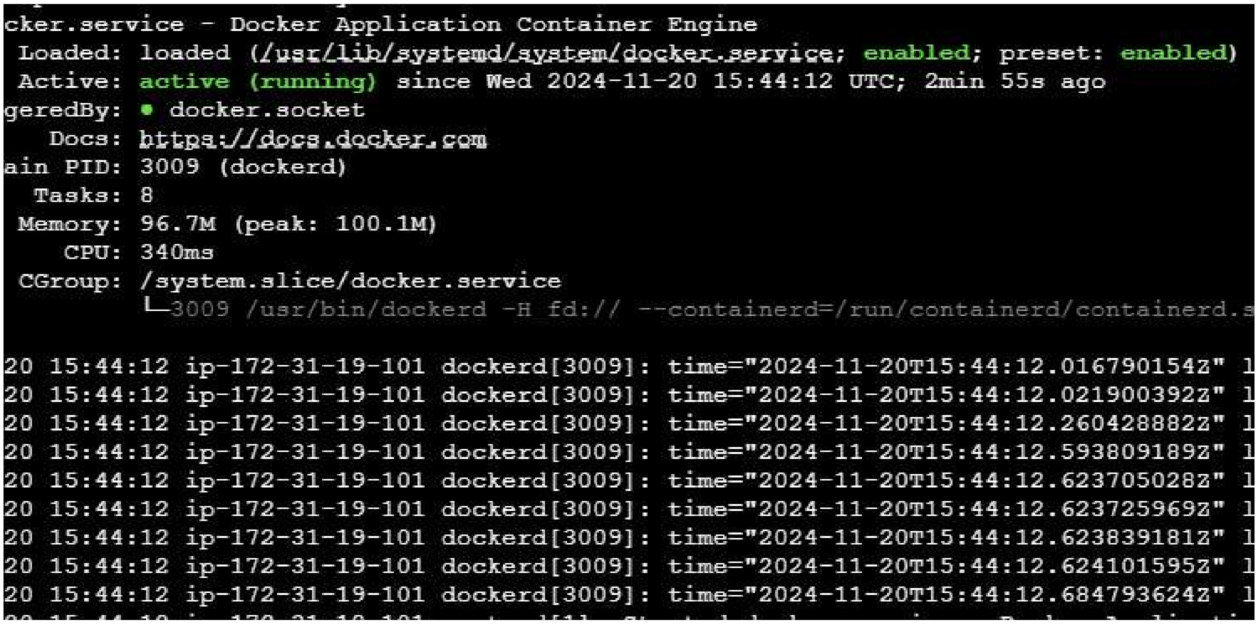
Apt update –y

curl -fsSL https://get.docker.com -o get-docker.sh

sh get-docker.sh



```
docker-init:
 Version:          0.19.0
 GitCommit:        de40ad0

================================================================================

o run Docker as a non-privileged user, consider setting up the
ocker daemon in rootless mode for your user:

   dockerd-rootless-setuptool.sh install

isit https://docs.docker.com/go/rootless/ to learn about rootless mode.


o run the Docker daemon as a fully privileged service, but granting non-root
sers access, refer to https://docs.docker.com/go/daemon-access/

ARNING: Access to the remote API on a privileged Docker daemon is equivalent
        to root access on the host. Refer to the 'Docker daemon attack surface'
        documentation for details: https://docs.docker.com/go/attack-surface/
```



