Steps to Set Up Kubernetes Clusters environment on CentOS 7

System Prerequisites

1. **Download & Install CentOS 7 Min**
2. Creating a master node
3. **System requirements (Master)**
4. 2048 mb - RAM
5. 2 core processor
6. 20GB Disk (safer side)

7. The master node will not get initiated if the RAM and virtual CPU cores not sufficient

   Note: You don't have to worry if you are using VirtualBox. You can always increase it.

8. If you are using VirtualBox, make sure you have a single Network Interface and sometimes **Adapter 1** in VirtualBox does not work on **Bridged** Setting. You have to enable the second adapter and disable the first one. (Explained this in Video)

Overview

This document will help you to install the k8 cluster. The document is purely for learning purposes. The cluster installation will give you an idea about the installation process and what to check if something goes wrong.

Please follow the steps:

9. **Download Minimal ISO for OS**
   CentOS 7
   http://centos.excellmedia.net/7.9.2009/isos/x86_64/CentOS-7-x86_64-Minimal-2009.iso
10. **Complete the CentOS installation - Basic Installation**

11. **Setup static IP Address**

```
nmtui edit
```

Update system hostname (all system): vim /etc/hostname

Example:

```
 k8-master
```

Update hosts file (all node info): $vim /etc/hosts

Example:

192.168.55.10	k8-master1
192.168.55.20	K8-worker1
192.168.55.30	k8-worker2

Update system packages

```
$Sudo yum update
```

Turn-Off Swap (all nodes)

```
$Sudo swapoff -a
```

To permanently off the swap we need to comment the swap partition form /etc/fstab

```
$ vim /etc/fstab
```

Disable Firewall (all nodes)

```
$sudo systemctl stop firewalld
```

To permanently disable the firewall:

```
$systemctl disable firewalld
```

Install below mentioned packages (All nodes):

```
$yum install -y vim
$yum install -y git
```

Install docker-runtime (All nodes)

Here we will set up Docker repository :

```
$sudo yum install -y yum-utils
$sudo yum-config-manager \--add-repo
https://d–ownload.docker.com/linux/centos/docker-ce.repo
```

Install docker engine

```
$sudo yum install containerd.io
```

Install required packages:

```
$ kubeadm, kubelet, kubectl
```

Installing kubeadm | Kubernetes

Init cluster

```
$kubeadm init
```

Verify the cluster info

```
$kubectl cluster-info
```

Your Kubernetes control-plane has initialized successfully!

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:

```
$export KUBECONFIG=/etc/kubernetes/admin.conf
```

```
$ init --cri-socket unix:///run/cri-dockerd.sock
```

If the Join command token is expired or you forgot:

```
$kubeadm token create --print-join-command
```

Remove node from the cluster (--ignore-daemonsets will be used if you do not care about pods running on that node)

```
$kubectl drain <node hostname> --ignore-daemonsets
```

```
$kubectl delete <node hostname>
```

On the Node: kubeadm reset

Errors:

Kubeadm unknown service runtime.v1alpha2.RuntimeService

Solution:

```
$rm /etc/containerd/config.toml
$systemctl restart containerd
$kubeadm init
```

FileContent--proc-sys-net-ipv4-ip_forward]:

Solution:

```
$echo '1' > /proc/sys/net/ipv4/ip_forward
$modprobe bridge
$modprobe br_netfilter
```