VirtualBox and Vagrant (HashiCorp) Configuration Setup

1. Update System Packages

Before installing, always make sure your system packages are up-to-date:

```
- sudo apt update && sudo apt upgrade -y
```

2. Install Required Dependencies

Vagrant requires some dependencies like **curl**, **gnupg**, and **software-properties-common**. Install them using:

```
- sudo apt install -y curl gnupg software-properties-common
```

3. Install VirtualBox

Vagrant uses **VirtualBox** as a default provider. Install it as follows:

Step 3.1: Add VirtualBox Repository

```
- wget -q
  https://www.virtualbox.org/download/oracle_vbox_2016
  .asc -O- | sudo gpg --dearmor -o
  /usr/share/keyrings/oracle-virtualbox.gpg
- echo "deb
  [signed-by=/usr/share/keyrings/oracle-virtualbox.gpg
  ] https://download.virtualbox.org/virtualbox/debian
  jammy contrib" | sudo tee
  /etc/apt/sources.list.d/virtualbox.list
```

Step 3.2: Update Packages

```
- sudo apt update
```

Step 3.3: Install VirtualBox

```
- sudo apt install -y virtualbox-7.0
```

Step 3.4: Verify Installation

```
- virtualbox --help
```

- You should see VirtualBox version and usage information.

4. Install Vagrant

Step 4.1: Download Latest Vagrant Package

Check [Vagrant official downloads](#) for the latest `.deb` package. Then download it:

```
- wget
  https://releases.hashicorp.com/vagrant/3.5.0/vagrant
  _3.5.0_linux_amd64.deb
```

Replace `3.5.0` with the latest version if newer.

Step 4.2: Install the Package

```
- sudo dpkg -i vagrant_3.5.0_linux_amd64.deb
```

Step 4.3: Fix Dependencies (if any)

```
- sudo apt install -f -y
```

Step 4.4: Verify Vagrant Installation

```
- vagrant --version
```

- Example output: `Vagrant 3.5.0`

5. Optional: Install Vagrant Plugins

Some Vagrant setups require plugins like **vagrant-vbguest** to automatically manage VirtualBox Guest Additions:

```
- vagrant plugin install vagrant-vbguest
```

- Check installed plugins:

```
- vagrant plugin list
```

6. Basic Vagrant Commands

- Initialize a Vagrant project:

```
- vagrant init
```

- Start Vagrant VMs:

```
- vagrant up
```

- SSH into VM:

```
- vagrant ssh
```

- Stop VM:

```
- vagrant halt
```

- Destroy VM:

```
- vagrant destroy -f
```

Vagrant Kubernetes Lab Setup Documentation

This document describes a **Vagrant-based lab environment for Kubernetes** on **Ubuntu 24.04**.

It includes:

- **One Master Node**
- **One Worker Node**
- **Private networking**
- **Port forwarding for services**
- **Resource allocation (CPU & RAM)**

7. Vagrantfile Overview

- Vagrant is configured with **bento/ubuntu-24.04** box and **VirtualBox** provider.
- Each VM has:
  - **Static private IP**
  - **Hostname**
  - **Resource allocation (CPU & RAM)**
  - **Port forwarding for testing services**

8. Master Node Configuration

```
config.vm.define "MasterNode" do |master|
  master.vm.box = "bento/ubuntu-24.04"
  master.vm.hostname = "MasterNode"
  master.vm.network "private_network", ip: "192.168.56.10"

  # Port Forwarding (Guest → Host)
  master.vm.network "forwarded_port", guest: 80, host: 8080
  master.vm.network "forwarded_port", guest: 8080, host: 8081

  master.vm.provider "virtualbox" do |vb|
    vb.name = "Master Node"
    vb.memory = 4096
```

```
    vb.cpus = 2
  end
end
```

Explanation:

- **hostname:** Sets the VM's hostname.
- **private_network:** Assigns a static IP for communication with worker nodes.
- **forwarded_port:** Maps guest ports to host ports for accessing services.
- **provider:** Allocates **2 CPUs** and **4GB RAM** to ensure Kubernetes can run smoothly.

9. Worker Node Configuration

```
config.vm.define "WorkerNode" do |worker|
  worker.vm.box = "bento/ubuntu-24.04"
  worker.vm.hostname = "WorkerNode"
  worker.vm.network "private_network", ip: "192.168.56.11"

  # Port Forwarding
  worker.vm.network "forwarded_port", guest: 8080, host: 8090
  worker.vm.network "forwarded_port", guest: 80, host: 8091

  worker.vm.provider "virtualbox" do |vb|
    vb.name = "Worker Node"
    vb.memory = 4096
    vb.cpus = 2
  end
end
```
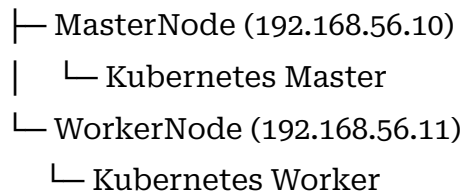
Explanation:

- Uses the **same Ubuntu box** as the Master node.
- Static private IP: **192.168.56.11**
- Port forwarding mapped to **different host ports** to avoid conflicts.
- **Resource allocation** is the same as Master node (**2 CPUs & 4GB RAM**).

