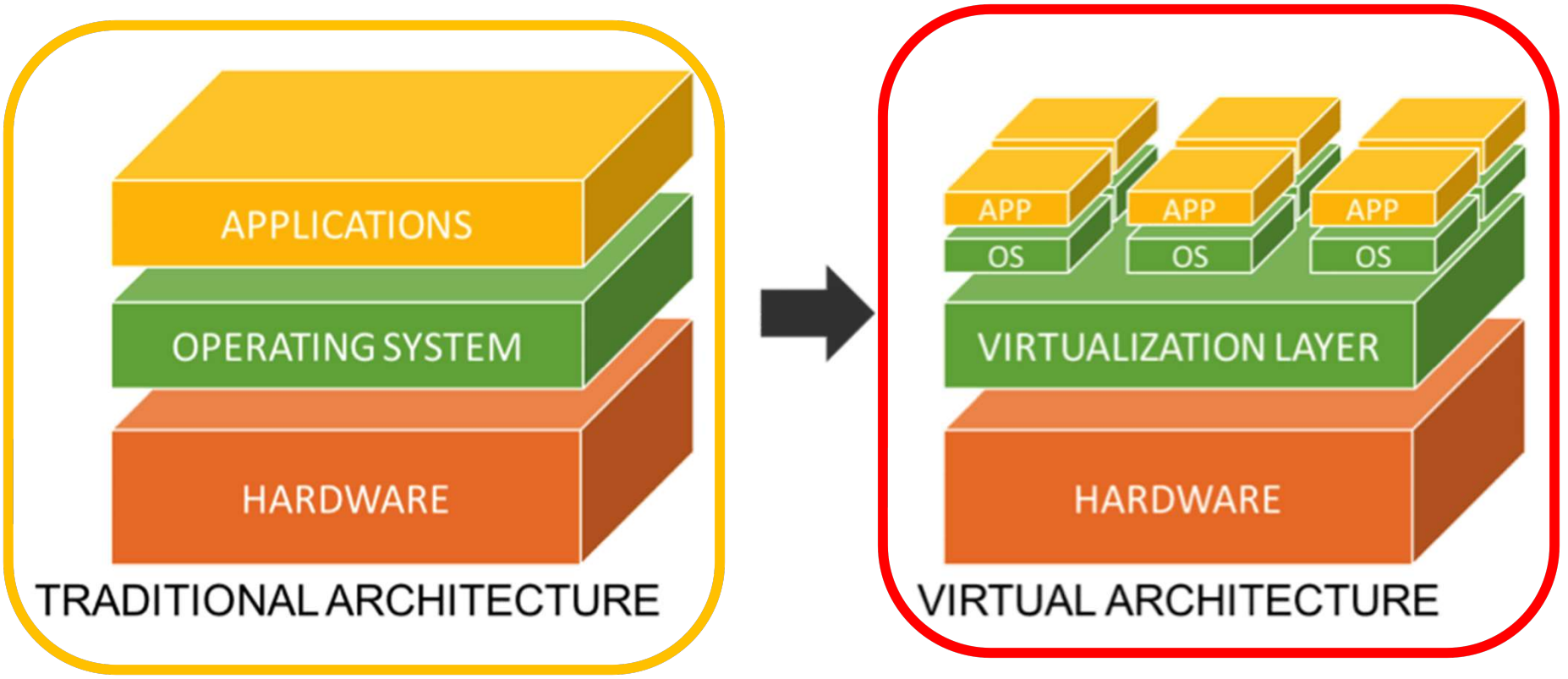
Virtualization

What is Virtualization?



What is Virtualization?

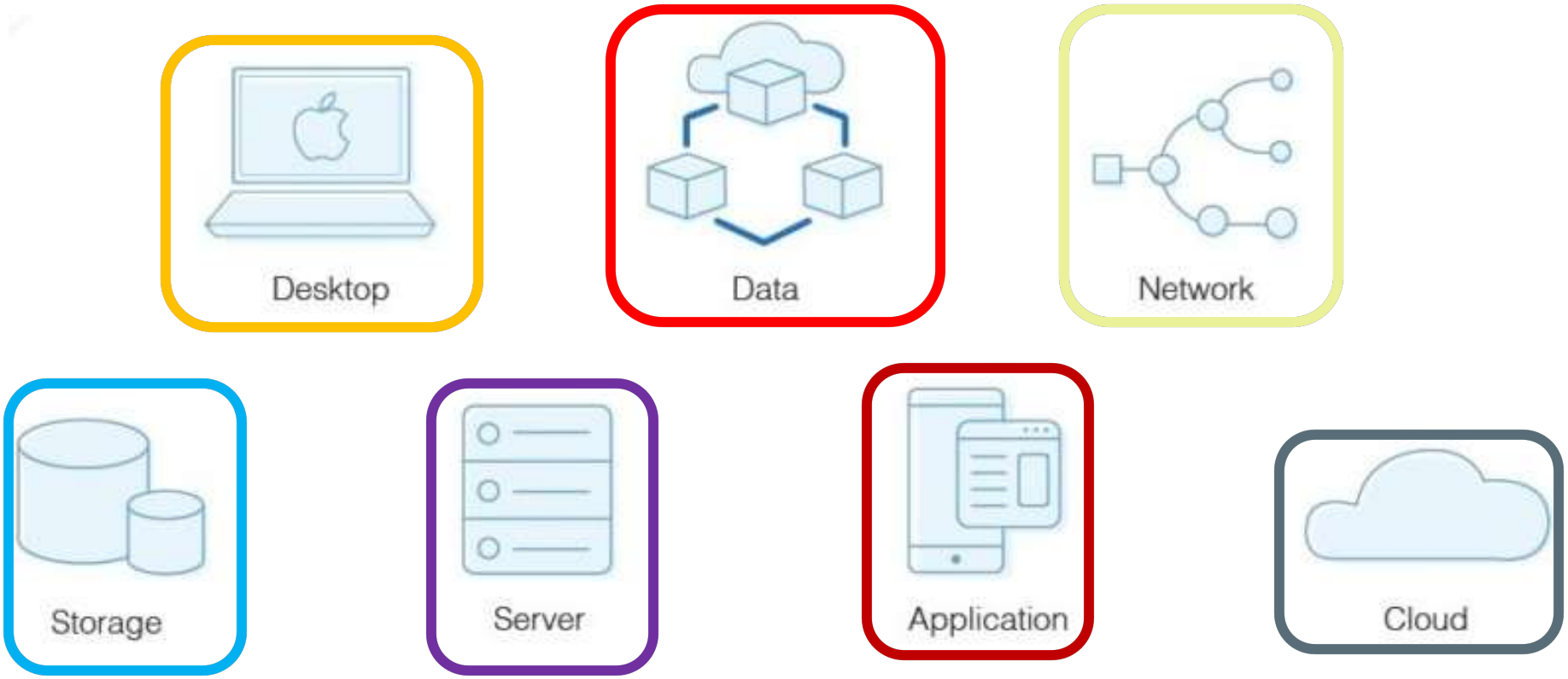
The act of creating a virtual version of

- Virtual computer hardware platforms,
- Storage devices, and
- Computer network resources etc

Single most effective way to reduce IT expenses

Boosts efficiency and agility for businesses

Types of Virtualization



Virtualization Technologies

HyperV

- By Microsoft for the Windows Server Virtualization

VMWare

- Cloud based virtulization technology
- Can run Multiple instances of virtual machines

KVM

- Linux based full virtulization technology which is free and Open source virtulization

Citrix server

- Provides virtulization for Desktop and Application based on Xen Hypervisor

Virtualization Objective and Benefits

Protection from System Failures

- Virtualization lets open the same work on another device
- Store all backup data through virtualization on cloud services
- Get easy access to it from any device

Hassle-free Transfer of Data

- Can easily transfer data from a physical storage to a virtual server, and vice versa
- Don’t have to waste time digging out hard drives to find data.

Firewall and Security

- With virtual firewalls, access to your data is restricted at much lower costs as compared to traditional methods

Smoother IT Operations

Cost-Effective Strategy

Applications for Businesses – Some Examples

Productivity

- Google Docs
- Office 365

Business Process

- CRM and ERP are now based on Cloud
- This method is cost-effective

Backup and recovery

- The responsibility of your information also lies with your service provider
- Recovery is also performed faster in the cloud
- Dropbox, Google Drive and Amazon S3

Big data analytics

- Cloud services make mining massive amounts of data possible by providing higher processing power

Social Networking

Applications for Businesses – Some Examples

Online Streaming Platform

- Netflix
- Amazon Prime
- Youtube

Application Development

- Github
- Cloud machines
- Online Backup
- Test machines for few hours

Deep Learning

Banking, Financial Services

HYPERVISORS

Software called hypervisors separate the physical resources from the virtual environments—the things that need those resources. Hypervisors can sit on top of an operating system (like on a laptop) or be installed directly onto hardware (like a server), which is how most enterprises virtualize. Hypervisors take your physical resources and divide them up so that virtual environments can use them.

