Kubernetes

Assignment – 1

Task to be performed

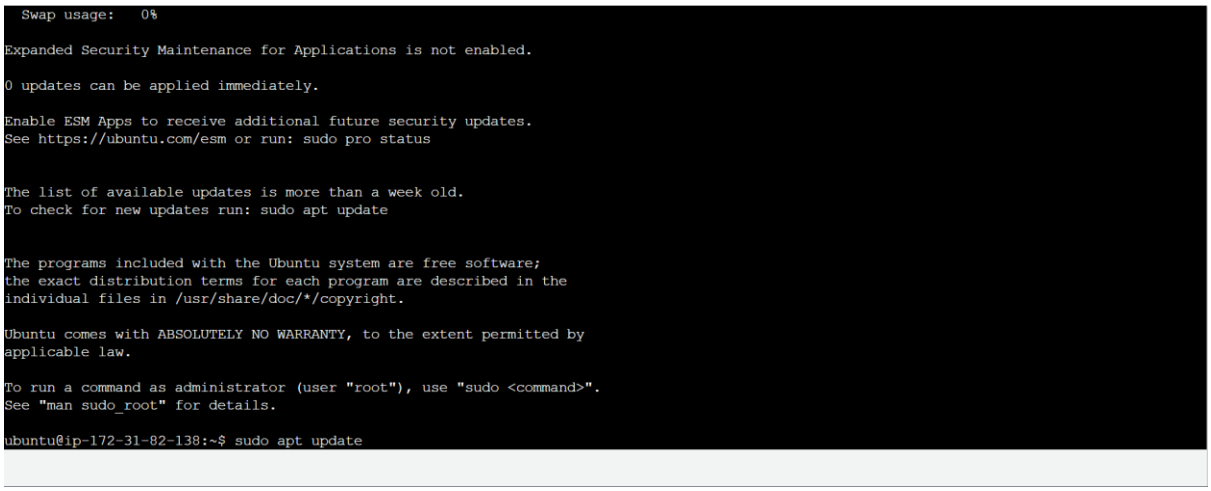
1. Deploy a Kubernetes cluster for 3 nodes
2. Create a NGINX deployment of 3 replicas

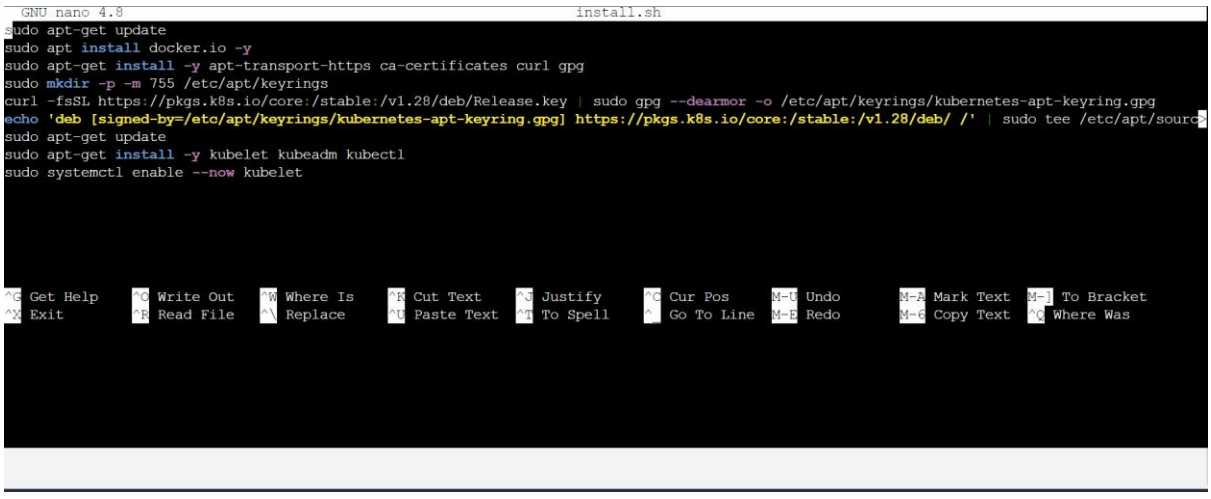
Steps and Commands

- Create 3 instance names them as master, slave1 and slave2
- Connect to all the 3 instance
- Update all the machine by running the command **sudo apt update**
- Next create a file in all the machines to install kubernetes by running the command **sudo nano (provide the name for creating file.sh)**
- Provide the necessary command to install kubernetes in all the machine
    a) **sudo apt-get update**
    b) **sudo apt install docker.io -y**
    c) **sudo apt-get install -y apt-transport-https ca-certificates curl gpg**
    d) **sudo mkdir -p -m 755 /etc/apt/keyrings**
    e) **curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg**
    f) **echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list**
    g) **sudo apt-get update**
    h) **sudo apt-get install -y kubelet kubeadm kubectl**
    i) **sudo systemctl enable --now kubelet**
- To run the commands which was provided in file by running the command **sudo bash (provide the name for created file.sh)**
- Navigate to master instance
- Next In order to generate a token and to make the master machines as administrator by running the command **sudo kubeadm init --apiserver-advertise-address=(privateip of master)**
- Navigate to slave1 and slave2 instance
- Next copy the token generated in mater and paste in both the slave1 and slave2 machine
- Navigate to master instance
- And next run the following in master machine command
    a) **mkdir -p $HOME/.kube**

b) **sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config**
c) **sudo chown $(id -u):$(id -g) $HOME/.kube/config**
d) **curl https://raw.githubusercontent.com/projectcalico/calico/v3.27.2/manifests/calico.yaml -O**
e) **kubectl apply -f calico.yaml**

