Kubernetes Hardware Details

Kubernetes Pre-requisites setup on CentOS

Kubernetes Installation:

Kubernetes Master HA Architecture Planned for Implementation

Setting Kubernetes Master Cluster

Adding Kubernetes Node

Name space Creation For different Environment (dev/prod)

Configuring Kubeconfig for client

Ingress Controller setup

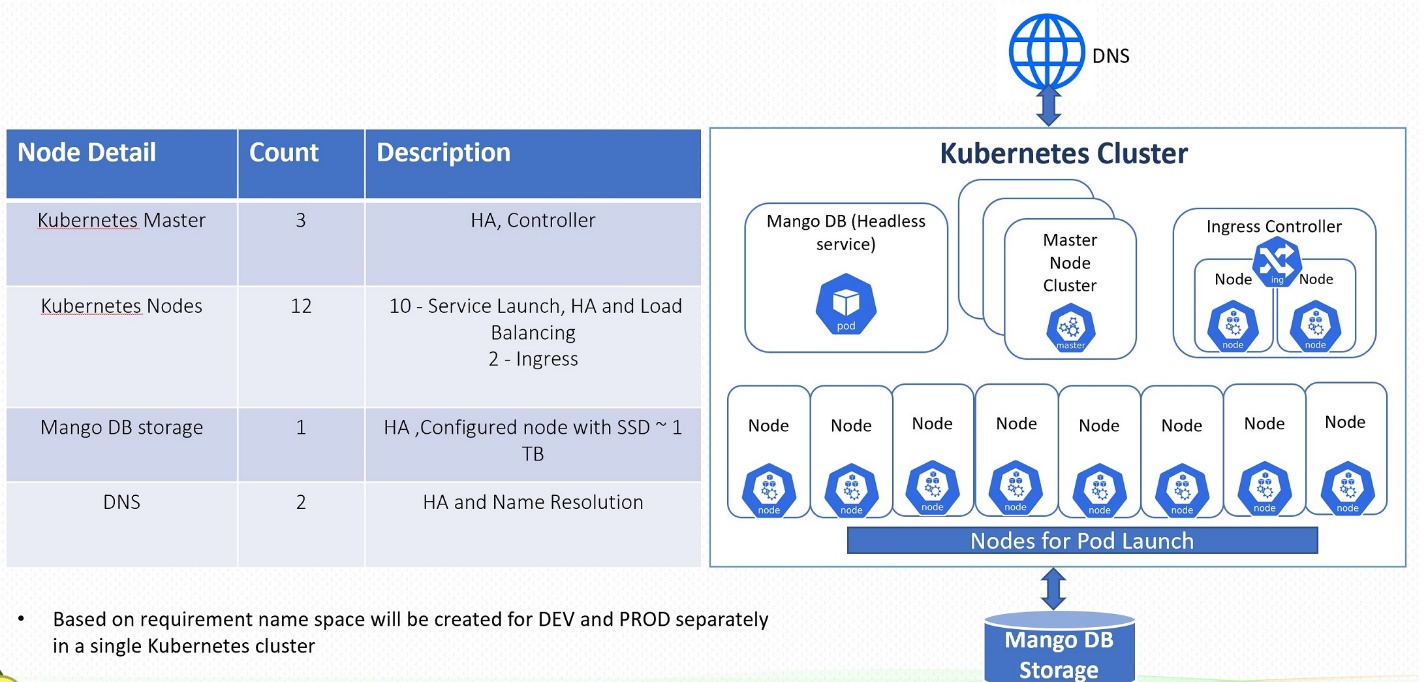
NFS and PV setup

Setting MongoDB ReplicaSet in Kubernetes

Kubernetes Hardware Details:

The below table provide the hardware details intend to implement the project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kubernetes Infrastructure** | **Node** | **CPU** | **RAM (GB)** | **Disk (GB)\*** |
| Master | Master1 | 4 | 8 | 60 |
| Master2 | 4 | 8 | 60 |
| Master3 | 4 | 8 | 60 |
| Application | Node1 | 4 | 6 | 60 |
| Node2 | 4 | 6 | 60 |
| Node3 | 4 | 6 | 60 |
| Node4 | 4 | 6 | 60 |
| Storage & MongoDB | Node5 | 4 | 6 | 60 |
| Node6 | 4 | 6 | 60 |
| DNS | DNS1 | 1 | 4 | 60 |
| DNS2 | 1 | 4 | 60 |
| NFS | NFS | 2 | 4 | 1TB |
| **Total** |  | **40** | **72** |  |