Types of Hypervisors

- **Type1**
- **Type2**

TYPE 1 Hypervisor

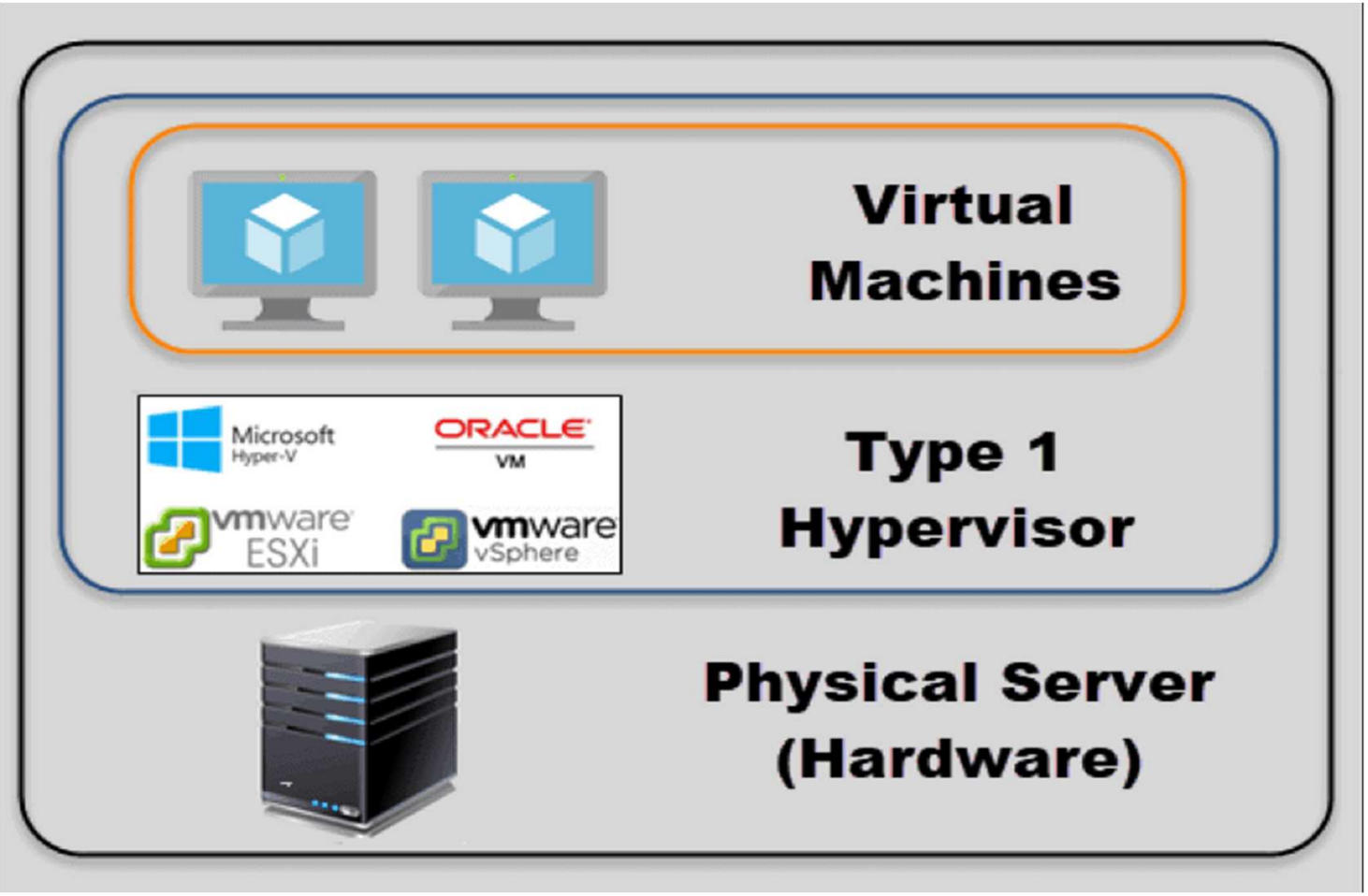
A layer of software we install directly on top of a physical server

Proven in providing excellent performance and stability since it does not run inside Windows or any other operating system.

Type 1 hypervisors are an OS themselves

The physical machine the hypervisor is running on servers virtualization purposes only

You cannot use it for anything else.


Virtual Machines
Microsoft Hyper-V
ORACLE VM
vmware ESXi
vmware vSphere
Type 1 Hypervisor
Physical Server (Hardware)

Type 1 Hypervisors examples

- KVM(Kernel-Based Virtual Machine)
- Vmware vSphere with Esxi
- Citrix Hypervisor (Xenserver)
- Hyper V

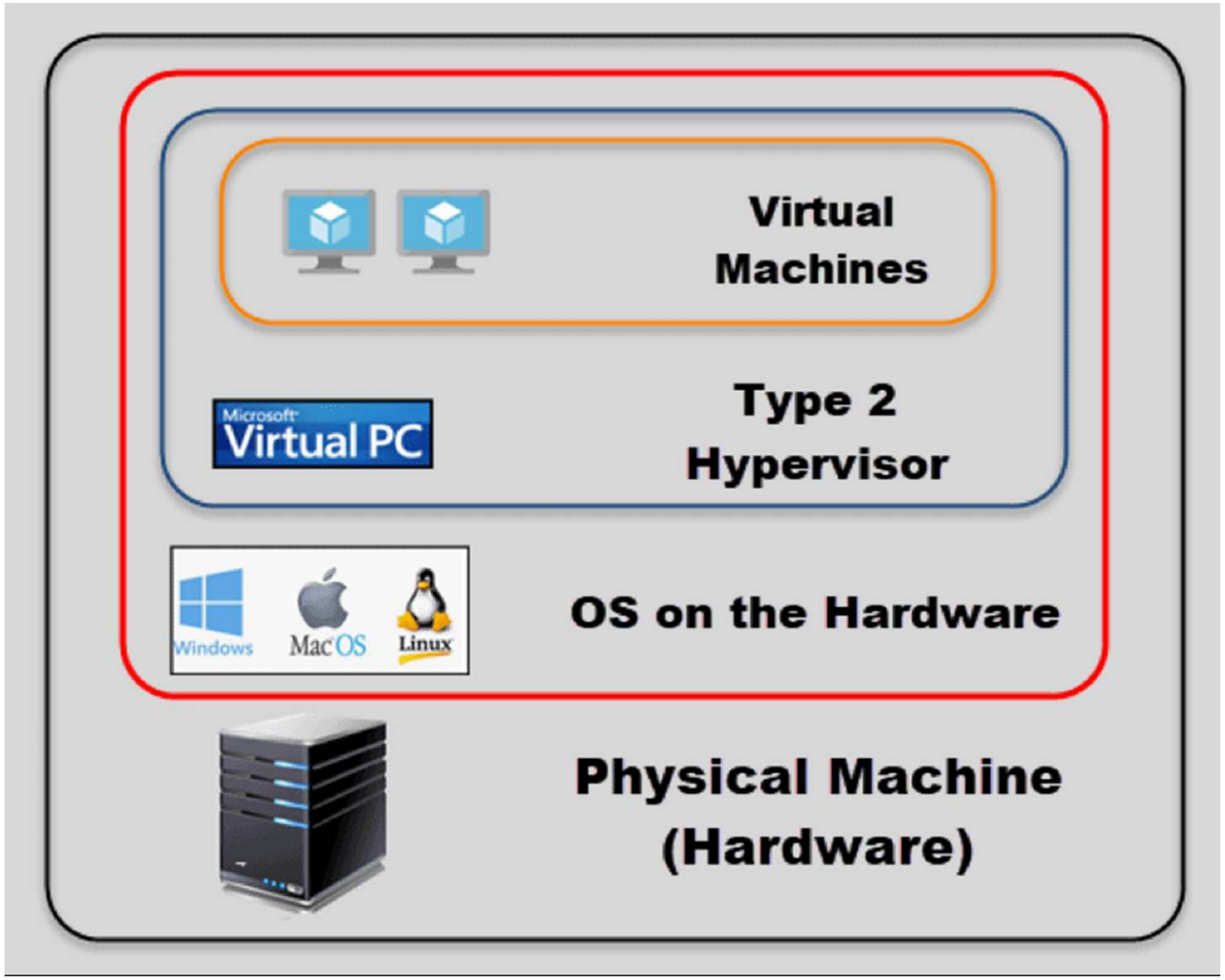
Type 2 Hypervisors

Doesn’t run directly on the underlying hardware

Instead, it runs as an application in an OS.

Rarely show up in server-based environments

Suitable for individual PC users needing to run multiple operating systems.


Virtual
Machines
Microsoft
Virtual PC
Type 2
Hypervisor
Windows
Mac OS
Linux
OS on the Hardware
Physical Machine
(Hardware)

Type 2 Hypervisors examples

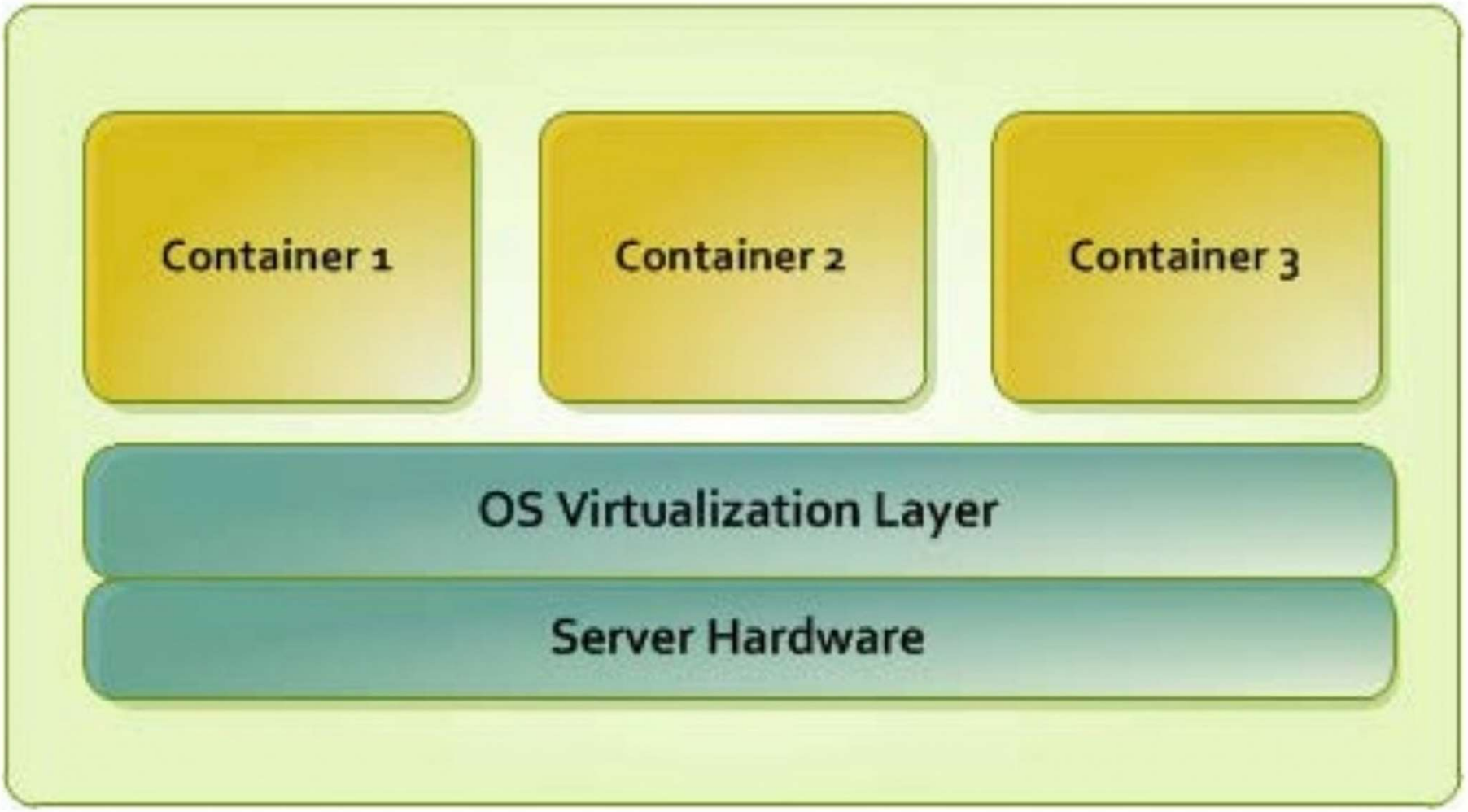
- Oracle VM virtualbox
- VMware Workstation Pro/VMware Fusion
- Parallels Desktop