10. Network Topology

Host Machine
```
├── MasterNode (192.168.56.10)
│     └── Kubernetes Master
└── WorkerNode (192.168.56.11)
      └── Kubernetes Worker
```

- Private network allows communication between Master & Worker.
- Port forwarding enables testing services from **host browser** or **curl**.

11. How to Bring Up the Vagrant Lab

1. Navigate to the directory with your **Vagrantfile**.

2. Run:
- vagrant up
- This will start both **Master** and **Worker nodes**.
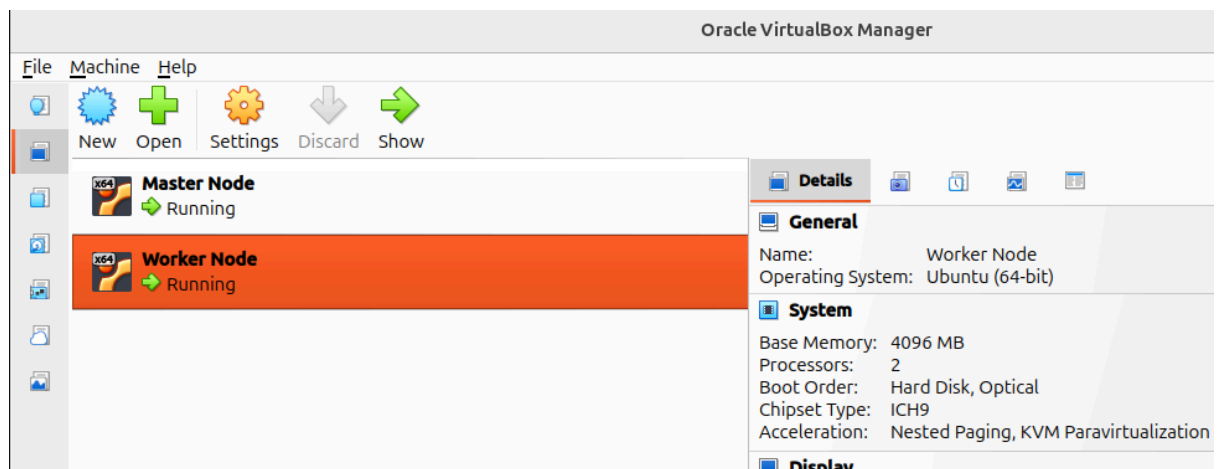
- vagrant status

3. SSH into nodes:

- vagrant ssh MasterNode
- vagrant ssh WorkerNode

```
hp@hp-HP-EliteBook-840-G1:~/Desktop/Kubernetes_Project$ vagrant ssh MasterNode
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-100-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Feb  9 02:35:07 PM UTC 2026

  System load:          1.71
  Usage of /:           29.6% of 30.34GB
  Memory usage:         23%
  Swap usage:           0%
  Processes:            191
  Users logged in:      1
  IPv4 address for eth0: 10.0.2.15
  IPv6 address for eth0: fd17:625c:f037:2:a00:27ff:fef8:c2eb


This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento

Use of this system is acceptance of the OS vendor EULA and License Agreements.
Last login: Mon Feb  9 04:38:28 2026 from 10.0.2.2
vagrant@MasterNode:~$ 
```

```
hp@hp-HP-EliteBook-840-G1:~/Desktop/Kubernetes_Project$ vagrant ssh WorkerNode
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-100-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Feb  9 02:36:00 PM UTC 2026

  System load:          0.03
  Usage of /:           26.3% of 30.34GB
  Memory usage:         13%
  Swap usage:           0%
  Processes:            188
  Users logged in:      1
  IPv4 address for eth0: 10.0.2.15
  IPv6 address for eth0: fd17:625c:f037:2:a00:27ff:fef8:c2eb


This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento

Use of this system is acceptance of the OS vendor EULA and License Agreements.
Last login: Mon Feb  9 04:38:40 2026 from 10.0.2.2
vagrant@WorkerNode:~$ hostname -I
10.0.2.15 192.168.56.11 172.17.0.1 192.168.190.192 fd17:625c:f037:2:a00:27ff:fef8:c2eb
vagrant@WorkerNode:~$ 
```

12. Recommended Resource Allocation

Node	CPU	RAM
Master	2	4GB
Worker	2	4GB