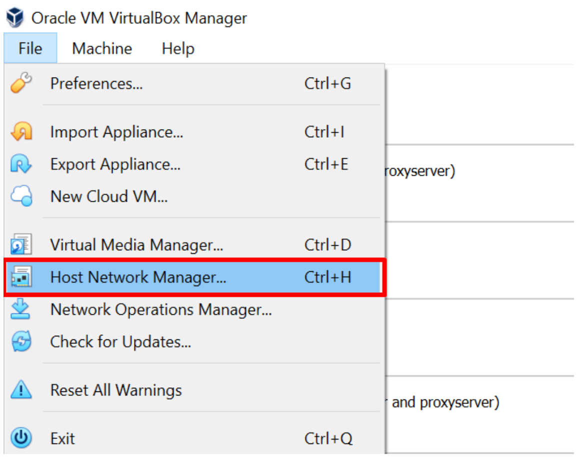
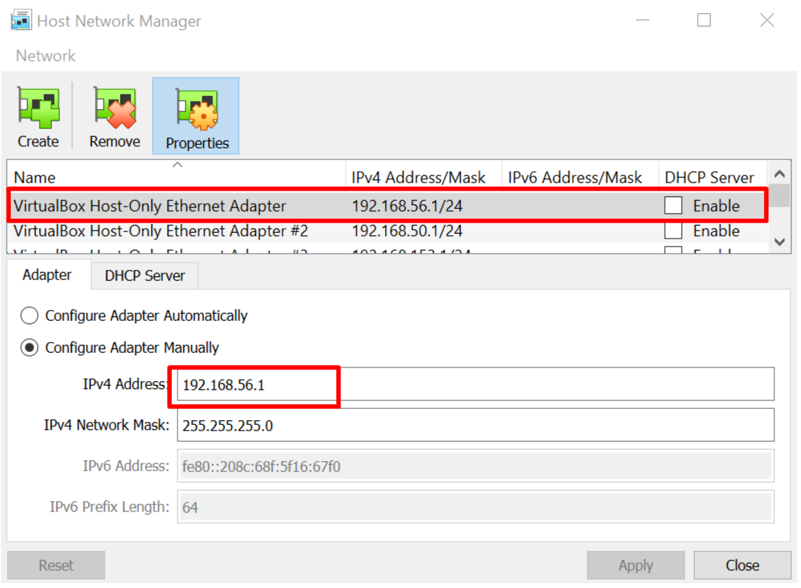
BareMetal Kubernetes cluster Setup on Windows

Prerequisites

- Hyper-V disabled on Windows Pro machines
- Admin rights
- VirtualBox installed
  https://www.virtualbox.org/wiki/Downloads
- Host network adapter in VirtualBox
  - ➢ Go to file -> Host Network Manager


Oracle VM VirtualBox Manager
File
Machine
Help
Preferences... Ctrl+G
Import Appliance... Ctrl+I
Export Appliance... Ctrl+E
New Cloud VM...
Virtual Media Manager... Ctrl+D
Host Network Manager... Ctrl+H
Network Operations Manager...
Check for Updates...
Reset All Warnings
Exit Ctrl+Q

  - ➢ By default, at least one host-only adapter should be available. If not create one using the create button
  - ➢ **Turn off DCHP server** and note down gateway IPv4 address that is assigned to the adapter. We are going to use this subnet IPs to assign static IPs to VMs. In my case, the gateway IP is 192.168.56.1
  - ➢ Click close once done



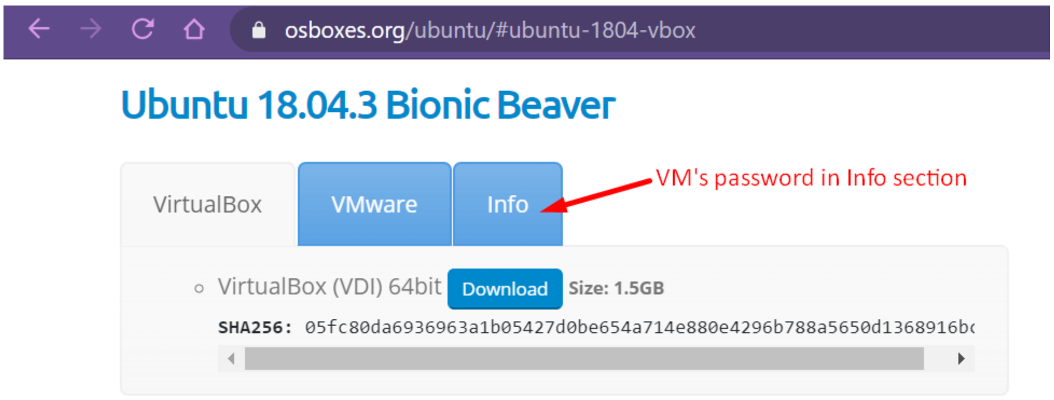
Name	IPv4 Address/Mask	IPv6 Address/Mask	DHCP Server
VirtualBox Host-Only Ethernet Adapter	192.168.56.1/24		☐ Enable
VirtualBox Host-Only Ethernet Adapter #2	192.168.50.1/24		☐ Enable