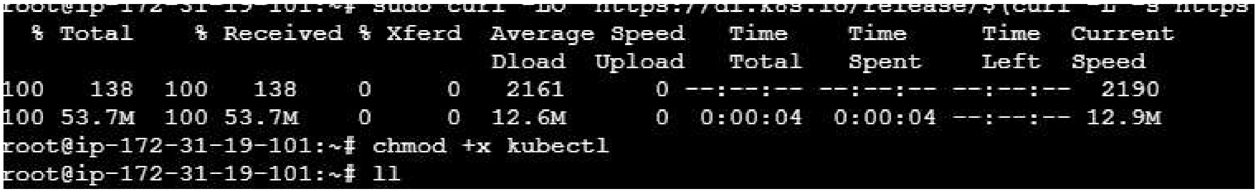
```
cker.service - Docker Application Container Engine
 Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
 Active: active (running) since Wed 2024-11-20 15:44:12 UTC; 2min 55s ago
geredBy: ● docker.socket
   Docs: https://docs.docker.com
ain PID: 3009 (dockerd)
  Tasks: 8
 Memory: 96.7M (peak: 100.1M)
    CPU: 340ms
 CGroup: /system.slice/docker.service
         └─3009 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.s

20 15:44:12 ip-172-31-19-101 dockerd[3009]: time="2024-11-20T15:44:12.016790154Z" l
20 15:44:12 ip-172-31-19-101 dockerd[3009]: time="2024-11-20T15:44:12.021900392Z" l
20 15:44:12 ip-172-31-19-101 dockerd[3009]: time="2024-11-20T15:44:12.260428882Z" l
20 15:44:12 ip-172-31-19-101 dockerd[3009]: time="2024-11-20T15:44:12.593809189Z" l
20 15:44:12 ip-172-31-19-101 dockerd[3009]: time="2024-11-20T15:44:12.623705028Z" l
20 15:44:12 ip-172-31-19-101 dockerd[3009]: time="2024-11-20T15:44:12.623725969Z" l
20 15:44:12 ip-172-31-19-101 dockerd[3009]: time="2024-11-20T15:44:12.623839181Z" l
20 15:44:12 ip-172-31-19-101 dockerd[3009]: time="2024-11-20T15:44:12.624101595Z" l
20 15:44:12 ip-172-31-19-101 dockerd[3009]: time="2024-11-20T15:44:12.684793624Z" l
```

Systemctl status docker

Setup kubectl:

sudo curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

Chmod +x kubectl

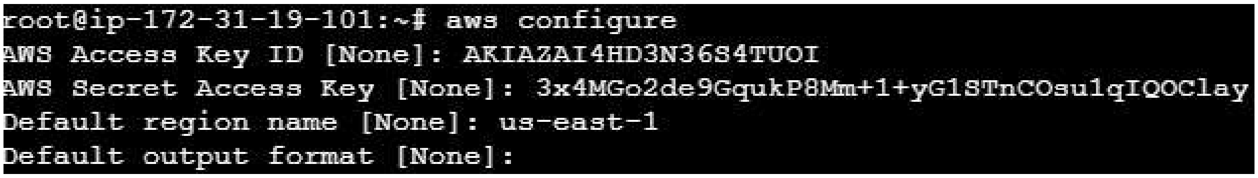


```
root@ip-172-31-19-101:~# sudo curl -LO "https://dl.k8s.io/release/$(curl -L -s https
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0   2161      0 --:--:-- --:--:-- --:--:--  2190
100 53.7M  100 53.7M    0     0  12.6M      0  0:00:04  0:00:04 --:--:-- 12.9M
root@ip-172-31-19-101:~# chmod +x kubectl
root@ip-172-31-19-101:~# ll
```

Aws configure

snap info aws-cli

snap install aws-cli --channel=v1/stable --classic



```
root@ip-172-31-19-101:~# aws configure
AWS Access Key ID [None]: AKIAZAI4HD3N36S4TUOI
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
```

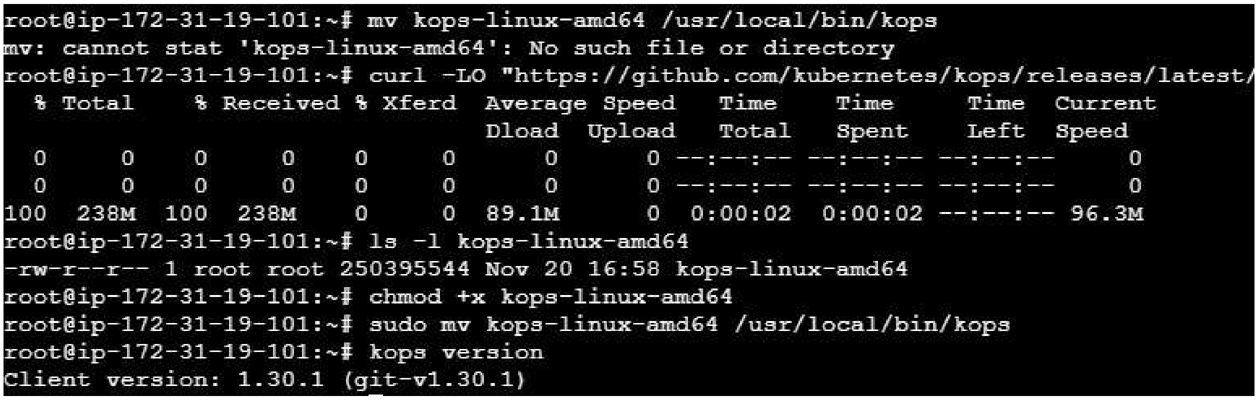
Install kops:

curl -Lo kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag_name | cut -d '"' -f 4)/kops-linux-amd64