Kubernetes Pre-requisites setup on CentOS

1. Set hostname for the master by running below command

hostnamectl set-hostname kmaster-1

1. Update the host file to resolve other hosts

cat >>/etc/hosts<<EOF

192.168.0.115 kmaster-1

192.168.0.116 kmaster-2

192.168.0.117 kmaster-3

192.168.0.118 knode-1

EOF

1. Install dependencies for kubernetes by executing below commands

yum install -y -q yum-utils device-mapper-persistent-data lvm2 > /dev/null 2>&1

yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo > /dev/null 2>&1

yum install -y -q docker-ce >/dev/null 2>&1

1. Enable the docker

systemctl enable docker

systemctl start docker

1. Disable SELinux : This will help to avoid port blocking

setenforce 0

sed -i --follow-symlinks 's/^SELINUX=enforcing/SELINUX=disabled/' /etc/sysconfig/selinux

1. Disable the firewall to overcome service block or port block issue

systemctl disable firewalld

systemctl stop firewalld

1. Disable the Swap Memory: This will avoid the kubernetes performance issue

sed -i '/swap/d' /etc/fstab

swapoff –a

1. Enable Kubernetes to create the Network Bridge

cat >>/etc/sysctl.d/kubernetes.conf<<EOF

net.bridge.bridge-nf-call-ip6tables = 1

net.bridge.bridge-nf-call-iptables = 1

EOF

sysctl --system

Repeat the above steps by changing hostname alone in all machine

Kubernetes Installation:

1. Add Package repository to existing environment

cat >>/etc/yum.repos.d/kubernetes.repo<<EOF

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg

https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

EOF

1. Install Kubernetes packages by running below command

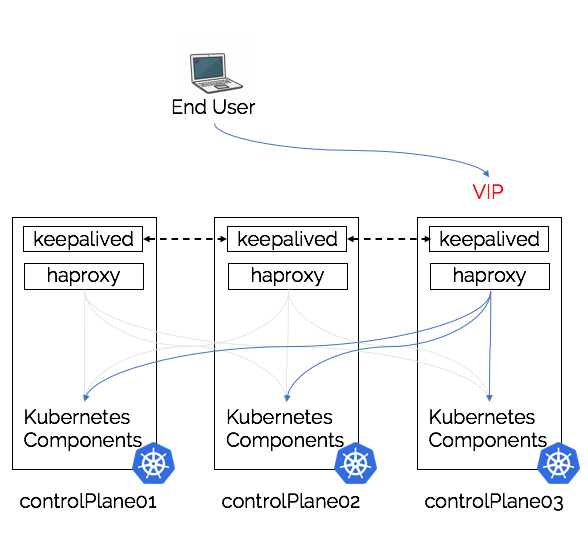
yum install -y kubeadm kubelet kubectl

1. Enable and start the kublet service on all machines

systemctl enable kubelet

systemctl start kubelet

Kubernetes Master HA Architecture Planned for Implementation:



**Setting Kubernetes Master Cluster:**

Primary Master:

1. Generate the Vitrtual IP configuration

Required Input: Ethernet Interface name , virtual IP address and Leader (true/false)

Command:

docker run --network host --rm plndr/kube-vip:0.1.5 kubeadm init –interface <interfacename> --vip <virtualip address> --startAsLeader=true | sudo tee /etc/kubernetes/manifests/vip.yaml

Example:

sudo docker run --network host --rm plndr/kube-vip:0.1.5 kubeadm init --interface ens192 --vip 192.168.0.81 --startAsLeader=true | sudo tee /etc/kubernetes/manifests/vip.yaml

This will generate /etc/kubernetes/manifests/vip.yaml file

Note: Set the leader as true for the first server

1. Initiate Kubernetes by running below command

Required Input: Generated VIP

Command:

kubeadm init --kubernetes-version 1.18.5 --control-plane-endpoint <virtual IP> --upload-certs

Example:

sudo kubeadm init --kubernetes-version 1.17.0 --control-plane-endpoint 192.168.0.81 --upload-certs