- Next to check the list of node by running the command **kubectl get nodes**
- Create a file for providing script for NGINX deployment by running the command **sudo nano nginx_deployment.yaml**
- Provide the necessary script for NGINX deployment
- To run the nginx deployment by running the command **kubectl apply –f nginx_deployment.yaml**
- To check the deployment by running the command **kubectl get deploy**
- To see the full details of deployment by running the command **kubectl get deploy –o wide, kubectl describe deploy nginx-deployment**





```
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [620 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [45.7 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main Translation-en [16.3 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [1420 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [25.0 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [16.3 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [880 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:30 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [431 kB]
Get:31 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [2730 kB]
Get:32 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [382 kB]
Get:33 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [952 kB]
Get:34 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [200 kB]
Get:35 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [19.2 kB]
Get:36 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [23.9 kB]
Get:37 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5904 B]
Get:38 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [548 B]
Fetched 30.8 MB in 5s (5840 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
35 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-82-138:~$ sudo nano install.sh
ubuntu@ip-172-31-82-138:~$ bash install .sh
```

```
Unpacking kubernetes-cni (1.2.0-2.1) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.7.3.3-2_amd64.deb ...
Unpacking socat (1.7.3.3-2) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.28.8-1.1_amd64.deb ...
Unpacking kubelet (1.28.8-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.28.8-1.1_amd64.deb ...
Unpacking kubectl (1.28.8-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.28.8-1.1_amd64.deb ...
Unpacking kubeadm (1.28.8-1.1) ...
Setting up conntrack (1:1.4.5-2) ...
Setting up kubectl (1.28.8-1.1) ...
Setting up ebtables (2.0.11-3build1) ...
Setting up socat (1.7.3.3-2) ...
Setting up cri-tools (1.28.0-1.1) ...
Setting up kubernetes-cni (1.2.0-2.1) ...
Setting up kubelet (1.28.8-1.1) ...
Setting up kubeadm (1.28.8-1.1) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@ip-172-31-82-138:~$ sudo kubeadm init --apiserver-advertise-address=172.31.82.138
I0411 12:16:33.331944    5350 version.go:256] remote version is much newer: v1.29.3; falling back to: stable-1.28
[init] Using Kubernetes version: v1.28.8
```

```
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 172.31.82.138:6443 --token m06hdn.79rprlrax6e9frht \
        --discovery-token-ca-cert-hash sha256:bbcbf6a64f7f06a0feld4ecf21df8c85cc6c4e3cd43ad0fc5ea4a0e60b2a7fe7
ubuntu@ip-172-31-82-138:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-82-138:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-82-138:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-82-138:~$ curl https://raw.githubusercontent.com/projectcalico/calico/v3.27.2/manifests/calico.yaml -O
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  246k  100  246k    0     0  2518k      0 --:--:-- --:--:-- --:--:-- 2518k
```

```
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-87-240:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
```

i-0e18244b00e6f765c (slave-1)

PublicIPs: 52.90.176.168 PrivateIPs: 172.31.87.240

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefer

```
 GNU nano 4.8                                  install.sh
sudo apt-get update
sudo apt install docker.io -y
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sourc
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo systemctl enable --now kubelet




                                [ Read 9 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo  M-A Mark Text  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo  M-6 Copy Text  ^Q Where Was
```

i-0e18244b00e6f765c (slave-1)

PublicIPs: 52.90.176.168 PrivateIPs: 172.31.87.240

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
Get:32 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [382 kB]
Get:33 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [952 kB]
Get:34 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [200 kB]
Get:35 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [19.2 kB]
Get:36 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [23.9 kB]
Get:37 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5904 B]
Get:38 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [548 B]
Fetched 30.8 MB in 5s (5992 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
35 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-87-240:~$ sudo nano install.sh
ubuntu@ip-172-31-87-240:~$ bash install.sh
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
```

i-0e18244b00e6f765c (slave-1)

PublicIPs: 52.90.176.168 PrivateIPs: 172.31.87.240

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefer

```
Unpacking kubelet (1.28.8-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.28.8-1.1_amd64.deb ...
Unpacking kubectl (1.28.8-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.28.8-1.1_amd64.deb ...
Unpacking kubeadm (1.28.8-1.1) ...
Setting up conntrack (1:1.4.5-2) ...
Setting up kubectl (1.28.8-1.1) ...
Setting up ebtables (2.0.11-3build1) ...
Setting up socat (1.7.3.3-2) ...
Setting up cri-tools (1.28.0-1.1) ...
Setting up kubernetes-cni (1.2.0-2.1) ...
Setting up kubelet (1.28.8-1.1) ...
Setting up kubeadm (1.28.8-1.1) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@ip-172-31-87-240:~$ sudo kubeadm join 172.31.82.138:6443 --token m06hdn.79rprlrax6e9frht \
>         --discovery-token-ca-cert-hash sha256:bbcbf6a64f7f06a0fe1d4ecf21df8c85cc6c4e3cd43ad0fc5ea4a0e60b2a7fe7
```

i-0e18244b00e6f765c (slave-1)

PublicIPs: 52.90.176.168 PrivateIPs: 172.31.87.240


```
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-89-235:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
```

i-0ab781b76e7e62b8d (slave-2)

PublicIPs: 3.85.128.175 PrivateIPs: 172.31.89.235


```
  GNU nano 4.8                                  install.sh
sudo apt-get update
sudo apt install docker.io -y
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sourc
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo systemctl enable --now kubelet




                              [ Read 9 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos   M-U Undo      M-A Mark Text  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo      M-6 Copy Text  ^Q Where Was
```

i-0ab781b76e7e62b8d (slave-2)

PublicIPs: 3.85.128.175 PrivateIPs: 172.31.89.235