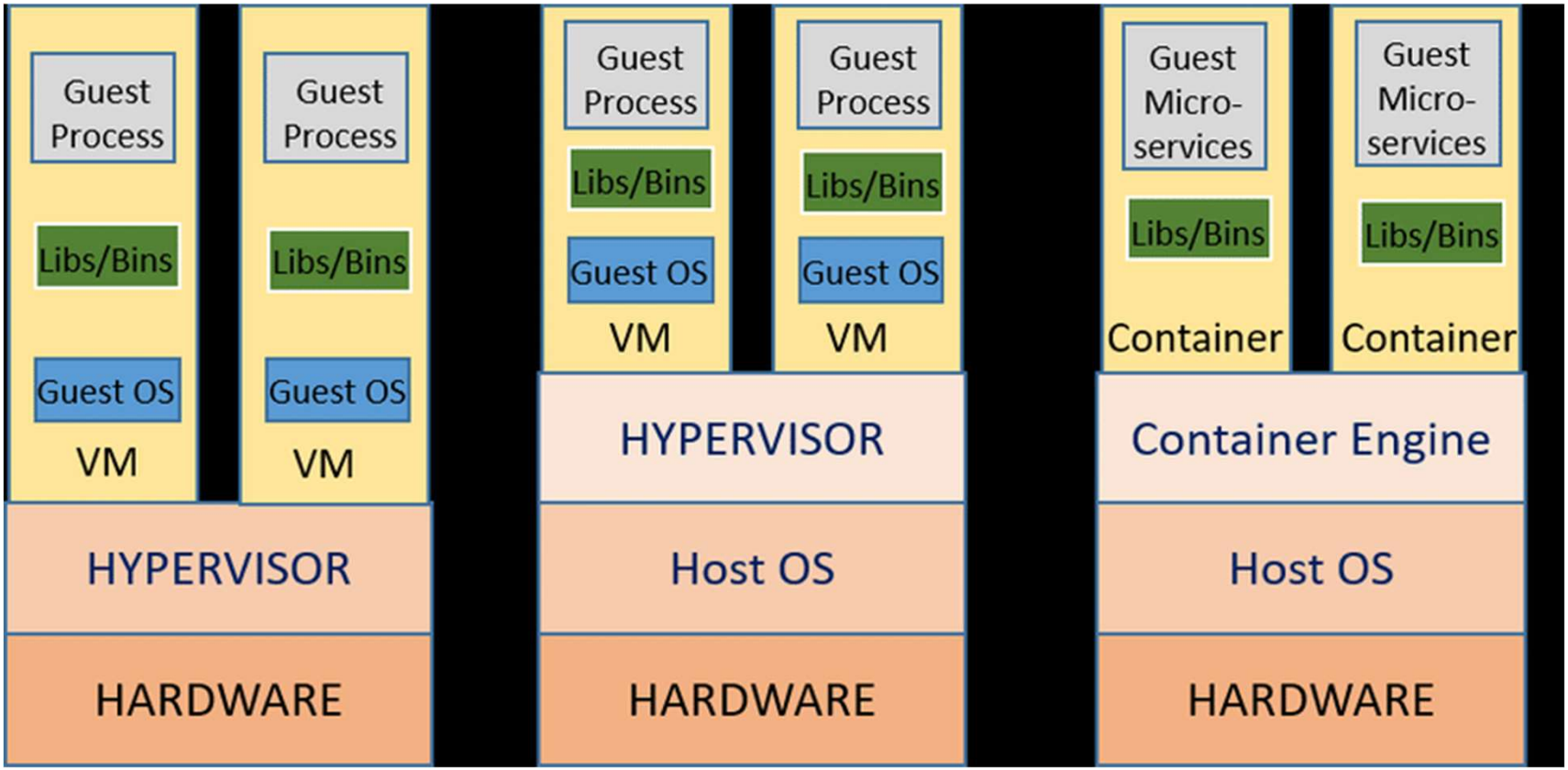
OS virtualization

Kernel of an operating system allows for multiple isolated user-space instances

Called as containers and software containers

Commonly used in virtual hosting environments


Container 1
Container 2
Container 3
OS Virtualization Layer
Server Hardware



TYPE 1
VS

VS
OS-LEVEL VIRT.

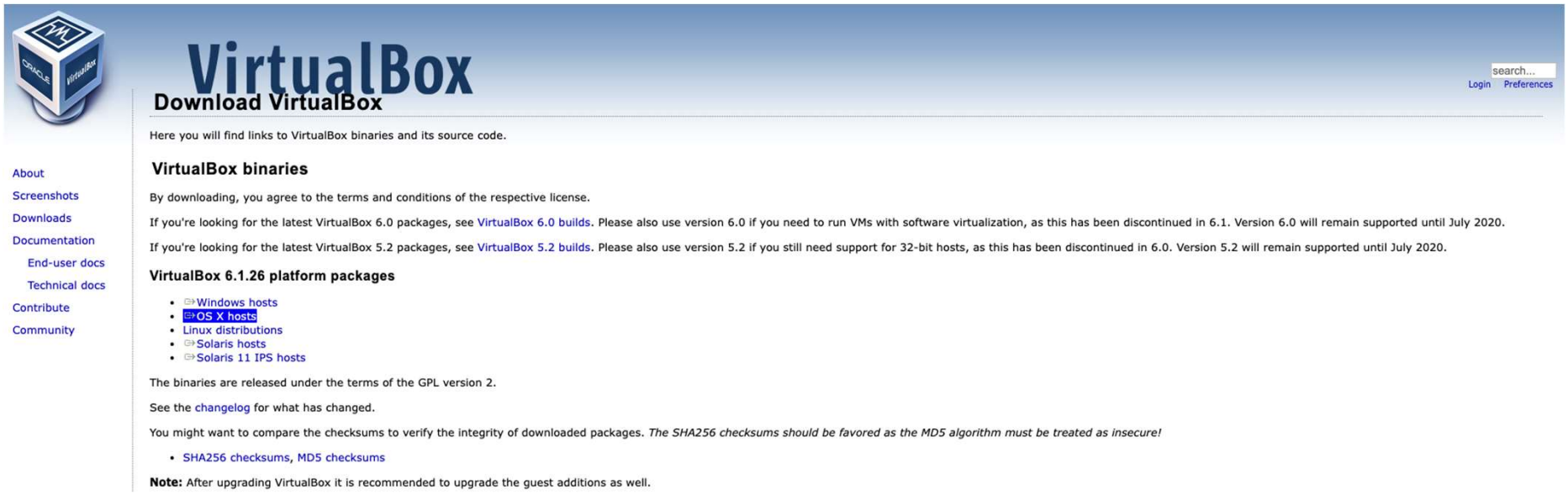
TYPE 2

VIRTUALBOX

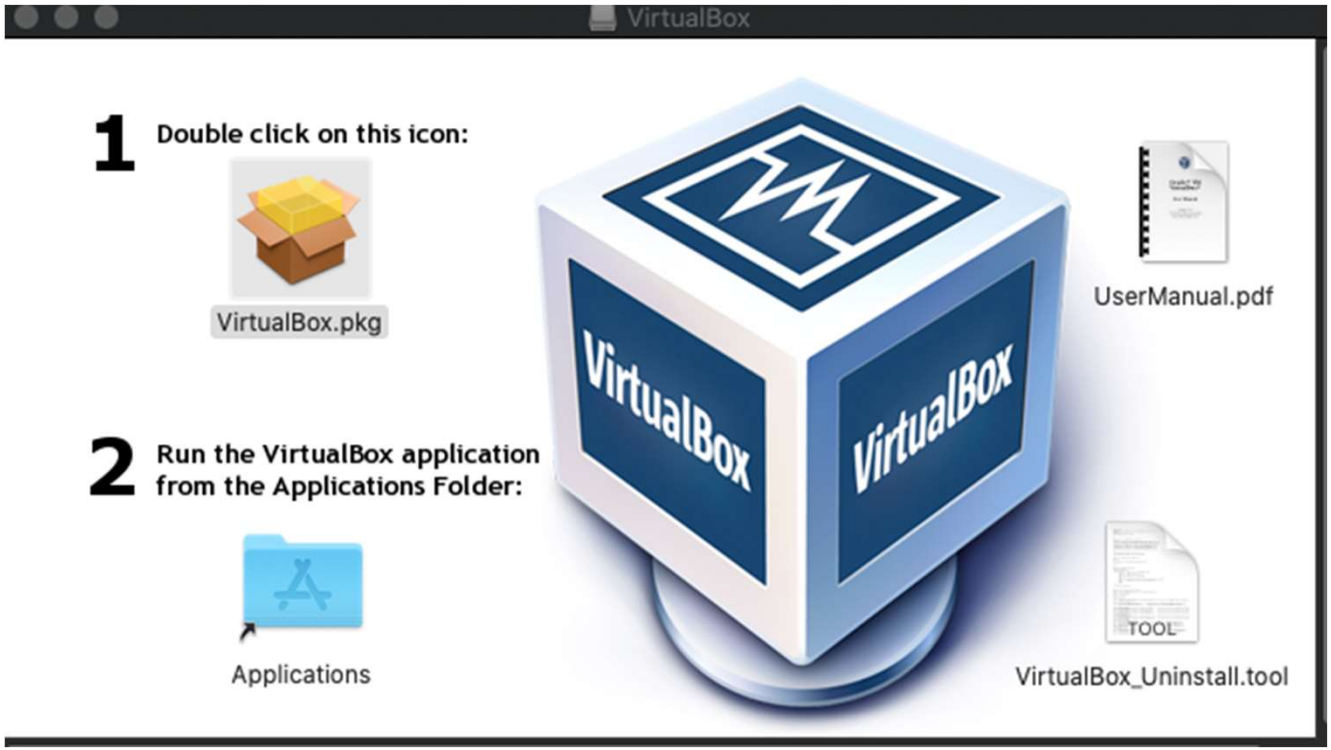
- VirtualBox is a virtualization tool used for enterprise as well as home.
- Created and managed by Oracle. It is used widely.
- It helps us to run virtual machines on our local desktops and laptops over already installed operating system.
- It supports all major desktop platforms including Windows, Macos, linux etc.

