**Step 1: Configure Kubernetes Repository**

Kubernetes packages are not available from official CentOS 7 repositories. This step needs to be performed on the Master Node, and each Worker Node you plan on utilizing for your container setup. Enter the following command to retrieve the Kubernetes repositories.

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

**Step 2: Install kubelet, kubeadm, and kubectl**

These 3 basic packages are required to be able to use Kubernetes. Install the following package(s) on each node:

sudo yum install -y kubelet kubeadm kubectl

systemctl enable kubelet

systemctl start kubelet

You have now successfully installed Kubernetes, [including its tools](https://phoenixnap.com/blog/kubernetes-tools) and basic packages.

Before deploying a cluster, make sure to set hostnames, configure the firewall, and kernel settings.

**Step 3: Set Hostname on Nodes**

To give a unique hostname to each of your nodes, use this command:

sudo hostnamectl set-hostname master-node

or

sudo hostnamectl set-hostname worker-node1

In this example, the master node is now named master-node, while a worker node is named worker-node1.

Make a host entry or DNS record to resolve the hostname for all nodes:

sudo vi /etc/hosts

With the entry:

192.168.1.10 master.phoenixnap.com master-node

192.168.1.20 node1. phoenixnap.com node1 worker-node

### Step 4: Configure Firewall

The nodes, containers, and pods need to be able to communicate across the cluster to perform their functions. Firewalld is enabled in CentOS by default on the front-end. Add the following ports by entering the listed commands.

On the Master Node enter:

sudo firewall-cmd --permanent --add-port=6443/tcp

sudo firewall-cmd --permanent --add-port=2379-2380/tcp

sudo firewall-cmd --permanent --add-port=10250/tcp

sudo firewall-cmd --permanent --add-port=10251/tcp

sudo firewall-cmd --permanent --add-port=10252/tcp

sudo firewall-cmd --permanent --add-port=10255/tcp

sudo firewall-cmd --reload

Enter the following commands on each worker node:

sudo firewall-cmd --permanent --add-port=10251/tcp

sudo firewall-cmd --permanent --add-port=10250/tcp

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

sudo firewall-cmd --permanent --add-port=10255/tcp

firewall-cmd --reload

**Step 5: Update Iptables Settings**

Set the net.bridge.bridge-nf-call-iptables to ‘1’ in your sysctl config file. This ensures that packets are properly processed by IP tables during filtering and port forwarding.

cat <<EOF > /etc/sysctl.d/k8s.conf

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

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

EOF

sysctl --system

### Step 6: Disable SELinux

The containers need to access the host filesystem. SELinux needs to be set to permissive mode, which effectively disables its security functions.

Use following commands to [disable SELinux](https://phoenixnap.com/kb/how-to-disable-selinux-on-centos-7):

sudo setenforce 0

sudo sed -i ‘s/^SELINUX=enforcing$/SELINUX=permissive/’ /etc/selinux/config

### Step 7: Disable SWAP

Lastly, we need to disable SWAP to enable the kubelet to work properly:

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

sudo swapoff -a

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

<https://kubernetes.io/docs/setup/production-environment/container-runtimes/>