```
Get:32 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [382 kB]
Get:33 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [952 kB]
Get:34 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [200 kB]
Get:35 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [19.2 kB]
Get:36 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [23.9 kB]
Get:37 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5904 B]
Get:38 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [548 B]
Fetched 30.8 MB in 5s (6003 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
35 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-89-235:~$ sudo nano install.sh
ubuntu@ip-172-31-89-235:~$ bash install.sh
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
```

i-0ab781b76e7e62b8d (slave-2)

PublicIPs: 3.85.128.175 PrivateIPs: 172.31.89.235


```
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.28.8-1.1_amd64.deb ...
Unpacking kubeadm (1.28.8-1.1) ...
Setting up conntrack (1:1.4.5-2) ...
Setting up kubectl (1.28.8-1.1) ...
Setting up ebtables (2.0.11-3build1) ...
Setting up socat (1.7.3.3-2) ...
Setting up cri-tools (1.28.0-1.1) ...
Setting up kubernetes-cni (1.2.0-2.1) ...
Setting up kubelet (1.28.8-1.1) ...
Setting up kubeadm (1.28.8-1.1) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@ip-172-31-89-235:~$ sudo kubeadm join 172.31.82.138:6443 --token m06hdn.79rpr1rax6e9frht \
>         --discovery-token-ca-cert-hash sha256:bbcbf6a64f7f06a0fe1d4ecf21df8c85cc6c4e3cd43ad0fc5ea4a0e60b2a7fe7
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
```

i-0ab781b76e7e62b8d (slave-2)

PublicIPs: 3.85.128.175 PrivateIPs: 172.31.89.235


```
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrole.rbac.authorization.k8s.io/calico-cni-plugin created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
ubuntu@ip-172-31-82-138:~$ kubectl get nodes
NAME              STATUS     ROLES           AGE     VERSION
ip-172-31-82-138   NotReady   control-plane   5m17s   v1.28.8
ip-172-31-87-240   NotReady   <none>          2m19s   v1.28.8
ip-172-31-89-235   NotReady   <none>          2m20s   v1.28.8
ubuntu@ip-172-31-82-138:~$ kubectl get nodes
NAME              STATUS   ROLES           AGE     VERSION
ip-172-31-82-138   Ready    control-plane   5m22s   v1.28.8
ip-172-31-87-240   Ready    <none>          2m24s   v1.28.8
ip-172-31-89-235   Ready    <none>          2m25s   v1.28.8
ubuntu@ip-172-31-82-138:~$  sudo nano install.sh
ubuntu@ip-172-31-82-138:~$ history
    1  sudo apt update
    2  sudo nano install.sh
    3  bash install .sh
```