VIRTUALBOX INSTALLATION

1. Go to website . Go to downloads and select the host which you using. I am using macos so I will click on OS X hosts.

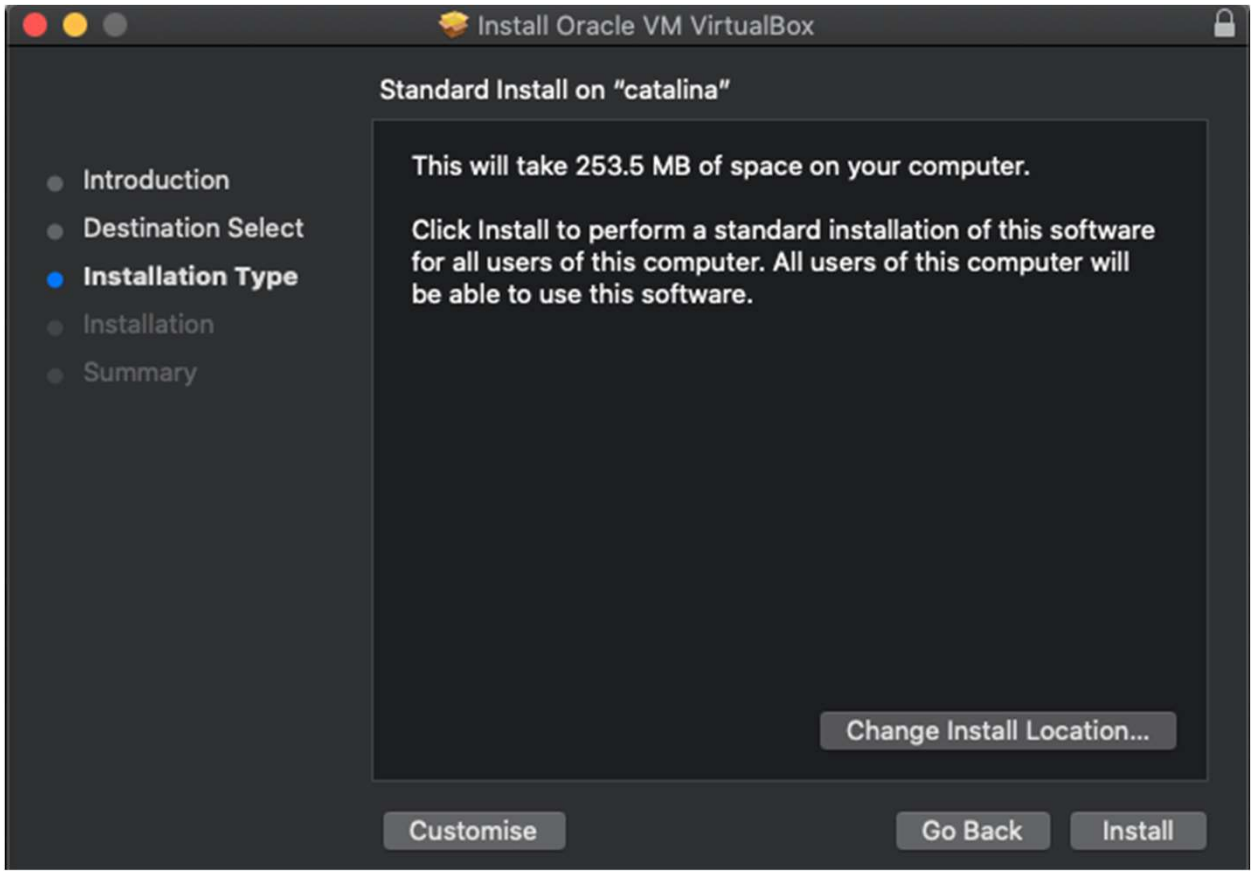


2. Double Click on VirtualBox.Pkg

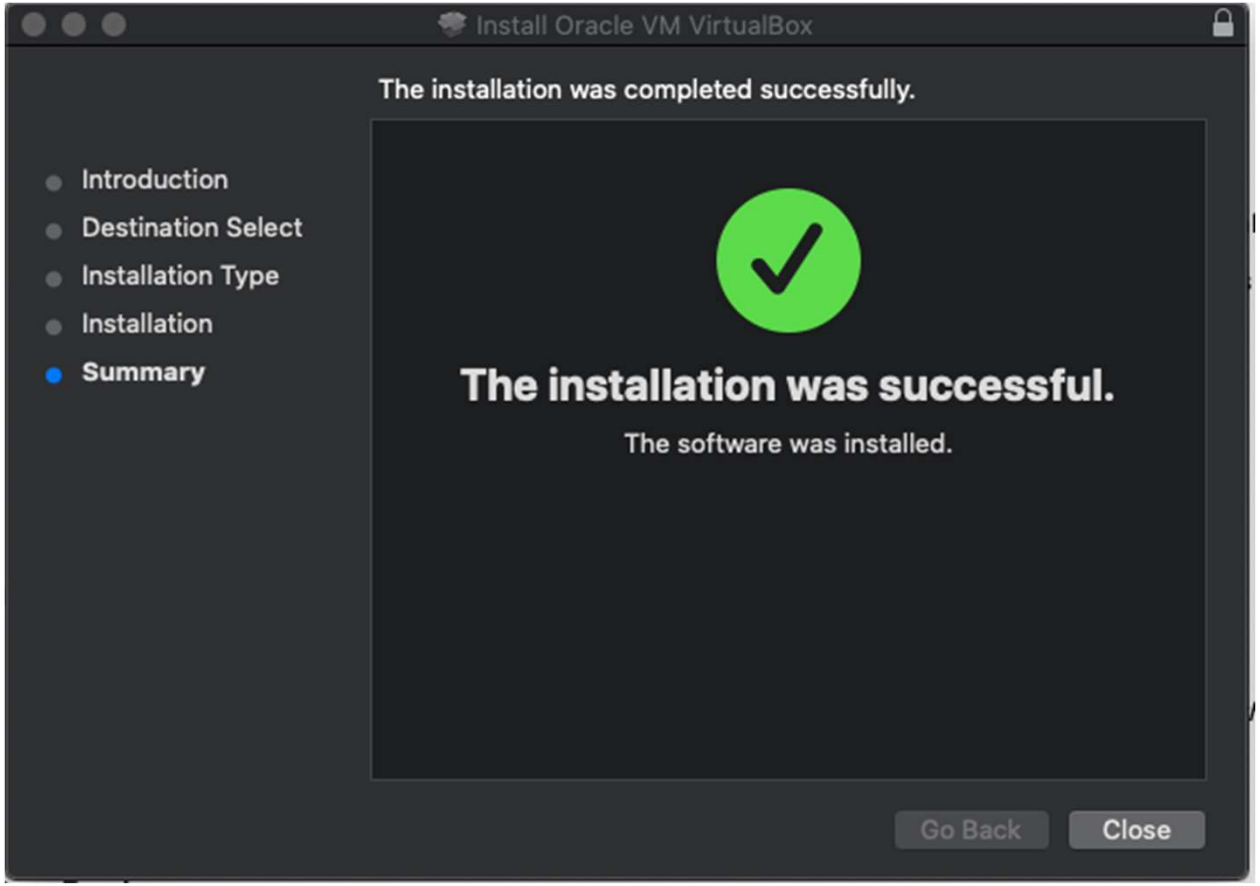


3. Click on the continue button

