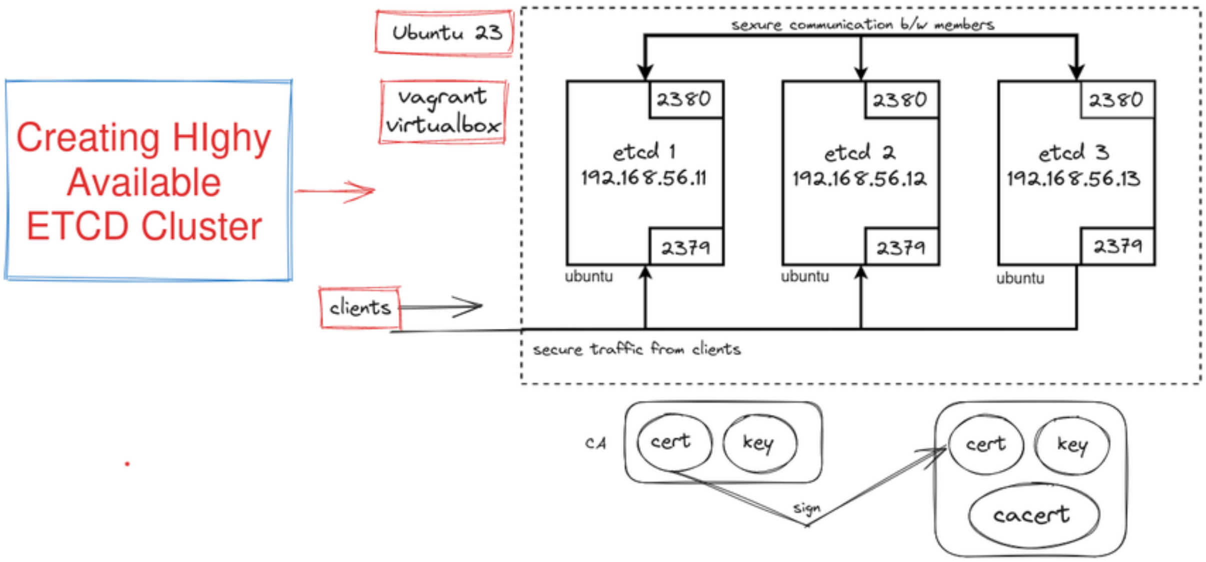
Al-Nafi Kubernetes Course

Setting up **Highly Available ETCD** cluster

⚠ Perform this practice on your local PC if possible, so that you can follow up easily



Step: 1 (Create ETCD Cluster using Vagrantfile)

⚠ Provision 3 VM’s using Vagrantfile.

✓ **Use below command to provision VM’s using vagrant.**

ℹ vagrant up --provider virtualbox

Step: 2 (Generate TLS Certificates)

⚠ First download required binaries.

✓ **Use below command to download binaries. Execute each command step by step.**

```
wget -q --show-progress https://github.com/cloudflare/cfssl/releases/download/v1.6.4/cfssl_1.6.4_linux_amd64
```

```
wget -q --show-progress https://github.com/cloudflare/cfssl/releases/download/v1.6.4/cfssljson_1.6.4_linux_amd64
```

```
chmod +x cfssl_1.6.4_linux_amd64 cfssljson_1.6.4_linux_amd64
```

```
sudo mv cfssl_1.6.4_linux_amd64 /usr/local/bin/cfssl
sudo mv cfssljson_1.6.4_linux_amd64 /usr/local/bin/cfssljson
```

Step: 3 (Create a Certificate Authority)

We will use this CA(Certificate Authority) to create other **TLS** certificates.

Use below commands to create certificate authority. Execute each command one after another

```
cat > ca-config.json <<EOF
{
    "signing": {
        "default": {
            "expiry": "8760h"
        },
        "profiles": {
            "etcd": {
                "expiry": "8760h",
                "usages": ["signing","key encipherment","server auth","client auth"]
            }
        }
    }
}
EOF
```

```
cat > ca-csr.json <<EOF
{
  "CN": "etcd cluster",
  "key": {
    "algo": "rsa",
    "size": 2048
  },
  "names": [
    {
      "C": "GB",
      "L": "England",
      "O": "Kubernetes",
      "OU": "ETCD-CA",
      "ST": "Cambridge"
    }
  ]
}
EOF
```

```
cfssl gencert -initca ca-csr.json | cfssljson -bare ca
```

Step: 4 (Create TLS certificates)

