



# Kubernetes installation help for Centos7

## Step 1: Disable SELinux & setup firewall rules.

Login to your kubernetes master node and set the hostname and disable selinux using following commands

```
~]# hostnamectl set-hostname 'k8s-master'

~]# exec bash

~]# setenforce 0

~]# sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g'
/etc/sysconfig/selinux
```

Set the following firewall rules.

```
[root@kubemaster ~]# firewall-cmd --permanent --add-port=6443/tcp

[root@kubemaster ~]# firewall-cmd --permanent --add-port=2379-2380/tcp

[root@kubemaster ~]# firewall-cmd --permanent --add-port=10250/tcp

[root@kubemaster ~]# firewall-cmd --permanent --add-port=10251/tcp

[root@kubemaster ~]# firewall-cmd --permanent --add-port=10252/tcp

[root@kubemaster ~]# firewall-cmd --permanent --add-port=10255/tcp

[root@kubemaster ~]# firewall-cmd --reload

[root@kubemaster ~]# modprobe br_netfilter

[root@kubemaster ~]# echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
```

**Note:** In case you don't have your own dns server then update /etc/hosts file on master and worker nodes

```
192.168.1.30 k8s-master

192.168.1.40 worker-node1

192.168.1.50 worker-node2
```

## Step 2: Configure Kubernetes Repository



# Kubernetes installation help for Centos7

Kubernetes packages are not available in the default CentOS 7 & RHEL 7 repositories, Use below command to configure its package repositories.

```
[root@kubemaster ~]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo

> [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

[root@kubemaster ~]# yum -y update
```

## Step 3: Install Kubectl and Docker

Once the package repositories are configured, run the beneath command to install kubectl and docker packages.

```
[root@kubemaster ~]# yum install kubectl docker -y
```

Start and enable kubectl and docker service

```
[root@kubemaster ~]# systemctl restart docker && systemctl enable docker

[root@kubemaster ~]# systemctl restart kubelet && systemctl enable kubelet
```

**Below Step to be executed only on KubeMaster server.**



# Kubernetes installation help for Centos7

## Step 4: Initialize Kubernetes Master with 'kubeadm init'

Run the beneath command to initialize and setup kubernetes master.

```
[root@k8s-master ~]# kubeadm init
```

Output of above command would be something like below

```
[kubeadm] WARNING: starting in 1.8, tokens expire after 24 hours by default (if you require a
[certificates] Generated CA certificate and key.
[certificates] Generated API server certificate and key.
[certificates] API Server serving cert is signed for DNS names [k8s-master kubernetes kuberne
es.default.svc.cluster.local] and IPs [10.96.0.1 192.168.1.30]
[certificates] Generated API server kubelet client certificate and key.
[certificates] Generated service account token signing key and public key.
[certificates] Generated front-proxy CA certificate and key.
[certificates] Generated front-proxy client certificate and key.
[certificates] Valid certificates and keys now exist in "/etc/kubernetes/pki"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/controller-manager.conf"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/scheduler.conf"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/admin.conf"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/kubelet.conf"
[apiclient] Created API client, waiting for the control plane to become ready
[apiclient] All control plane components are healthy after 880.002703 seconds
[token] Using token: a3bd48.1bc42347c3b35851
[apiconfig] Created RBAC rules
[addons] Applied essential addon: kube-proxy
[addons] Applied essential addon: kube-dns
```

Your Kubernetes master has initialized **successfully!**

To start using your cluster, you need to run (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
```

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:  
<http://kubernetes.io/docs/admin/addons/>

You can now join any number of machines by running the following on each node  
as root:

```
kubeadm join --token a3bd48.1bc42347c3b35851 192.168.1.30:6443
```

```
[root@k8s-master ~]#
```

As we can see in the output that kubernetes master has been initialized successfully. Execute the beneath commands to use the cluster as root user.

```
[root@kubemaster ~]# mkdir -p $HOME/.kube
```

```
[root@kubemaster ~]# cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

```
[root@kubemaster ~]# chown $(id -u):$(id -g) $HOME/.kube/config
```



# Kubernetes installation help for Centos7

```
[root@kubemaster ~]# export kubever=$(kubectl version | base64 | tr -d '\n')

[root@kubemaster ~]# kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-
version=$(kubectl version | base64 | tr -d '\n')"
```

```
[root@kubemaster ~]# kubectl get nodes
```

## Execute below commands only on both Node VMs...

### Step 1: Disable SELinux & configure firewall rules on both the nodes

Before disabling SELinux set the hostname on the both nodes as 'kubenode1' and 'kubenode2' respectively

```
~]# setenforce 0

~]# sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g'
/etc/sysconfig/selinux

~]# firewall-cmd --permanent --add-port=10250/tcp

~]# firewall-cmd --permanent --add-port=10255/tcp

~]# firewall-cmd --permanent --add-port=30000-32767/tcp

~]# firewall-cmd --permanent --add-port=6783/tcp

~]# firewall-cmd --reload

~]# echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
```

### Step 2: Configure Kubernetes Repositories on both worker nodes

```
~]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo

> [kubernetes]
```



# Kubernetes installation help for Centos7

```
> 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
```

## Step 3: Install kubeadm and docker package on both nodes

```
[root@kubenode1 ~]# yum install kubeadm docker -y

[root@kubenode2 ~]# yum install kubeadm docker -y
```

Start and enable docker service

```
[root@kubenode1 ~]# systemctl restart docker && systemctl enable docker

[root@kubenode2 ~]# systemctl restart docker && systemctl enable docker
```

## Step 4: Now Join worker nodes to master node

To join worker nodes to Master node, a token is required.

After we initialized the Kubeadm on kubemaster, then in the output we get command and token. Copy that command and run on both nodes.

**Below command will help the nodes to join the Kubernetes cluster.**

```
kubeadm join 10.142.0.7:6443 --token ykqof2.h1pek7ffl07z87cv \
```



# Kubernetes installation help for Centos7

---

```
--discovery-token-ca-cert-hash  
sha256:46d5230915b73e8567bc1fedb4d22e1bb3c7d1fdeadb64d624308b0234979f7f
```

Here the **Token** and **certificate** values will change as per installation server.