

Introduction to Virtualisation

Frank Walsh

Agenda

- What's machine virtualisation?
- Why would you use virtualisation
- Implementing virtualisation
- Managing virtualisation

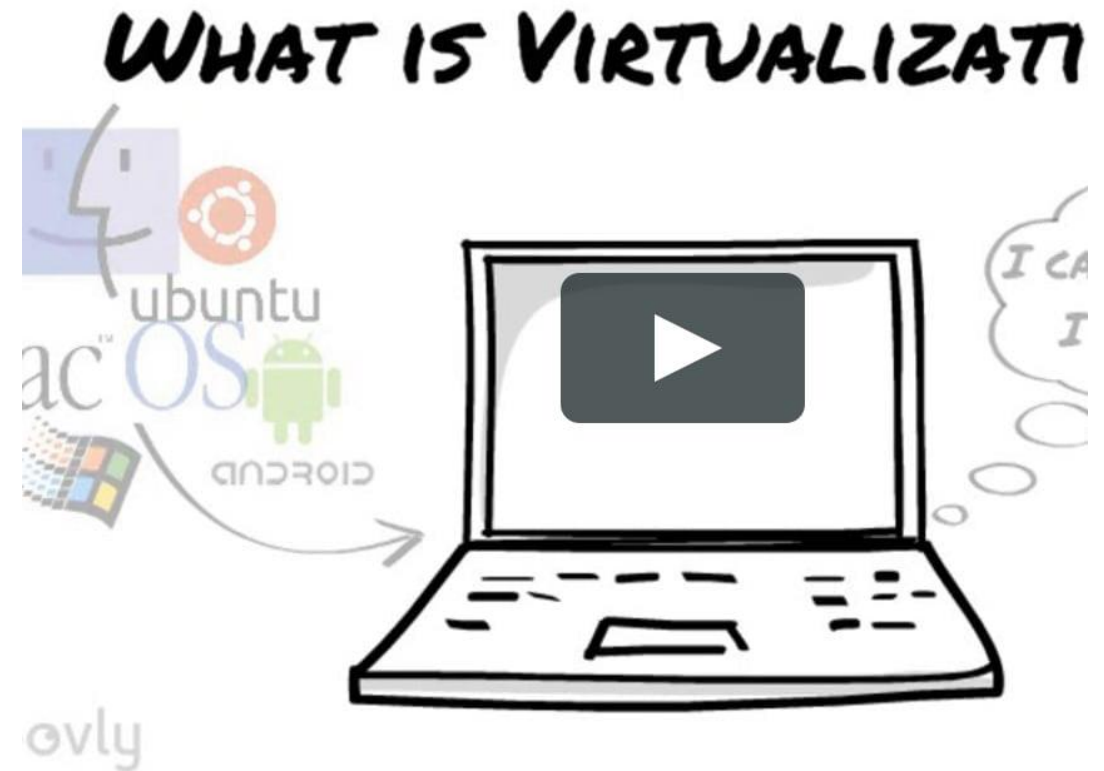
WHAT IS VIRTUALIZATION?



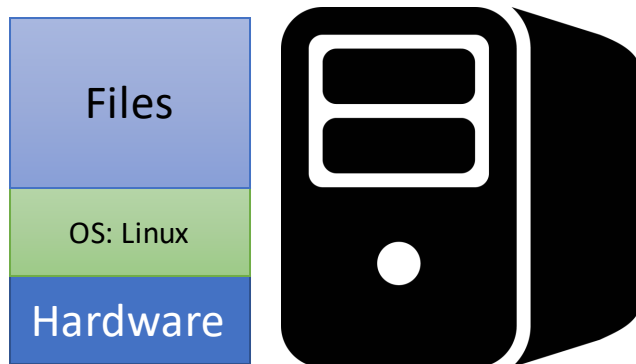
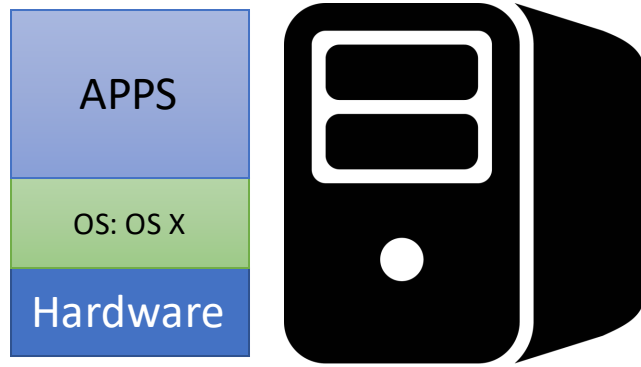
Source: <https://vimeo.com/89444438>