4. Click on the install button

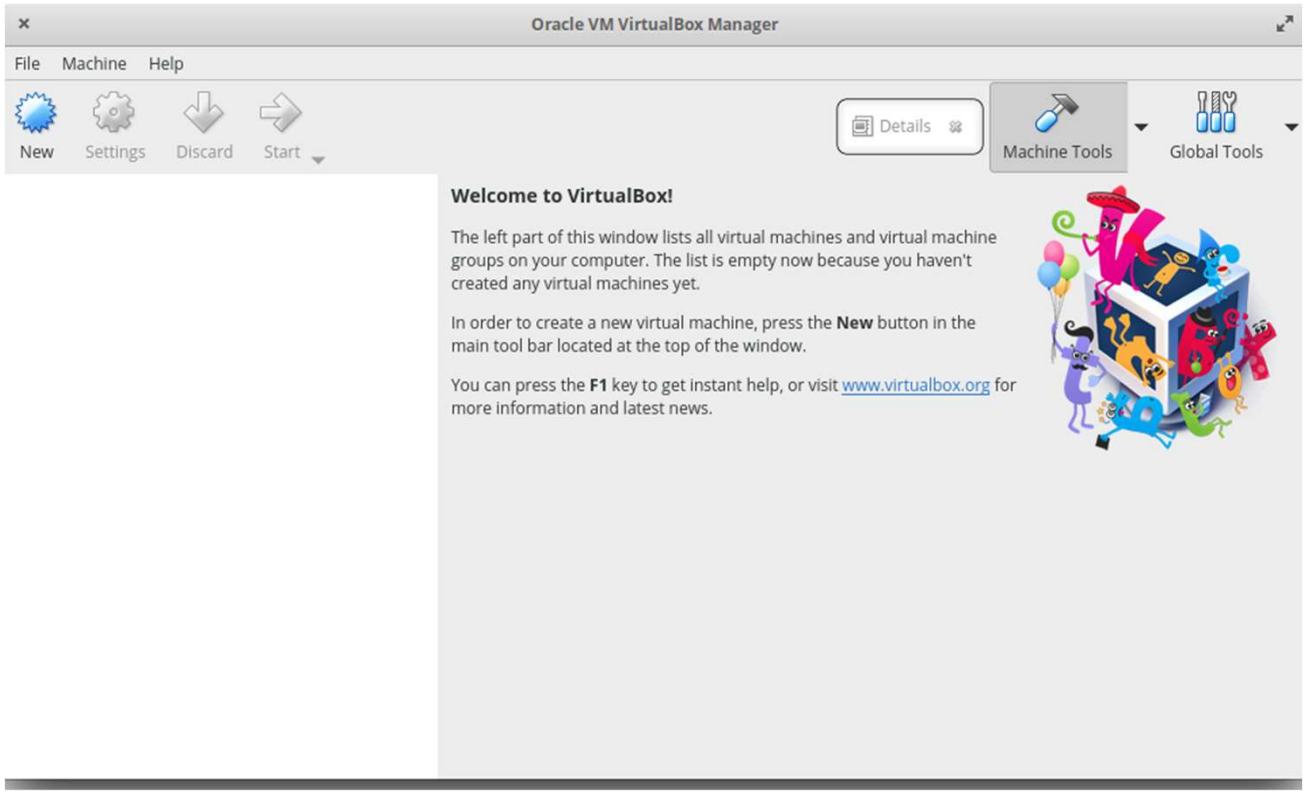


5. On successful installation we will get this message.

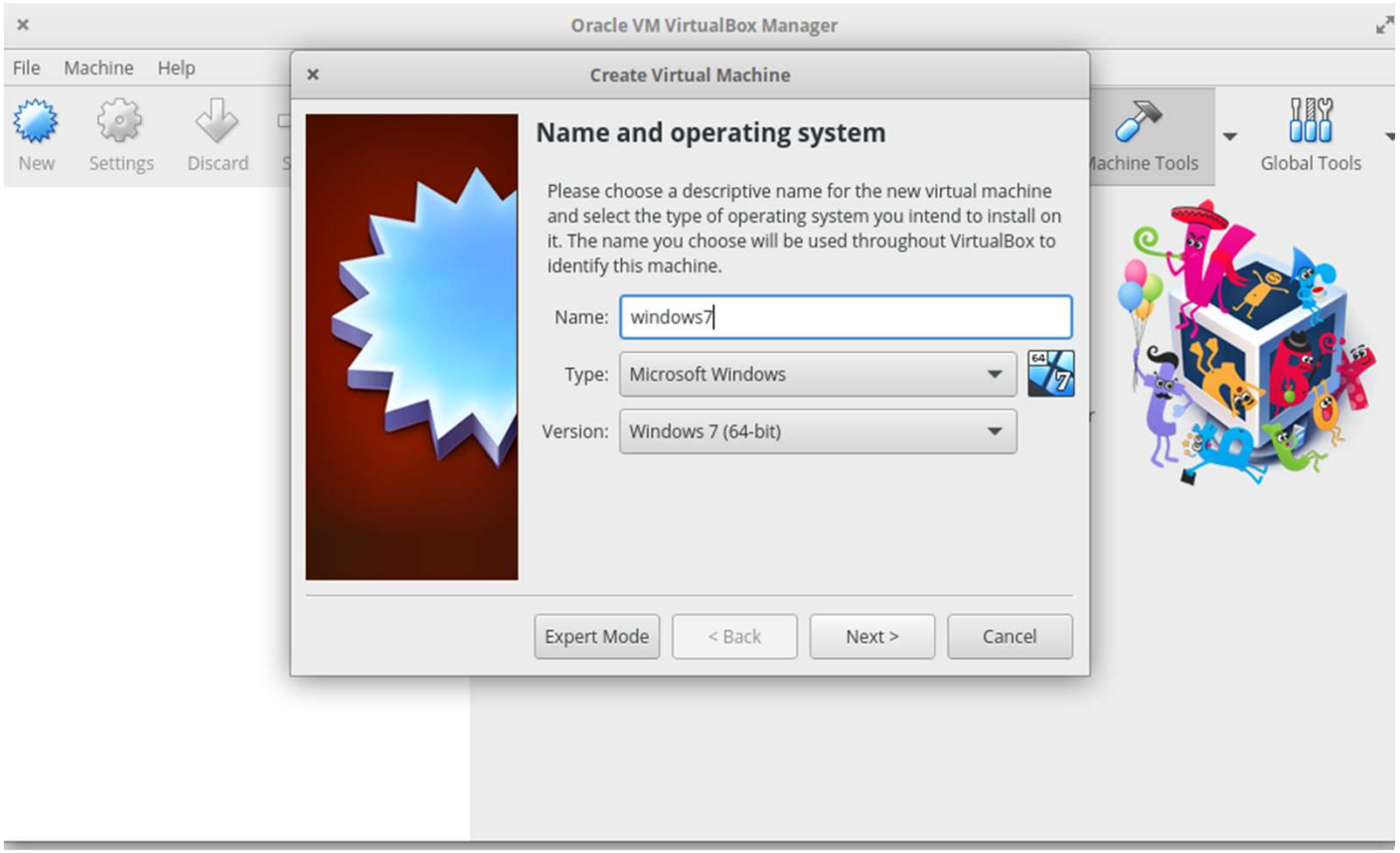


CREATING VIRTUAL MACHINES

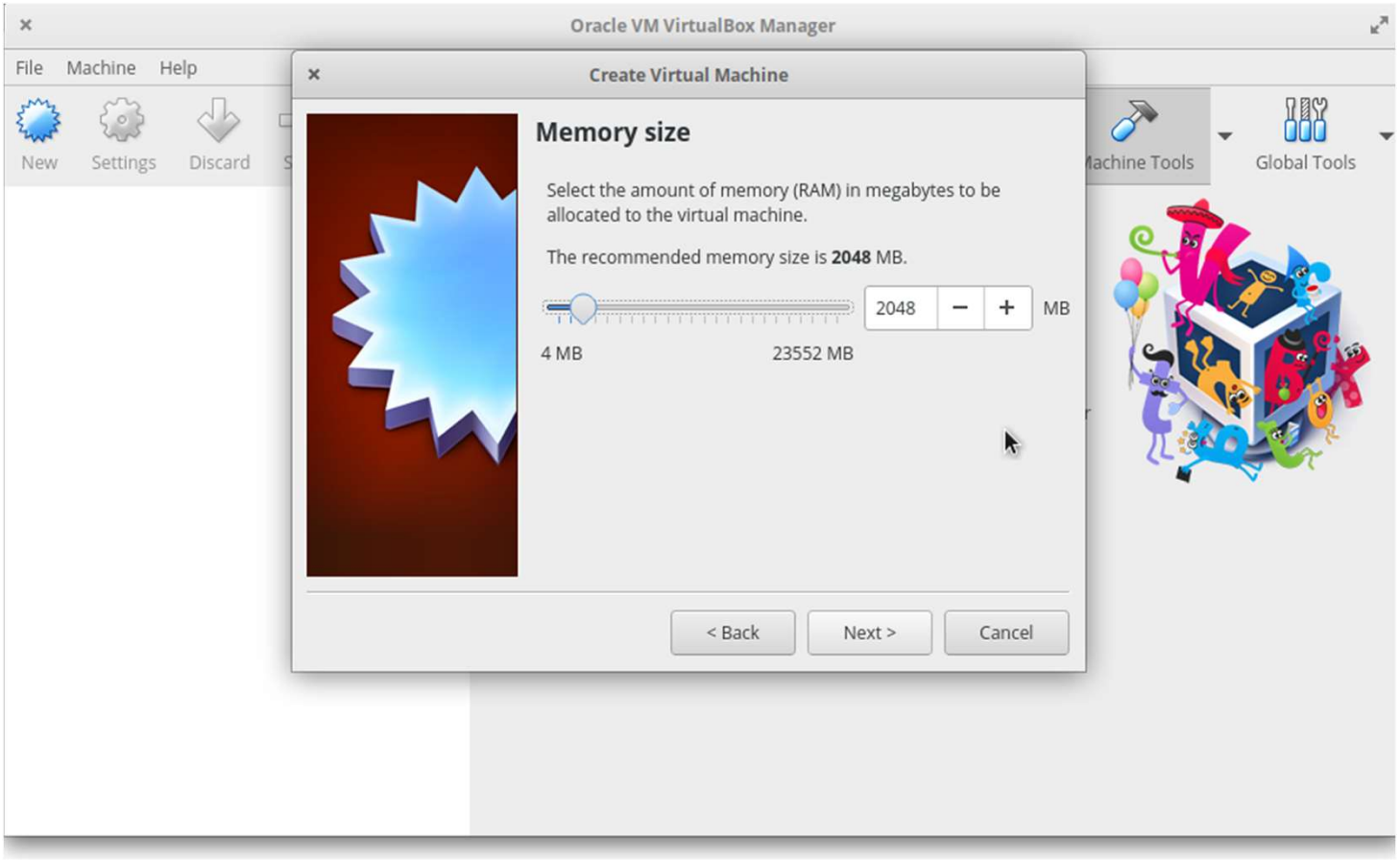
1. Click on “New” to create a virtual machine.



