



VM Setup for Ansible Demo

Mukul Katiyar
PS No. 10671391



A Larsen & Toubro
Group Company

CONTEXT:

Ansible is an open-source IT engine which automates the IT tools such as intra service orchestration, application deployment, cloud provisioning, etc.

Ansible is easy to deploy because it does not use any **agents** or **custom security** infrastructure on the client-side, and by pushing modules to the clients. These modules are executed locally on the client-side, and the output is pushed back to the Ansible server. It can easily connect to clients using **SSH-Keys**, simplifying though the whole process. Client details, such as **hostnames** or **IP addresses** and **SSH ports**, are stored in the files, which are called inventory files. If you created an inventory file and populated it, then Ansible can use it.

Ansible uses the playbook to describe automation jobs, and playbook, which uses simple language, i.e., **YAML**. **YAML** is a human-readable data serialization language & commonly used for configuration files, but it can be used in many applications where data is being stored.

Ansible is designed for multi-tier deployment. Ansible does not manage one system at a time, and it models IT infrastructure by describing all of your systems are interrelated. Ansible is entirely agentless, which means Ansible works by connecting your nodes through **SSH** (by default). Ansible gives the option to you if you want another method for the connection like **Kerberos**.

Ansible pushes small programs after connecting to your nodes which are known as "**Ansible Modules**". Ansible runs that module on your nodes and removes them when finished. Ansible manages the inventory in simple text files (These are the host's files). Ansible uses the host file where one can group the hosts and can control the actions on a specific group in the playbooks.

However to view Ansible in all its glory and have a hands on experience with it we need to setup a lab environment simulating the server and node machines infrastructure Ansible works best in. The setup will have VMs running **Ubuntu** distributions of GNU/Linux and comprise 1 **WorkstationVM** running Ansible and 3 **Host VMs** which act as target machines for Ansible. This document goes into the step by step process of setting these VMs.

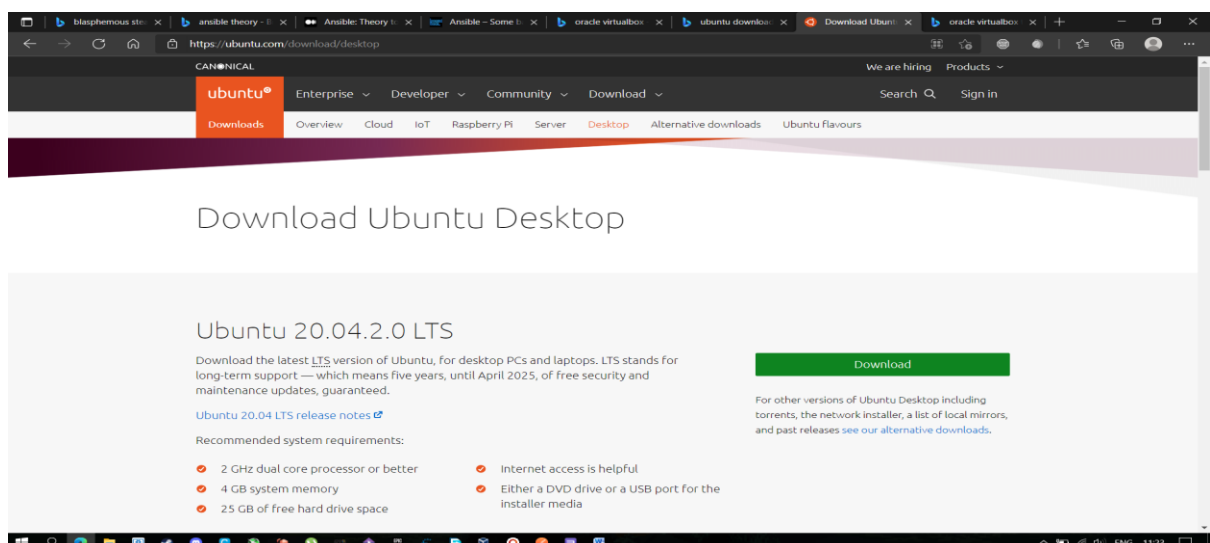
SOLUTION:

The setting up of VMs is done using **Oracle VirtualBox**. VirtualBox is a free and open-source software program for virtualizing the **x86** computing architecture. Oracle Corporation developed it. It works as a hypervisor and develops a Virtual Machine where the user can run another operating system. The "**host**" OS is the operating system where VirtualBox runs. The "**guest**" OS is the operating system running on the Virtual Machine. As the host OS, VirtualBox supports Windows, Linux, Solaris, Open Solaris, and MacOS.