Introduction

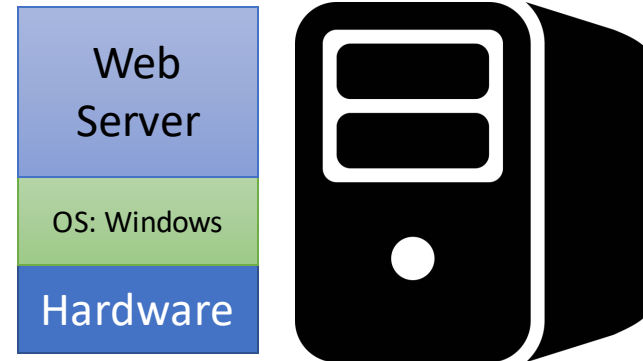
- Virtualisation: a technology used to simultaneously run multiple operating systems which are isolated from each other.
 - e.g. run both Linux and Windows on the same machine at the same time
- Virtualisation is NOT the same as dual boot:
 - Both OSes run **simultaneously**
- Note: lots of different types of virtualisation these days (network virtualisation, application visualisation)
- In this topic, we'll be talking about Operating System/Hardware Virtualisation, **aka Creating "Virtual Machines"**



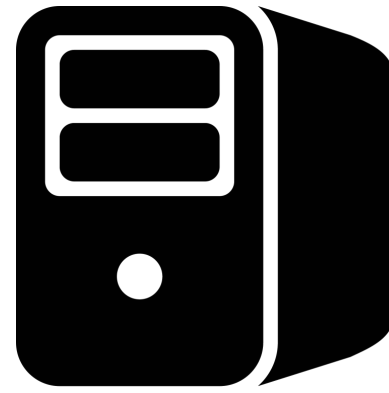
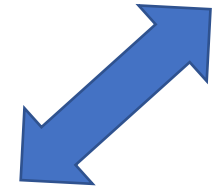
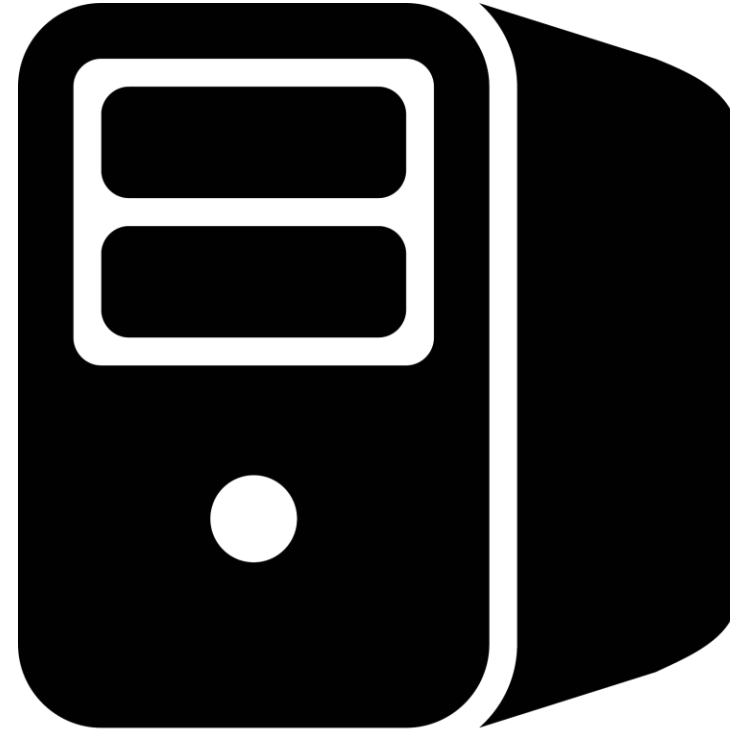
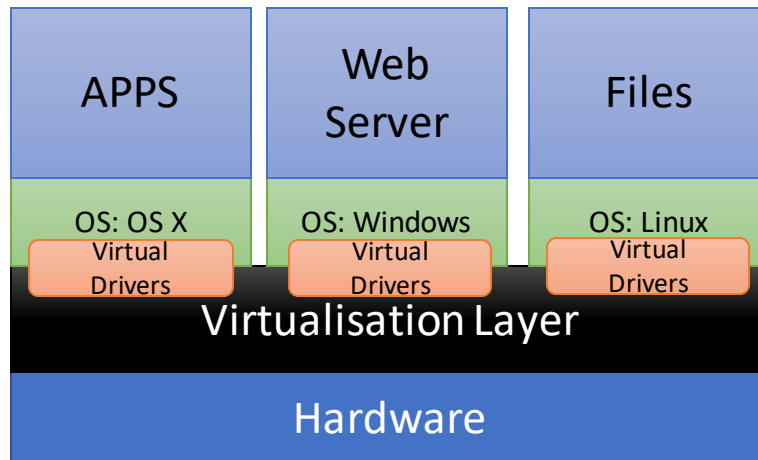
Before Virtualisation