Execute below command to create TLS certificates for VMs.

```
ETCD1_IP="192.168.56.11"
ETCD2_IP="192.168.56.12"
ETCD3_IP="192.168.56.13"
```

```
cat > etcd-csr.json <<EOF
{
  "CN": "etcd",
  "hosts": [
    "localhost",
    "127.0.0.1",
    "${ETCD1_IP}",
    "${ETCD2_IP}",
    "${ETCD3_IP}"
  ],
  "key": {
    "algo": "rsa",
    "size": 2048
  },
  "names": [
    {
      "C": "GB",
      "L": "England",
      "O": "Kubernetes",
      "OU": "etcd",
      "ST": "Cambridge"
    }
  ]
}
EOF
```

```
cfssl gencert -ca=ca.pem -ca-key=ca-key.pem -config=ca-config.json -profile=etcd etcd-csr.json | cfssljson -bare etcd
```

Step: 5 (Copy certificate to ETCD nodes)

Use below command to copy certificate to ETCD nodes

```
{

declare -a NODES=(192.168.56.11 192.168.56.12 192.168.56.13)

for node in ${NODES[@]}; do
  scp ca.pem etcd.pem etcd-key.pem root@$node:
done

}
```

Use below steps on each ETCD node

Step: 6 (Copy the certificate to a standard location)

Use below commands to copy certificates to standard location

Use these commands on all ETCD nodes

```
{
  mkdir -p /etc/etcd/pki
  mv ca.pem etcd.pem etcd-key.pem /etc/etcd/pki/
}
```

Step: 7 (Download etcd & etcdctl binaries from Github)

Use below commands to download etcd and etcdctl binaries from the github.

Use these commands on all ETCD nodes

```
{
  ETCD_VER=v3.5.1
  wget -q --show-progress "https://github.com/etcd-io/etcd/releases/download/${ETCD_VER}/etcd-${ETCD_VER}-linux-amd64.tar.gz"
  tar zxf etcd-v3.5.1-linux-amd64.tar.gz
  mv etcd-v3.5.1-linux-amd64/etcd* /usr/local/bin/
  rm -rf etcd*
}
```

Step: 8 (Create systemd unit file for etcd service)

Use below commands to create systemd unit file for ETCD service.

Use these commands on all ETCD nodes

```
{
NODE_IP="192.168.56.12"

ETCD_NAME=$(hostname -s)

ETCD1_IP="192.168.56.11"
ETCD2_IP="192.168.56.12"
ETCD3_IP="192.168.56.13"

cat <<EOF >/etc/systemd/system/etcd.service
[Unit]
Description=etcd

[Service]
Type=notify
ExecStart=/usr/local/bin/etcd \\
  --name ${ETCD_NAME} \\
  --cert-file=/etc/etcd/pki/etcd.pem \\
  --key-file=/etc/etcd/pki/etcd-key.pem \\
  --peer-cert-file=/etc/etcd/pki/etcd.pem \\
  --peer-key-file=/etc/etcd/pki/etcd-key.pem \\
  --trusted-ca-file=/etc/etcd/pki/ca.pem \\
  --peer-trusted-ca-file=/etc/etcd/pki/ca.pem \\
  --peer-client-cert-auth \\
  --client-cert-auth \\
  --initial-advertise-peer-urls https://${NODE_IP}:2380 \\
  --listen-peer-urls https://${NODE_IP}:2380 \\
  --advertise-client-urls https://${NODE_IP}:2379 \\
  --listen-client-urls https://${NODE_IP}:2379,https://127.0.0.1:2379 \\
  --initial-cluster-token etcd-cluster-1 \\
  --initial-cluster etcd1=https://${ETCD1_IP}:2380,etcd2=https://${ETCD2_IP}:2380,etcd3=https://${ETCD3_IP}:2380 \\
  --initial-cluster-state new
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
EOF
}
```

Step: 9 (Enable and Start etcd service)

Use below commands to enable and start etcd service

Use these commands on all ETCD nodes

```
{
  systemctl daemon-reload
  systemctl enable --now etcd
}
```

Step: 10 (Verify Etcd cluster status)

Use below commands to verify ETCD cluster status

Use these commands on all ETCD nodes

```
ETCDCTL_API=3 etcdctl \
  --endpoints=https://192.168.56.11:2379,https://192.168.56.12:2379,https://192.168.56.13:2379 \
  --cacert=/etc/etcd/pki/ca.pem \
  --cert=/etc/etcd/pki/etcd.pem \
  --key=/etc/etcd/pki/etcd-key.pem \
  member list
```

Step: 11 (Export these as environment variables)

Use below commands to export these as environment variables.

Use these commands on all ETCD nodes

```
export ETCDCTL_API=3
```

```
export ETCDCTL_ENDPOINTS=https://192.168.56.11:2379,https://192.168.56.12:2379,https://192.168.56.13:2379
```

```
export ETCDCTL_CACERT=/etc/etcd/pki/ca.pem
export ETCDCTL_CERT=/etc/etcd/pki/etcd.pem
export ETCDCTL_KEY=/etc/etcd/pki/etcd-key.pem
```

Step: 12 (Status Check Commands)

```
etcdctl member list
etcdctl endpoint status
etcdctl endpoint health
```