Vm Prerequisite

① Enable virtualization in bios.
- VTx / virtualization / secure virtual machi

# to check
↳ ① seach for 'Task manger'
② go to performance
③ check "virtiialization" is "enabled" or not

② Disable
- microsoft hyperv
- window hypervisor platform
- windows subsytem for linux
- Docker desktop
- virtual machine platform

# to check
↳ ① turn windows feature on & off
② check the following options.

Virtualization

- it allow one computer to run multiple OS
- it run on isolated enviroment.
- each vm needs its own OS
- server virtualization is the most common virtualization

Terminology —

- Host OS - primary os that run directly on hardware
- Guest OS - OS run within vm.
- vm - virtual machine
- Snapshot - backup of the current vm
- hypervisor - software on which we run multiple or single virtual machine.

Type of Hypervisor

Type 1
- Baremetal
- run on base OS
- production
- eg:- vmware esxi, Xen Hypervisor

Type 2
- Runs a software
- Learn & test
- eg. oracle virtual box, vmware server
-

Manual setup

To connect it
- open, git bash
- type 'ssh username@ip
- to get this in terminal add write 'ip addr show'

① create new vm
② enter name
③ attach iso file in 'ISO image'
④ check ☑ skip unattached installation
⑤ Go to Hardware > basememory = 4096MB
Processor : 2 CPU
enable EFI
⑥ Go to HardDisk > allocate 20 to 25 gm storage
uncheck the pre-allocate full size
⑦ click on finish.
⑧ click on ~~show~~ setting > network > Adapter 2
- check in the enable box
- select Bridge adapter in 'attached to'
- click ok

setting > storage > empty > • check LiveCD

⑨ click 'Start' ⇨
⑩ select install (first option for installation)
⑪ select every option as per your need
⑫ when the option shows for 'reboot' or 'login'
⑬ power off the vm. than click on setting > storage
⑭ under controller IDE click the option > optical drive
- remove
- uncheck Live CD

⑮ Start.

now with keyboard

Automate the VM ~~step~~ setup

① o install 'vagrant' through 'choco'.

'choco install vagrant--version=2.4.3-y'

② open 'git bash'!

③ redirect to the paticular folder by entering the cd+ path.

④ check current location 'pwd'

⑤ create two directory. and then redirect to that directory

⑥ vagrant init [———] → vagrant box code

⑦ cat Vagrantfile - display the content of the file

⑧ vagrant up - power onn vm

⑨ vagrant reload - reload the onn vm for update

⑩ vagrant status - statues of vm

⑪ vagrant global-status. - statues of all the vm present

⑫ vagrant halt - power off vm

⑬ vagrant destroy - delete vm

⑭ vagrant ssh - get into vm

whoami - show the use
pwd — current location
sudo-i - transfer to root user
exit - exit from root use
exit. - exit from vm

⑮ Vagrant box list. - show a list of all the vagrant boxes that have been installed on your system.

Vagrant - VM automation tool which use hypervisor to mange / create / destroy VM on ~~[illegible]~~ computer

① mkdir vagrant-vm → vm folder.
② cd vagrant-vm
③ mkdir centos → linux folder
④ cd centos
⑤ vagrant init [linux flav ~~fam~~] → from vagrant cloud.
⑥ cat Vagrantfile → to see details of vagrant vm.
⑦ vagrant up
{ if any error show ~~turn~~ off your antivirus }
⑧ vagrant ssh → log in to the vm
⑨ vagrant halt → turn off the vm
⑩ vagrant status → show the status
vagrant global-status → show the status of all the vms

↳ by ~~following~~ above steps we can create and check a vm in git bash.

* first we need to redirect our self to the folder where we need to setup the vm using 'cd'; 'ls'; 'pwd'

* second we need to follow the steps.