Physical Machines



After Virtualisation



Hypervisor

- Used to create and run Virtual Machines
- Software layer that sits between Hardware and the Operating System
- Interacts with hardware
- One physical machine(**Host**) runs the Hypervisor software.
 - Provides interface to share available resources to Virtual machines

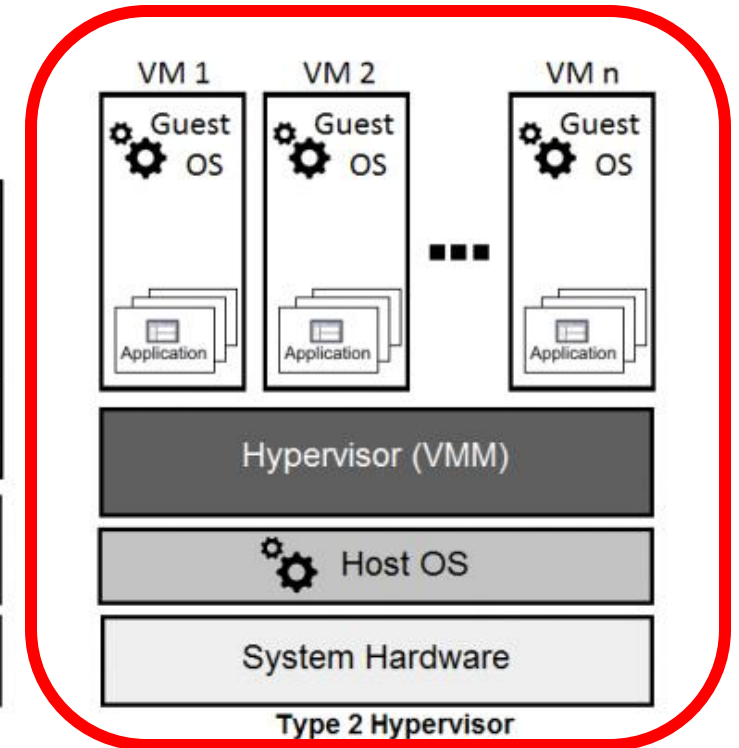
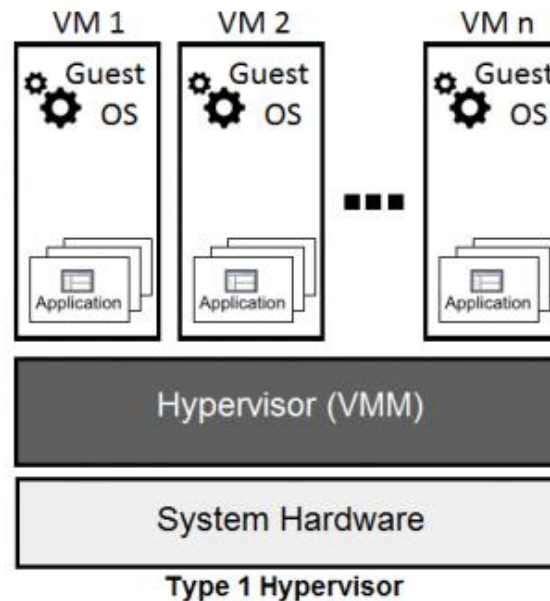
Hypervisor Types

