

Virtualization

September 2020

ABOUT ME

Miguel Ángel Sotomayor

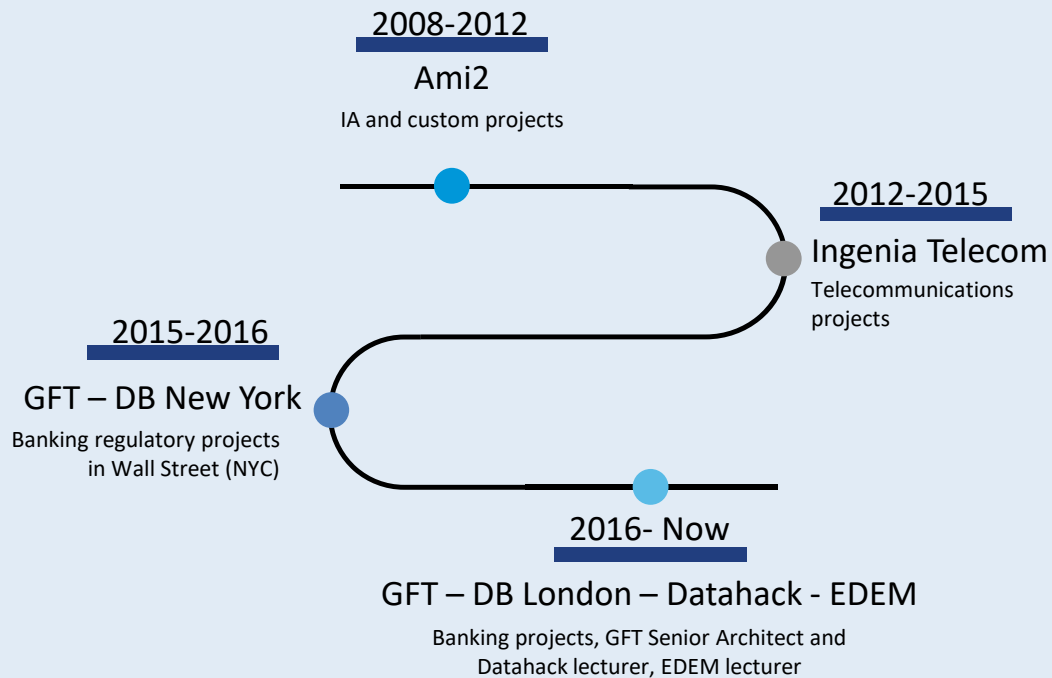
Senior Data Engineer – Data Architect

mlsy@gft.com

<https://github.com/masfworld>

<https://www.linkedin.com/in/miguelsotomayorf/>







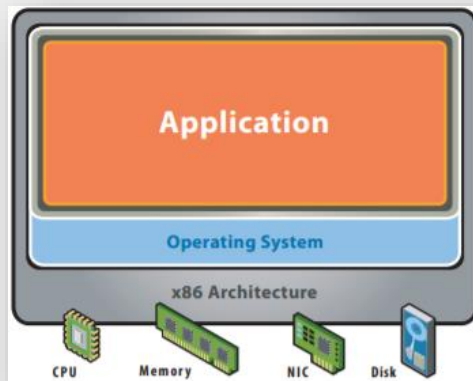
Agenda

- 1. Virtual Machines**
- 2. Hypervisor**
- 3. Enabling Virtualization**
- 4. CI/CD**
- 5. Vagrant**
- 6. Virtualization in Cloud**

Virtual machines

Before Virtualization

- Servers would traditionally run one application on one server with one operating system
 - Even one or more applications and an operating system would run on their own unique physical server
- Expensive hardware were being purchased, but not used
 - Depending on application, most of resources were unused