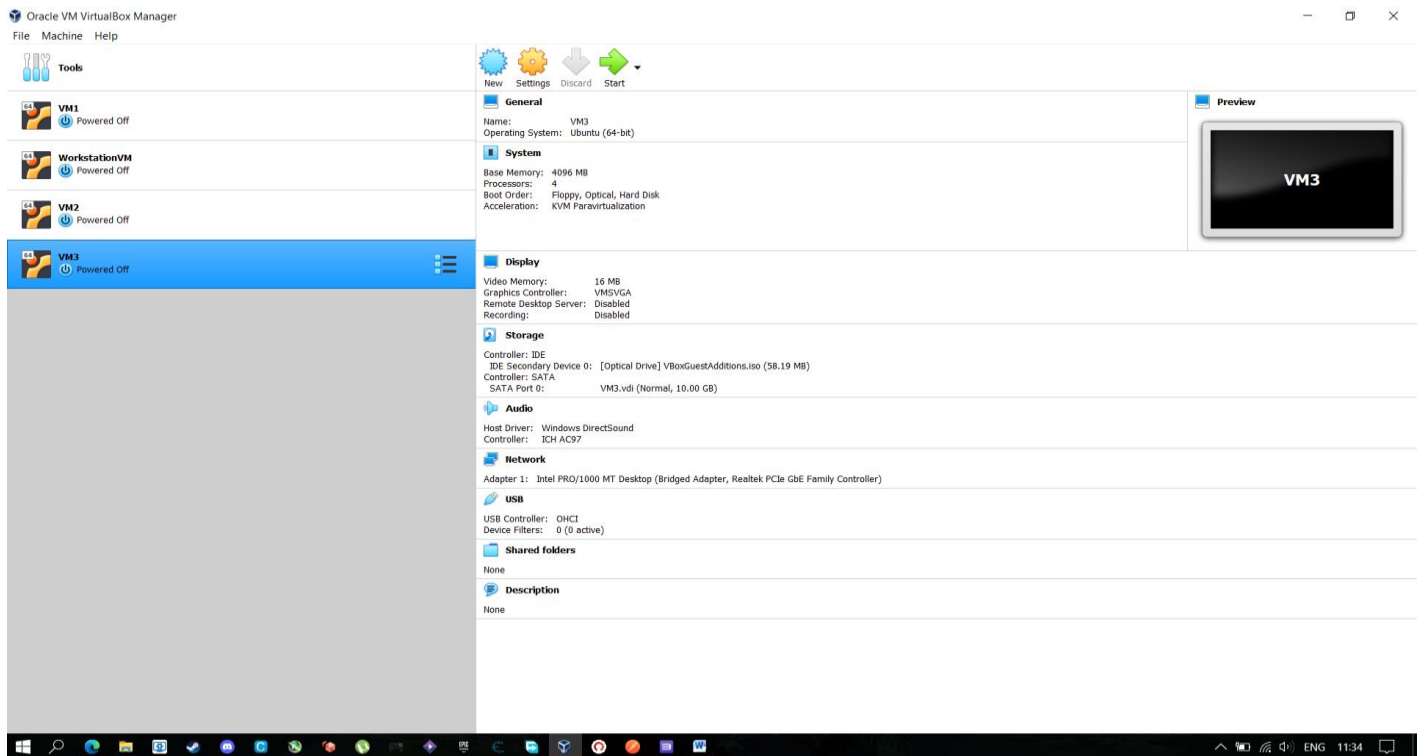
Step 1- Installation of VirtualBox is basically your run of the mill installation. To download VirtualBox use the following link: [Downloads – Oracle VM VirtualBox](#).

Download the appropriate version based on the platform you need VirtualBox to run on. Note that this is independent on the OS you want to set your VMs on. Before installing VirtualBox ensure that you have Virtualization enabled in your BIOS.

Step 2- The next step is to download the ISO of the OS you want to run your VMs on. Since we are using the Ubuntu distribution of GNU/Linux as it is open source we will download the ISO from [Download Ubuntu Desktop | Download | Ubuntu](#). Select the most recent version to download.



Step 3- After downloading the GNU/Linux distribution ISO we can start VirtualBox and go about setting up our first Workstation VM. When we open VirtualBox the GUI looks like this.



Click on the **New** icon to initiate the setup of creating the VM. You will find the following configuration menu.

? X

← 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.

