KOPS

What are kops

1. Kops is an open-source tool designed to perform the deployment, management, and maintenance of Kubernetes clusters
2. Kops can be created and manage the autoscaling and load balancers automatically

Steps to create kops

Step 1:

1. Create the server with ubuntu
    - Select the ubuntu
    - Select the t2 micro
    - Increase the storage to 25
2. Change to the root user
3. Update the server apt update -y

Step 2:

1. Install the docker packages by using commands

```
sudo apt install curl wget apt-transport-https -y
sudo curl -fsSL https://get.docker.com -o get-docker.sh
chmod 777 get-docker.sh
sh get-docker.sh
```

2. Start the docker by command systemctl start docker

3. Check the status of docker by command systemctl status docker

Step 3:

1. Install the kubectl packages

```
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
```

2. It can provide the executable permissions to the kubectl file

Step 4:

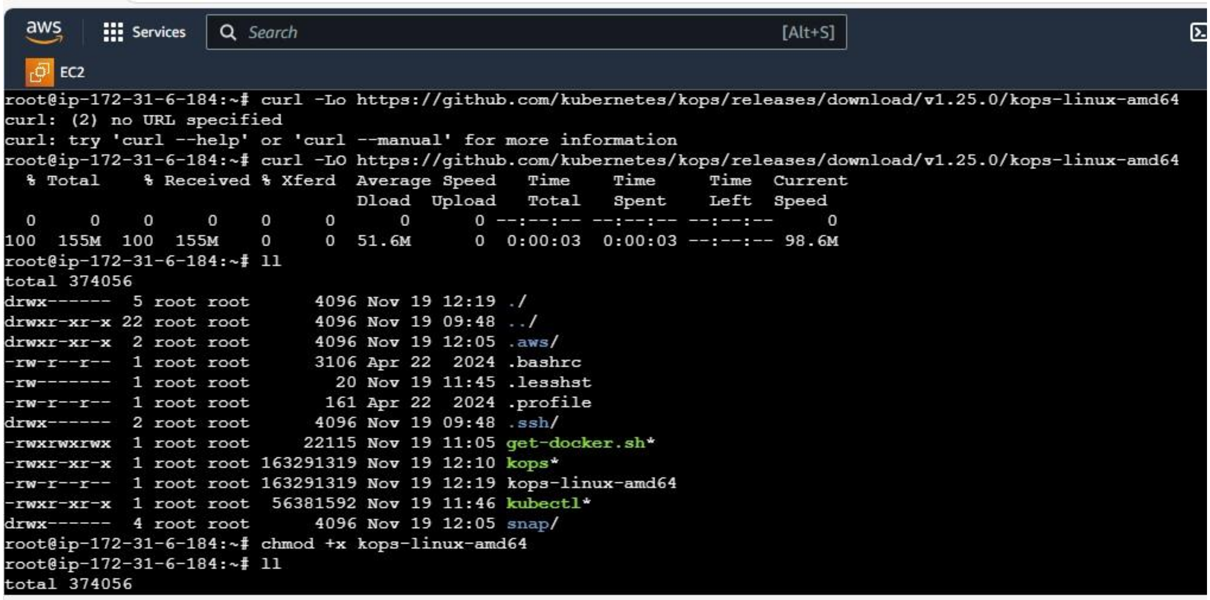
1. Give aws configure
2. Give snap info aws-cli
3. Give snap install aws-cli --channel=v1/stable --classic to install the aws cli
4. Create user in iam and give access permissions to the user
5. Provide access key and secret key in cli

Step 5:

1. To install the kops by command is

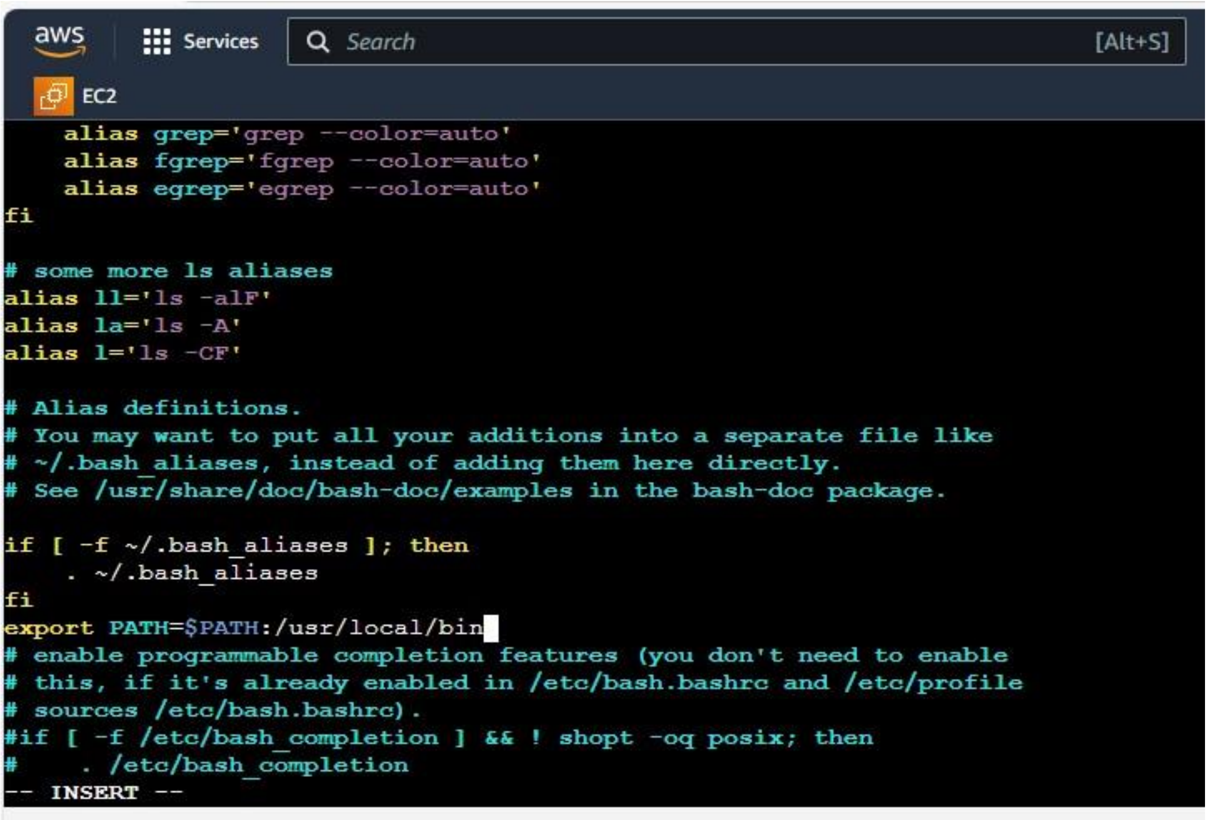
```
curl -LO https://github.com/kubernetes/kops/releases/download/v1.25.0/kops-linux-amd64
chmod +x kops-linux-amd64
mv kops-linux-amd64 /usr/local/bin/kops
kops version
mv kubectl /usr/local/bin/kubectl
```