( ) Configure Adapter Automatically
(●) Configure Adapter Manually

IPv4 Address: 192.168.56.1
IPv4 Network Mask: 255.255.255.0
IPv6 Address: fe80::208c:68f:5f16:67f0
IPv6 Prefix Length: 64

Preparing Master node VM

1. Download Ubuntu 18.04.3 Bionic Beaver(preferred) VirtualBox image
   https://www.osboxes.org/ubuntu/#ubuntu-1804-vbox
   Password for this VM will be in the info section as shown below. Password is usually **osboxes.org**



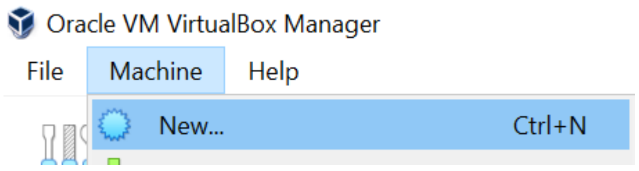
osboxes.org/ubuntu/#ubuntu-1804-vbox

Ubuntu 18.04.3 Bionic Beaver

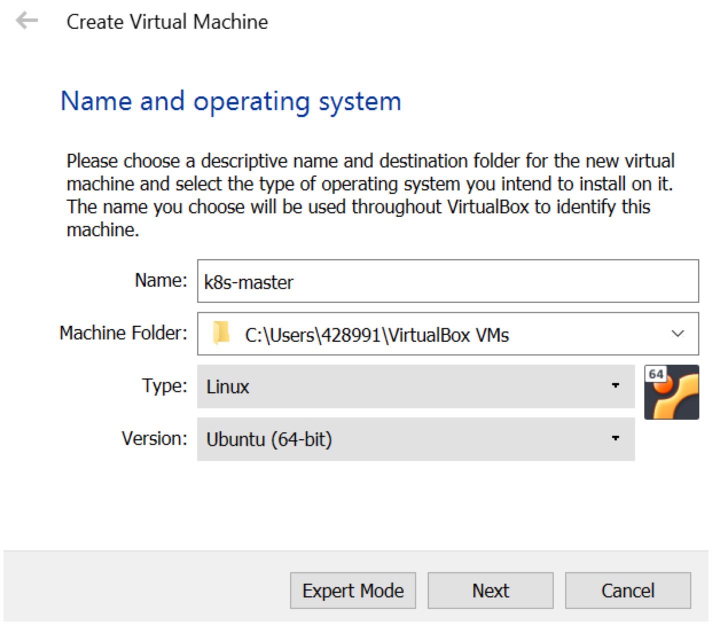
VirtualBox (VDI) 64bit Download Size: 1.5GB

SHA256: 05fc80da6936963a1b05427d0be654a714e880e4296b788a5650d1368916b

2. Once downloaded, extract the image
3. Create a new VM in VirtualBox
   Go to Machine -> New



Give a name to your VM and select options as in screenshot and click Next
I have chosen name as k8s-master as this node will be acting as **Kubernetes master**



Create Virtual Machine

Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

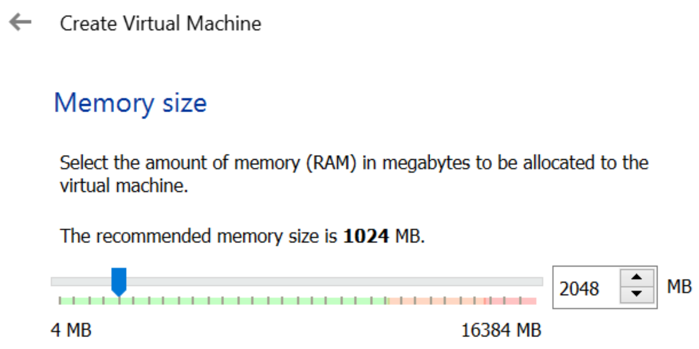
Name: k8s-master

Machine Folder: C:\Users\428991\VirtualBox VMs

Type: Linux

Version: Ubuntu (64-bit)