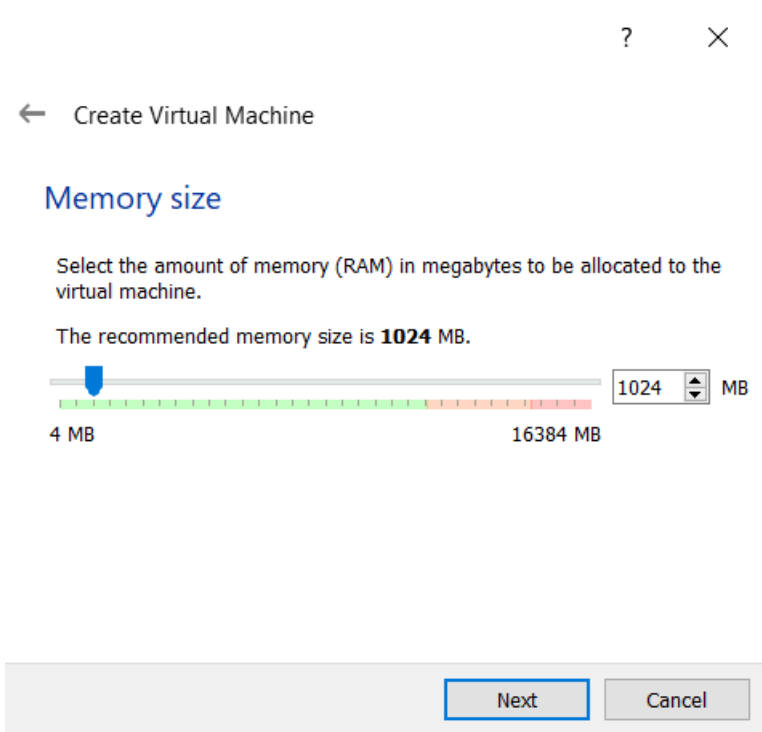
Name:

Machine Folder:

Type:

Version:

Select the name of the VM and type as Linux with Version as Ubuntu (64 bit) and proceed to the next step. The next steps are shown via images.



← Create Virtual Machine

Memory size

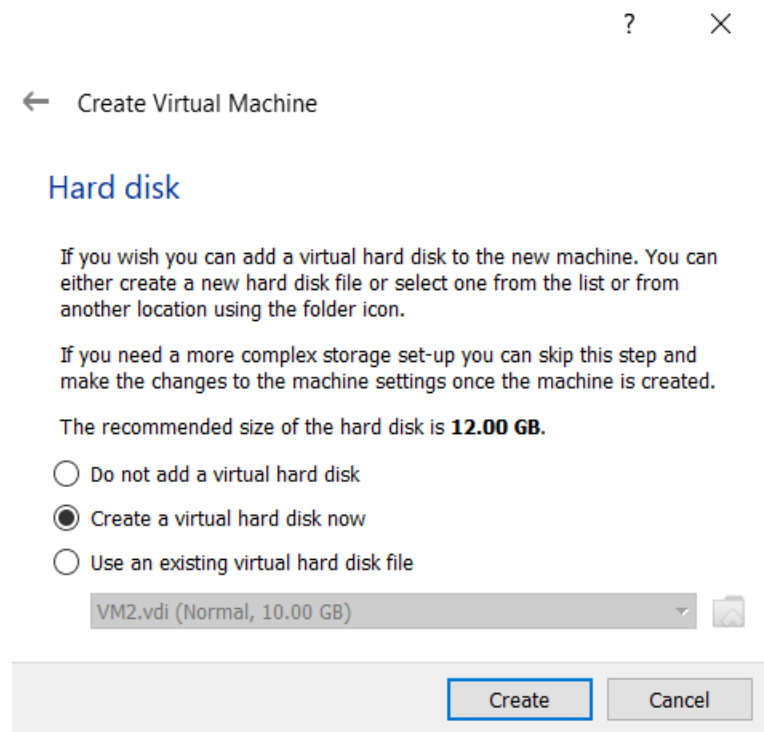
Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

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

4 MB 16384 MB

1024 MB

Next Cancel



← Create Virtual Machine

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 **12.00 GB**.

