

# Setting up kubernetes on Rocky Linux

## Install kubernetes cluster and nginx

### What is needed and how they will be used

We will be using 2 control plan nodes, 4 worker/agent nodes and eventually 3 nodes for storage. One will also be used as a reverse proxy using Nginx.

The DNS server that will be used is currently running on a raspberry pi, it is running PiHole and has a local DNS server.

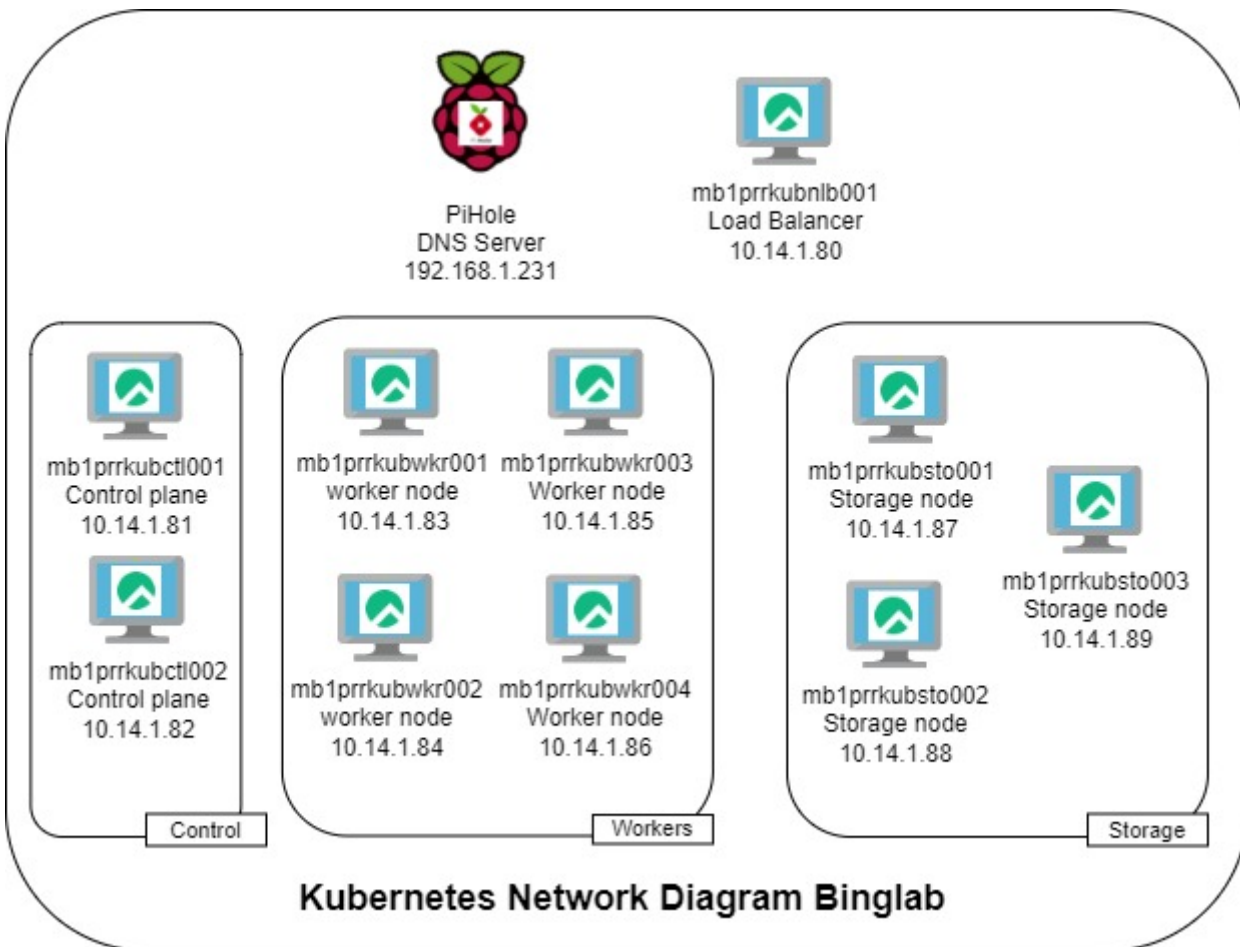
This is an initial install of Kubernetes on Rocky Linux 8.5.

- 1 x Nginx proxy running Rocky Linux 8.5, 2CPU, 4 GB RAM, 32 GB disk
- 2 x VM's for control plane Rocky Linux 8.5, 2CPU, 4 GB RAM, 32 GB disk
- 4 x VM's for Worker nodes Rocky Linux 8.5, 2CPU, 4 GB RAM, 32 GB disk
- 3 x VM's for storage nodes Rocky Linux 8.5, 2CPU, 4 GB RAM, 32 GB disk

Hostname	IP Address	Description
nlb.binglab.lan	10.14.1.80	Load balancer
mb1prrkubctl001.binglab.lan	10.14.1.81	Kubernetes Control plane
mb1prrkubctl002.binglab.lan	10.14.1.82	Kubernetes Control plane
mb1prrkubwkr001.binglab.lan	10.14.1.83	Kubernetes Worker node
mb1prrkubwkr002.binglab.lan	10.14.1.84	Kubernetes Worker node
mb1prrkubwkr003.binglab.lan	10.14.1.85	Kubernetes Worker node

mb1prrkubwkr004.binglab.lan	10.14.1.86	Kubernetes Worker node
mb1prrkubsto001.binglab.lan	10.14.1.87	Kubernetes Storage node
mb1prrkubsto002.binglab.lan	10.14.1.88	Kubernetes Storage node
mb1prrkubsto003.binglab.lan	10.14.1.89	Kubernetes Storage node

## Network Diagram



## Software/Stack used

- Rocky Linux 8.5
- kubelet
- Kubectl
- Kubernetes-cni
- docker-ce

1 First install the load balancer

1.1 Install package

```
yum install nginx
```

## 1.2 configure Nginx

```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

stream {
    upstream k3s_servers {
        server 10.14.1.81:6443;
        server 10.14.1.82:6443;
    }
    server {
        listen 6443;
        proxy_pass k3s_servers;
    }
}
```

## 2 Install Docker on the other nodes 6 for now

```
yum update
yum install -y yum-utils
sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
yum remove docker docker-client docker-client-latest docker-common docker-latest docker-
latest-logrotate docker-logrotate docker-engine
yum install docker-ce docker-ce-cli containerd.io
systemctl start docker
```

```
systemctl enable docker
systemctl status docker
docker ps
```

### 3 Install Kubernetes but first we need to add the repo on all the nodes

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-`basearch`
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
exclude=kubelet kubeadm kubectl
EOF
```

Once the repo is added we need to install the kubernetes packages: on all the nodes

```
dnf install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
systemctl enable --now kubelet
echo "KUBELET_EXTRA_ARGS=--cgroup-driver=cgroupfs" | sudo tee /etc/sysconfig/kubelet
systemctl restart kubelet
systemctl status kubelet
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee /etc/sysctl.d/k8s-iptables.conf
echo "net.bridge.bridge-nf-call-ip6tables=1" | sudo tee /etc/sysctl.d/k8s-ip6tables.conf
sysctl --system
```

### 4 Install the first control plain node of cluster

```
kubeadm init --kubernetes-version "1.23.4" --pod-network-cidr "192.168.1.0/16" --service-dns-
domain "apps.binglab.lan" --control-plane-endpoint "mb1prrkubnlb001.binglab.lan:6443" --
upload-certs
```

---

Revision #1

Created 14 March 2022 02:30:36 by mbtechadmin

Updated 14 March 2022 02:30:36 by mbtechadmin