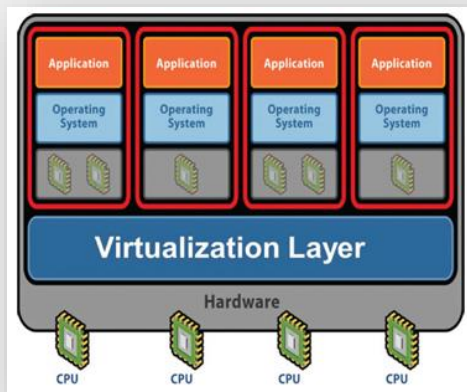
Before Virtualization

- It was not unusual to see a physical server using **less than five percent**, or even ten percent, of its CPU and/or memory
- Multiple applications in a single OS, in one operating system have an impact in terms of **security**



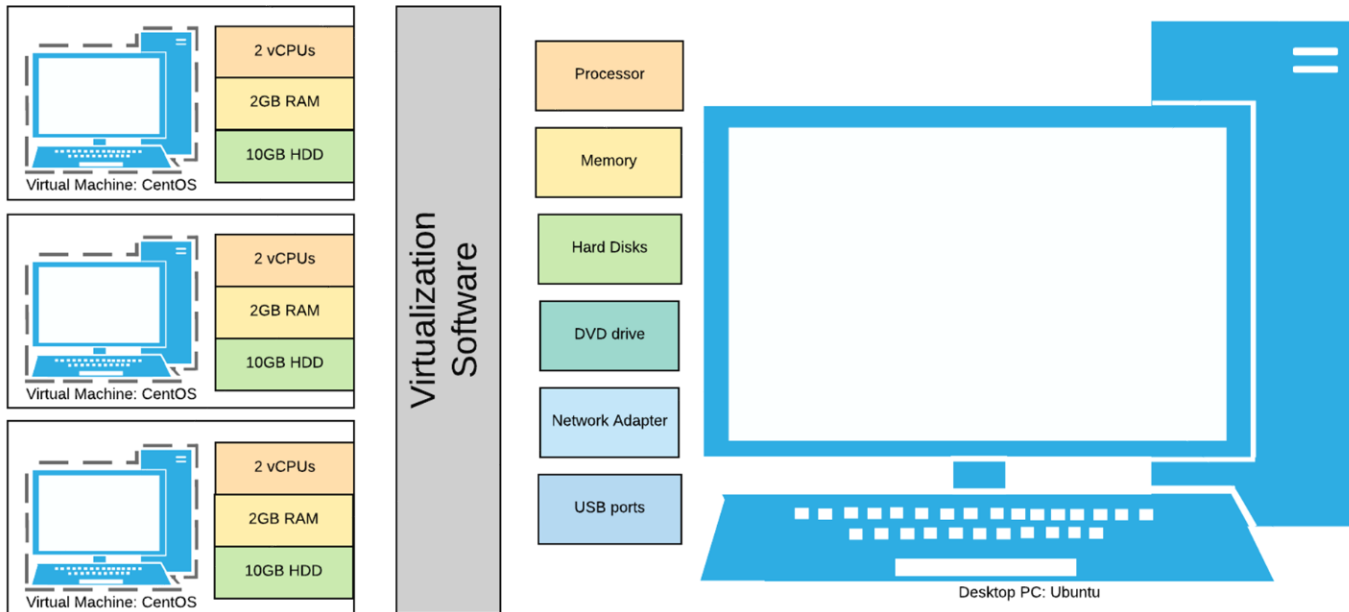
What is Virtualization?

- Virtualization uses software to create abstraction layer over computer hardware that allows the hardware elements of a single computer to be divided into multiple virtual computers, commonly called virtual machines (VMs)
- In general, the idea behind virtualization is to make many from one
- It's the single most effective way to reduce IT expenses while boosting efficiency and agility for all size businesses



What is Virtualization?