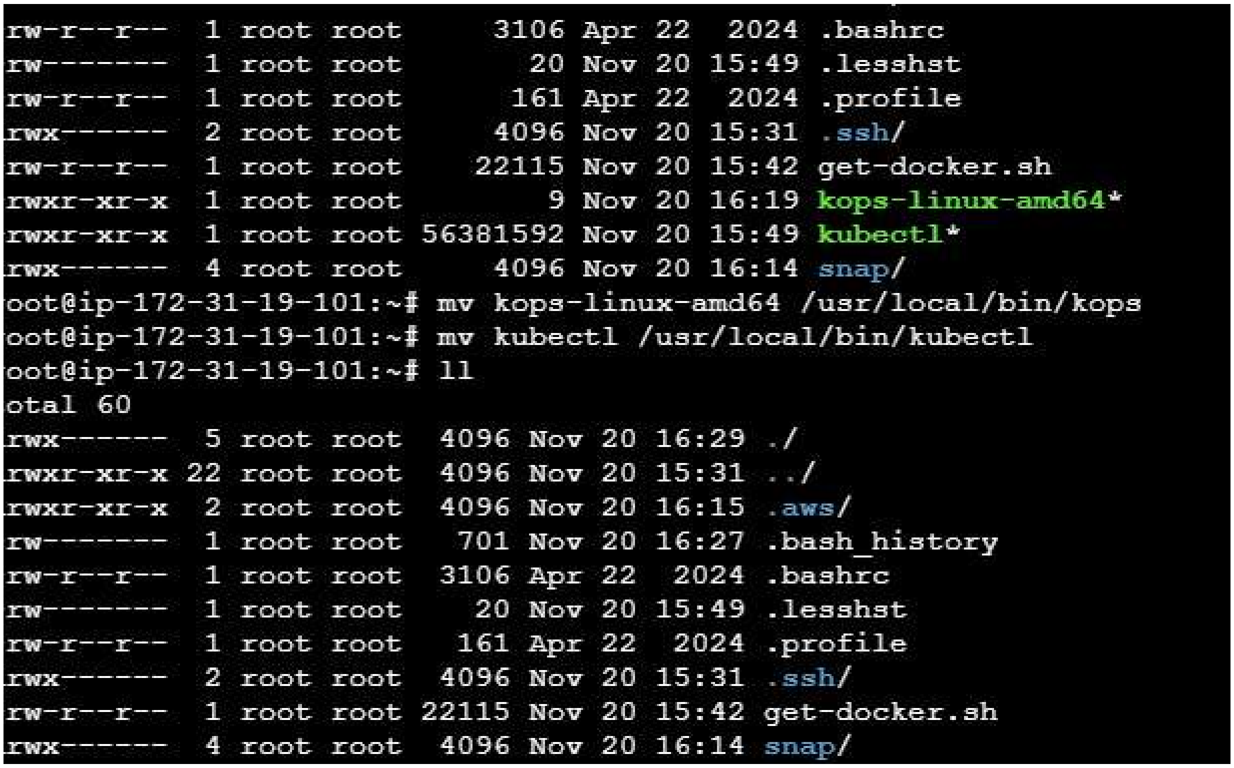
chmod +x ./kops

sudo mv ./kops /usr/local/bin/

mv kubectl /usr/local/bin/kubectl

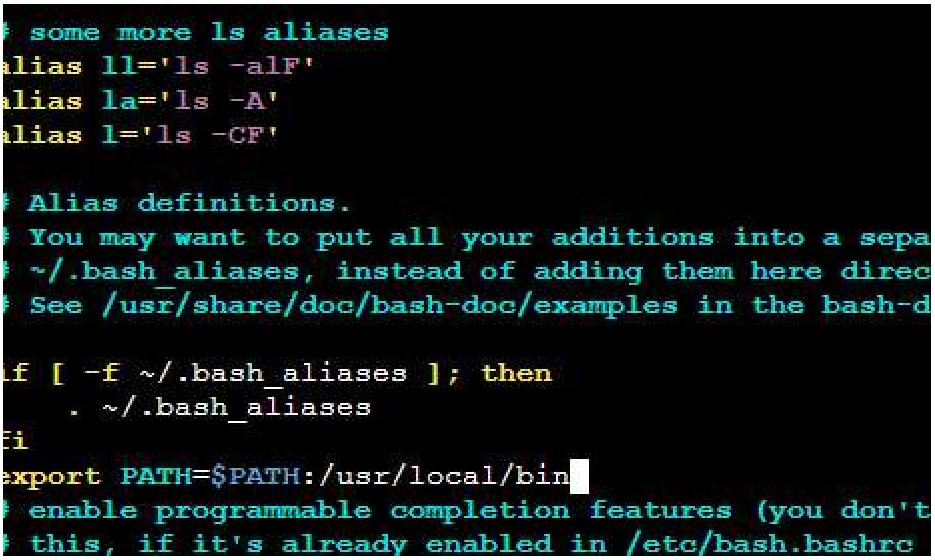


```
root@ip-172-31-19-101:~# mv kops-linux-amd64 /usr/local/bin/kops
mv: cannot stat 'kops-linux-amd64': No such file or directory
root@ip-172-31-19-101:~# curl -LO "https://github.com/kubernetes/kops/releases/latest/
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  238M  100  238M    0     0  89.1M      0  0:00:02  0:00:02 --:--:-- 96.3M
root@ip-172-31-19-101:~# ls -l kops-linux-amd64
-rw-r--r-- 1 root root 250395544 Nov 20 16:58 kops-linux-amd64
root@ip-172-31-19-101:~# chmod +x kops-linux-amd64
root@ip-172-31-19-101:~# sudo mv kops-linux-amd64 /usr/local/bin/kops
root@ip-172-31-19-101:~# kops version
Client version: 1.30.1 (git-v1.30.1)
```



```
rw-r--r--  1 root root     3106 Apr 22  2024 .bashrc
rw-------  1 root root       20 Nov 20 15:49 .lesshst
rw-r--r--  1 root root      161 Apr 22  2024 .profile
rwx------  2 root root     4096 Nov 20 15:31 .ssh/
rw-r--r--  1 root root    22115 Nov 20 15:42 get-docker.sh
rwxr-xr-x  1 root root        9 Nov 20 16:19 kops-linux-amd64*
rwxr-xr-x  1 root root 56381592 Nov 20 15:49 kubectl*