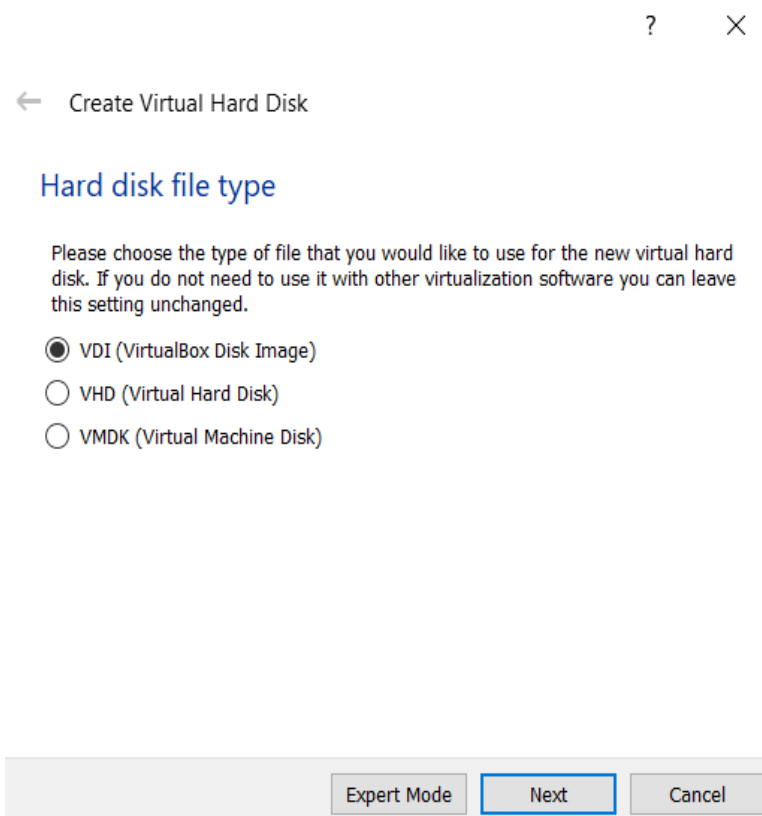
☐ Do not add a virtual hard disk

☒ Create a virtual hard disk now

☐ Use an existing virtual hard disk file

VM2.vdi (Normal, 10.00 GB)

Create Cancel



← Create Virtual Hard Disk

Hard disk file type

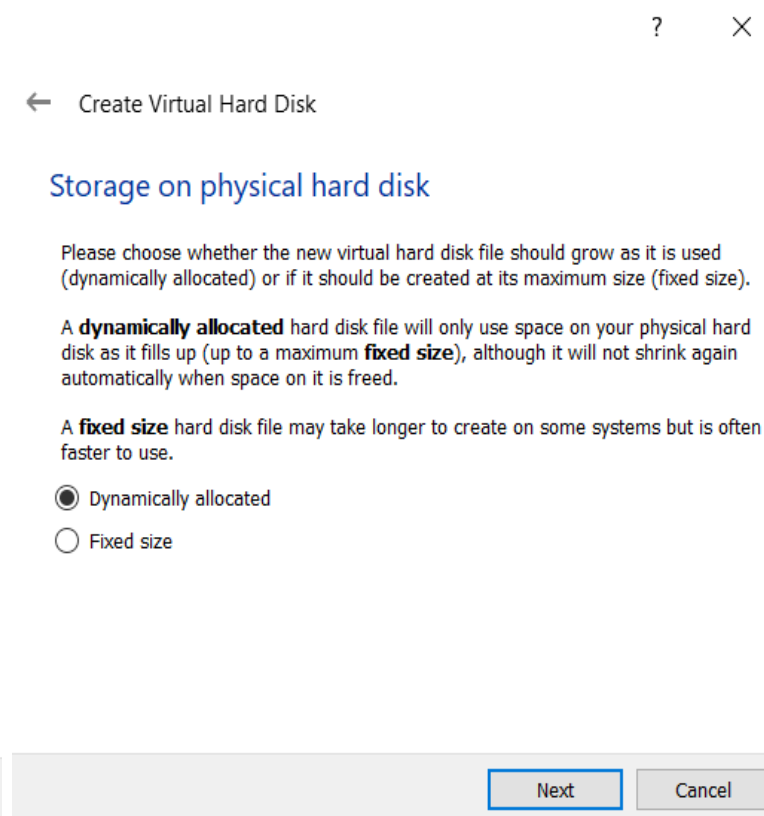
Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

☒ VDI (VirtualBox Disk Image)

☐ VHD (Virtual Hard Disk)

☐ VMDK (Virtual Machine Disk)