Type 1:

run directly on the host's hardware to control the hardware and to manage guest operating systems. Sometimes referred to as "Bare Metal Hypervisor"

Type 2:

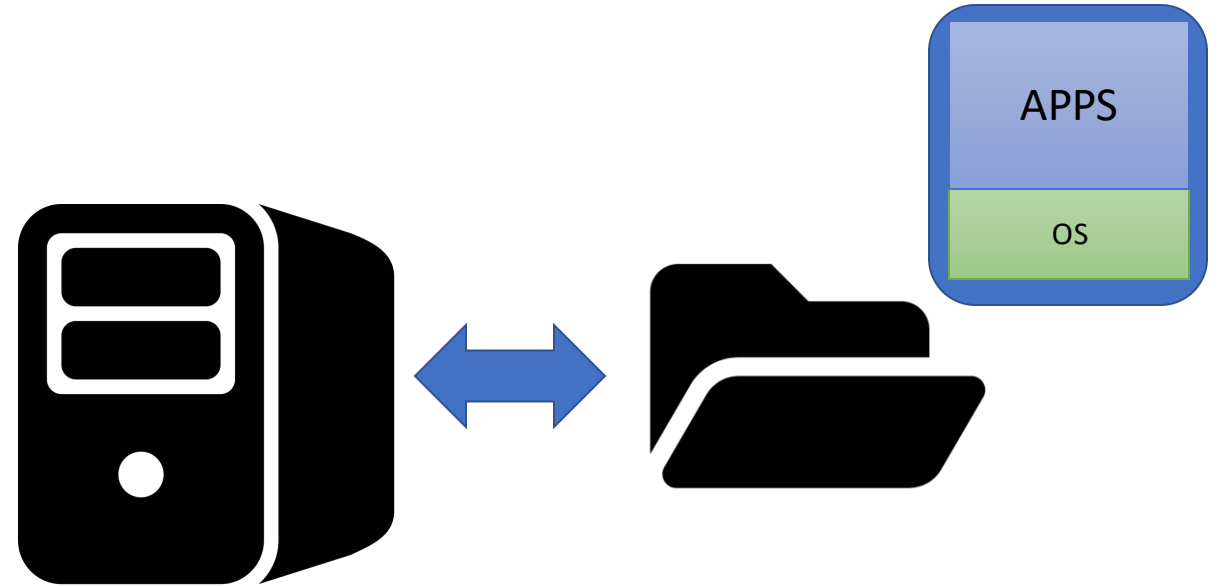
Runs on a conventional operating system (OS) just as other computer programs do. Sometimes referred to as "Hosted Hypervisor"



***This is what we're doing
with Virtual Box***

Why Virtualise

- Optimise resource usage
 - Your file server might only be at 10%
 - Your web server might be at 95%
 - Consolidate to one virtual infrastructure
- Simplification
- Partitioning
 - Share physical resources
- Portability
 - Entire virtual machine saved to file (e.g. *fxwalsh/labvm*)
- Security
 - Hardware isolated from operating systems
 - Recovery of OS can be easy – just like restoring a file
- Hardware/Platform Agnostic
 - Same Virtual Machine on different physical hosts (e.g. Mac and PC).
- Since 2007, more virtual servers were deployed than physical servers
 - source: "Virtualization Essentials", M. Portnoy, Wiley, ISBN: 978-1-118-17671-9

