Experiment No. 3

Objective: To understand the Kubernetes Cluster Architecture and to set up a Kubernetes Cluster on Linux Machines/Cloud.

Procedure:

1. **Instance Setup:**
    - Create three EC2 instances using Amazon Linux as the operating system.
    - Ensure that SSH traffic is allowed from any source.
    - For optimal performance, choose an instance type of at least `t2.medium`, as Kubernetes recommends a minimum of 2 vCPUs.

Name	Instance ID	Instance state	Instance type
kube-master	i-00aa79ac09d7462c0	Running	t2.medium
kube-worker1	i-0bab86cd3fbfcb40a	Running	t2.medium
kube-worker2	i-00dcfd302ffd80dda	Running	t2.medium

2. **SSH Access:**
    - SSH into each of the three machines using separate terminal windows: `ssh -i <keyname>.pem ubuntu@<public_ip_address>`

```
quantum@machine ~/Downloads ssh -i "ec-2-ubuntu.pem" ec2-user@ec2-3-88-111-183.compute-1.amazonaws.com
The authenticity of host 'ec2-3-88-111-183.compute-1.amazonaws.com (3.88.111.183)' can't be established.
ED25519 key fingerprint is SHA256:pQu+xs9foYbY3de1twjZcVVAOzmGwGv6PHmVruF/Q1s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-88-111-183.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
```

3. **Docker Installation and Configuration:**
    - On all three machines, install Docker with the command: `sudo yum install docker -y`

    - Configure Docker to use `systemd` as the cgroup driver by creating and editing the `daemon.json` file:
        - Change directory to `/etc/docker`

        - Use the command `cat <<EOF | sudo tee /etc/docker/daemon.json` followed by the JSON configuration details and end with `EOF`

    - Enable and restart Docker: `sudo systemctl enable docker` `sudo systemctl daemon-reload` `sudo systemctl restart docker` `docker -v`

```
[ec2-user@ip-172-31-92-18 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:09:56 ago on Wed Sep 11 15:19:39 2024.
Dependencies resolved.
================================================================================
 Package                                                    Architecture
================================================================================
Installing:
 docker                                                     x86_64
Installing dependencies:
 containerd                                                 x86_64
 iptables-libs                                              x86_64
 iptables-nft                                               x86_64
 libcgroup                                                  x86_64
 libnetfilter_conntrack                                     x86_64
 libnfnetlink                                               x86_64
 libnftnl                                                   x86_64
 pigz                                                       x86_64
 runc                                                       x86_64

Transaction Summary
```

4. **Kubernetes Installation:**
    - Disable SELinux before configuring `kubelet`: `sudo setenforce 0` `sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config`

    - Add the Kubernetes repository and install Kubernetes components:
        - Use the command `cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo` followed by the repository configuration details and end with `EOF` `sudo yum update` `sudo yum install -y kubelet kubeadm kubectl 0--disableexcludes=kubernetes`

- Configure networking for bridging: `sudo swapoff -a` `echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf` `sudo sysctl -p`

```
[ec2-user@ip-172-31-81-63 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v

Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-81-63 docker]$
```

```
[ec2-user@ip-172-31-81-63 docker]$ sudo setenforce 0
[ec2-user@ip-172-31-81-63 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
[ec2-user@ip-172-31-81-63 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:01:34 ago on Wed Sep 11 15:39:05 2024.
Dependencies resolved.
================================================================================================================================================
 Package                                                 Architecture                            Version
================================================================================================================================================
Installing:
 kubeadm                                                 x86_64                                  1.30.4-150500.1.1
 kubectl                                                 x86_64                                  1.30.4-150500.1.1
 kubelet                                                 x86_64                                  1.30.4-150500.1.1
Installing dependencies:
 conntrack-tools                                         x86_64                                  1.4.6-2.amzn2023.0.2
 cri-tools                                               x86_64                                  1.30.1-150500.1.1
 kubernetes-cni                                          x86_64                                  1.4.0-150500.1.1
 libnetfilter_cthelper                                   x86_64                                  1.0.0-21.amzn2023.0.2
 libnetfilter_cttimeout                                  x86_64                                  1.0.0-19.amzn2023.0.2
 libnetfilter_queue                                      x86_64                                  1.0.5-2.amzn2023.0.2
 socat                                                   x86_64                                  1.7.4.2-1.amzn2023.0.2

Transaction Summary
================================================================================================================================================
Install  10 Packages
```