2. After performing all the commands in terminal as shown in below figure



```
root@ip-172-31-6-184:~# curl -Lo https://github.com/kubernetes/kops/releases/download/v1.25.0/kops-linux-amd64
curl: (2) no URL specified
curl: try 'curl --help' or 'curl --manual' for more information
root@ip-172-31-6-184:~# curl -LO https://github.com/kubernetes/kops/releases/download/v1.25.0/kops-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  155M  100  155M    0     0  51.6M      0  0:00:03  0:00:03 --:--:-- 98.6M
root@ip-172-31-6-184:~# ll
total 374056
drwx------  5 root root      4096 Nov 19 12:19 ./
drwxr-xr-x 22 root root      4096 Nov 19 09:48 ../
drwxr-xr-x  2 root root      4096 Nov 19 12:05 .aws/
-rw-r--r--  1 root root      3106 Apr 22  2024 .bashrc
-rw-------  1 root root        20 Nov 19 11:45 .lesshst
-rw-r--r--  1 root root       161 Apr 22  2024 .profile
drwx------  2 root root      4096 Nov 19 09:48 .ssh/
-rwxrwxrwx  1 root root     22115 Nov 19 11:05 get-docker.sh*
-rwxr-xr-x  1 root root 163291319 Nov 19 12:10 kops*
-rw-r--r--  1 root root 163291319 Nov 19 12:19 kops-linux-amd64
-rwxr-xr-x  1 root root  56381592 Nov 19 11:46 kubectl*
drwx------  4 root root      4096 Nov 19 12:05 snap/
root@ip-172-31-6-184:~# chmod +x kops-linux-amd64
root@ip-172-31-6-184:~# ll
total 374056
```

3. Give ll -a to list the files
4. Go inside of the file vi .bashrc to add the content in file as shown in below figure



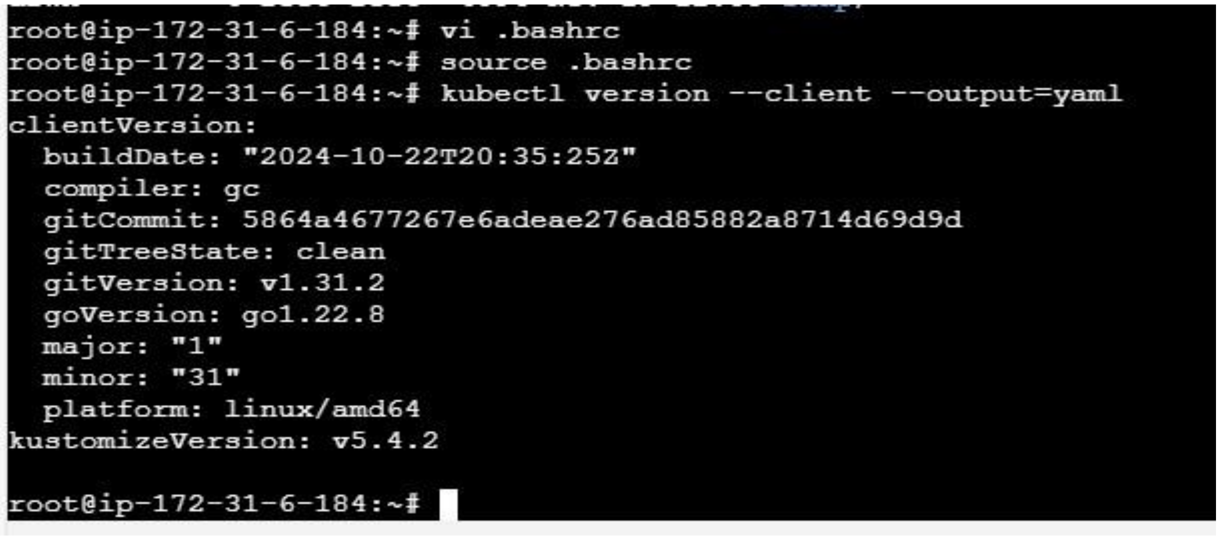
```
    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# 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 separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
export PATH=$PATH:/usr/local/bin
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#    . /etc/bash_completion
-- INSERT --
```

5. Save and quit from the editor
6. Type source .bashrc command
7. Type kubectl version --client --output=yaml it will show the version in yaml file format as shown in below figure



```
root@ip-172-31-6-184:~# vi .bashrc
root@ip-172-31-6-184:~# source .bashrc
root@ip-172-31-6-184:~# kubectl version --client --output=yaml
clientVersion:
  buildDate: "2024-10-22T20:35:25Z"
  compiler: gc
  gitCommit: 5864a4677267e6adeae276ad85882a8714d69d9d
  gitTreeState: clean
  gitVersion: v1.31.2
  goVersion: go1.22.8
  major: "1"
  minor: "31"
  platform: linux/amd64
kustomizeVersion: v5.4.2

root@ip-172-31-6-184:~#
```

