

# VM Setup

Inside our computer we can create a virtual computer.

## What is virtualization

One computer does the job of multiple computers (multiple OS).

### Life Before Virtualization

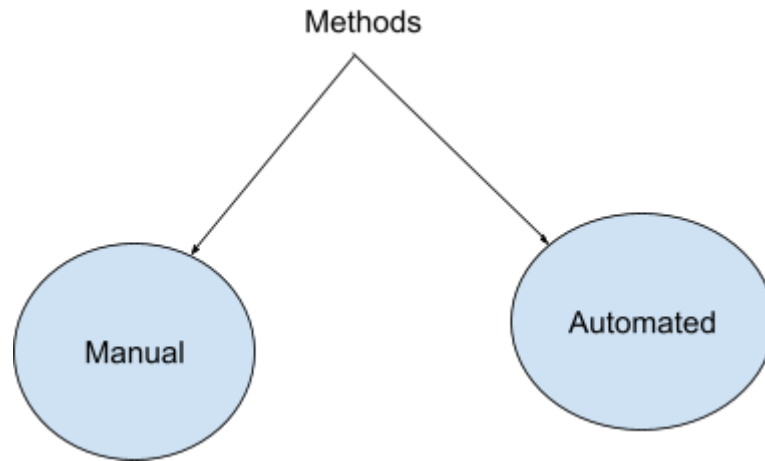
- To run app/services we need server
  - Physical computers (Servers in Datacenter)
  - One service- One Server (Isolation)
  - Servers are always overprovisioned
  - Server Resources mostly underutilized
  - Huge Cap Ex & Op Ex.
- 
- ❖ Enter VMware.
  - ❖ Allow the same computer to run multiple OS.
  - ❖ Partition physical resource in virtual resource.
  - ❖ Virtual Machines runs in Isolated Env.
  - ❖ Each virtual machine need its own OS.
  - ❖ Server Virtualization is the most common virtualization.

### Terminologies:

- Host OS
- Guest OS
- VM
- Snapshot: Is a way of taking backup of VM
- Hypervisor: Tool/software to create VM
  - Type 1
    - Bare Metal
    - Runs as a Base OS
    - Only for production
    - E.g. VMware esxi, Xen Hypervisor
  - Type 2
    - Runs a software
    - Learn & Test
    - E.g. Oracle virtualbox, Vmware server

## Virtual Machine Setup

We will set up two Linux OS that is CentOS and Ubuntu.



### VM- Manually

Prerequisites:

- Enable Virtualization in BIOS
  - VTx
  - Secure virtual machine
  - Virtualization
- Disable others windows virtualization
  - Microsoft Hyperv
  - Windows Hypervisor platform
  - Windows Subsystem for linux
  - Docker Desktop
  - Virtual Machine Platform
  -

<Sometime VM not get ip address>

Steps to install:

- Click new
- Give name choose type and version
- Allocate resource
- Uncheck preallocated section in storage section
- Finish

Yep, now we have a virtual computer but the OS is not installed in it for this we will **download iso file and install it.**

ISO files contain the installation files for OS

- After downloading the ISO file
- Attach the ISO
- Change the Network Interface
- ACPI Shutdown
- Start → start the installation
- Shutdown → remove the iso image → start the vm

Type command: `ip addr show` → Note down the ip

Now we can do ssh to the machine.

`ssh <username>@<ip_addr>`

Enter the password.

## VM- Automatically

**VAGRANT:** It is a VM automation tool/ manages VM lifecycle.

It is a command line tool

Steps:

`mkdir /c/vagrant-vms`

`cd /c/vagrant-vms/`

Create a folder

`mkdir /ubuntu`

`cd /ubuntu/`

Vagrant has a vagrant cloud which stores the image of all the virtual boxes.

So if we want to use Ubuntu we use that image to proceed further.

Go to the vagrant cloud and search for the image.

Copy and use the box name

**vagrant init ubuntu/jammy64** : Vagrant file will be created

**vagrant up**

**vagrant status**

Current machine states:

default                      running (virtualbox)

The VM is running. To stop this VM, you can run ``vagrant halt`` to shut it down forcefully, or you can run ``vagrant suspend`` to simply suspend the virtual machine. In either case, to restart it again, simply run ``vagrant up``.

To list the status of all the virtual box

`vagrant global-status`

**vagrant ssh** to login the vm

`sudo -i` : to switch to root user

Exit: to logout

**vagrant halt**: to power off the vm

To power on: `vagrant up`

**vagrant destroy:** to delete the vm

**vagrant reload :** to reload the vm

**vagrant box list:** to list the available boxes in the computer