```
  GNU nano 4.8                          nginx_deployment.yaml                          Modified
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo  M-A Mark Text
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo  M-6 Copy Text
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ sudo nano nginx_deployment.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f nginx_deployment.yaml
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ sudo nano nginx_deployment.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f nginx_deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           35s
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ sudo nano nginx_deployment.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f nginx_deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           35s
ubuntu@ip-172-31-82-138:~$ kubectl get deploy -o wide
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138


```
ubuntu@ip-172-31-82-138:~$ sudo nano nginx_deployment.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f nginx_deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           35s
ubuntu@ip-172-31-82-138:~$ kubectl get deploy -o wide
NAME               READY   UP-TO-DATE   AVAILABLE   AGE    CONTAINERS   IMAGES   SELECTOR
nginx-deployment   3/3     3            3           2m6s   nginx        nginx    app=nginx
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

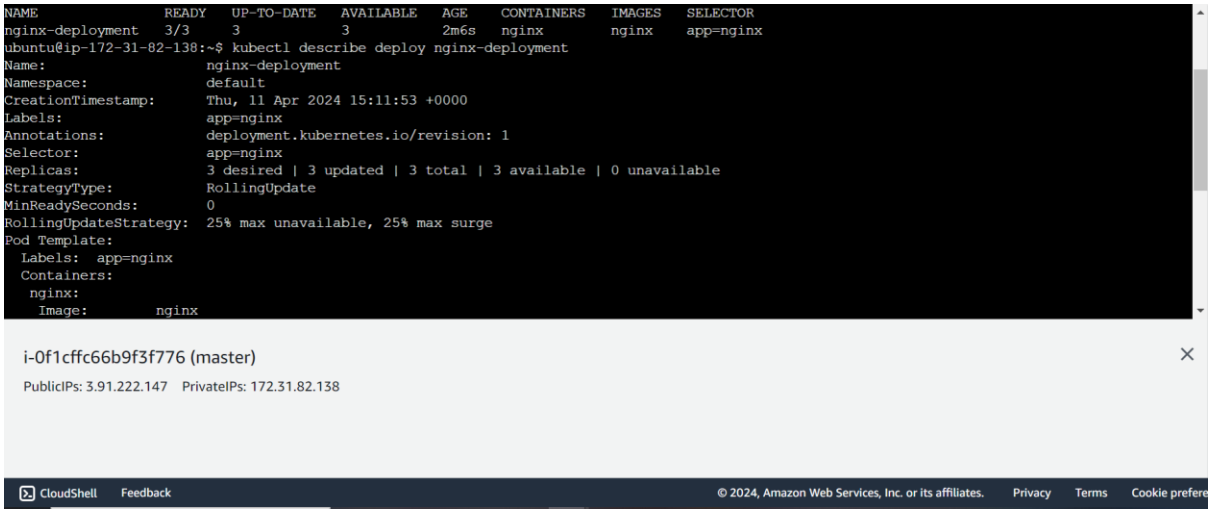
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ sudo nano nginx_deployment.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f nginx_deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           35s
ubuntu@ip-172-31-82-138:~$ kubectl get deploy -o wide
NAME               READY   UP-TO-DATE   AVAILABLE   AGE    CONTAINERS   IMAGES   SELECTOR
nginx-deployment   3/3     3            3           2m6s   nginx        nginx    app=nginx
ubuntu@ip-172-31-82-138:~$ kubectl describe deploy nginx-deployment
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere


NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR
nginx-deployment 3/3 3 3 2m6s nginx nginx app=nginx
ubuntu@ip-172-31-82-138:~$ kubectl describe deploy nginx-deployment
Name: nginx-deployment
Namespace: default
CreationTimestamp: Thu, 11 Apr 2024 15:11:53 +0000
Labels: app=nginx
Annotations: deployment.kubernetes.io/revision: 1
Selector: app=nginx
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
Labels: app=nginx
Containers:
nginx:
Image: nginx
i-0f1cffc66b9f3f776 (master)
PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie prefere

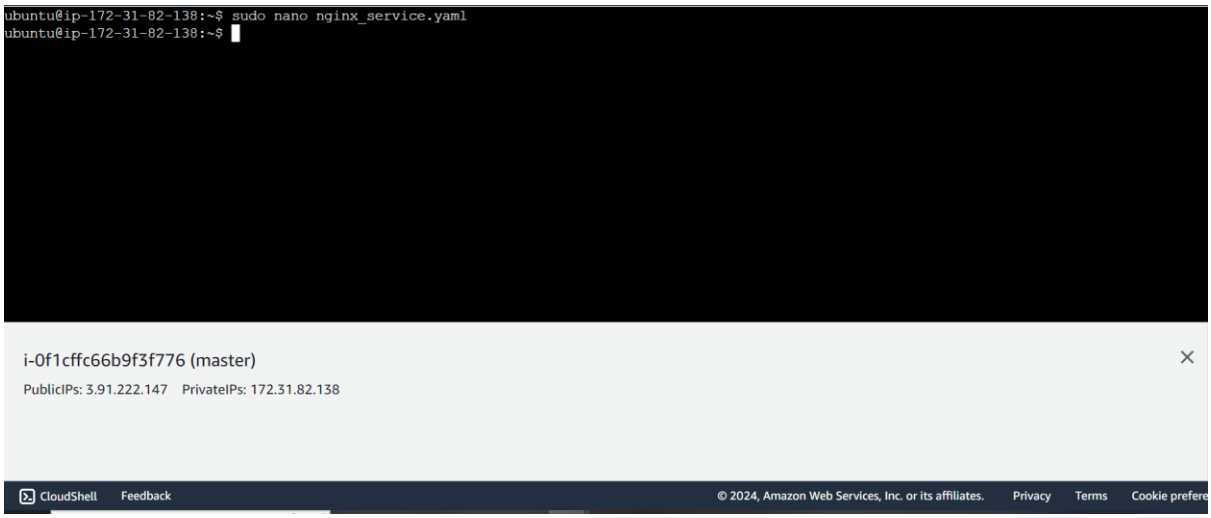
Assignment – 2

Task to be performed

1. Use the previous deployment
2. Create a service of type Nodeport for NGINX deployment
3. Check the Nodeport service on a browser to verify

Steps and Commands

- Create a file for providing script for creating NGINX service by running the command **sudo nano nginx_service.yaml**
- Provide the necessary script for NGINX service
- To run the nginx service by running the command **kubectl apply –f nginx_service.yaml**
- To check the service status by running the command **kubectl get svc**
- To see the Nginx page on browser put the (**public IP address:30008)** any machine master, slave1, slave2 of instance in the Brower the Nginx page will be reflected.



```
  GNU nano 4.8                    nginx_service.yaml                          Modified
apiVersion: v1
kind: Service
metadata:
  name: my-nginx-deployment

spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:
    app: nginx

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo  M-A Mark Text
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo  M-6 Copy Text
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138


```
ubuntu@ip-172-31-82-138:~$ sudo nano nginx_service.yaml
ubuntu@ip-172-31-82-138:~$ kubectl create -f nginx_service.yaml
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ sudo nano nginx_service.yaml
ubuntu@ip-172-31-82-138:~$ kubectl create -f nginx_service.yaml
service/my-nginx-deployment created
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

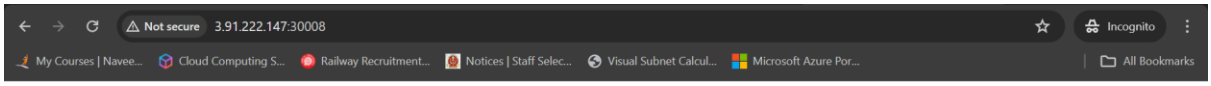
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ sudo nano nginx_service.yaml
ubuntu@ip-172-31-82-138:~$ kubectl create -f nginx_service.yaml
service/my-nginx-deployment created
ubuntu@ip-172-31-82-138:~$ kubectl get svc
NAME                 TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes           ClusterIP   10.96.0.1      <none>        443/TCP          3h57m
my-nginx-deployment  NodePort    10.98.213.98   <none>        80:30008/TCP     114s
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

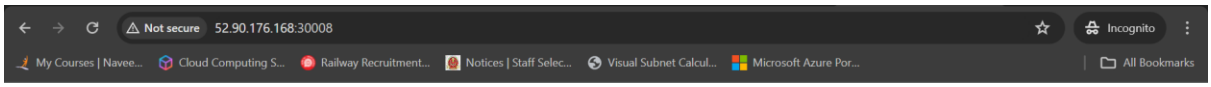


Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

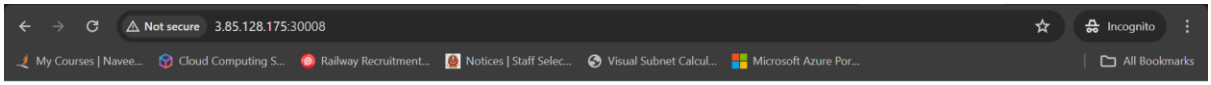


Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

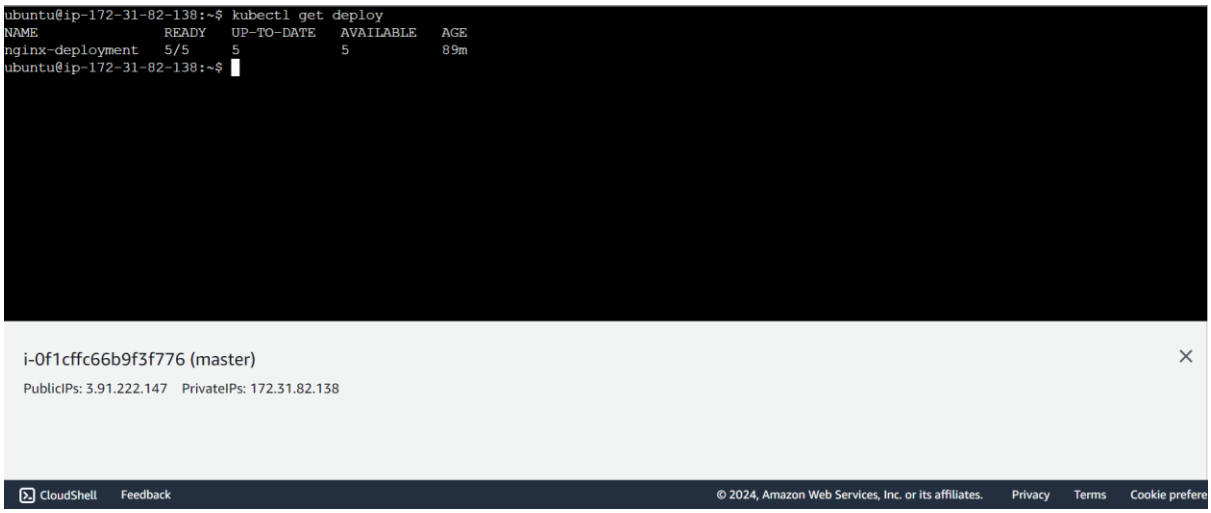