4. Select RAM size as 2048



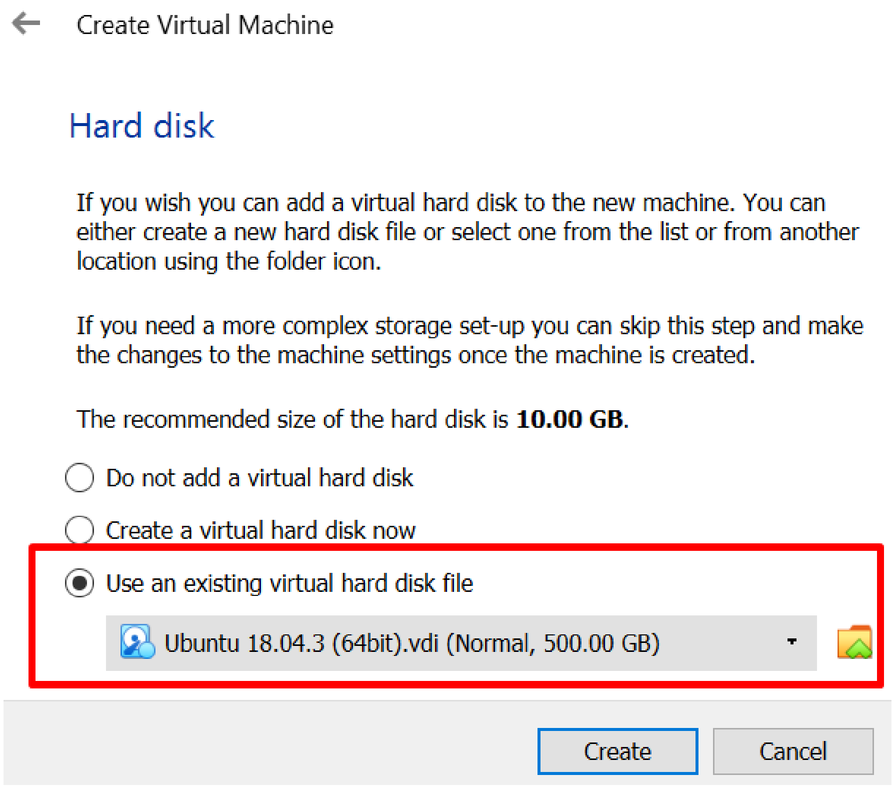
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

2048 MB

5. Choose the image that is downloaded and extracted from osboxes.org website and click create



Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

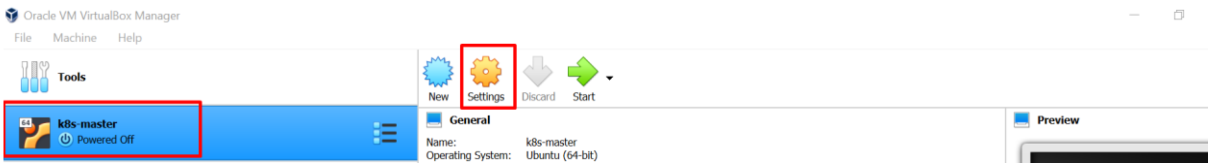
If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **10.00 GB**.

- ○ Do not add a virtual hard disk
- ○ Create a virtual hard disk now
- ● Use an existing virtual hard disk file

Ubuntu 18.04.3 (64bit).vdi (Normal, 500.00 GB)

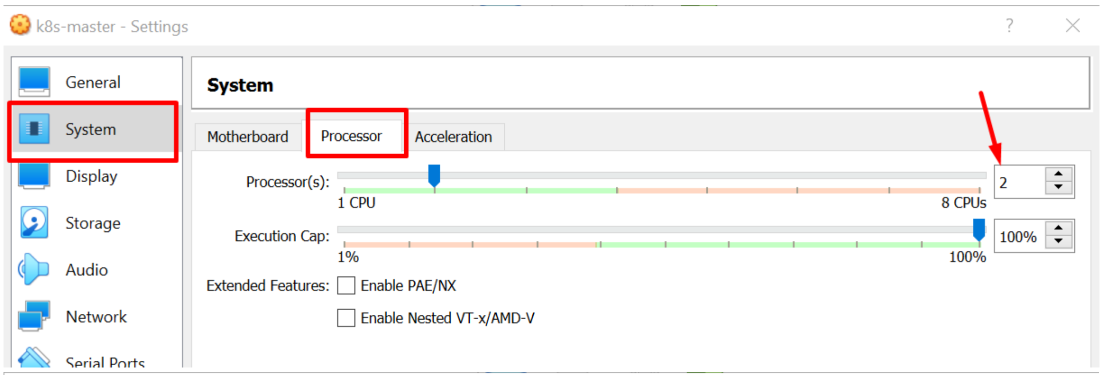
6. Once VM is created, we need to do some additional changes in the settings before running it. Go to settings after selecting the image



k8s-master
Powered Off

General
Name: k8s-master
Operating System: Ubuntu (64-bit)