2. Enter the name of the virtual machine we want to set.

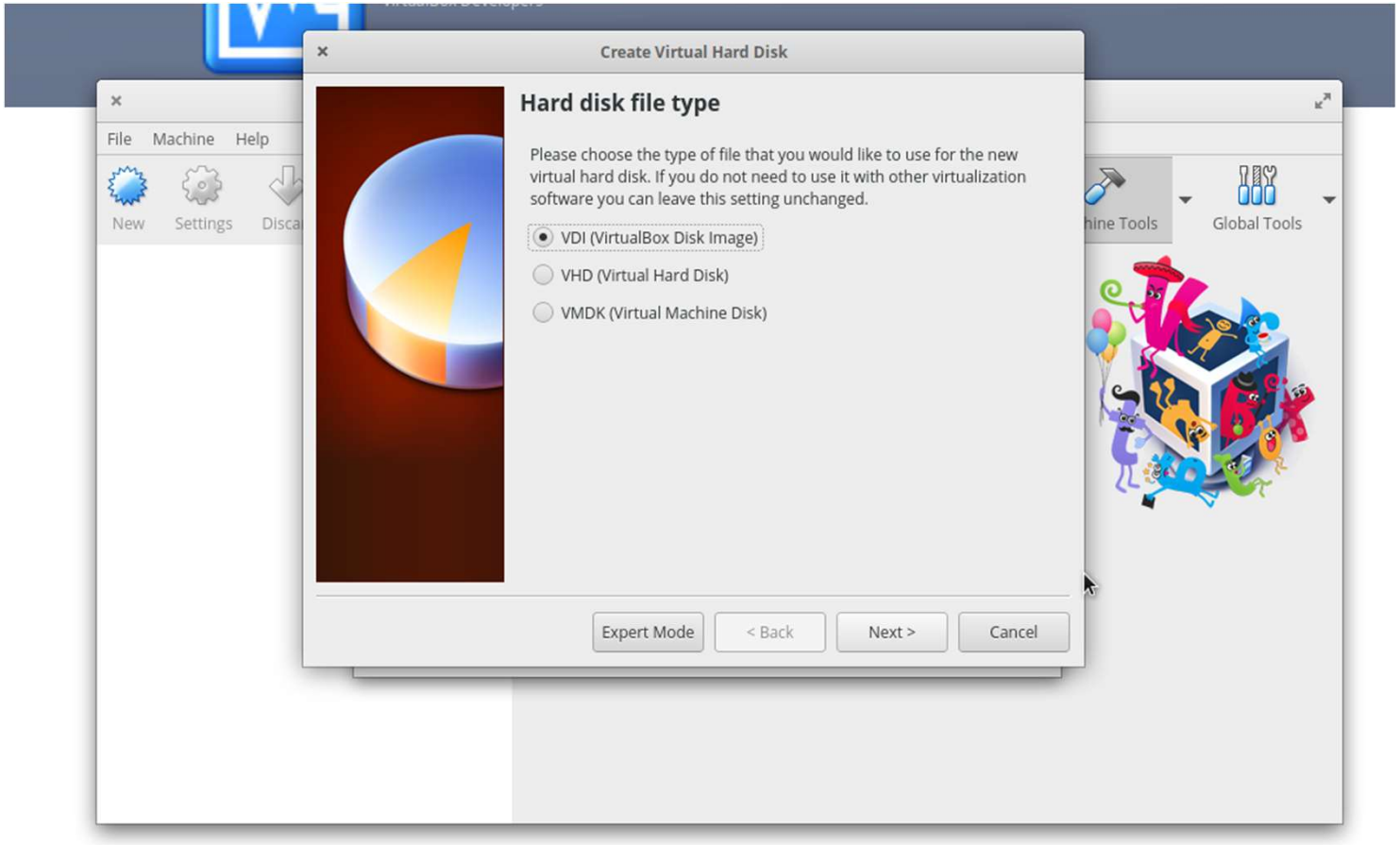


3. Set the RAM size for the virtual machine.

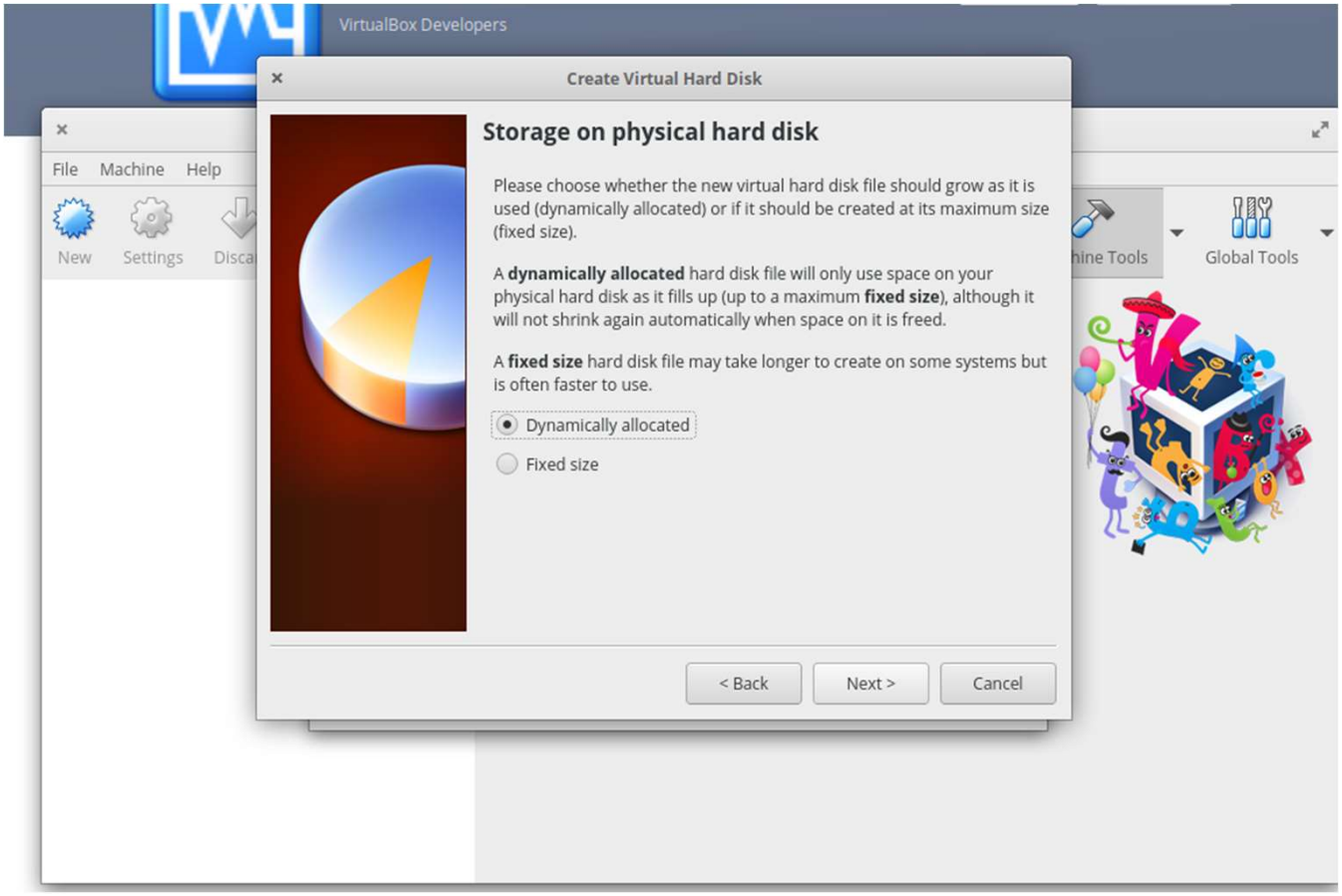


4. Select “Create a virtual hard disk now” and click on “Create”.

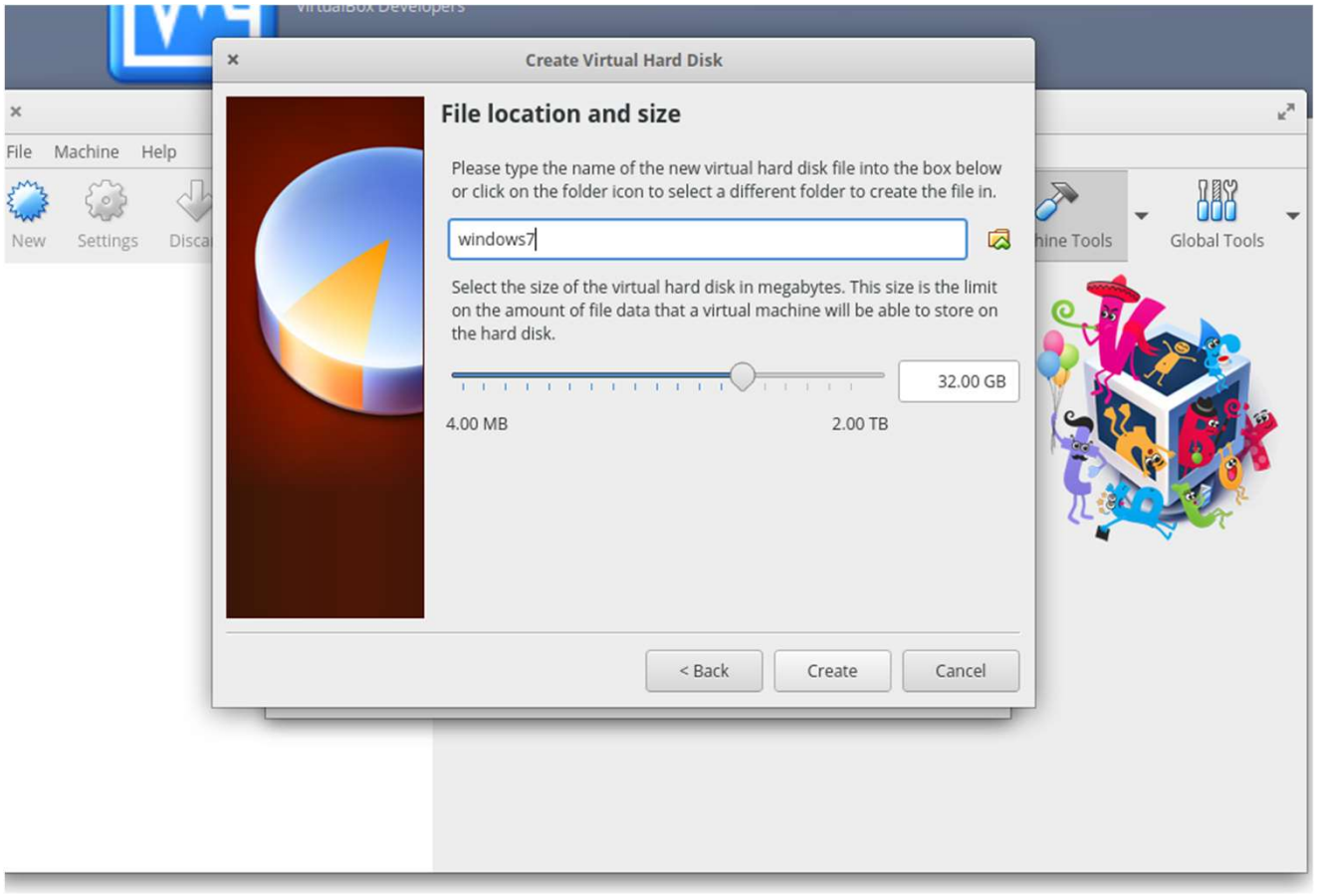
5. Select “VDI” and click on “Next”.



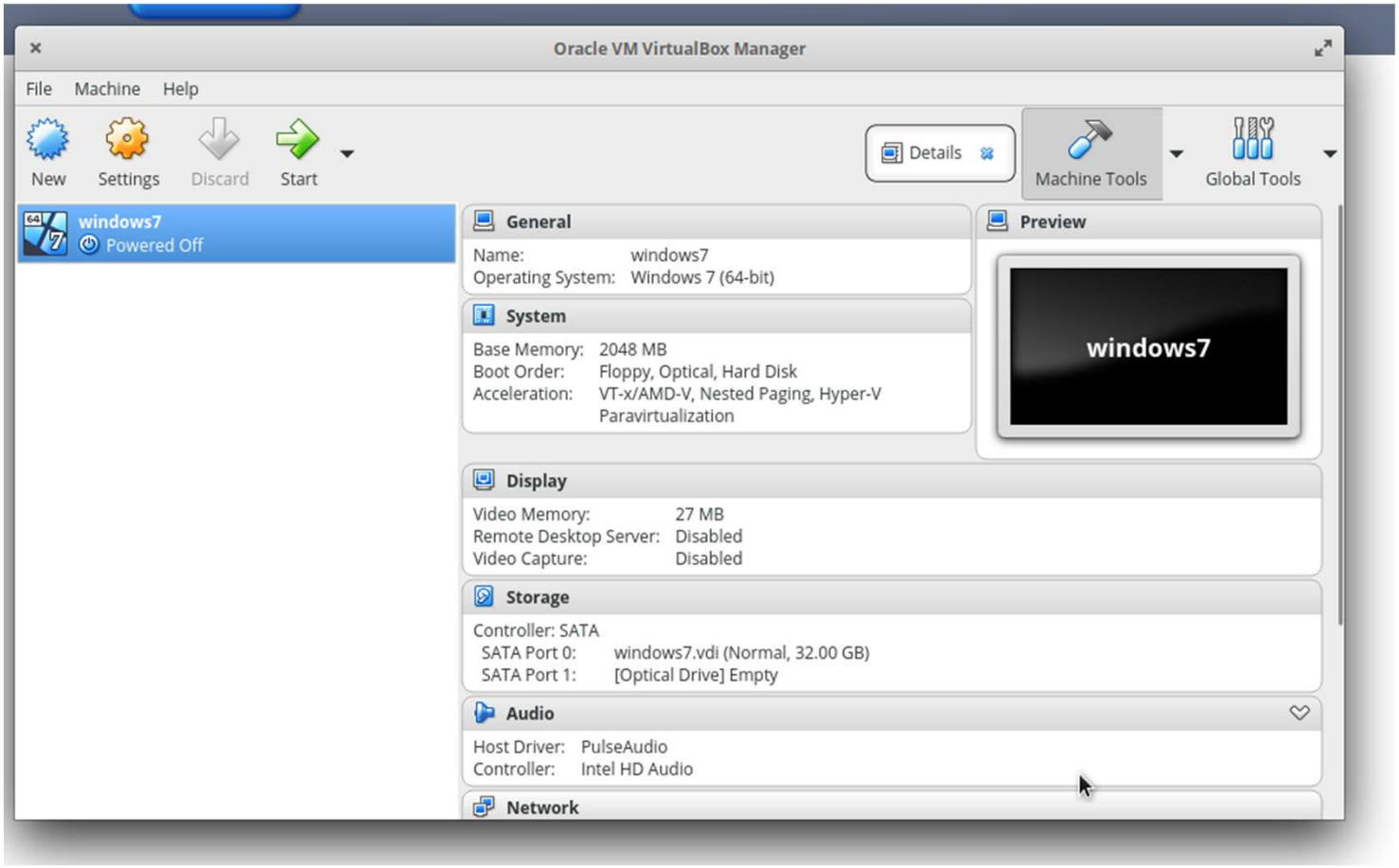
6. Select “Dynamically Allocated” and click on “Next”.