Assignment – 3

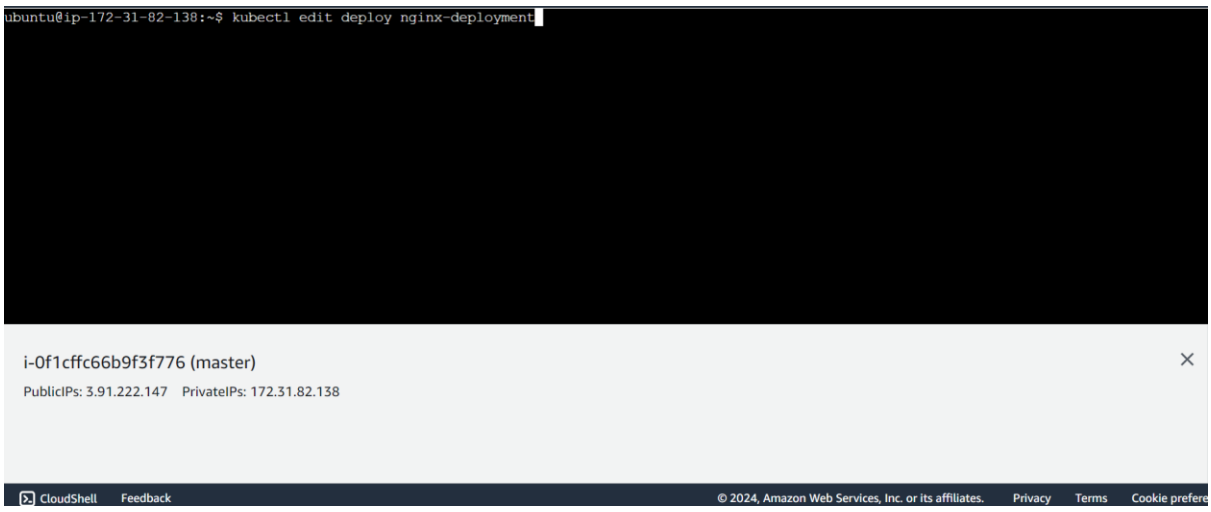
Task to be performed

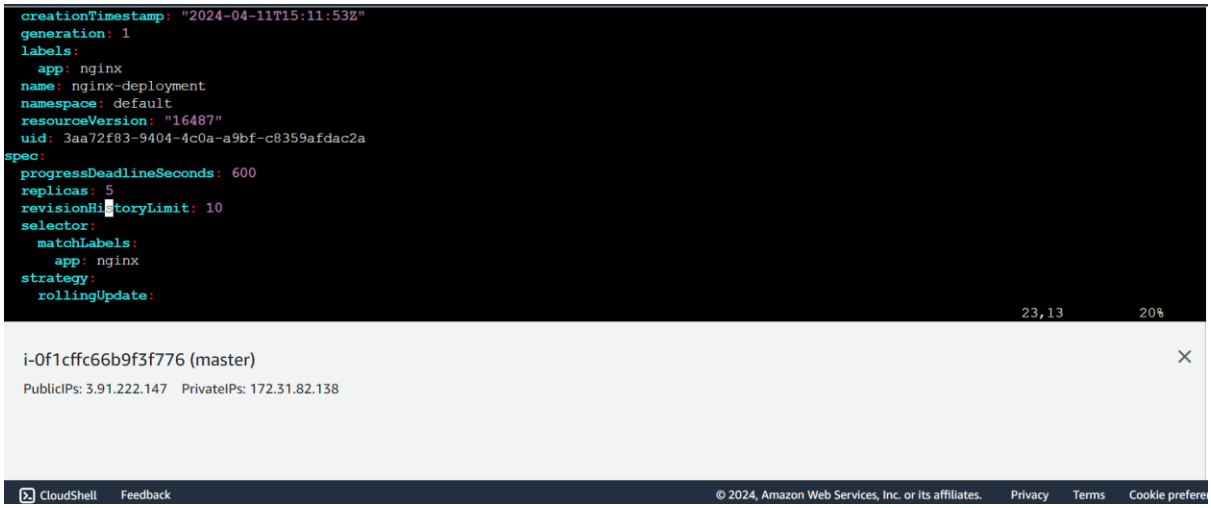
1. Use the previous deployment
2. Change the replicas to 5 for the deployment

Steps and Commands

- To open the nginx-deployment by running the command kubectl edit deploy nginx-deployment
- Edit the necessary script in replicas provide 5
- Save and exit from nginx-deployment
- To check replicas status by running the command kubectl get pods






creationTimestamp: "2024-04-11T15:11:53Z"
generation: 1
labels:
app: nginx
name: nginx-deployment
namespace: default
resourceVersion: "16487"
uid: 3aa72f83-9404-4c0a-a9bf-c8359afdac2a
spec:
progressDeadlineSeconds: 600
replicas: 5
revisionHistoryLimit: 10
selector:
matchLabels:
app: nginx
strategy:
rollingUpdate:
23,13 20%
i-0f1cffc66b9f3f776 (master)
PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

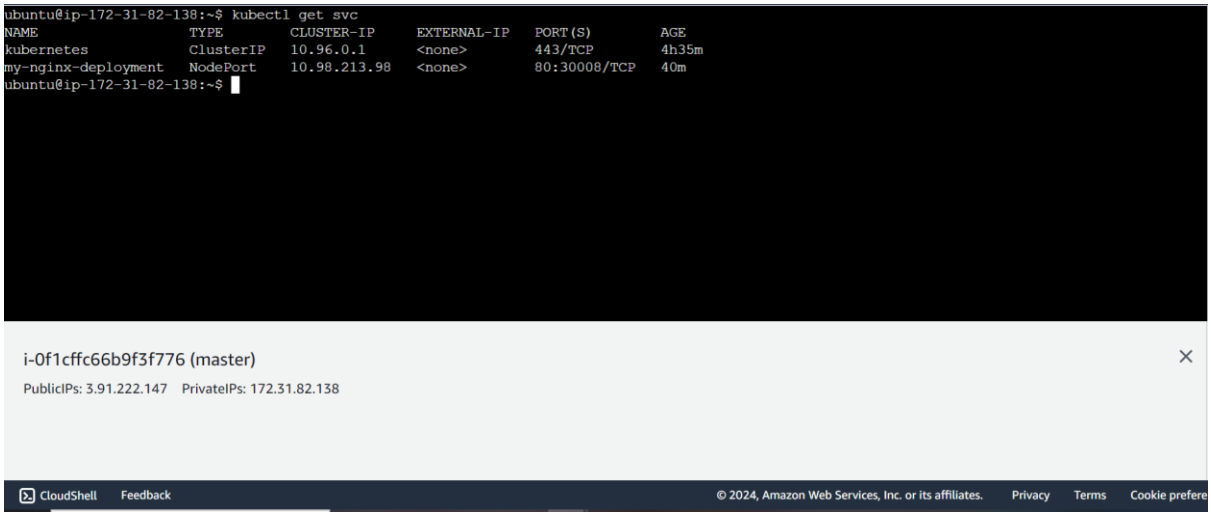
Assignment – 4

Task to be performed

1. Use the previous deployment
2. Change the service type to clusterIP

Steps and Commands

- To open svc my-nginx-deployment by running the command **kubectl edit svc my-nginx-deployment**
- Provide the necessary script by providing type as ClusterIP
- Save and exit from the svc my-nginx-deployment
- To check the Type status by running the command **kubectl get svc**
- To check the clusterIP work in machine my running the command **curl clusterIP**