Expert Mode Next Cancel



← Create Virtual Hard Disk

Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

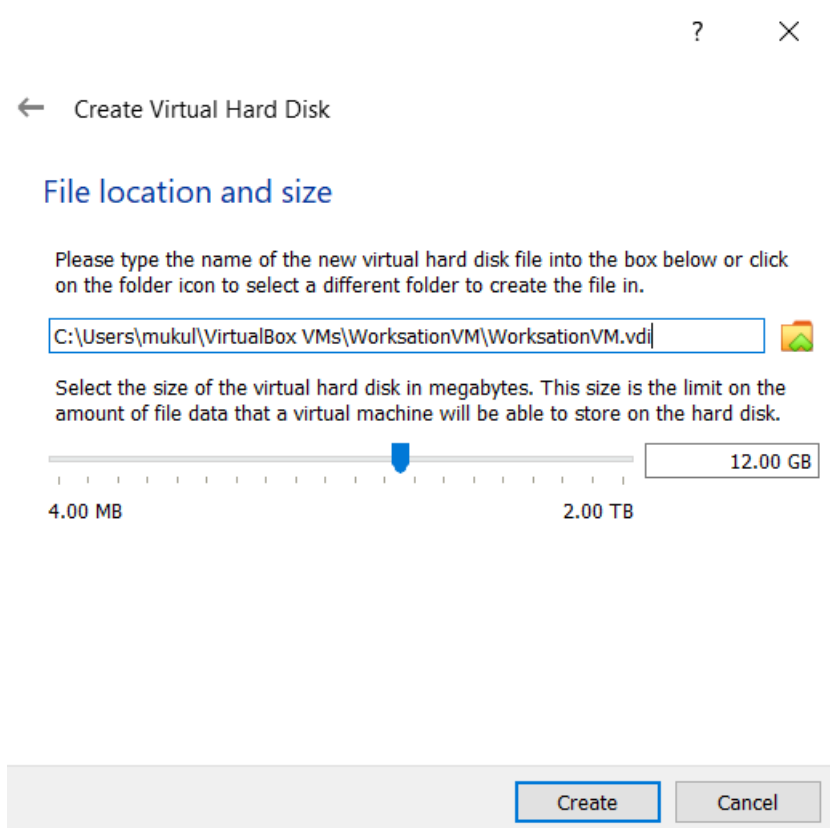
A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