This will generate Master Kubernetes and give set of instruction to set

1. Please execute below command to set the network on Primary Master alone

mkdir -p $HOME/.kube

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

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

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

Secondary Master:

1. Run below command to join secondary master to priimay

sudo kubeadm join 192.168.0.81:6443 --token w5atsr.blahblahblah --control-plane --certificate-key abc123

1. Enable the Keep Alive setup by running below command

sudo docker run -v /etc/kubernetes/admin.conf:/etc/kubernetes/admin.conf --network host --rm plndr/kube-vip:0.1.5 kubeadm join --interface ens192 --vip 192.168.0.81 --startAsLeader=false | sudo tee /etc/kubernetes/manifests/vip.yaml

Adding Kubernetes Node:

1. Run below to integrate Node with Kubernetes master

kubeadm join --discovery-token-unsafe-skip-ca-verification --token=102952.1a7dd4cc8d1f4cc5 172.17.0.50:6443

Name space Creation For different Environment (dev/prod)

**nsrback.yaml**

---

apiVersion: v1

kind: Namespace

metadata:

name: stage-ns

---

apiVersion: v1

kind: ServiceAccount

metadata:

name: stage-service-account

namespace: stage-ns

---

kind: Role

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: stage-role

namespace: stage-ns

rules:

- apiGroups: [""]

resources: ["pods", "pods/log"]

verbs: ["get", "list", "watch"]

---

kind: RoleBinding

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: stage-role-binding

namespace: stage-ns

subjects:

- kind: ServiceAccount

name: stage-service-account

namespace: stage-ns

roleRef:

kind: Role

name: stage-role

apiGroup: rbac.authorization.k8s.io

SECRET\_NAME=$(kubectl get sa stage-service-account --namespace stage-ns -o json | jq -r .secrets[].name)

kubectl get secret --namespace stage-ns "${SECRET\_NAME}" -o json | jq -r '.data["ca.crt"]' | base64 --decode > ca.crt

USER\_TOKEN=$(kubectl get secret --namespace stage-ns "${SECRET\_NAME}" -o json | jq -r '.data["token"]' | base64 --decode)

abigirl\_p@cloudshell:~ (mythic-cinema-275706)$ **kubectl apply -f nsrbac.yaml**

namespace/stage-ns created

serviceaccount/stage-service-account created

role.rbac.authorization.k8s.io/stage-role created

rolebinding.rbac.authorization.k8s.io/stage-role-binding created

abigirl\_p@cloudshell:~ (mythic-cinema-275706)$ **SECRET\_NAME=$(kubectl get sa stage-service-account --namespace stage-ns -o json | jq -r .secrets[].name)**

abigirl\_p@cloudshell:~ (mythic-cinema-275706)$ echo $SECRET\_NAME

stage-service-account-token-kr4dd

abigirl\_p@cloudshell:~ (mythic-cinema-275706)$ **kubectl get secret --namespace stage-ns "${SECRET\_NAME}" -o json | jq -r '.data["ca.crt"]' | base64 --decode > ca.crt**