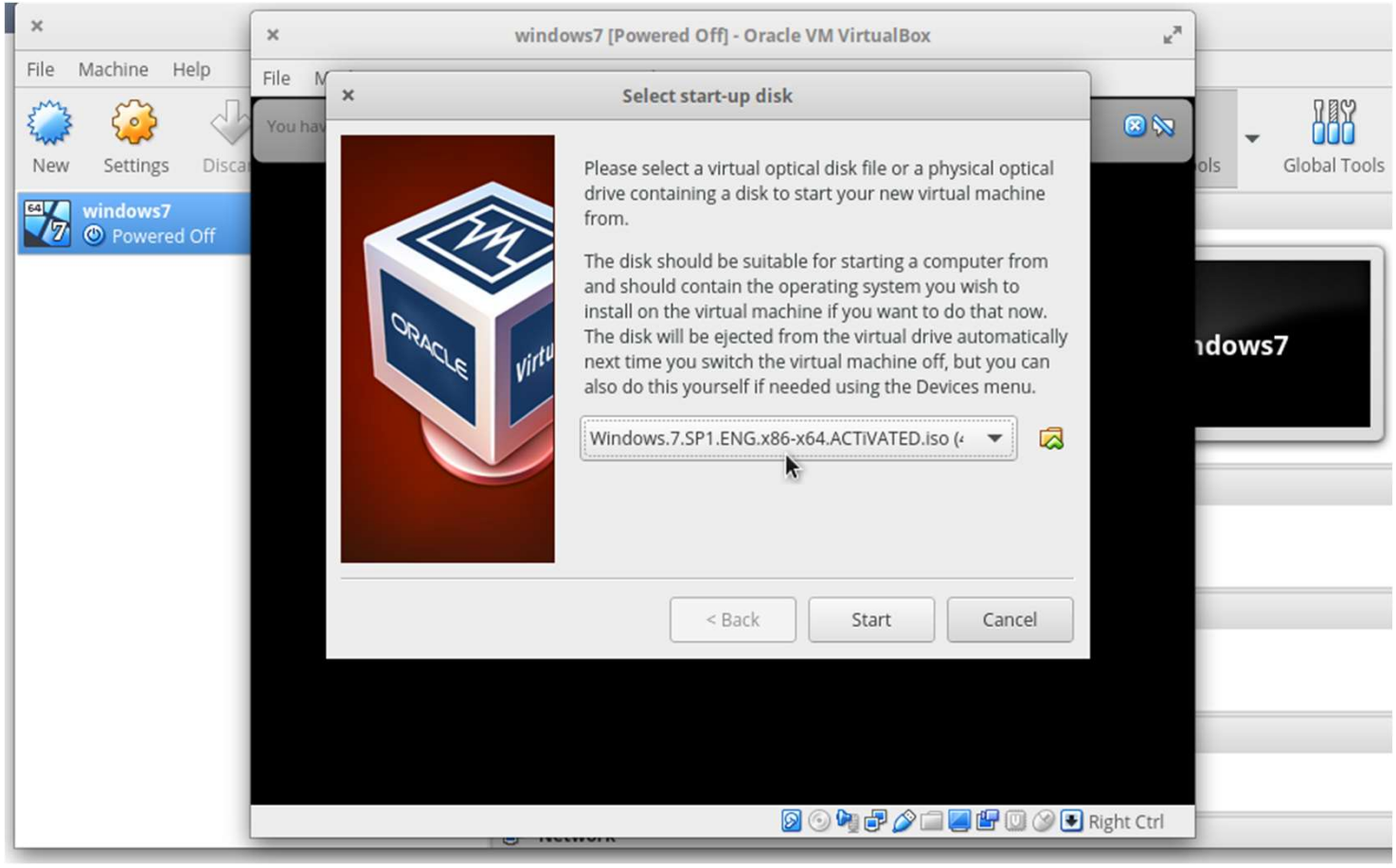
7. Enter the disk name,location and size. Then click on “Create”.



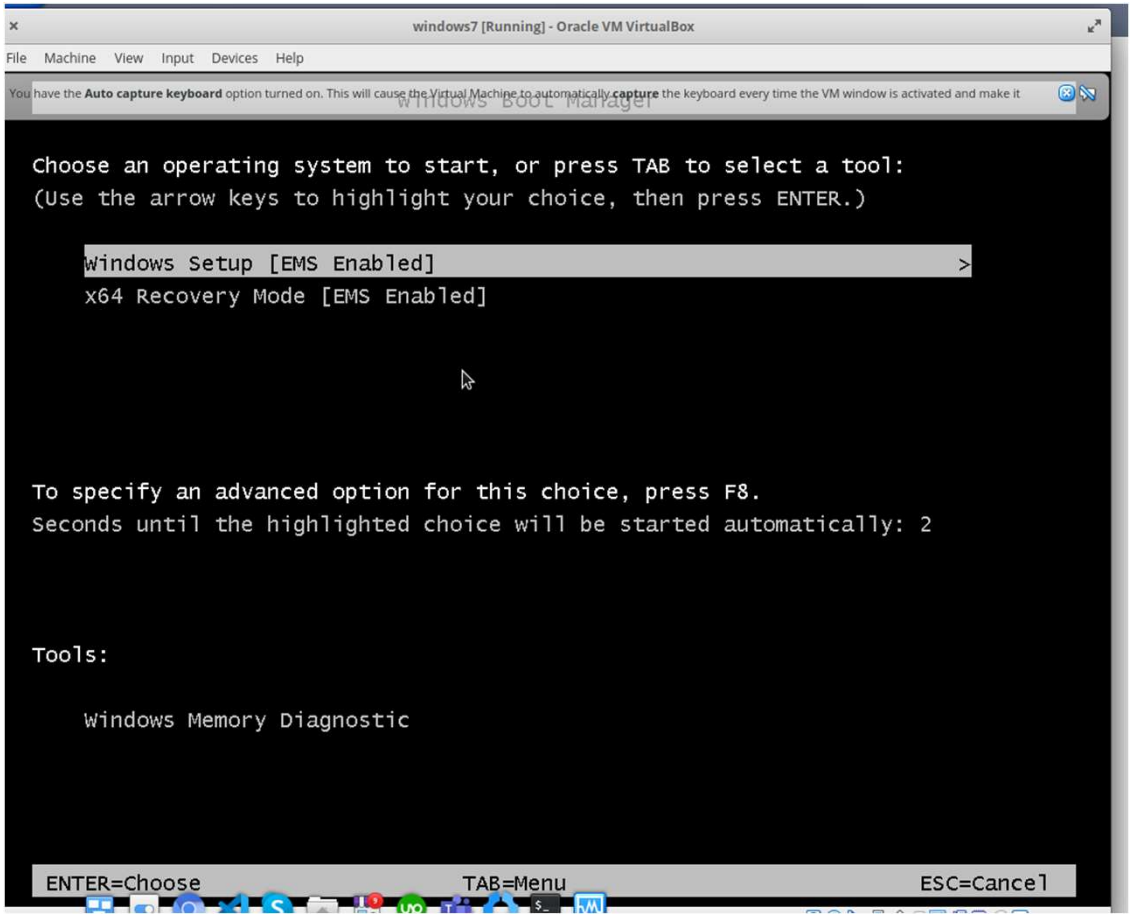
8. Select the create VM and click on “Start”.