rwx------  4 root root     4096 Nov 20 16:14 snap/
oot@ip-172-31-19-101:~# mv kops-linux-amd64 /usr/local/bin/kops
oot@ip-172-31-19-101:~# mv kubectl /usr/local/bin/kubectl
oot@ip-172-31-19-101:~# ll
otal 60
rwx------  5 root root  4096 Nov 20 16:29 ./
rwxr-xr-x 22 root root  4096 Nov 20 15:31 ../
rwxr-xr-x  2 root root  4096 Nov 20 16:15 .aws/
rw-------  1 root root   701 Nov 20 16:27 .bash_history
rw-r--r--  1 root root  3106 Apr 22  2024 .bashrc
rw-------  1 root root    20 Nov 20 15:49 .lesshst
rw-r--r--  1 root root   161 Apr 22  2024 .profile
rwx------  2 root root  4096 Nov 20 15:31 .ssh/
rw-r--r--  1 root root 22115 Nov 20 15:42 get-docker.sh
rwx------  4 root root  4096 Nov 20 16:14 snap/
```

vi .bashrc



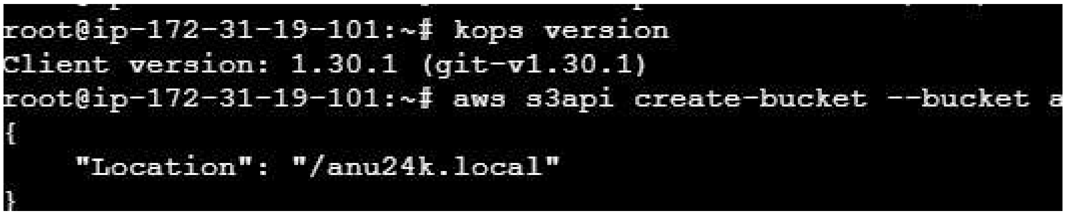
```
# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Alias definitions.
# You may want to put all your additions into a sepa
# ~/.bash_aliases, instead of adding them here direc
# See /usr/share/doc/bash-doc/examples in the bash-d

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
export PATH=$PATH:/usr/local/bin
# enable programmable completion features (you don't
# this, if it's already enabled in /etc/bash.bashrc
```