7. Make changes as shown in the screenshots below and click close/ok.
(We will have 2 network adapters connected to the VM and hence the following changes)

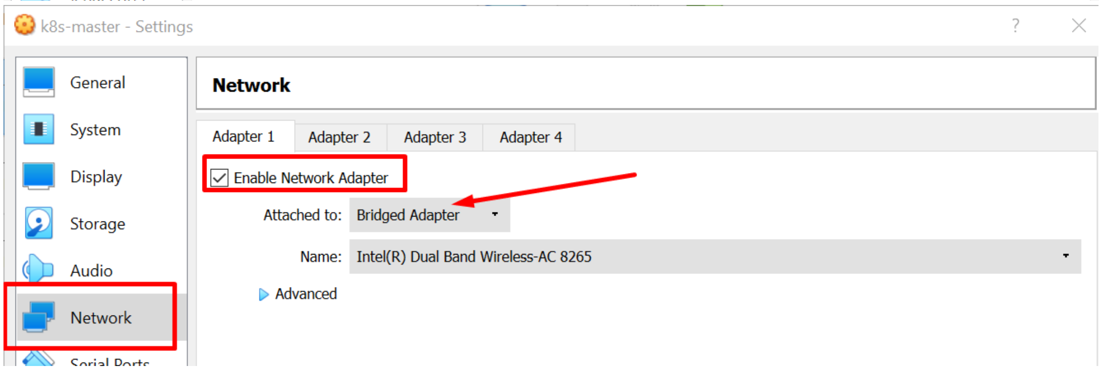

k8s-master - Settings

System

Processor(s): 2

Execution Cap: 100%

Extended Features: ☐ Enable PAE/NX ☐ Enable Nested VT-x/AMD-V


k8s-master - Settings

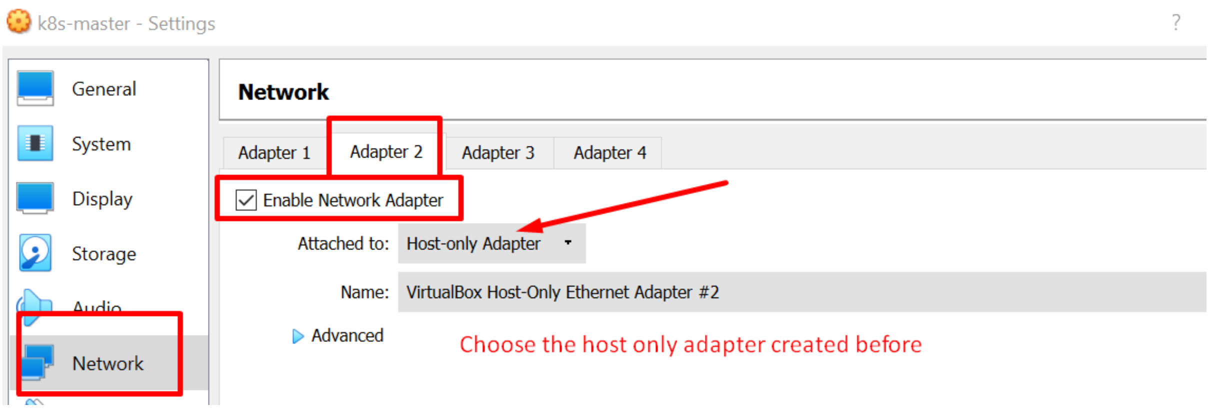
Network

Adapter 1

☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: Intel(R) Dual Band Wireless-AC 8265


k8s-master - Settings

Network

Adapter 2

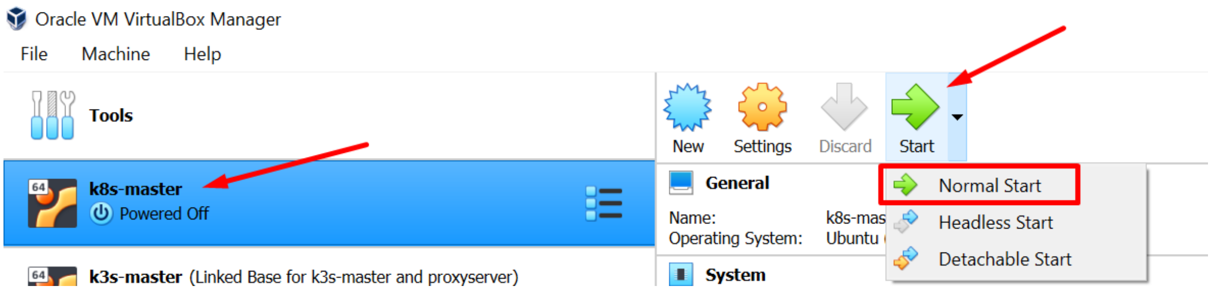
☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter #2