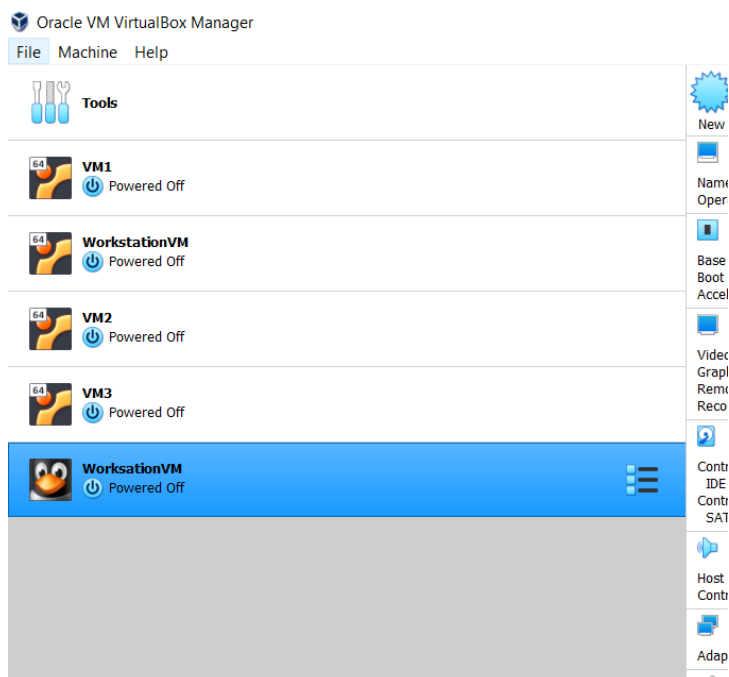
☒ Dynamically allocated

☐ Fixed size

Next Cancel



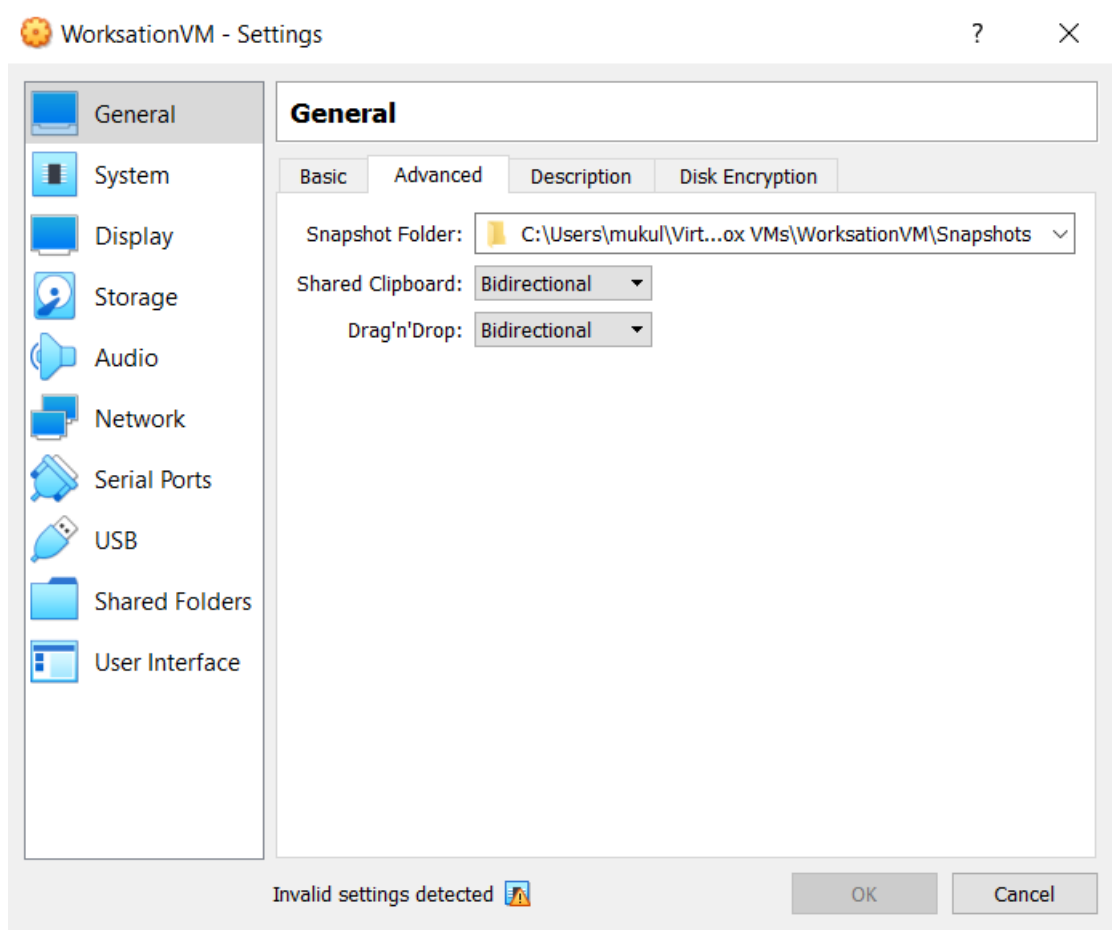
In the above images we can see the configuration that has been kept during the setup of the VMs for this demo. All the configuration options are self explanatory and can be altered to fit the custom needs on a case by case basis. Now we have the bare VM created and can be seen in the left pane showing our created VMs.



What we have now is the bare provisioned resources for the VM. It doesn't have an OS installed in it right now so it won't be able to boot up. The next steps involve the installation of OS in the VM as well as additional configurations.

