**Kubernetes Installation on Bare-Metal**

The following is a step-by-step guide to install Kubernetes on a cluster of machines running *Debian 10* (Buster) using the ***Linux 4.19.0*** kernel.

Text before a yellow background serves as a friendly reminder.

Text before a light blue background indicates a choice.

### Check Availability of Container Runtime

Make sure ***docker*** (latest version) is installed.

If not, follow the instruction on [**Install Docker Engine on Ubuntu**](https://docs.docker.com/engine/install/ubuntu/)

### Verify Host Names

Check ***/etc/hosts*** for this content in addition to the *local host* entry:

**128.200.135.194 c16080100** **c16080100.ss2k.uci.edu**

**128.200.135.204 c16080102 c16080102.ss2k.uci.edu**

**128.200.135.203 c11100502 c11100502.ss2k.uci.edu**

**128.200.135.220 c13031900 c13031900.ss2k.uci.edu**

### Check Open Ports

*Kubernetes* ports (***179***/tcp, ***4789***/udp, ***5473***/tcp, ***443***/tcp, ***6443***/tcp, ***2379***/tcp, ***4149***/tcp, ***10250***/tcp, ***10255***/tcp, 1***0256***/tcp, ***9099***/tcp, ***6443***/tcp) are not used, i.e. they are *open*.

This can be checked by running:  
**sudo netstat -tulpn**

### Disable Swapping

Disable *swapping* by running:  
**sudo swappoff -a**

To disable the swapping permanently, edit the ***/etc/fstab*** file and comment-out the line in **bold**-face:

UUID=6880a28d-a9dc-4bfb-ba47-0876b50e96b3 / ext4 errors=remount-ro 0 1

UUID=7350e6f2-e3a7-4d80-9a95-8741c7db118f /home ext4 defaults 0 2

UUID=E2E26AD1E26AAA0D /media/windows ntfs defaults,umask=007,gid=46 0 0

# Swap a usb extern (3.7 GB):

**#/dev/sdb1 none swap sw 0 0**

### Turn Off NetworkManager

Disable *network-manager* by running:  
**sudo systemctl disable network-manager**

and **reboot** the machine!

### Turn Off UFW

Disable *UFW* by running:  
**service ufw stop**

### Pre-Configure IP-Bridging

Check if installed:  
**sudo modprobe br\_netfilter**

Check on the current Kubernetes relevant ***IP-bridging*** configuration by viewing the contents of the file called **/etc/sysctl.d/kubernetes.conf**

If the file does not exist, create it with the following contents:

**net.bridge.bridge-nf-call-ip6tables = 1  
net.bridge.bridge-nf-call-iptables = 1  
net.ipv4.ip\_forward = 1**

To [re-]activate the new configuration run **sudo sysctl --system**.

### Reset System for Master Installation

In case it becomes necessary to uninstall and purge Kubernetes completely from a Debian Linux node follow these instructions:

kubeadm reset --force

sudo rm -rf /etc/cni /opt/cni /var/lib/cni /etc/kubernetes /var/lib/dockershim /var/lib/etcd /var/lib/kubelet /var/run/kubernetes

(the following *iptables* commands need to be executed as **root**)

iptables -F && iptables -X

iptables -t nat -F && iptables -t nat -X

iptables -t raw -F && iptables -t raw -X

iptables -t mangle -F && iptables -t mangle -X

(exit ***root*** mode)

sudo systemctl restart docker

### IP Packet Forwarding

Make sure ***br-netfilter***  is loaded:

**sudo modprobe br\_netfilter**

Ensure the proper bridging of IP packets by configuring the ***iptables***:

**cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf**

**br\_netfilter**

**EOF**

**cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf**

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

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

**EOF**

**sudo sysctl --system**

### Initialize Master Node

To deploy the Kubernetes *master* node using the ***kubeadm*** run the following command:

**sudo kubeadm –-config=/path-to-file/Kubeadm\_min-config.yaml**

**sudo kubeadm --apiserver-advertise-address=128.200.135.194 --pod-network-cidr=192.168.0.0/16 --service-cidr=10.98.0.0/16**

### Initialize Calico Networking

CALICO\_IPV4POOL\_CIDR

To install Calico v3.17.0 use kubectl and run:

**kubectl apply -f Calico-3.16.5\_OnPremise.yaml**

Apply the following changes:

* line ***3710***: enter value ‘**192.168.0.0/16**’ to match data in *Kubeadm* manifest file
* line ***3678***: enter NIC port value string “**interface=enp0\*, eth\***”

### To Add Worker Nodes to Cluster

For adding worker nodes, start by following the instructions in step **1.** through **8.**

Then use command output on the master node when it was initialized.

Alternatively run:  
**sudo kubeadm token create --print-join-command**

**Note**: Depending on what has been done prior, it may not hurt to execute  
**sudo kubeadm reset**