5. **Master Node Setup:**
   - Initialize the Kubernetes master node (perform only on the master machine): `sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all`

   - After initialization, set up the Kubernetes configuration on the master node: `mkdir -p $HOME/.kube` `sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config` `sudo chown $(id -u):$(id -g) $HOME/.kube/config`

- Save the generated join command from the output for worker nodes. This command is unique and specific to your cluster setup: `kubeadm join 172.31.91.120:6443 --token r8j60r.n1j6h0klbewvoka5\--discovery-token-ca-cert-hashsha256:dd8426260174d673303aef17717f740772fcf7ee782245bc653eecf4a1305da7`

- Deploy the Flannel networking plugin to enable pod communication: `kubectl apply -f` https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

- Check the status of the pods to ensure they are running: `kubectl get pods --all-namespaces`

```
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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 172.31.81.63:6443 --token zh5jbb.a6ty3eujzc51d15d \
        --discovery-token-ca-cert-hash sha256:0822f656bf52a17a2b6686c123f811306f41495ca650a0aed9bf6cd2d2f6f8c5
[ec2-user@ip-172-31-81-63 docker]$  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-81-63 docker]$
```

```
[ec2-user@ip-172-31-81-63 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

(error)

```
[ec2-user@ip-172-31-88-200 docker]$ kubectl get pods
No resources found in default namespace.
[ec2-user@ip-172-31-88-200 docker]$ kubectl get pods --all-namespaces
NAMESPACE      NAME
 READY   STATUS             RESTARTS         AGE
kube-flannel   kube-flannel-ds-lptnz
 0/1     Pending            0                45s
kube-system    coredns-55cb58b774-bddbl
 0/1     Pending            0                20m
kube-system    coredns-55cb58b774-tk4v6
 0/1     Pending            0                20m
kube-system    etcd-ip-172-31-88-200.ec2.internal
 1/1     Running            11 (2m38s ago)   20m
kube-system    kube-apiserver-ip-172-31-88-200.ec2.internal
 1/1     Running            15 (114s ago)    21m
kube-system    kube-controller-manager-ip-172-31-88-200.ec2.internal
 1/1     Running            3 (3m49s ago)    5m58s
kube-system    kube-proxy-chdf8
 1/1     Running            11 (113s ago)    20m
kube-system    kube-scheduler-ip-172-31-88-200.ec2.internal
 0/1     CrashLoopBackOff   18 (43s ago)     20m
```

6. **Worker Node Setup:**
    - On each worker node, install the required package and configure `kubelet`: `sudo yum install iproute-tc -y` `sudo systemctl enable kubelet` `sudo systemctl restart kubelet`

    - Join the worker nodes to the Kubernetes cluster using the join command from the master node: `kubeadm join 172.31.91.120:6443 --token r8j60r.n1j6h0klbewvoka5\--discovery-token-ca-cert-hashsha256:dd8426260174d673303aef17717f740772fcf7ee782245bc653eecf4a1305da7`

(error)

```
[ec2-user@ip-172-31-81-158 docker]$ sudo kubeadm join 172.31.88.200:6443 --token
 24afv3.qqvxo2z9kpka99jr --discovery-token-ca-cert-hash sha256:397d893742317bc80
2d145e7c5e41114a646bad88631730bd27e46edd1c05d05
[preflight] Running pre-flight checks
^C
```

7. **Verify Node Status:**
    - On the master node, verify that the worker nodes have successfully joined the cluster by running: `watch kubectl get nodes`

Conclusion:

Setting up the Kubernetes cluster involved several challenges. Network configuration issues initially hindered the deployment of the Flannel plugin, requiring open ports and a functional Kubernetes API server. Disabling SELinux and adjusting firewall rules were essential for proper communication between

components. Worker nodes experienced difficulties with the kubelet service, which needed to be correctly configured and restarted. Additionally, accurate copying of the join command, including the token and discovery-token-ca-cert-hash, was crucial for integrating worker nodes into the cluster. These issues underscored the need for precise configuration and troubleshooting to achieve a stable Kubernetes setup.