```
ubuntu@ip-172-31-82-138:~$ kubectl get svc
NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes           ClusterIP   10.96.0.1       <none>        443/TCP          4h35m
my-nginx-deployment  NodePort    10.98.213.98    <none>        80:30008/TCP     40m
ubuntu@ip-172-31-82-138:~$ kubectl edit svc my-nginx-deployment
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138


```
  - 10.98.213.98
  externalTrafficPolicy: Cluster
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - nodePort: 30008
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
                                                                31,7          Bot
```

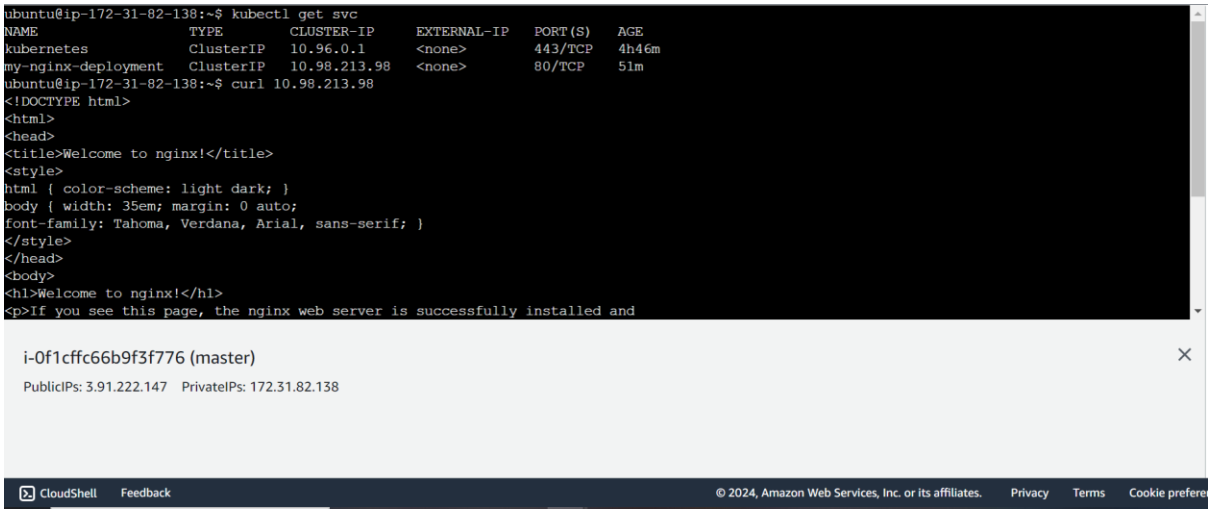
i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138


```
ubuntu@ip-172-31-82-138:~$ kubectl get svc
NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes           ClusterIP   10.96.0.1       <none>        443/TCP   4h46m
my-nginx-deployment  ClusterIP   10.98.213.98    <none>        80/TCP    51m
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138


ubuntu@ip-172-31-82-138:~$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 4h46m
my-nginx-deployment ClusterIP 10.98.213.98 <none> 80/TCP 51m
ubuntu@ip-172-31-82-138:~$ curl 10.98.213.98
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
i-0f1cffc66b9f3f776 (master)
PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie prefere

Assignment – 5

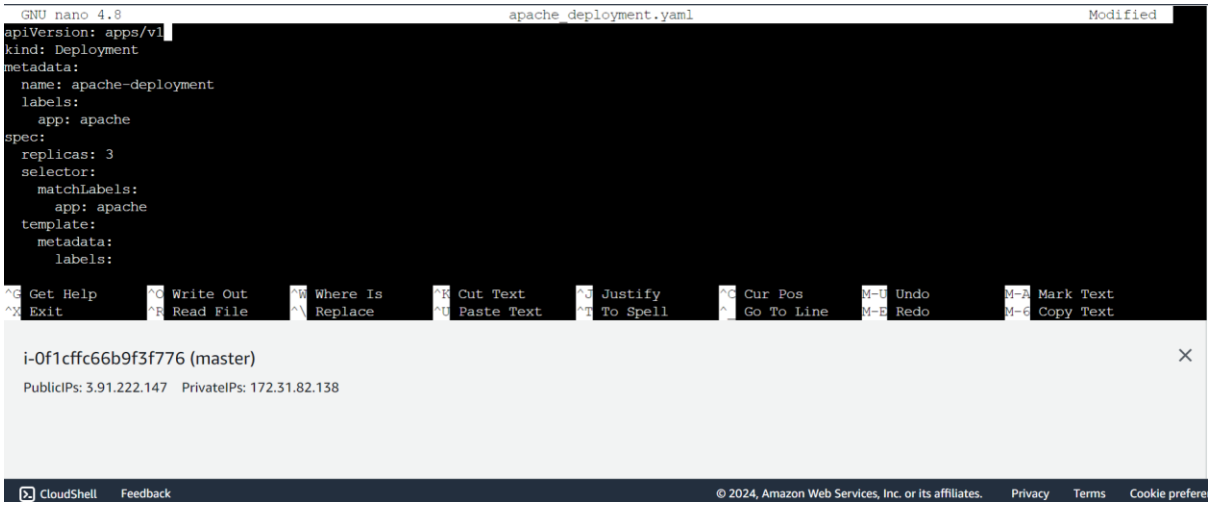
Task to be performed

1. Use the previous deployment
2. Deploy an NGINX deployment of 3 replicas
3. Create an NGINX service of type ClusterIP
4. Create an ingress service/ Apache to Apache service/ NGINX to NGINX service

Steps and Commands

- Create a file for providing script for Apache deployment by running the command **sudo nano apache_deployment.yaml**
- Provide the necessary script for apache deployment
- To run the nginx deployment by running the command **kubectl apply –f apache_deployment.yaml**
- To check the deployment by running the command **kubectl get deploy**
- To see the full details of deployment by running the command **kubectl get deploy –o wide, kubectl describe deploy nginx-deployment**
- Create a file for providing script for creating Apache service by running the command **sudo nano apache_service.yaml**
- Provide the necessary script for Apache service
- To run the nginx service by running the command **kubectl apply –f apache_service.yaml**
- To check the service status by running the command **kubectl get svc**
- To open the nginx-deployment by running the command kubectl edit deploy nginx-deployment
- Edit the necessary script in replicas provide 3
- Save and exit from nginx-deployment
- To check replicas status by running the command kubectl get pods
- To create a file for providing script for Ingress by running the command **sudo nano ingress.yaml**
- Provide the necessary script in ingress.yaml
- Next save and exit form ingress.yaml
- To run the Ingress.yaml by running the command **kubectl apply –f ingress.yaml**
- To check the Ingress status by running the command **kubectl get ingress**
- To see the Nginx page on browser put the (**public IP address:Ingress nodeport/nginx)** any machine master, slave1, slave2 of instance in the Brower the Nginx page will be reflected.

- To see the Apache page on browser put the (**public IP address:Ingress nodeport/apache)** any machine master, slave1, slave2 of instance in the Brower the Nginx page will be reflected.