Choose the host only adapter created before

8. Now start the VM in normal mode(GUI mode)


Oracle VM VirtualBox Manager

k8s-master Powered Off

k3s-master (Linked Base for k3s-master and proxyserver)

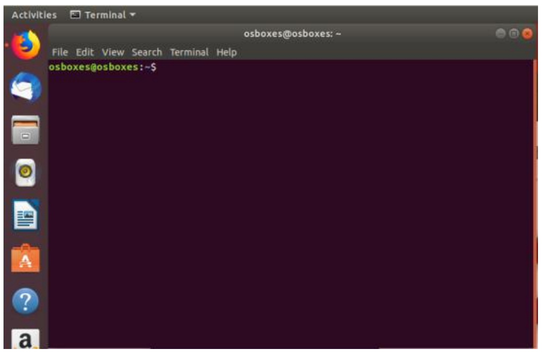
General

Name: k8s-mas

Operating System: Ubuntu

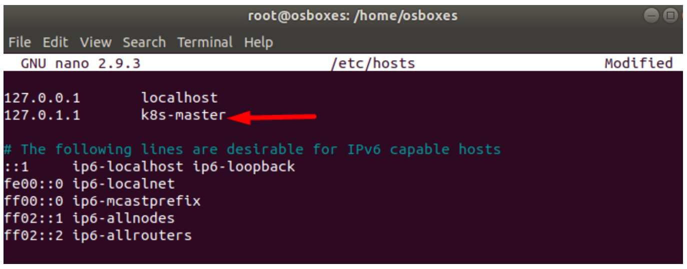
Normal Start

9. Once GUI is available, login using password(**osboxes.org** in most cases)
10. Open terminal in the Ubuntu VM and become root user



```
osboxes@osboxes:~$
```

11. Change hostname and hosts file
    - **nano /etc/hostname** -> change hostname to **k8s-master** (because this VM will be used as master, choose your own name of choice)
    - Update hosts file with new hostname: **nano /etc/hosts**



```
  GNU nano 2.9.3                    /etc/hosts                        Modified

127.0.0.1       localhost
127.0.1.1       k8s-master

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

12. Update and add related packages
    - **apt update -y**
    - **apt install net-tools**
    - **apt install openssh-server -y**

13. Adding static IP to interface

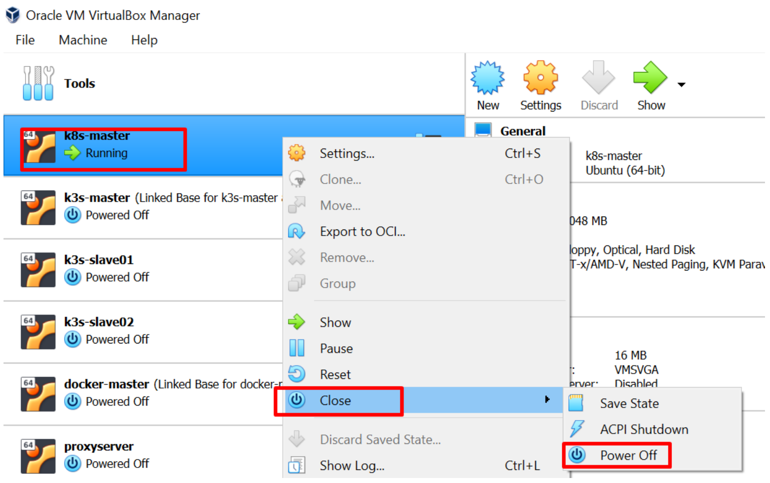
    This is required as VMs change their IP after every reboot(most of the time). We cannot use these dynamic IPs to start the cluster as worker node fails to find master every time master node changes its IP. In order to avoid starting the cluster every time with a new IP it's better to have a fixed IP set to an Interface and have the cluster running forever. In this case we will use ***enp0s8*** as the interface with fixed IP.

    Also, we will choose static IPs from host-adapter subnet range created before.
    - **ifconfig enp0s8 192.168.56.2**
    - **nano /etc/network/interfaces** and append these lines

