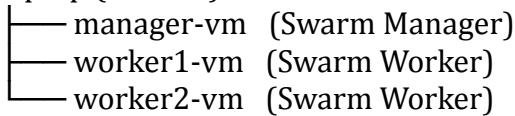


Docker swarm cluster setup on Ubuntu laptop using VirtualBox + Vagrant:

Laptop (Ubuntu)



1. Install VirtualBox (Host Ubuntu)

```
sudo apt update  
sudo apt install -y virtualbox
```

Verify:

```
vboxmanage --version
```

a. Add HashiCorp GPG key

```
curl -fsSL https://apt.releases.hashicorp.com/gpg | \  
sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
```

b. Add HashiCorp repository

```
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \  
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \  
sudo tee /etc/apt/sources.list.d/hashicorp.list
```

c. Update & install Vagrant

```
sudo apt update  
sudo apt install -y vagrant
```

d. Verify

```
vagrant --version
```

output:

Vagrant 2.x.x

2. Create Project Directory

```
mkdir docker-swarm-vms  
cd docker-swarm-vms
```

3. Create Vagrantfile (3 VMs)

```
vagrant init
```

Edit the file:

```
nano Vagrantfile
```

Replace entire content with this

```
Vagrant.configure("2") do |config|  
  config.vm.box = "ubuntu/jammy64"
```

```

# Swarm Manager
config.vm.define "manager-vm" do |manager|
  manager.vm.hostname = "manager-vm"
  manager.vm.network "private_network", ip: "192.168.56.10"
  manager.vm.provider "virtualbox" do |vb|
    vb.memory = 2048
    vb.cpus = 2
  end
end

# Worker 1
config.vm.define "worker1-vm" do |worker1|
  worker1.vm.hostname = "worker1-vm"
  worker1.vm.network "private_network", ip: "192.168.56.11"
  worker1.vm.provider "virtualbox" do |vb|
    vb.memory = 1024
    vb.cpus = 1
  end
end

# Worker 2
config.vm.define "worker2-vm" do |worker2|
  worker2.vm.hostname = "worker2-vm"
  worker2.vm.network "private_network", ip: "192.168.56.12"
  worker2.vm.provider "virtualbox" do |vb|
    vb.memory = 1024
    vb.cpus = 1
  end
end

```

Save and exit (CTRL+O, ENTER, CTRL+X).

4. Create All 3 VMs (One Command)

vagrant up

if you get error:

Blacklist KVM modules

sudo nano /etc/modprobe.d/blacklist-kvm.conf

Add:

blacklist kvm

blacklist kvm_intel

blacklist kvm_amd

Save & exit.

Update initramfs and reboot

sudo update-initramfs -u

sudo reboot

After reboot:

lsmod | grep kvm

output: should show nothing.

Then retry

cd ~/docker-swarm-vms

vagrant up

final output should be

vagrant status

manager-vm running

worker1-vm running

worker2-vm running

Then:

vagrant ssh manager-vm

5. Login into Each VM

You are already in [**vagrant@manager-vm**](#) so goto step 6 directly

Manager

vagrant ssh manager-vm

to run worker1 open new terminal and type

cd ~/docker-swarm-vms

Worker 1:

vagrant ssh worker1-vm

to run worker2 open new terminal and type

cd ~/docker-swarm-vms

Worker 2:

vagrant ssh worker2-vm

6. Install Docker (Run on ALL VMs)

Inside **each VM**:

```
sudo apt update  
sudo apt install -y docker.io  
sudo systemctl enable docker  
sudo systemctl start docker  
sudo usermod -aG docker $USER  
logout
```

Re-login:

```
vagrant ssh <vm-name> // <vm-name> to be replaced with the actual VM name
```

Verify:

```
docker --version
```

7. Initialize Swarm (Manager VM)

On **manager-vm**:

```
docker swarm init --advertise-addr 192.168.56.10
```

Copy the **worker join token** and paste it in the workers respectively

Verify:

```
docker node ls
```

8. Join Workers to Swarm

On worker1-vm

```
docker swarm join \  
--token SWMTKN-1-xxxx \  
192.168.56.10:2377
```

On worker2-vm

```
docker swarm join \  
--token SWMTKN-1-xxxx \  
192.168.56.10:2377
```

9. Verify Full Cluster

On **manager-vm**:

```
docker node ls
```

Expected:

HOSTNAME	STATUS	MANAGER STATUS
manager-vm	Ready	Leader
worker1-vm	Ready	
worker2-vm	Ready	

10. Test with a Service

```
docker service create --name web --replicas 3 -p 8080:80 nginx  
docker service ps web
```

Access from host:

<http://192.168.56.10:8080>

11. Stop / Delete Everything

```
vagrant halt      # stop VMs  
vagrant destroy -f # delete VMs
```