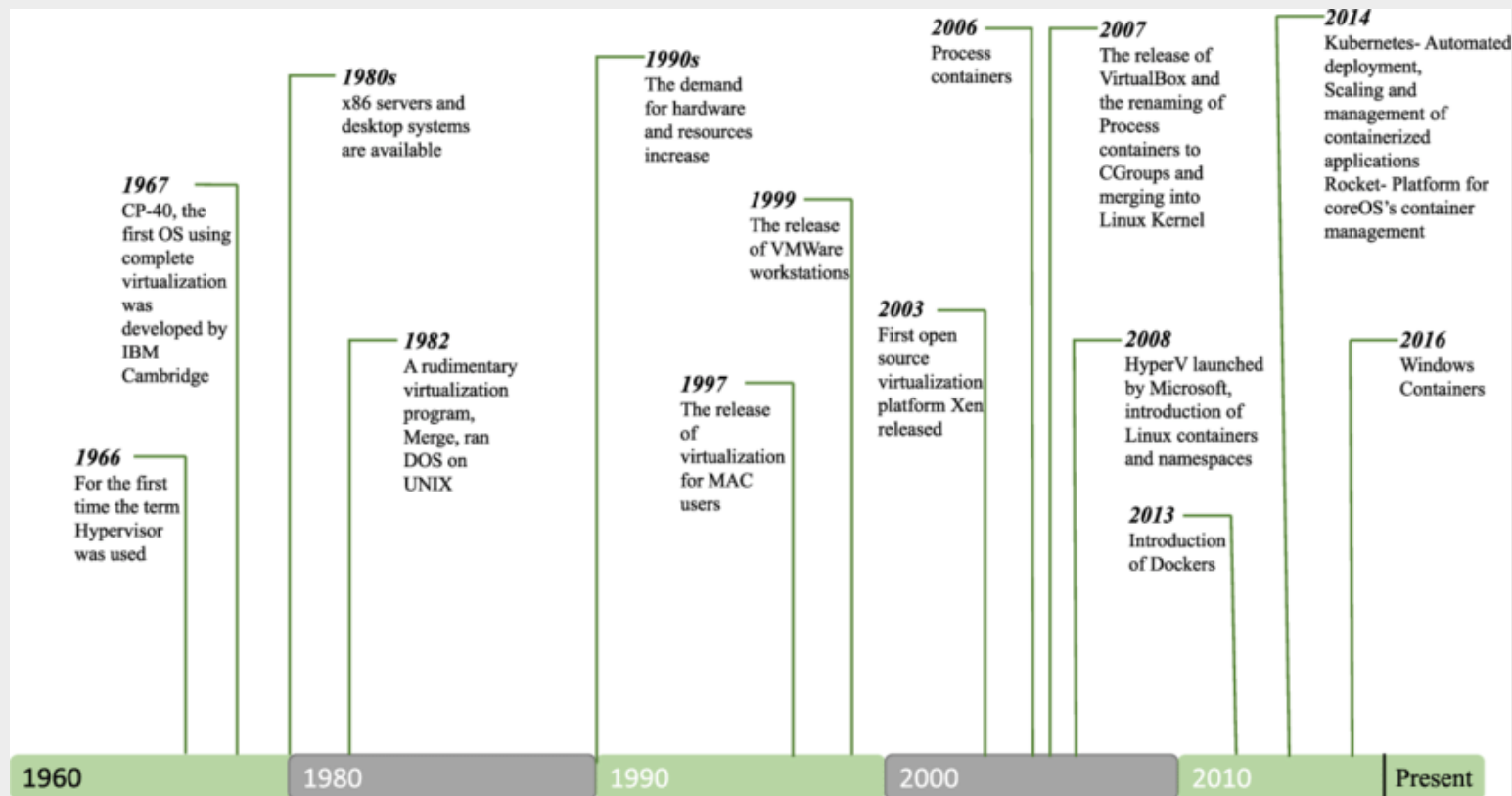
Hardware Virtualization: a Desktop Virtualization Example