abigirl\_p@cloudshell:~ (mythic-cinema-275706)$ **cat ca.crt**

-----BEGIN CERTIFICATE-----

MIIDDDCCAfSgAwIBAgIRAO4N/SRJLPpF8UZU/6rMXLAwDQYJKoZIhvcNAQELBQAw

LzEtMCsGA1UEAxMkODMxNDE0OTMtYWI3MS00MTdjLTg3OWQtNjY3MTI2MWFiYTkx

MB4XDTIwMDQyOTA4MzYyOVoXDTI1MDQyODA5MzYyOVowLzEtMCsGA1UEAxMkODMx

NDE0OTMtYWI3MS00MTdjLTg3OWQtNjY3MTI2MWFiYTkxMIIBIjANBgkqhkiG9w0B

AQEFAAOCAQ8AMIIBCgKCAQEAs+j7qtAg3BjYWWYbO51F5/VECcW3GXP58XPm8q5b

/iMTS33NV8SEKgbN0w6g5ubn4q1bLoaawguW97iYRv+zp2oYKEuNzLkz+Owc2qnp

MdRiGAbs8oSMgqRYkR+R63UNoa/gM7Cd/Ij0UTFtHrniDy017sEsfXEq5+GHj682

GhBWFvTW3BdGXwVF0pSGoRKaW9gmdy/g32z81ISPywWN/FJDcOzR01ivOM1ivuGS

EPusqdOHy61FTH5E2YbjKc2WwGdQeFuEplMgzmnWCTNcRmIl+yNATI4Zw14grCEy

8bzgUk2UUqqNHq0exvYwPMiCm/7GPDqlacacKlv6zUoJNQIDAQABoyMwITAOBgNV

HQ8BAf8EBAMCAgQwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEA

Mi+XoFgei7bilLLasHKCcWsuNBMi8iiVr8TLsqjWdFn4uYCY24ANYy5UDIn8YiRd

2if0xETbpjbNCqnDiS9T5aHIS+pItJx58Fb5gpFuTXlg85gDlOhUD/GZFu/nyIEG

7W6SNBTpzeF8b4VBGCjJdzDC9wgeuPO/asNcKugiX3RRb21XRdGV2cvahsB3OPxZ

VU4bStBedVqKM2ABr3zo1rHJOltMN08Ea3Zq3w9p71y1LKh4oAzbWOfW1OELUut9

0bPfj774ygG+6Gw3H9jHFb8SDkYNLWXnZvNH0UYBYvcyqSccC/Q780V/undF8DEV

lMydMpIC3UZBGqoHz0SxCg==

-----END CERTIFICATE-----

abigirl\_p@cloudshell:~ (mythic-cinema-275706)$ **USER\_TOKEN=$(kubectl get secret --namespace stage-ns "${SECRET\_NAME}" -o json | jq -r '.data["token"]' | base64 --decode)**

abigirl\_p@cloudshell:~ (mythic-cinema-275706)$ echo $USER\_TOKEN

eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJzdGFnZS1ucyIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJzdGFnZ

<https://github.com/scotty-c/kubernetes-security-workshop/blob/master/rbac-namespaces-clusterroles/scripts/kubectl.sh>

S1zZXJ2aWNlLWFjY291bnQtdG9rZW4ta3I0ZGQiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoic3RhZ2Utc2VydmljZS1hY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiMDM

zNDA1M2UtYzExYy0xMWVhLTgwZGYtNDIwMTBhOGUwMDkzIiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OnN0YWdlLW5zOnN0YWdlLXNlcnZpY2UtYWNjb3VudCJ9.WSnsCmr8GfzixPgvBWZOIIzMrm5O6h0Y7oHJVecoZJ2tryNHwqmD0KeVMY0aUwZFmgod9X4UjNp6J7Sz5ym3

tqmXPLLijIzs0VXv6Uv42QeNHr81S-heQtnpBVAOR2QonzktqiKLJ5I7Lu3hQYQ40\_-41VTSwqkRgP-qcOCcwpi9rGAqETBhEeglkJ26RgmRz3ypuNeX1ZVzAePRmnYl3M95Xogx9DkIvbTwOMkhdhqgoRhhEy8t3YX-xOSYVYRG36Ck\_ForGRvozoRh\_4Z5o9JMPbzmqnlKAfok8hEiVC

el-L\_ix0uKVZmSJcUYS2veoDpWgpd8dC\_8LoKKKhPjpA

Configuring Kubeconfig for client

Ingress Controller setup

NFS and PV setup

NFS Server Setup:

1. Install the server by running below command

yum install nfs-utils

1. Create directory and provide permission which need to be exported

mkdir /var/nfsshare

chmod -R 755 /var/nfsshare

chown nfsnobody:nfsnobody /var/nfsshare

1. Start the Service and make the service as permenant by running below command

systemctl start nfs-server

systemctl enable nfs-server

1. Update the configuration like the folder which need to be exported

echo '/var/nfsshare \*(rw)'>>/etc/exports

exportfs –r

1. Check the folder are shared by running below command

showmount -e <server\_ip>

Try to mount the folder from client. If you have problem, please try to disable SElinux and firewall by following the steps provided in above document. After disable, kindly check the same

PV Setup:

Setting MongoDB ReplicaSet in Kubernetes

Reference:

<https://github.com/justmeandopensource/kubernetes/blob/master/docs/install-cluster-centos-7.md>

<https://kube-vip.io/control-plane/>

<https://codelabs.developers.google.com/codelabs/cloud-mongodb-statefulset/index.html?index=..%2F..index#4>