source .bashrc

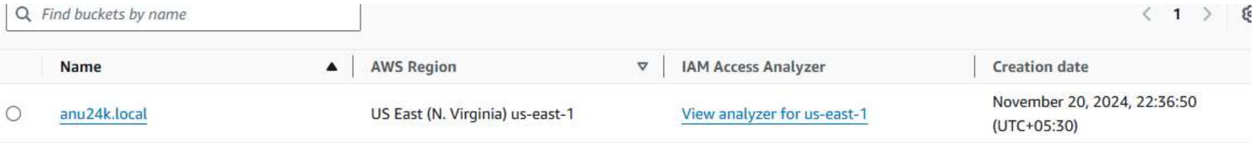
kubectl version --client --output=yaml

Create the s3 bucket by using below command

aws s3api create-bucket --bucket anu24k.local --region us-east-1



```
root@ip-172-31-19-101:~# kops version
Client version: 1.30.1 (git-v1.30.1)
root@ip-172-31-19-101:~# aws s3api create-bucket --bucket a
{
    "Location": "/anu24k.local"
}
```



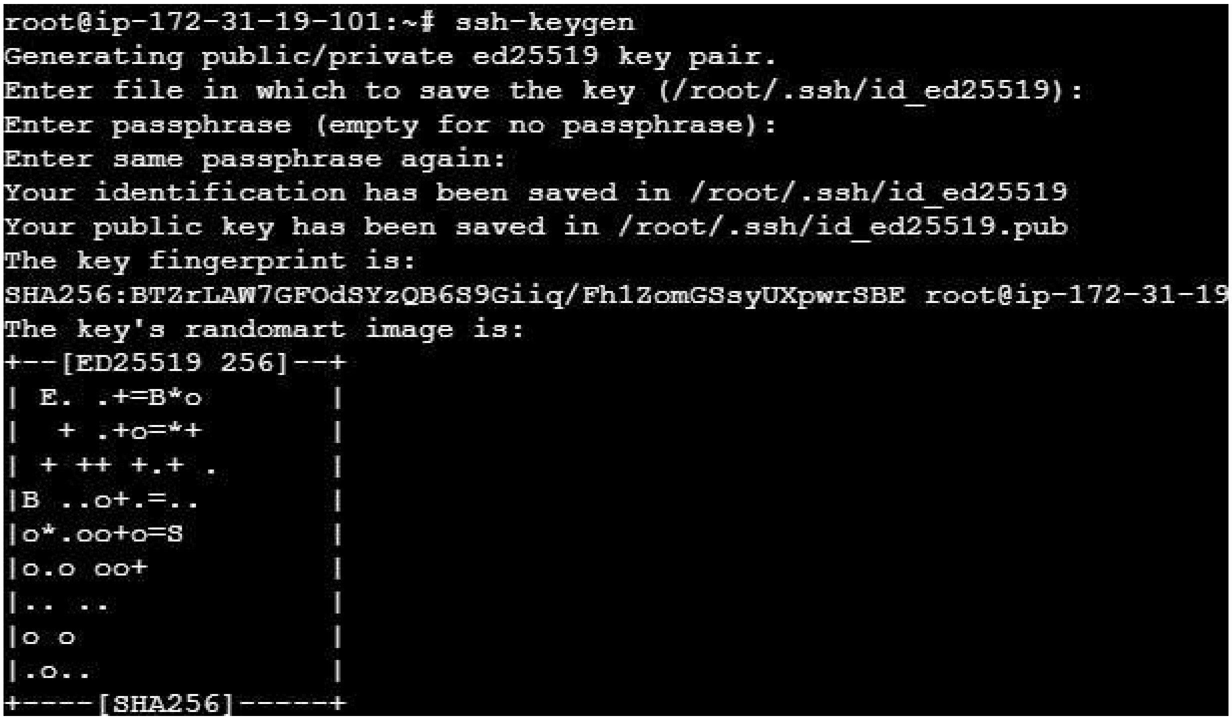
Name	AWS Region	IAM Access Analyzer	Creation date
○ anu24k.local	US East (N. Virginia) us-east-1	View analyzer for us-east-1	November 20, 2024, 22:36:50 (UTC+05:30)