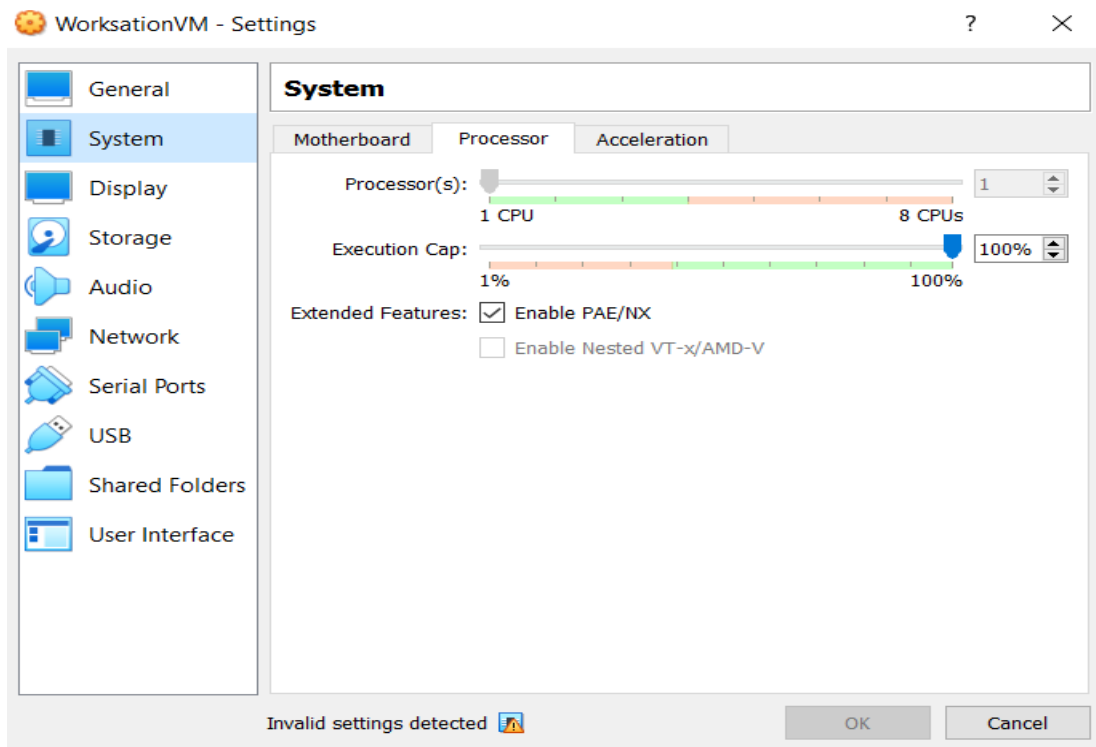
Select the VM name in the left pane of VirtualBox and click on the **Settings** icon.

Under the **Advanced** section of the **General** tab make sure to change the following settings.

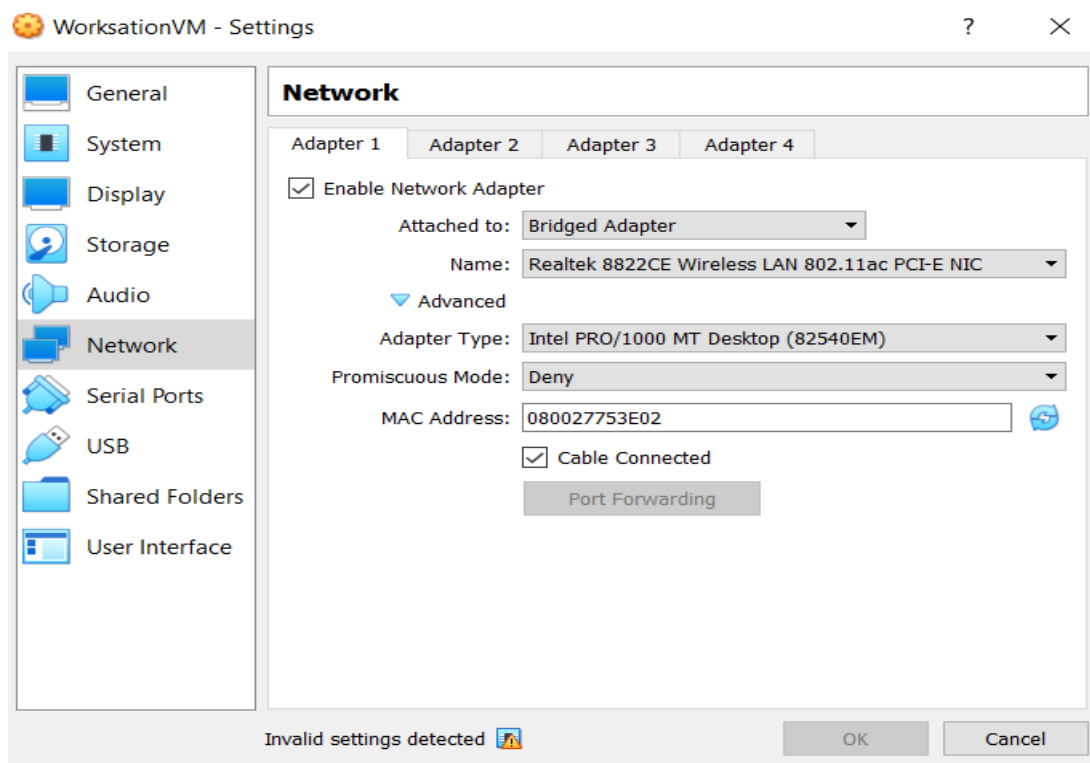


These settings allow you to copy paste data between your host system and VMs as well as between different VMs.