```
auto enp0s8
iface enp0s8 inet static
        address 192.168.56.2
        netmask 255.255.255.0
```

Note the IP 192.168.56.2 is used for master. Going forward we will use 192.168.56.3 for worker node 01(k8s-slave01) and 192.168.56.4 for worker node 02(k8s-slave02) and so on…

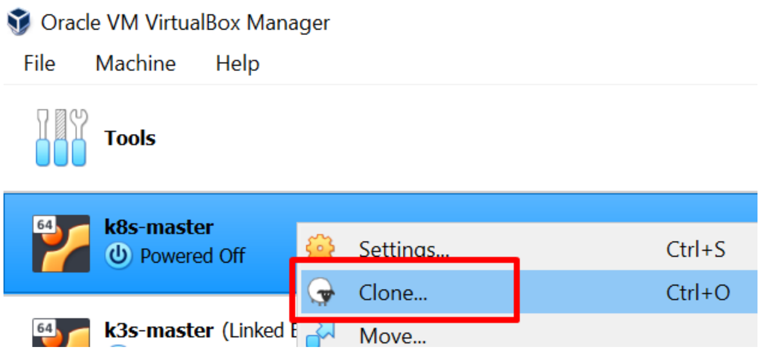
14. Installing Docker. Run these commands individually as root user
    - **sudo apt update**
    - **sudo apt install apt-transport-https ca-certificates curl software-properties-common**
    - **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -**
    - **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"**
    - **sudo apt update**
    - **apt-cache policy docker-ce**
    - **sudo apt install docker-ce**
    - **sudo systemctl status docker**
15. Check if docker is installed properly by running **docker version**
16. Shut down the VM by doing **init 0** in the terminal or from VirtualBox itself by



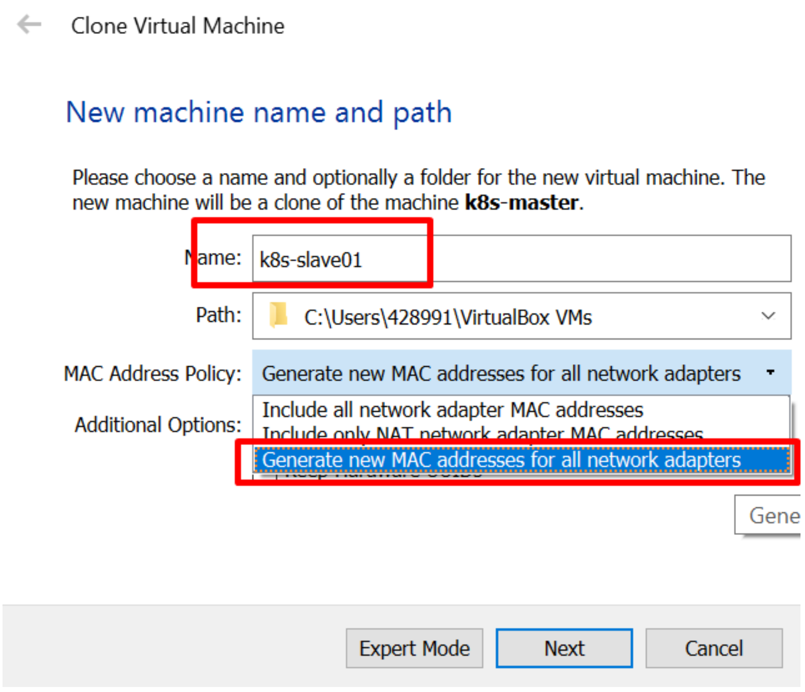
Preparing Worker node 01 VM

We will clone the master node for slave nodes as this will save significant amount of disk space on the host.

1. Cloning the master node
   Make sure master node is powered off. Right click on master VM and select clone



2. Rename the VM to **k8s-slave01**, select the appropriate options as shown in screenshots and click Next



Clone Virtual Machine

New machine name and path

Please choose a name and optionally a folder for the new virtual machine. The new machine will be a clone of the machine **k8s-master**.

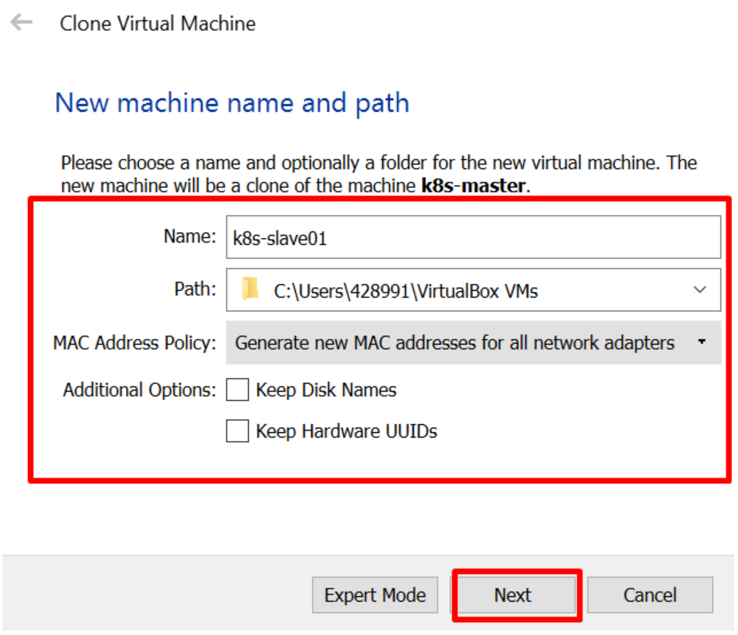
Name: k8s-slave01

Path: C:\Users\428991\VirtualBox VMs

MAC Address Policy: Generate new MAC addresses for all network adapters

Additional Options:

Include all network adapter MAC addresses
Include only NAT network adapter MAC addresses
Generate new MAC addresses for all network adapters



Clone Virtual Machine

New machine name and path

Please choose a name and optionally a folder for the new virtual machine. The new machine will be a clone of the machine **k8s-master**.

Name: k8s-slave01

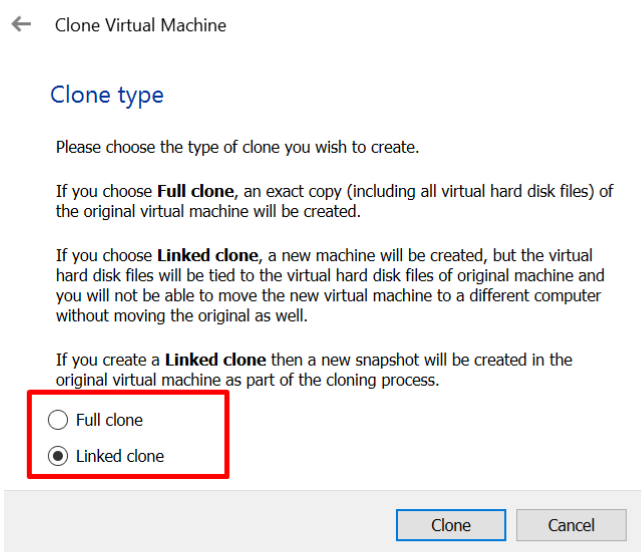
Path: C:\Users\428991\VirtualBox VMs

MAC Address Policy: Generate new MAC addresses for all network adapters

Additional Options: ☐ Keep Disk Names