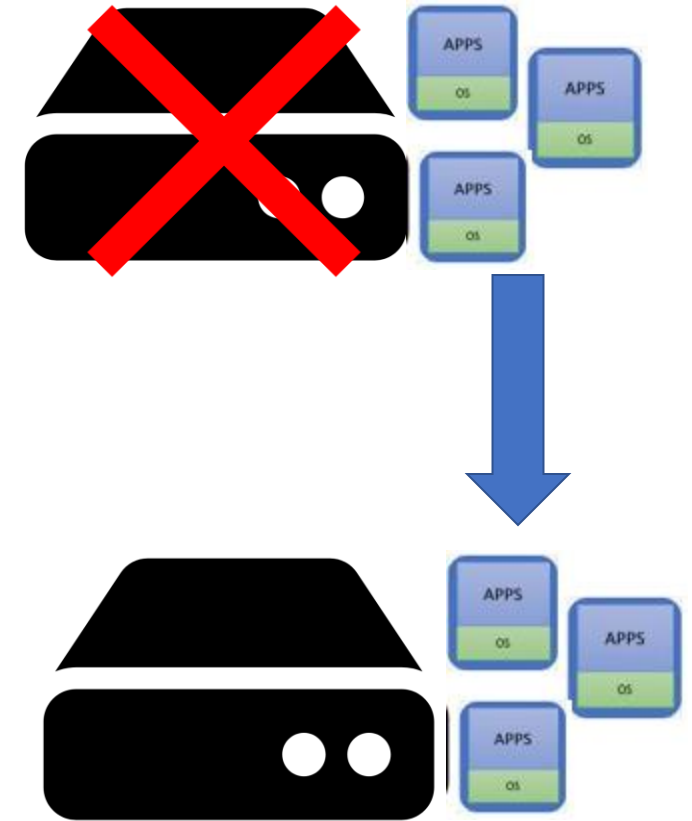


Advantages: Optimise resources

- Use the resources you need, no more no less(e.g. memory, no. of processors, hard drive size).
- Can configure some Hypervisors to dynamically optimise a Virtual Machine automatically
 - e.g. allocate more memory if needed or remove memory if not needed
- Using Hypervisor, can easily scale up/scale down resources allocated to VM
 - e.g. commercial web sites will have increased traffic at certain times such as Christmas.

Advantages: Portability/Increased availability

- What if a host machine is having problems and needs to be taken offline...
- Virtualisation enables Live Migration:
 - You can move all the running Virtual machines on a host to a different host
- End user should not notice
- Advanced feature – not all hypervisors do this
 - Usually requires planning



Advantages: Disaster Recovery

- Most Hypervisors allow you to save the state of a virtual machine
 - Often known as a "snap shot"
- Can use the snap shot to return the VM to that state.
- Take snap shot before doing something big (e.g. install service pack, new drivers etc.)
- Can always revert/roll back to snapshot if it's a disaster.
- Snap shots are a copy of the virtual hard drive
 - Takes a lot of space – get rid if not needed



Advantages: Energy

- Running less hardware usually means less power consumption
 - Virtualisation, by its nature, does this
- Hypervisors, through optimisation, can save energy
 - e.g. slow down processor if not required.
- Some hypervisors monitor and report energy use
 - Can be used for data center analysis and optimisation

Disadvantages: Complexity

- Network:
 - Creating connected VMs increases network load
 - Hypervisors use "virtual switch" and virtual network interfaces – can get complicated
- Storage:
 - VMs can be large – may need lots of storage

Disadvantages: Sprawl

- Sometimes too easy to spin up a Virtual Machine
- Sprawl is when you have too many VMs
- Remember to audit/decommission unused VMs
 - Take up space and will be an expense if in public cloud like Amazon

Virtualisation: Desktop Computer

- **Problem:** organisation can have many desktop computer environments (developers/admins/support)
 - Difficult to maintain/upgrade
- **Solution:** Virtual Desktop Infrastructure (VDI)
 - Central server pushes correct VM to each user

