INT334

CA-1

Submitted By
Nijo Luca
11704688
39

Question 1)

Answer:a)Make three instances on AWS One Master and Two Slaves.

☐	Master	i-00449c2625957fe0e	t2.micro	us-east-2c	running	2/2 checks ...	None	
☐	Master	i-044504b982f37de7e	t2.micro	us-east-2b	terminated		None	
☐	Slave 1	i-053a3fde7450a01b1	t2.micro	us-east-2c	running	2/2 checks ...	None	
☑	Slave 2	i-055dfab64135e92d0	t2.micro	us-east-2c	running	2/2 checks ...	None	

On the master Install kubernetes

Command for installing kubernetes

sudo apt-get update && sudo apt-get install -y apt-transport-https curl

```
Unpacking libcurl4:amd64 (7.58.0-2ubuntu3.10) over (7.58.0-2ubuntu3.9)
Setting up apt-transport-https (1.6.12ubuntu0.1) ...
Setting up libcurl4:amd64 (7.58.0-2ubuntu3.10) ...
Setting up curl (7.58.0-2ubuntu3.10) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
```

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add –

```
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
root@ip-172-31-36-223:~# curl -s https://packages.cloud.google.com/apt/doc/apt-k
ey.gpg | sudo apt-key add -
OK
root@ip-172-31-36-223:~#
```

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

```
root@ip-172-31-36-223:~# cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes
.list
> deb https://apt.kubernetes.io/ kubernetes-xenial main
> EOF
deb https://apt.kubernetes.io/ kubernetes-xenial main
root@ip-172-31-36-223:~#
```

```
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
```

```
Setting up kubelet (1.18.8-00) ...
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /l
ib/systemd/system/kubelet.service.
Setting up kubectl (1.18.8-00) ...
Setting up kubeadm (1.18.8-00) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
root@ip-172-31-36-223:~#
```

```
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
root@ip-172-31-36-223:~#
```

Do the same on the other Two slaves

1st Slave

```
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
root@ip-172-31-36-169:~# sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
```

2nd Slave

```
root@ip-172-31-47-11:~# sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
root@ip-172-31-47-11:~#
```

On Master

Command to initialize kubeadm

kubeadm init --apiserver-advertise-address=<ip-address-of-master> --pod-network-cidr=192.168.0.0/16 --ignore-preflight-errors=all

```
kubeadm join 172.31.36.223:6443 --token 259pnv.qvsuvytchevoux92 \
    --discovery-token-ca-cert-hash sha256:d1006fc71433e1dbbe6090362d9ecff3ba7e7c
7e0dcf204cec87eeb6e48dbba0
root@ip-172-31-36-223:~#
```

Copy paste on the slave to join the cluster

```
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/ku
belet/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 apiserver 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.

root@ip-172-31-47-11:~#
root@ip-172-31-47-11:~#
```

```
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver 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.

root@ip-172-31-36-169:~#
```

Add this command on the Master

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Command to install network plugin

curl https://docs.projectcalico.org/manifests/calico.yaml -O

kubectl apply -f calico.yaml

curl https://docs.projectcalico.org/manifests/calico-typha.yaml -o calico.yaml

kubectl apply -f calico.yaml

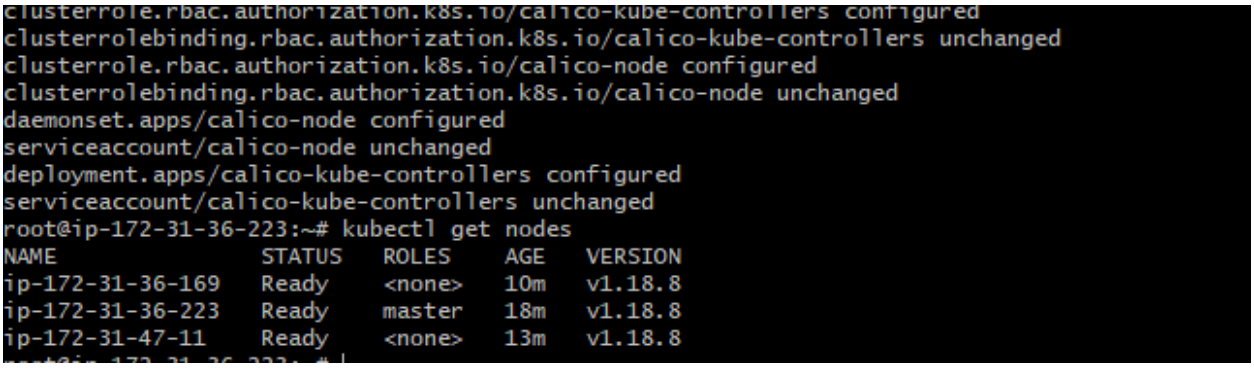
curl https://docs.projectcalico.org/manifests/calico-etcd.yaml -o calico.yaml

kubectl apply -f calico.yaml

Hence Installation is complete

Answer b)

kubect get nodes



Answer c)Create a yaml file

Named nginx1.yaml and add this contents to it. This will create 2 pod with two images

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: rss-site
  labels:
    app: web
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: front-end
          image: nginx
          ports:
            - containerPort: 80
        - name: rss-reader
          image: nickchase/rss-php-nginx:v1
          ports:
            - containerPort: 88
```

Run the following command to c

create the deployment

Kubectl create -f nginx1.yaml

(Note: Here I have used an online environment for creating deployment)

```
controlplane $ kubectl create -f nginx1.yaml
deployment.apps/rss-site created
```

```
rss-site-cd4cf7dcd-7sqsf            2/2     Running   0          68s
rss-site-cd4cf7dcd-vdzqf            2/2     Running   0          68s
```

2)Create a service and display the content of nginx file on command prompt and also display it on web.

Answer)

Create a service name nginx1

```
controlplane $ kubectl create service nodeport nginx1 --tcp=80:80
service/nginx1 created
```

```
controlplane $ kubectl get svc nginx1
NAME     TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
nginx1   NodePort   10.101.3.56    <none>        80:31853/TCP   68s
controlplane $
```

Use Cat Command to view the content of nginx file we have created

Cat nginx1.yaml

```
ubuntu@ip-172-31-33-45:~$ cat nginx1.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: rss-site
  labels:
    app: web
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: front-end
          image: nginx
          ports:
            - containerPort: 80
        - name: rss-reader
          image: nickchase/rss-php-nginx:v1
          ports:
            - containerPort: 88
```

To verify the service on browser add in the format

<Ip address of master or slave>: <port id>

This will give the output as follows

(Note: Here I have displayed this using My GCP Kubernetes Engine)