

Kubernetes Installation

```
subscription-manager register --force --username qa@redhat.com --password  
MTQj5t3n5K86p3gH --auto-attach
```

1. `sudo yum remove podman`
2. `sudo yum install -y yum-utils`
3. `sudo yum-config-manager \`
 `> --add-repo \`
 `> sudo yum-config-manager --add-repo`
 `https://download.docker.com/linux/centos/docker-ce.repo`
4. `sudo yum install docker-ce docker-ce-cli containerd.io`
5. `sudo systemctl start docker.service`
6. `systemctl start firewalld`
7. `swapoff -a`
8. `firewall-cmd --permanent --add-port=6443/tcp`
9. `firewall-cmd --permanent --add-port=10250/tcp`
10. `cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf`
 `net.bridge.bridge-nf-call-ip6tables = 1`
 `net.bridge.bridge-nf-call-iptables = 1`
 `EOF`
11. `sudo sysctl --system`
12. `cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo`
 `[kubernetes]`
 `name=Kubernetes`
 `baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7- $\$ $basearch`
 `enabled=1`
 `gpgcheck=1`
 `repo_gpgcheck=1`
 `gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg`
 `https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg`
 `exclude=kubelet kubeadm kubectl`
 `EOF`
13. `sudo setenforce 0`
14. `sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config`
15. `yum install -y yum-utils device-mapper-persistent-data lvm2`
16. `cat /etc/docker/daemon.json`
 `{`
 `"exec-opts": ["native.cgroupdriver=systemd"],`
 `"log-driver": "json-file",`
 `"log-opts": {`
 `"max-size": "100m"`
 `},`
 `"storage-driver": "overlay2",`
 `"storage-opts": [`

```

        "overlay2.override_kernel_check=true"
    ]
}
17. mkdir -p /etc/systemd/system/docker.service.d
18. systemctl daemon-reload
19. systemctl restart docker
20. systemctl enable docker
21. systemctl status docker
22. sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
23. sudo systemctl enable --now kubelet
24. kubeadm init --pod-network-cidr 10.244.0.0/16 --apiserver-advertise-address=<Ip
    address of controller node>

```

Output:

To start using your cluster, you need to run the following as a regular user:

```

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

```

Alternatively, if you are the root user, you can run:

```

export KUBECONFIG=/etc/kubernetes/admin.conf

```

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

<https://kubernetes.io/docs/concepts/cluster-administration/addons/>

Then you can join any number of worker nodes by running the following on each as root:

```

kubeadm join 10.8.129.210:6443 --token bda59n.7th1y04t7bo4143n \
    --discovery-token-ca-cert-hash
sha256:e501cfb6f512d22fecb0b7ad012b17f70dffab6d2e12866de689e79d3e5d37a9
[root@bruuni010 docker]#

```

```

25. mkdir -p $HOME/.kube
26. sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
27. sudo chown $(id -u):$(id -g) $HOME/.kube/config
28. kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version |
    base64 | tr -d '\n')"
29. yum install bash-completion
30. echo "source <(kubectl completion bash)" >> ~/.bashrc

```

On Worker node:

```
kubeadm join 10.8.129.210:6443 --token bda59n.7th1y04t7bo4143n
--discovery-token-ca-cert-hash
sha256:e501cfb6f512d22fecb0b7ad012b17f70dffab6d2e12866de689e79d3e5d37a9(Not
exactly refer after executing kubeadm init command)
```

Creating Cluster using Rook

1. git clone https://github.com/rook/rook.git
2. cd rook/deploy/examples
3. kubectl create -f crds.yaml -f common.yaml -f operator.yaml -f osd-env-override.yaml
4. Modify the cluster-test.yaml
image: quay.io/ceph-ci/ceph:<sha1-crimson>
Example:
quay.io/ceph-ci/ceph:29e1fc1722aa5915b44828a5ad02ec45ce760aa3-crimson
5. kubectl create -f cluster-test.yaml
6. kubectl create -f toolbox.yaml

References:

Kubernetes Installation ref:-

https://www.youtube.com/watch?v=E3h8_MJmkVU&t=517s

<https://github.com/tunetolinux/Kubernetes-Installation/wiki>

Useful commands:

```
kubectl get node
kubectl get pod --all-namespaces
kubectl describe
```