Enable the versioning

aws s3api put-bucket-versioning --bucket anusha24k.local --region us-east-1 --versioning-configuration Status=Enabled

export kops_state_store=s3://anusha24k.local

ssh-keygen

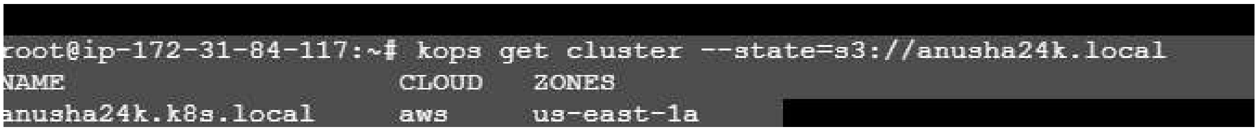


```
root@ip-172-31-19-101:~# ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:BTZrLAW7GFOdSYzQB6S9Giiq/Fh1ZomGSsyUXpwrSBE root@ip-172-31-19
The key's randomart image is:
+--[ED25519 256]--+
| E.  .+=B*o      |
|  + .+o=*+       |
| + ++ +.+ .      |
|B ..o+.=..       |
|o*.oo+o=S        |
|o.o oo+          |
|.. ..            |
|o o              |
|.o..             |
+----[SHA256]-----+
```

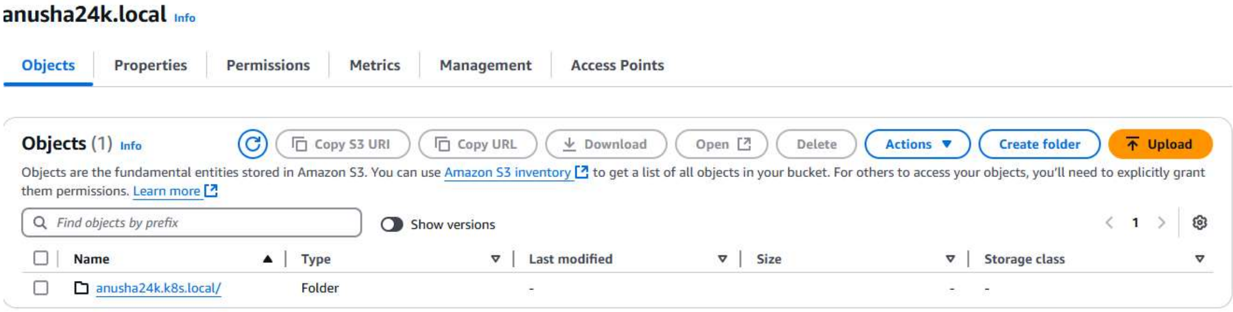
Create the cluster:

kops create cluster --name anusha24k.k8s.local --state=s3://anusha24k.local --zones us-east-1a --master-size t2.medium --node-size t2.micro

kops get cluster --state=s3://anusha24k.local



```
root@ip-172-31-84-117:~# kops get cluster --state=s3://anusha24k.local
NAME                    CLOUD   ZONES
anusha24k.k8s.local     aws     us-east-1a
```



anusha24k.local Info

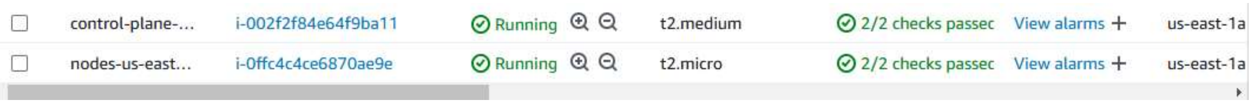
Objects (1) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type	Last modified	Size	Storage class
□	anusha24k.k8s.local/	Folder	-	-	-