☐ Keep Hardware UUIDs

3. Select Linked clone and click clone



Clone Virtual Machine

Clone type

Please choose the type of clone you wish to create.

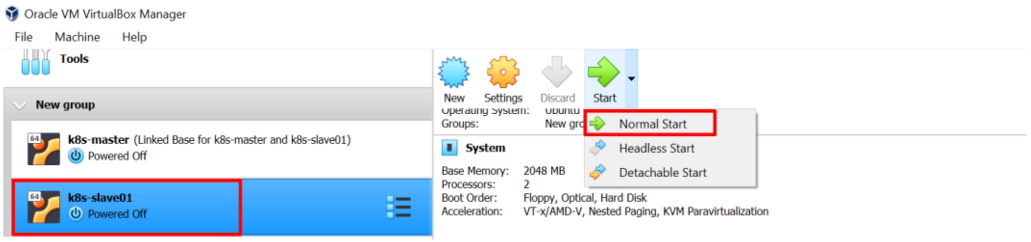
If you choose **Full clone**, an exact copy (including all virtual hard disk files) of the original virtual machine will be created.

If you choose **Linked clone**, a new machine will be created, but the virtual hard disk files will be tied to the virtual hard disk files of original machine and you will not be able to move the new virtual machine to a different computer without moving the original as well.

If you create a **Linked clone** then a new snapshot will be created in the original virtual machine as part of the cloning process.

Full clone [✓ Linked clone]

4. Select the slave VM and start it in GUI mode



New group

k8s-master (Linked Base for k8s-master and k8s-slave01)
Powered Off

k8s-slave01
Powered Off

Operating System: Ubuntu
Groups: New gro

System

Base Memory: 2048 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

5. Repeat Steps 11 & 13 used in setting up **master VM**. But this time use
   K8s-slave01 instead of k8s-master
   192.168.56.3 instead of 192.168.56.2
   Basically, we must change the hostname and hosts file with name of the slave chosen and giving static IP to VM which is different from static IP assigned to master
6. Shutdown this VM

Preparing Worker node 02 VM

Same steps as in **Preparing Worker node 01 VM** but use hostname as **k8s-slave02** and static IP as **192.168.56.4**

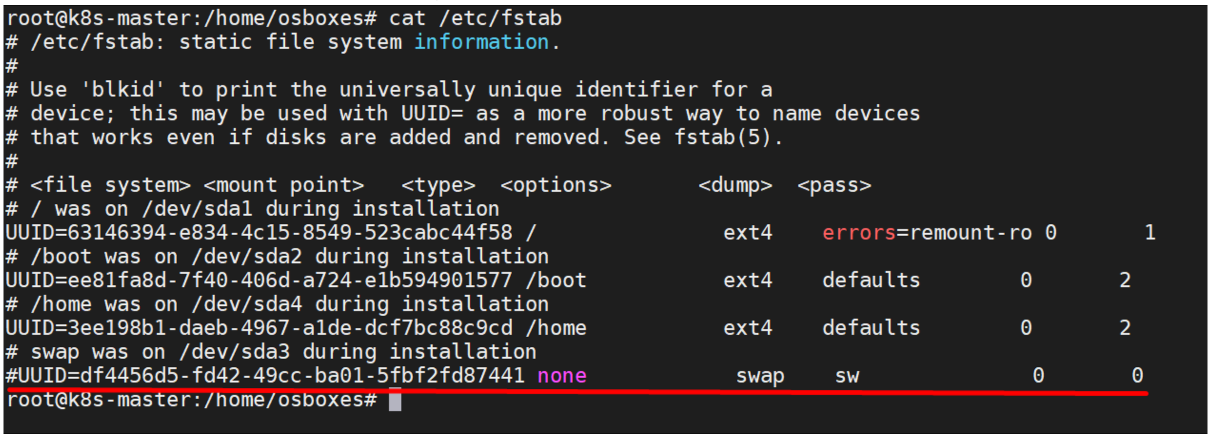
What is done so far…

- We have configured 3 VMs with names k8s-master, k8s-slave01 and k8s-slave02
- We have given them static IPs 192.168.56.2, 192.168.56.3 and 192.168.56.4
- Since docker is installed on master and slaves are nothing but clones of master, docker should be available on all VMs

Bootstrapping the cluster

1. Start all 3 VMs
2. Start terminal on all VMs and become root

3. Run below commands on all 3 VMs(prefer opening multiple putty sessions and running these commands on all nodes at once).
   These commands include turning off swap memory and installing Kubernetes components
   - **swapoff -a**
   - **nano /etc/fstab** (comment out swap line)



