**Kubernetes HA Cluster Setup with HAProxy and Nginx**

**Prerequisites:**

**1. Servers:** Ensure you have 7 servers with the following IPs:

- Bastion Host: 172.16.106.217

- Master Nodes: 172.16.106.218 to 172.16.106.220

- Worker Nodes: 172.16.106.221 to 172.16.106.223

**2. Software Requirements:**

- Ubuntu 24.04 installed on all servers.

- SSH access to all servers configured with the bastion host accessible via `govadmin` user.

- Internet access for package installation.

**3. Networking:**

- Ensure all servers can communicate with each other over the network.

- Configure firewall rules to allow necessary ports (e.g., SSH, Kubernetes API ports).

**Steps to Setup Kubernetes HA Cluster with HAProxy and Nginx:**

**1. Install Docker on All Nodes:**

SSH into each node including bastion host

ssh govadmin@172.16.106.217

Install Docker

>> apt update

>> apt install -y docker.io

>> systemctl enable --now docker

**2. Configure Kubernetes Repository and Install kubeadm, kubelet, and kubectl:**

**Add Kubernetes repository and install tools**

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

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

>> echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | tee /etc/apt/sources.list.d/kubernetes.list

>> apt-get update

>> apt-get install -y kubelet kubeadm kubectl

>> apt-mark hold kubelet kubeadm kubectl

**3. Initialize Kubernetes Control Plane (Master Nodes):**

**On master node 1 (172.16.106.218)**

**>>** kubeadm init --control-plane-endpoint="esds-kubernetes" --apiserver-advertise-address=172.16.106.218 --upload-certs --pod-network-cidr=10.244.0.0/16

**Follow the kubeadm init output instructions to set up kubeconfig for non-root user**

>> mkdir -p $HOME/.kube

>> cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

>> chown $(id -u):$(id -g) $HOME/.kube/config

**Join additional master nodes (172.16.106.219, 172.16.106.220)**

>> kubeadm join 172.16.106.218:6443 --token <token> \

--discovery-token-ca-cert-hash sha256:<hash> \

--control-plane --certificate-key <cert-key>

**4. Join Worker Nodes to the Cluster:**

**On each worker node (172.16.106.221 to 172.16.106.223)**

>> kubeadm join 172.16.106.218:6443 --token <token> \

--discovery-token-ca-cert-hash sha256:<hash>

**5. Set Up Networking (Calico):**

**On any master node (after kubeadm init)**

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

**6. Configure HAProxy for Load Balancing Master Nodes:**

**Install HAProxy on the bastion host or a dedicated server for load balancing the Kubernetes API requests to the master nodes.**

**Install HAProxy**

>> apt-get update && apt-get install -y haproxy

Configure HAProxy (/etc/haproxy/haproxy.cfg)

>> vi /etc/haproxy/haproxy.cfg

Configure HAProxy configuration file (`/etc/haproxy/haproxy.cfg`) as follows:

global

log /dev/log local0

log /dev/log local1 notice

chroot /var/lib/haproxy

stats socket /run/haproxy/admin.sock mode 660 level admin

stats timeout 30s

user haproxy

group haproxy

daemon

defaults

log global

mode tcp

option tcplog

option dontlognull

timeout connect 5000

timeout client 50000

timeout server 50000

frontend k8s-api

bind \*:6443

mode tcp

default\_backend k8s-api-servers

backend k8s-api-servers

mode tcp

balance roundrobin

server k8s-master-1 172.16.106.218:6443 check

server k8s-master-2 172.16.106.219:6443 check

server k8s-master-3 172.16.106.220:6443 check

Restart HAProxy after configuration:

>> systemctl restart haproxy

**7. Configure Nginx for Load Balancing Kubernetes Services:**

**Install Nginx on the bastion host or a dedicated server to load balance requests to Kubernetes services.**

Install Nginx

>> apt-get update && apt-get install -y nginx

Configure Nginx (/etc/nginx/nginx.conf)

>> vi /etc/nginx/nginx.conf

Configure Nginx configuration file (`/etc/nginx/nginx.conf`) to include upstream configuration for Kubernetes services:

http {

upstream kubernetes {

server 172.16.106.218:80;

server 172.16.106.219:80;

server 172.16.106.220:80;

}

server {

listen 80;

location / {

proxy\_pass http://kubernetes;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

}

}

Restart Nginx after configuration:

>> systemctl restart nginx

**8. Verify Cluster Setup:**

**On bastion host or any master node**

>> kubectl get nodes -o wide

>> kubectl get pods --all-namespaces