Step 6:

1. Create the s3 bucket from cli and store the entire cluster information in the created bucket
2. Create the bucket from cli by the command

```
aws s3api create-bucket --bucket mani420 --region us-east-1
```

3. To enable the versioning from cli by through command

```
aws s3api put-bucket-versioning --bucket mani420 --region
us-east-1 --versioning-configuration Status=Enabled
```

4. Exporting the bucket in kops by the command

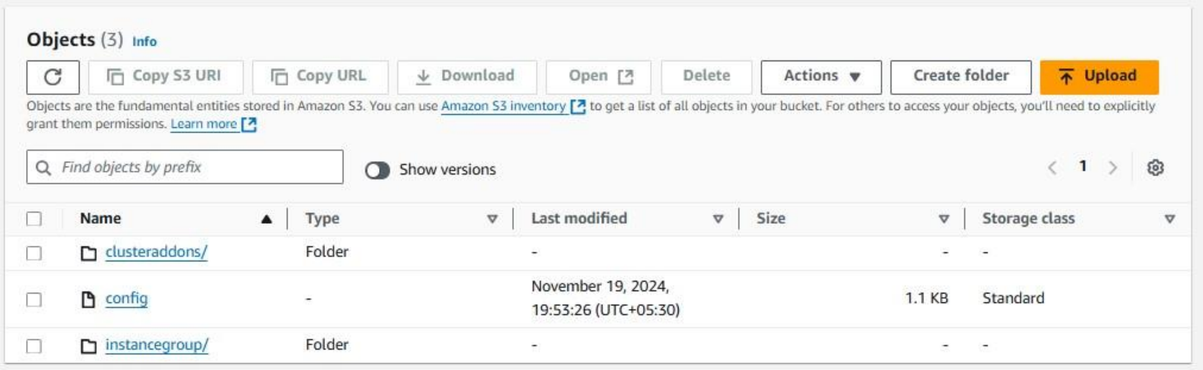
```
export kops_state_store=s3://mani420
ssh-keygen – to generate public and private keys
```

Step 7:

1. We need to create the cluster
2. We can create the cluster by the command

```
kops create cluster --name raj.k8s.local --
state=s3://mani420 --zones us-east-1a --master-size
t2.medium --node-size t2.micro
```

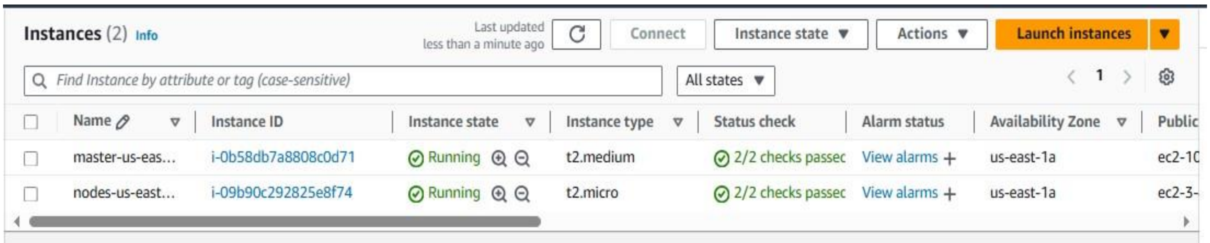
3. It can create the s3 bucket, autoscaling, load balancer automatically in N.virginia as shown in below figure



