…..................FOR KUBERNETES.................

KUBECTL:

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

* chmod +x kubectl
* mv kubectl /usr/local/bin/kubectl

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
* Sudō mv kops /usr/local/bin/kops

KBS\_CLUSTER:

1. export NAME=dptcluster.k8s.local
2. export KOPS\_STATE\_STORE=s3://dpt-k8s-state-store

KEYS:

ssh-keygen -t rsa

CREATING CLUSTER:

1. kops create cluster --zones=us-east-1a ${NAME}

OPEN A FILE IN ROOT LOCATION:

NAMED :

Vi deploy.yaml

In Deploy.yaml:

# Deploy ReplicaSet Controller

apiVersion: apps/v1

kind: Deployment

metadata:

name: myapp

spec:

replicas: 3

selector:

matchLabels:

app: myapp

template:

metadata:

name: myapp

labels:

app: myapp

spec:

containers:

- name: myapp-container

image:nan2917/Kunal: latest

2.kubectl apply –f deploy.yaml

3.kubectl get pods/nodes

4.create a replica.yaml

Vi replia.yaml

# Deploy ReplicaSet Controller

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: myapp

spec:

replicas: 3

selector:

matchLabels:

app: myapp

template:

metadata:

name: myapp

labels:

app: myapp

spec:

containers:

- name: myapp-container

image:nan2917/kunal:latest

Again apply it

Kubectl apply –f replica.yaml