Origins of Virtualization

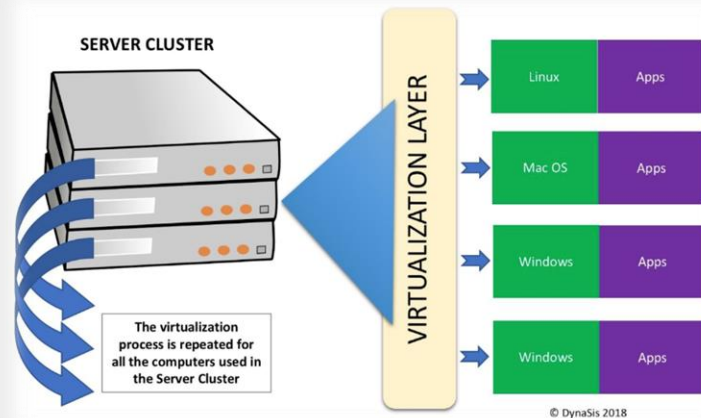
- Paper presented on time-shared computers, by **Cristopher Strachey** at the June 1959 UNESCO Information Processing Conference
- After this conference, new researches were done:
 - **MIT** developed a Compatible Time-Sharing system
 - **IBM** was pioneered in the early 1960s in terms of virtualization to solve several problems
 - The main problem that IBM wanted to solved was that each new system that they introduced was incompatible with previous systems.

Origins of Virtualization



Why Virtualization?

- Issues with traditional systems:
 - Software and hardware tightly coupled
 - Running multiple applications on same machine often creates conflict
 - Underutilized resources
 - Inflexible and costly infrastructure
- Virtualization lets you run more applications on fewer physical servers.
 - Rather than **one** application running on **one** server with **one** operating system, **multiple** VMs run **multiple** apps and operating systems on **one** physical server.



Types of Virtualization

- **Server Virtualization**

- Enables multiple operating systems to run on a single physical server
- Reduced operating costs
- Higher server availability

- **Network Virtualization**

- Reproducing a physical network
- Allows applications to run on a virtual network

- **Desktop Virtualization**

- Enables IT organizations to respond faster to changing workplace needs and emerging opportunities

- **Storage Virtualization**

- Logical view of the physical storage resources

Hypervisor

Hypervisor

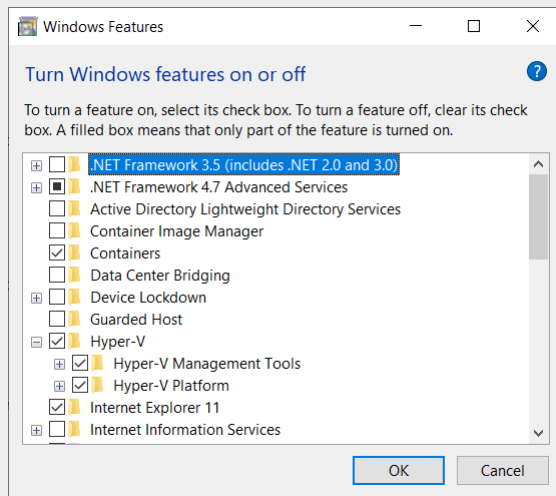
- A hypervisor is computer software, firmware or hardware that **creates and runs virtual machines**
- It's a process that **separates** a computer's operating system and applications from the underlying physical hardware
 - Even though VMs can run on the same physical hardware, they are still logically separated from each other
 - That means that if one VM experiences an error, crash or malware attack, it doesn't extend to other VMs on the same machine



Enabling virtualization

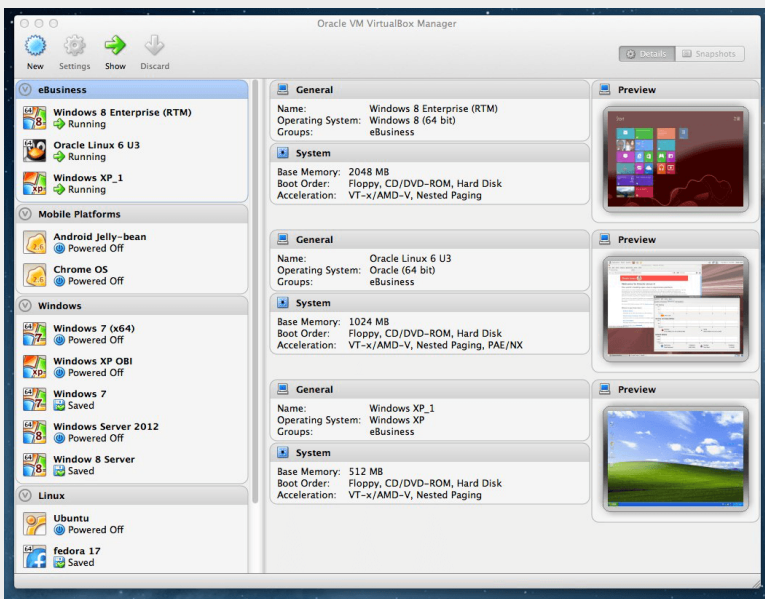
Enabling Hyper-V in Windows 10 Pro

- <https://docs.microsoft.com/es-es/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v>