Typically, most of the work done on remote server

- Virtual processor on server from which it was pushed

Virtualisation: Application

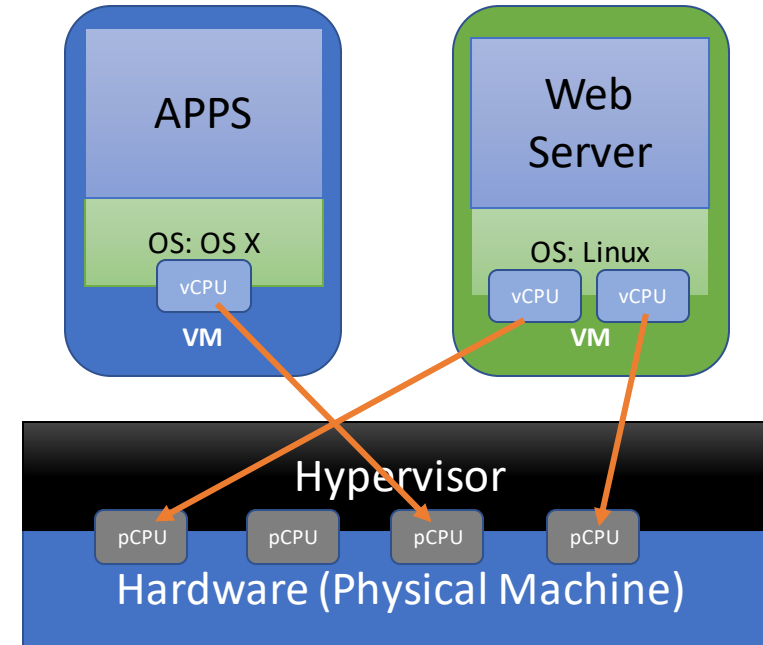
- Certain programs may not run very well, or at all, on some machines
- Install hypervisor and build VM with application you want to run
- 32 bit app might not run very well on 64 bit machine
- Run Windows app on Mac

Virtualisation: Development and Testing

- Many companies use virtualised test/dev environments
- Permanent test environment:
 - Set up to replicate "live" environment
 - Practice roll outs
 - Use snapshots to reverse/examine issues when testing
 - Assess resource requirements for certain loads
- Temporary
 - Evaluate new application/product

Virtual CPU

- Virtual Processor – vCPU
- Physical Processor – pCPU
- Pool of vCPUs sit on top of pCPUs
- Each pCPU gives some cycles to a pool of vCPUs
- Not necessarily 1 to 1 relationship
 - Oversubscription possible...



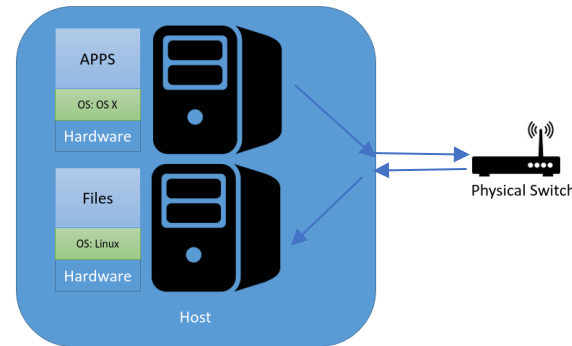
Source: <https://www.l4ka.org/77.php>

Virtual Machines: Memory

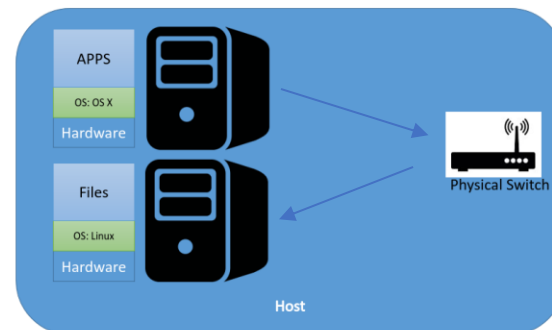
- Hypervisor provides access to physical memory
 - Guest machine gets its own section of physical memory from host
- Set fixed value or range (min and max) in hypervisor
 - Hypervisor will allocate memory as necessary
- Hypervisor can reclaim memory for host
 - Uses ballooning: insert small program in each guest that grows in memory; all memory used by ballooning program can be reclaimed by hypervisor for host

Virtual Machines: Networking

- Two VMs connecting on the same host:
 - Physical Switch: information leaves host to physical switch and returned to host

