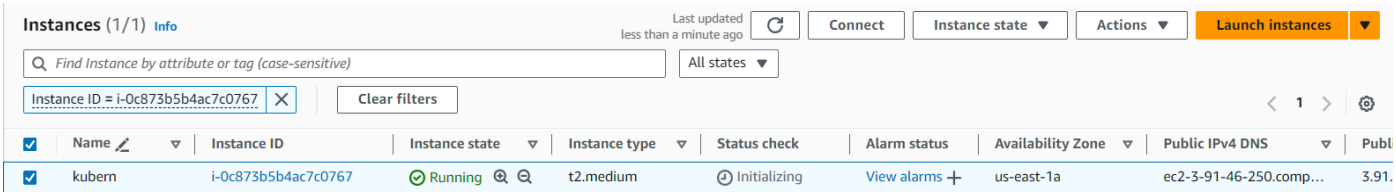
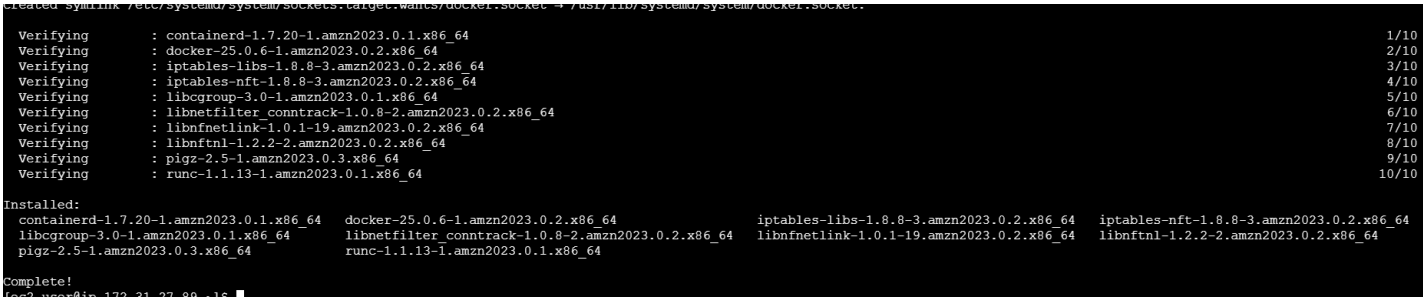
Veydant Sharma D15C 50 AdvDevops Exp4

Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

1. Create an EC2 instance with OS as Amazon Linux and make sure to allow SSH traffic.



2. To install docker run the following command:
   sudo yum install docker -y



3. Configure cgroup in daemon.json file using the following commands:

```
cd /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
EOF
```

```
[ec2-user@ip-172-31-27-89 ~]$ cd /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
EOF
```

4. Run the following command after this:

   sudo systemctl enable docker
   sudo systemctl daemon-reload
   sudo systemctl restart docker

```
[ec2-user@ip-172-31-27-89 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-27-89 docker]$
```

5. Verify Installation

```
[ec2-user@ip-172-31-27-89 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

6. Install Kubernetes

   Disable SELinux before configuring kubelet
   sudo setenforce 0
   sudo sed -i ‘s/^SELINUX=enforcing$/SELINUX=permissive/’ /etc/selinux/config

   Add kubernetes repository
   cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
   [kubernetes]
   name=Kubernetes
   baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
   enabled=1

Veydant Sharma D15C 50 AdvDevops Exp4

gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF

```
[ec2-user@ip-172-31-27-89 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-27-89 docker]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[ec2-user@ip-172-31-27-89 docker]$
```

sudo yum update
sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

```
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[ec2-user@ip-172-31-27-89 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes                                                              59 kB/s | 9.4 kB     00:00
Dependencies resolved.
=====================================================================================================
 Package                   Architecture     Version                      Repository          Size
=====================================================================================================
Installing:
 kubeadm                   x86_64           1.31.1-150500.1.1            kubernetes          11 M
 kubectl                   x86_64           1.31.1-150500.1.1            kubernetes          11 M
 kubelet                   x86_64           1.31.1-150500.1.1            kubernetes          15 M
Installing dependencies:
 conntrack-tools           x86_64           1.4.6-2.amzn2023.0.2         amazonlinux        208 k
 cri-tools                 x86_64           1.31.1-150500.1.1            kubernetes         6.9 M
 kubernetes-cni            x86_64           1.5.1-150500.1.1             kubernetes         7.1 M
 libnetfilter_cthelper     x86_64           1.0.0-21.amzn2023.0.2        amazonlinux         24 k
 libnetfilter_cttimeout    x86_64           1.0.0-19.amzn2023.0.2        amazonlinux         24 k
 libnetfilter_queue        x86_64           1.0.5-2.amzn2023.0.2         amazonlinux         30 k

Transaction Summary
=====================================================================================================
```

Configure internet options to allow bridging

- sudo swapoff -a
- echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
- sudo sysctl -p

```
[ec2-user@ip-172-31-27-89 docker]$ sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p

net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-iptables = 1
[ec2-user@ip-172-31-27-89 docker]$
```

7. Initialize the kubecluster

   sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```
[ec2-user@ip-172-31-27-89 docker]$ kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml
pod/nginx created
[ec2-user@ip-172-31-27-89 docker]$
```

Run 3 commands

```
[ec2-user@ip-172-31-27-89 docker]$ mkdir -p $HOME/.kube && sudo cp -f /etc/kubernetes/admin.conf $HOME/.kube/config && sudo chown $(id -u):$(id -g) $HOME/.kube/config
cp: cannot stat '/etc/kubernetes/admin.conf': No such file or directory
[ec2-user@ip-172-31-27-89 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
      [WARNING FileExisting-socat]: socat not found in system path
      [WARNING FileExisting-tc]: tc not found in system path
      [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
```

8. Deploy nginx server on this cluster using the command
   kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml
   Also run kubectl get pods to check creation of pod

```
pod/nginx created
[ec2-user@ip-172-31-27-89 docker]$ kubectl describe pod nginx
Name:             nginx
Namespace:        default
Priority:         0
Service Account:  default
Node:             <none>
Labels:           <none>
Annotations:      <none>
Status:           Pending
IP:
IPs:              <none>
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-sbq47 (ro)
Conditions:
  Type           Status
```

9. Use kubectl get pods to check status od pods
10. Also mention the port that u want to host

```
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

Conclusion:

In this experiment, we set up a Kubernetes cluster on an Amazon Linux EC2 instance. We installed Docker, configured it to use systemd for cgroup management, and prepared Kubernetes by disabling SELinux and installing necessary components. After initializing the cluster and deploying the Flannel network plugin, we successfully launched an Nginx server. We ensured the Nginx pod was accessible on port from the local machine.