Virtualbox

- <https://www.virtualbox.org/wiki/Downloads>



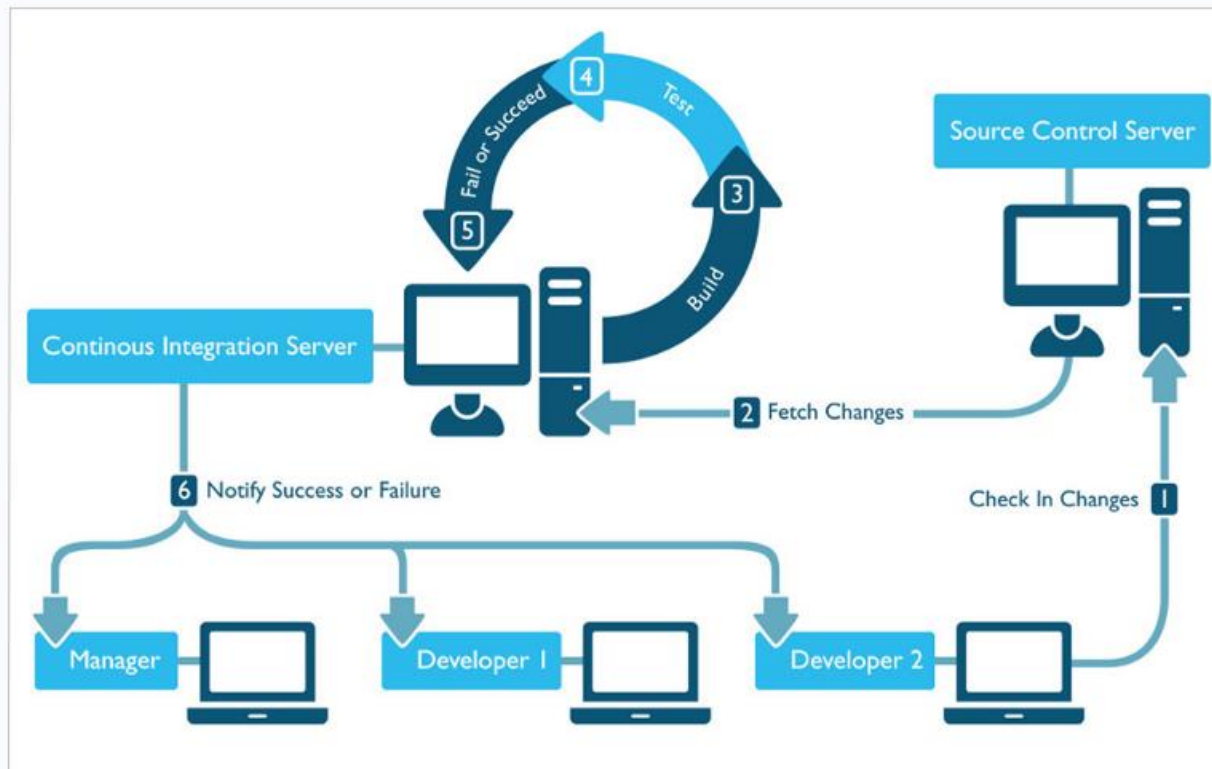
CI/CD

Continuous Integration - CI

- **Continuous integration** is a coding philosophy and set of practices that drive development teams to implement small changes and check in code to version control repositories frequently
- The idea is to establish a **consistent and automated** way to build, package, and test applications
- The developer's changes are validated by creating a build and running **automated test** against the build.