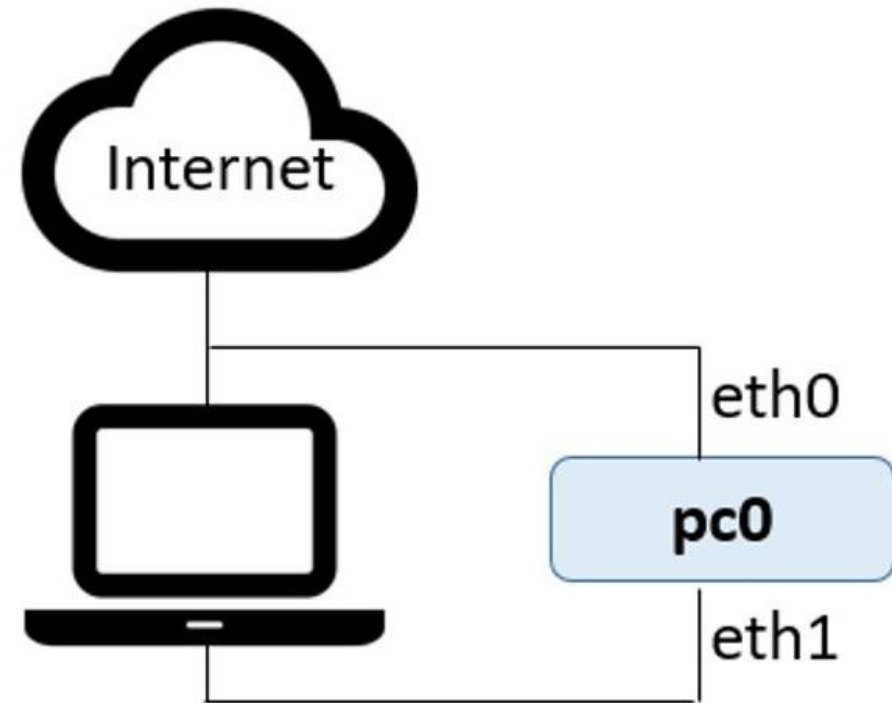


- Virtual Switch (most probable)



Virtual Machines: Networking

- Each guest will have its own IP and MAC and (virtual) network adapter(s)
- Traffic sent out through hosts physical adapter
- One traffic outside host, it's no different from any other physical machine or device.
- Two main modes for connected VMs:
 - Network Address Translation(NAT)
 - simplest way of accessing an external network from a virtual machine.
 - Bridged
 - effectively creates a new network interface in software



Virtualbox & Vagrant

Virtualbox

- *"The World's Most Popular Open Source, Cross-Platform Workstation Virtualization.."*
- Available for all common platforms (Linux/Windows/Mac)
- Hosted (Type 2) hypervisor
- Labs will do installation and VM creation
- Worth checking out chapter 1 of manual for features

<https://www.virtualbox.org/manual/ch01.html>



Vagrant

- Tool for building development environments
- Initially built on Virtualbox
- Idea to create environments as similar as possible to production servers
- Reason we're using it:
 - Idea is to create identical environment irrespective of underlying OS/hardware
 - No more "well it works on my machine..." (hopefully!)



Vagrant interaction

- Done through the **command-line interface(CLI)**.
- Vagrant interface is available using the **vagrant** command
 - vagrant command in turn has many subcommands, such as vagrant up,
- Run **vagrant** by itself to display all available subcommands.
- Run any Vagrant sub command with the -h flag to output help about that specific command.

```
frank@frank-Latitude-E5540 ~$ vagrant up -h
==> vagrant: A new version of vagrant is available: 2.1.2!
==> vagrant: To upgrade visit: https://www.vagrantup.com/downloads.html

Usage: vagrant up [options] [name|id]

Options:
  --[no-]provision              Enable or disable provisioning
  --provision-with x,y,z        Enable only certain provisioners, by type or by name.
  --[no-]destroy-on-error       Destroy machine if any fatal error happens (default to true)
  --[no-]parallel               Enable or disable parallelism if provider supports it
  --provider PROVIDER           Back the machine with a specific provider
  --[no-]install-provider       If possible, install the provider if it isn't installed
  -h, --help                    Print this help
```

Vagrant Box

- Previously built virtual machine, ready to go with installed packages/software
- Can use previously/curated boxes or you can make your own