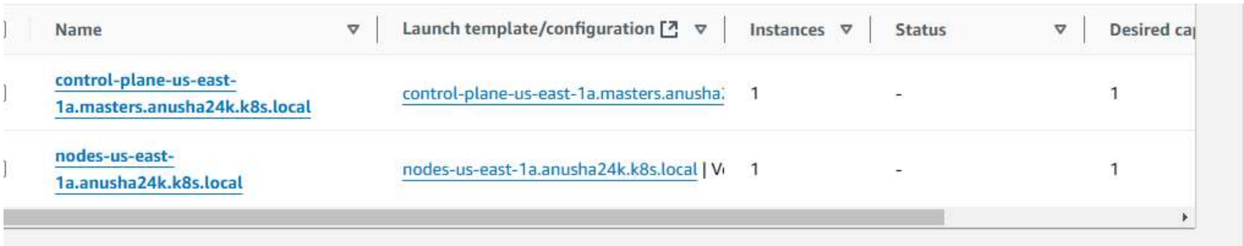
kops update cluster --name anusha24k.k8s.local --yes --admin --state=s3://anusha24k.local

Kubectl get node



□	control-plane-…	i-002f2f84e64f9ba11	✓ Running	t2.medium	✓ 2/2 checks passec	View alarms +	us-east-1a
□	nodes-us-east…	i-0ffc4c4ce6870ae9e	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1a

autoscaling

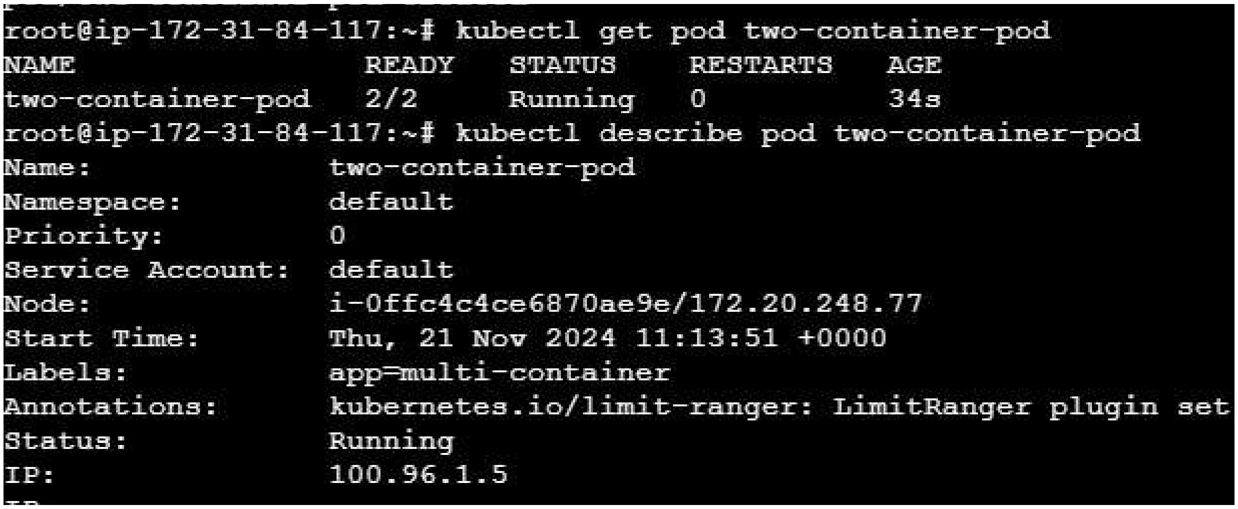


Name	Launch template/configuration	Instances	Status	Desired ca
control-plane-us-east-1a.masters.anusha24k.k8s.local	control-plane-us-east-1a.masters.anusha:	1	-	1
nodes-us-east-1a.anusha24k.k8s.local	nodes-us-east-1a.anusha24k.k8s.local | V	1	-	1

In single pod how to create two diffrent containers

vi multi-container-pod.yaml

kubectl apply -f multi-container-pod.yaml



```
root@ip-172-31-84-117:~# kubectl get pod two-container-pod
NAME                READY   STATUS    RESTARTS   AGE
two-container-pod   2/2     Running   0          34s
root@ip-172-31-84-117:~# kubectl describe pod two-container-pod
Name:             two-container-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             i-0ffc4c4ce6870ae9e/172.20.248.77
Start Time:       Thu, 21 Nov 2024 11:13:51 +0000
Labels:           app=multi-container
Annotations:      kubernetes.io/limit-ranger: LimitRanger plugin set
Status:           Running
IP:               100.96.1.5
```

Kubectl get pod