Commands : SetUp K8s HA Cluster (Updated)

Kubernetes Installation Process

************* Install Kubernetes on Master Node *************

1. Upgrade apt packages
   ```
   sudo apt-get update
   ```

2. Create a configuration file for containerd:

   ```
   cat <<EOF | sudo tee /etc/modules-load.d/containerd.conf
   overlay
   br_netfilter
   EOF
   ```

3. Load modules:

   ```
   sudo modprobe overlay
   sudo modprobe br_netfilter
   ```

4. Set system configurations for Kubernetes networking:

   ```
   cat <<EOF | sudo tee /etc/sysctl.d/99-kubernetes-cri.conf
   net.bridge.bridge-nf-call-iptables = 1
   net.ipv4.ip_forward = 1
   net.bridge.bridge-nf-call-ip6tables = 1
   EOF
   ```

5. Apply new settings:
   ```
   sudo sysctl –system
   ```

6. Install containerd:

   ```
   sudo apt-get update && sudo apt-get install -y containerd
   ```

7. Create a default configuration file for containerd:

```
sudo mkdir -p /etc/containerd
```

8. Generate default containerd configuration and save to the newly created default file:

```
sudo mkdir -p /etc/containerd
containerd config default | sudo tee /etc/containerd/config.toml
```

9. Restart containerd to ensure new configuration file usage:

```
sudo systemctl restart containerd
```

10. Verify that containerd is running.

```
sudo systemctl status containerd
```

11. Disable swap:

```
sudo swapoff -a
```

12. Disable swap on startup in `/etc/fstab`:

```
sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
```

13. Install dependency packages:

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

14. Download and add the GPG key:

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

15. Add Kubernetes to the repository list:

```
cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
```

16. Update package listings:

```
sudo apt-get update
```

17. Install Kubernetes packages (***Note: If you get a dpkg lock message, just wait a minute or two before trying the command again***):

```
sudo apt-get install -y kubelet=1.23.0-00 kubeadm=1.23.0-00
kubectl=1.23.0-00
```

18. Turn off automatic updates:

```
sudo apt-mark hold kubelet kubeadm kubectl
```

Log into both Worker Nodes to perform previous steps 1 to 18.

19. **Initialize the Cluster-**
20. Initialize the Kubernetes cluster on the control plane node using kubeadm (***Note: This is only performed on the Control Plane Node***):

```
sudo kubeadm init --pod-network-cidr 192.168.0.0/16 --kubernetes-version
1.23.0
```

21. Set kubectl access:

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

22. Test access to cluster:

```
kubectl get nodes
```

23. **Install the Calico Network Add-On -**
24. On the Control Plane Node, install Calico Networking:

```
kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
```

25. Wait for 2-4 Min and Check the status of the control plane node:

```
kubectl get nodes
```

26. **Join the Worker Nodes to the Cluster**
    a. In the Control Plane Node, create the token and copy the kubeadm join command *(NOTE: The join command can also be found in the output from `kubeadm init` command)*:

```
kubeadm token create --print-join-command
```

    b. In both Worker Nodes, paste the kubeadm join command to join the cluster. Use sudo to run it as root:

```
sudo kubeadm join ...
```

    c. In the Control Plane Node, view cluster status (Note: You may have to wait a few moments to allow all nodes to become ready):

```
kubectl get nodes
```