```
root@k8s-master:/home/osboxes# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=63146394-e834-4c15-8549-523cabc44f58 /               ext4    errors=remount-ro 0       1
# /boot was on /dev/sda2 during installation
UUID=ee81fa8d-7f40-406d-a724-e1b594901577 /boot           ext4    defaults        0       2
# /home was on /dev/sda4 during installation
UUID=3ee198b1-daeb-4967-a1de-dcf7bc88c9cd /home           ext4    defaults        0       2
# swap was on /dev/sda3 during installation
#UUID=df4456d5-fd42-49cc-ba01-5fbf2fd87441 none            swap    sw              0       0
root@k8s-master:/home/osboxes#
```

   - **sudo apt-get update && sudo apt-get install -y apt-transport-https curl**
   - **curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -**
   - **cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list**
     deb https://apt.kubernetes.io/ kubernetes-xenial main
     EOF
   - **sudo apt-get update**
   - **sudo apt-get install -y kubelet kubeadm kubectl**
   - **sudo apt-mark hold kubelet kubeadm kubectl**

4. Run these commands ONLY ON MASTER NODE(k8s-master) as root user
   - **kubeadm config images pull**
   - **kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=192.168.56.2**
     Please note 192.168.56.2 is the static IP assigned to master while master VM is being setup. If you have assigned any other IP, replace it in the command
     Once above command is executed, a token of the form (kubeadm join 192.168.56.2:6443 --token oqp357.wexmezpiszo5zeeg \
       --discovery-token-ca-cert-hash
     sha256:67577a18d7abcd1815cf5086e2196dadf5ac39ab3b18fb705694fdafac84580b)
     will be generated. save this token for later use
   - mkdir -p $HOME/.kube
   - sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
   - sudo chown $(id -u):$(id -g) $HOME/.kube/config
     The above 3 commands will also be shown in the terminal once kubeadm init …. command is run
5. Choose a CNI plugin for the cluster. For VirtualBox ‘flannel’ CNI is advised. You can also choose other CNIs(refer References section of this document for more info)

6. Run this ONLY ON MASTER as root(basically applying cluster networking interface plugin)
   kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
7. Paste the token, that is saved earlier from step 4 in this section, on all worker nodes(k8s-slave 01 & k8s-slave02). This should join them to the cluster
8. Run **kubectl get nodes** on MASTER to see the list of nodes in the cluster and their status

References

- https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/
- https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/