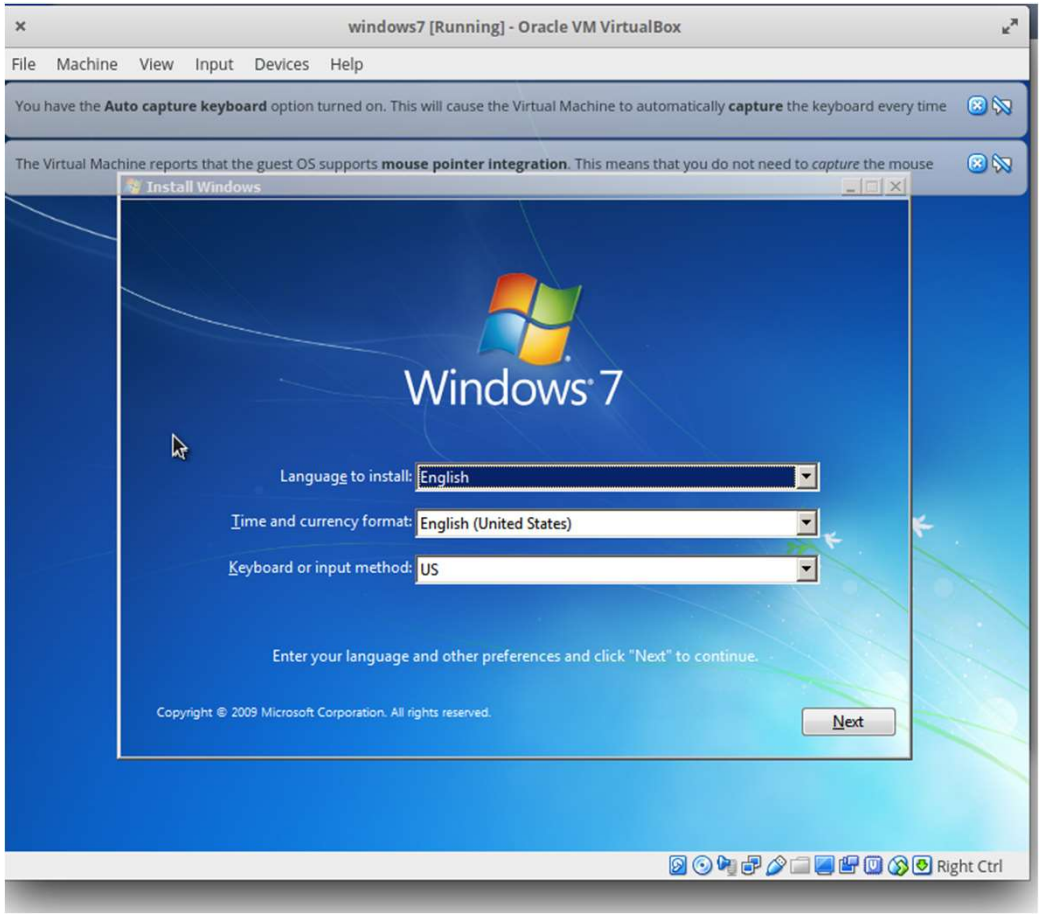
9. Select the iso and click on “Start”.



10. Select the Windows Setup



11. Start the Operating System installation procedure



VAGRANT

- Vagrant is a tool used to build development environments based on virtual machines(virtualbox).
- It is focused to create environments that are similar as possible or identical with production servers.
- It also supports Vmware Fusion.
- It is developed and managed by HashiCorp.
- In order to download and install vagrant follow this link.

1. Create a Local directory

```
devops@a-ThinkPad-T480s:~$ mkdir ubuntu
devops@a-ThinkPad-T480s:~$ cd ubuntu/
devops@a-ThinkPad-T480s:~/ubuntu$ mkdir shared
devops@a-ThinkPad-T480s:~/ubuntu$ cd shared/
```

2 . Install Ubuntu Box

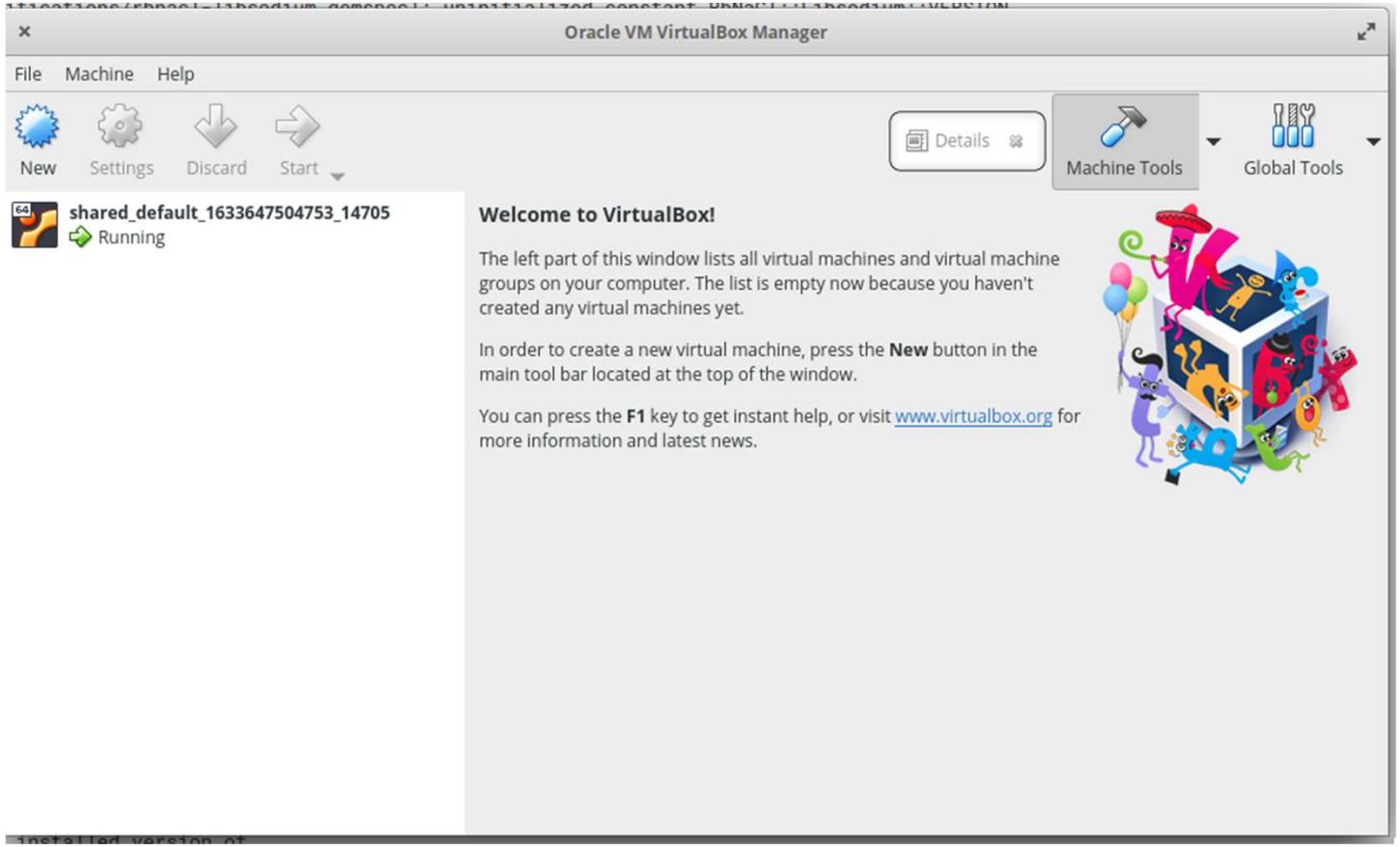
These are already created virtual machine image from which we will create our VM. More images can be found under this link.

```
devops@a-ThinkPad-T480s:~$ /usr/bin/vagrant box add ubuntu/trusty64
Invalid gemspec in [/usr/share/rubygems-integration/all/specifications/rbnacl-libsodium.gemspec]: uninitialized constant RbNaCl::Libsodium::VERSION
Invalid gemspec in [/usr/share/rubygems-integration/all/specifications/rbnacl-libsodium.gemspec]: uninitialized constant RbNaCl::Libsodium::VERSION
==> box: Loading metadata for box 'ubuntu/trusty64'
    box: URL: https://vagrantcloud.com/ubuntu/trusty64
==> box: Adding box 'ubuntu/trusty64' (v20190514.0.0) for provider: virtualbox
    box: Downloading: https://vagrantcloud.com/ubuntu/boxes/trusty64/versions/20190514.0.0/providers/virtualbox.box
    box: Progress: 59% (Rate: 4374k/s, Estimated time remaining: 0:00:43)
==> box: Successfully added box 'ubuntu/trusty64' (v20190514.0.0) for 'virtualbox'!
```

3. Now lets provision our Virtual machine.

```
devops@a-ThinkPad-T480s:~/ubuntu/shared$ /usr/bin/vagrant init ubuntu/trusty64
Invalid gemspec in [/usr/share/rubygems-integration/all/specifications/rbnacl-libsodium.gemspec]: uninitialized constant RbNaCl::Libsodium::VERSION
Invalid gemspec in [/usr/share/rubygems-integration/all/specifications/rbnacl-libsodium.gemspec]: uninitialized constant RbNaCl::Libsodium::VERSION
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
devops@a-ThinkPad-T480s:~/ubuntu/shared$ /usr/bin/vagrant up
Invalid gemspec in [/usr/share/rubygems-integration/all/specifications/rbnacl-libsodium.gemspec]: uninitialized constant RbNaCl::Libsodium::VERSION
Invalid gemspec in [/usr/share/rubygems-integration/all/specifications/rbnacl-libsodium.gemspec]: uninitialized constant RbNaCl::Libsodium::VERSION
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/trusty64' is up to date...
==> default: Setting the name of the VM: shared_default_1633647504753_14705
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
    default:
    default: Vagrant insecure key detected. Vagrant will automatically replace
    default: this with a newly generated keypair for better security.
    default:
    default: Inserting generated public key within guest...
    default: Removing insecure key from the guest if it's present...
    default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
    default: The guest additions on this VM do not match the installed version of
    default: VirtualBox! In most cases this is fine, but in rare cases it can
    default: prevent things such as shared folders from working properly. If you see
    default: shared folder errors, please make sure the guest additions within the
    default: virtual machine match the version of VirtualBox you have installed on
    default: your host and reload your VM.
    default:
    default: Guest Additions Version: 4.3.40
    default: VirtualBox Version: 5.2
==> default: Mounting shared folders...
    default: /vagrant => /home/devops/ubuntu/shared
```

We can also view the running VM using VirtualBox Manager.



4. Lets now login into the vagrant machine

```
devops@a-ThinkPad-T480s:~/ubuntu/shared$ /usr/bin/vagrant ssh
Invalid gemspec in [/usr/share/rubygems-integration/all/specifications/rbnacl-libsodium.gemspec]: uninitialized constant RbNaCl::Libsodium::VERSION
Invalid gemspec in [/usr/share/rubygems-integration/all/specifications/rbnacl-libsodium.gemspec]: uninitialized constant RbNaCl::Libsodium::VERSION
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 3.13.0-170-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Thu Oct  7 22:58:46 UTC 2021

  System load:  0.36              Processes:           82
  Usage of /:   3.6% of 39.34GB   Users logged in:     0
  Memory usage: 25%               IP address for eth0: 10.0.2.15
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

UA Infrastructure Extended Security Maintenance (ESM) is not enabled.

0 updates can be installed immediately.
0 of these updates are security updates.

Enable UA Infrastructure ESM to receive 64 additional security updates.
See https://ubuntu.com/advantage or run: sudo ua status

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


vagrant@vagrant-ubuntu-trusty-64:~$
```

Some other useful commands for vagrant :-

- To Pause the VM, navigate to the Vagrantfile and use :-
  - **vagrant suspend**
- To turn off the VM, navigate to the Vagrantfile and use :-
  - **vagrant halt**
- To destroy the VM, navigate to the Vagrantfile and use :-
  - **vagrant destroy**

More commands can be found under this link.

Thanks