```
ubuntu@ip-172-31-82-138:~$ sudo nano apache_deployment.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f apache_deployment.yaml
deployment.apps/apache-deployment created
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138


```
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
apache-deployment   3/3     3            3           6m52s
nginx-deployment    5/5     5            5           164m
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

```
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
apache-deployment   3/3     3            3           6m52s
nginx-deployment    5/5     5            5           164m
ubuntu@ip-172-31-82-138:~$ sudo nano apache_svc.yaml
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

```
  GNU nano 4.8                          apache_svc.yaml                          Modified
apiVersion: v1
kind: Service
metadata:
  name: my-apache

spec:
  type: ClusterIP
  ports:
    - targetPort: 80
      port: 80

  selector:
    app: apache

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo  M-A Mark Text  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo  M-6 Copy Text  ^Q Where Was
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
apache-deployment   3/3     3            3           6m52s
nginx-deployment    5/5     5            5           164m
ubuntu@ip-172-31-82-138:~$ sudo nano apache_svc.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f apache_svc.yaml
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
apache-deployment   3/3     3            3           6m52s
nginx-deployment    5/5     5            5           164m
ubuntu@ip-172-31-82-138:~$ sudo nano apache_svc.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f apache_svc.yaml
service/my-apache created
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
apache-deployment  3/3     3            3           6m52s
nginx-deployment   5/5     5            5           164m
ubuntu@ip-172-31-82-138:~$ sudo nano apache_svc.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f apache_svc.yaml
service/my-apache created
ubuntu@ip-172-31-82-138:~$ kubectl get svc
NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes           ClusterIP   10.96.0.1       <none>        443/TCP   5h48m
my-apache            ClusterIP   10.109.186.67   <none>        80/TCP    47s
my-nginx-deployment  ClusterIP   10.98.213.98    <none>        80/TCP    113m
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138


```
ubuntu@ip-172-31-82-138:~$ kubectl edit deploy nginx-deployment
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
  creationTimestamp: "2024-04-11T15:11:53Z"
  generation: 2
  labels:
    app: nginx
  name: nginx-deployment
  namespace: default
  resourceVersion: "24491"
  uid: 3aa72f83-9404-4c0a-a9bf-c8359afdac2a
spec:
  progressDeadlineSeconds: 600
  replicas: 3
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
                                                                23,13         20%
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ kubectl edit deploy nginx-deployment
deployment.apps/nginx-deployment edited
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ kubectl edit deploy nginx-deployment
deployment.apps/nginx-deployment edited
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
apache-deployment   3/3     3            3           24m
nginx-deployment    3/3     3            3           3h2m
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ kubectl edit deploy nginx-deployment
deployment.apps/nginx-deployment edited
ubuntu@ip-172-31-82-138:~$ kubectl get deploy
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
apache-deployment   3/3     3            3           24m
nginx-deployment    3/3     3            3           3h2m
ubuntu@ip-172-31-82-138:~$ kubectl get svc
NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes           ClusterIP   10.96.0.1       <none>        443/TCP   6h
my-apache            ClusterIP   10.109.186.67   <none>        80/TCP    12m
my-nginx-deployment  ClusterIP   10.98.213.98    <none>        80/TCP    124m
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

```
ubuntu@ip-172-31-82-138:~$ sudo nano ingress.yaml
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138


```
  GNU nano 4.8                                ingress.yaml                                      Modified
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
 name: ingress
 annotations:
   nginx.ingress.kubernetes.io/rewrite-target: /
spec:
 ingressClassName: nginx
 rules:
  - http:
      paths:
      - path: /apache
        pathType: Prefix
        backend:

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo  M-A Mark Text  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo  M-6 Copy Text  ^Q Where Was
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138


```
ubuntu@ip-172-31-82-138:~$ sudo nano ingress.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f ingress.yaml
```

i-0f1cffc66b9f3f776 (master)

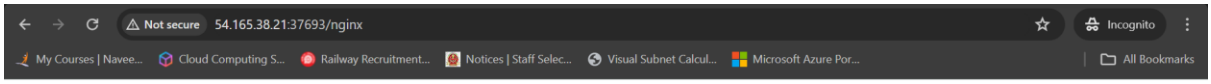
PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

```
ubuntu@ip-172-31-82-138:~$ sudo nano ingress.yaml
ubuntu@ip-172-31-82-138:~$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/ingress created
ubuntu@ip-172-31-82-138:~$ kubectl get ingress
NAME      CLASS   HOSTS   ADDRESS   PORTS   AGE
ingress   nginx   *                 80      35s
ubuntu@ip-172-31-82-138:~$
```

i-0f1cffc66b9f3f776 (master)

PublicIPs: 3.91.222.147 PrivateIPs: 172.31.82.138

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

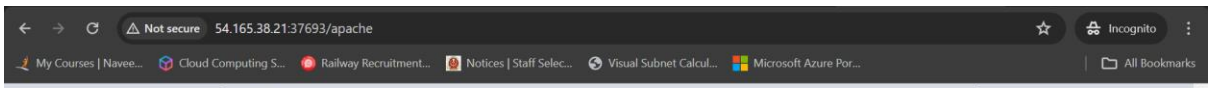


Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Ubuntu Logo

Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
```