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

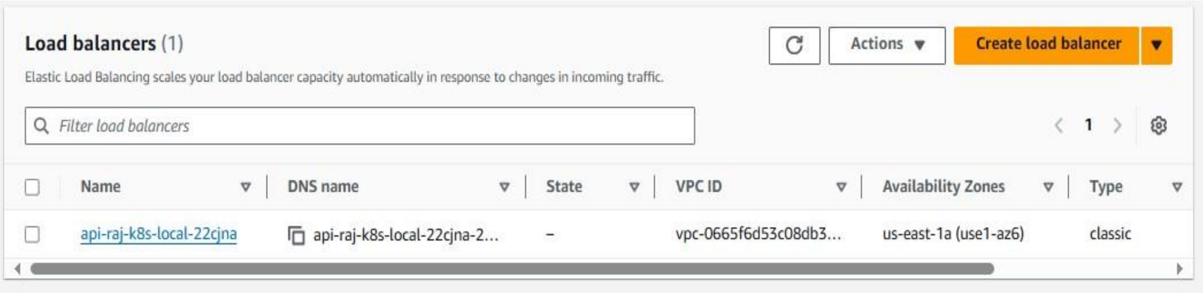
□	Name	Type	Last modified	Size	Storage class
□	clusteraddons/	Folder	-	-	-
□	config	-	November 19, 2024, 19:53:26 (UTC+05:30)	1.1 KB	Standard
□	instancegroup/	Folder	-	-	-

4. instances as shown in below figure



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
□	master-us-eas...	i-0b58db7a8808c0d71	Running	t2.medium	2/2 checks passec	View alarms +	us-east-1a	ec2-10
□	nodes-us-east...	i-09b90c292825e8f74	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	ec2-3-

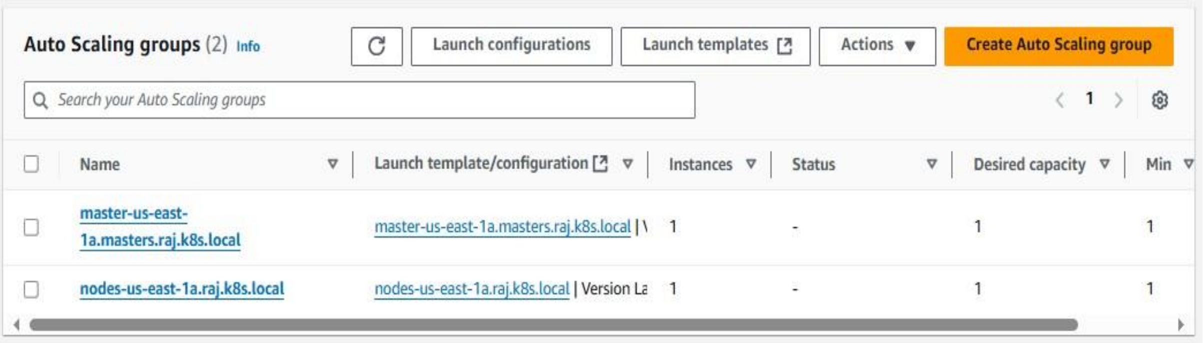
5. Load balancer created



Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

□	Name	DNS name	State	VPC ID	Availability Zones	Type
□	api-raj-k8s-local-22cjna	api-raj-k8s-local-22cjna-2...	-	vpc-0665f6d53c08db3...	us-east-1a (use1-az6)	classic

6. Autoscaling created


Auto Scaling groups (2) Info
Launch configurations
Launch templates
Actions ▼
Create Auto Scaling group
Search your Auto Scaling groups
1

□	Name	Launch template/configuration	Instances	Status	Desired capacity	Min
□	master-us-east-1a.masters.raj.k8s.local	master-us-east-1a.masters.raj.k8s.local | \	1	-	1	1
□	nodes-us-east-1a.raj.k8s.local	nodes-us-east-1a.raj.k8s.local | Version La	1	-	1	1

7. To see the cluster information by the command

```
kops get clusters --state=s3://mani420
```

8. To edit the cluster by the command

```
kops edit cluster raj.k8s.local --state=s3://mani420
```

9. If the instances are not created then update the cluster then the instances, load balancers, autoscaling will be created by the command

```
kops update cluster --name raj.k8s.local --yes --admin --
state=s3://mani420
```