Under the **Processors** section of the **System** tab make changes to the number of CPU cores you want in your VM.

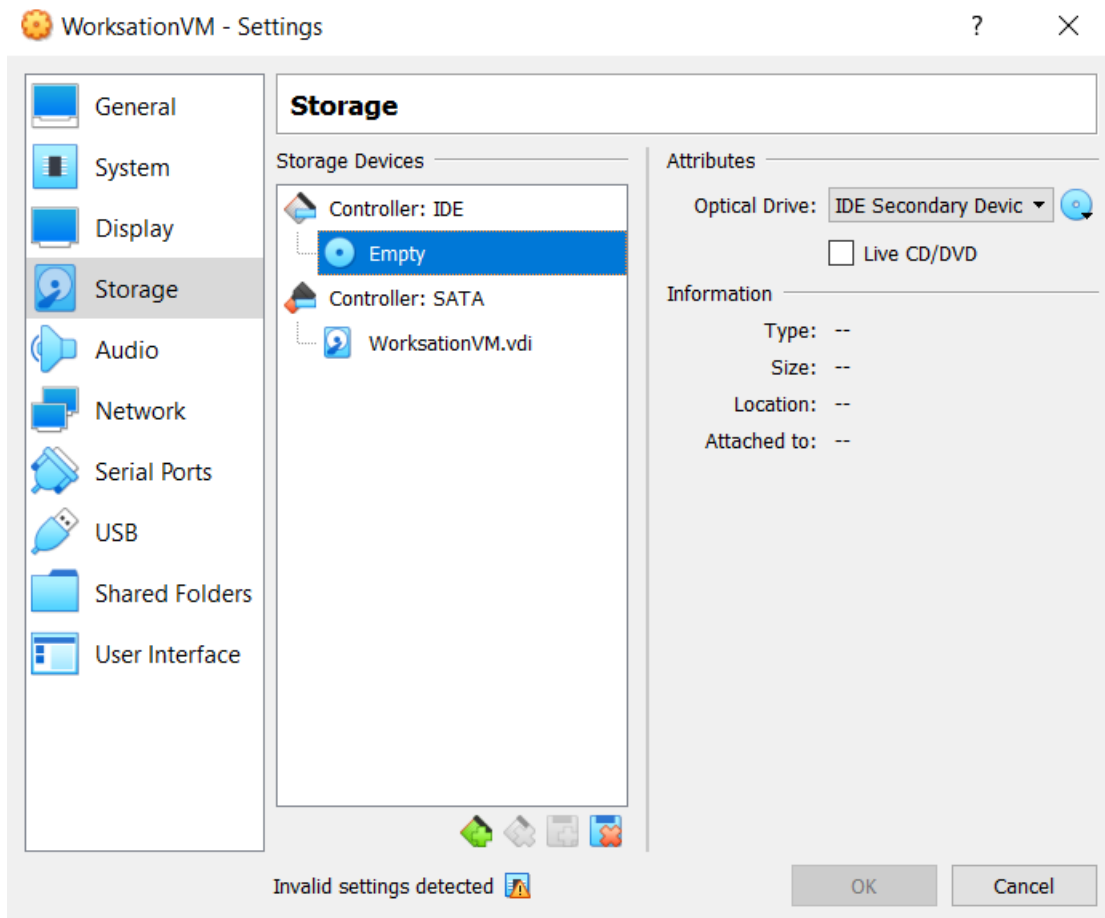


For the purpose of this demo we would like to have all the VMs separate IP addresses so that they are treated as separate machines in Ansible. To accomplish this we use **Bridged Adapter** as well as the Network Adapter we use on the host machine.



The final step is to tell the VM which OS ISO it has to use as the boot device. After this when it is booted for the first time we install the OS from the ISO and then the VM is ready to be used.

Under the **Storage** section of the settings locate this window.



Click on the small CD like icon next to the Optical Drive attribute and browse and select the ISO of the GNU/Linux distribution we downloaded earlier. Save the settings and power on the machine. It will take you to installing Ubuntu on the machine. Install Ubuntu using the GUI and then the machine is ready for use.

Repeat the steps for 3 more VMs. When installing Ubuntu try keep the Sudo passwords for all machines same for the purpose of this demo. An indepth guide to install Ubuntu (although installing Ubuntu and knowing Linux commands is a prerequisite for Ansible) can be found at this link: [Install Ubuntu](#).