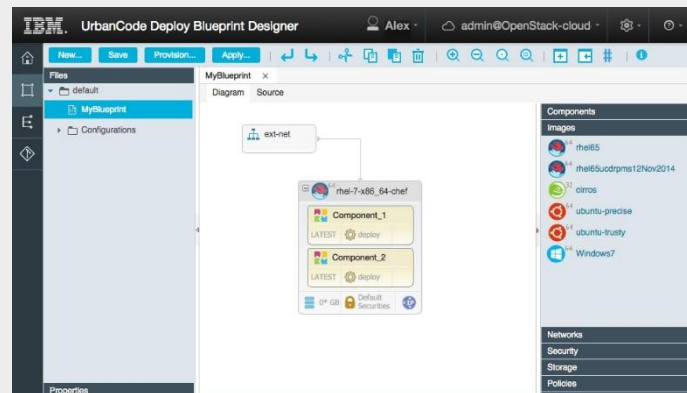


Continuous Integration - CI

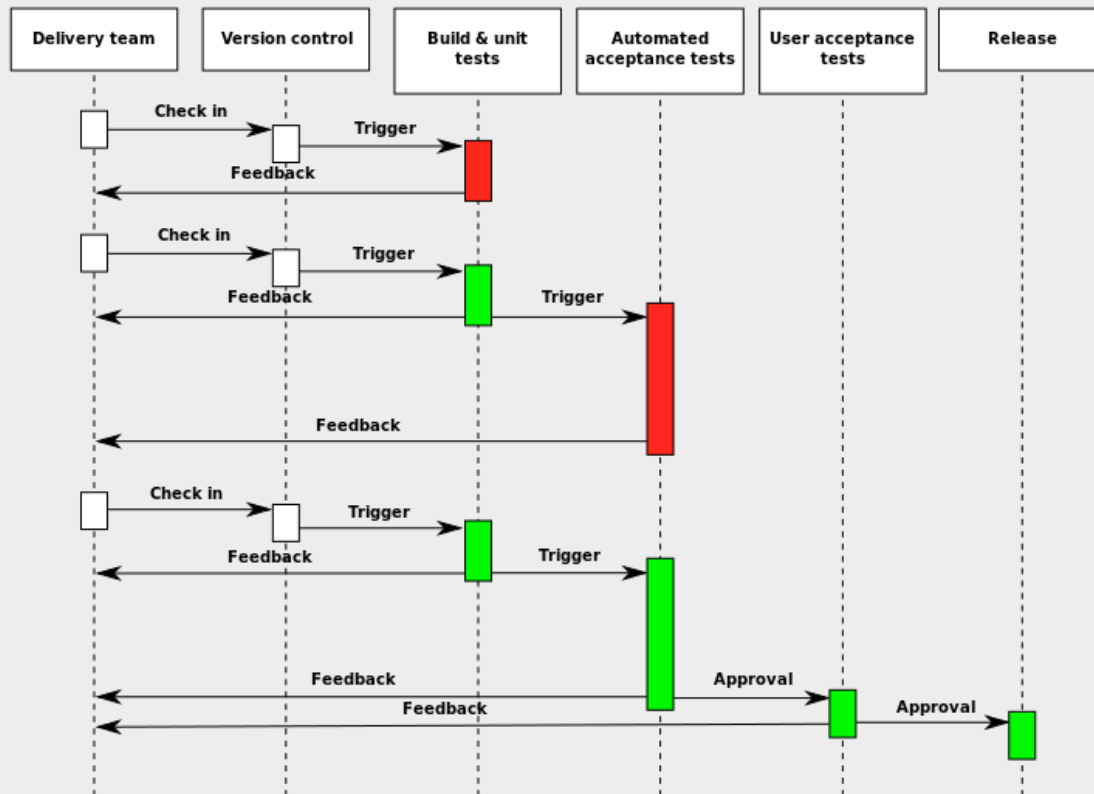


Continuous Delivery - CD

- **Continuous delivery** picks up where CI ends. CD **automates** the delivery of applications to selected infrastructure environments.
- Most teams work with multiple environments other than productions, such as development and testing environments
- CD ensures there is an automated way to push code changes to them
- You can **decide to release** daily, weekly or whatever suits your business requirements

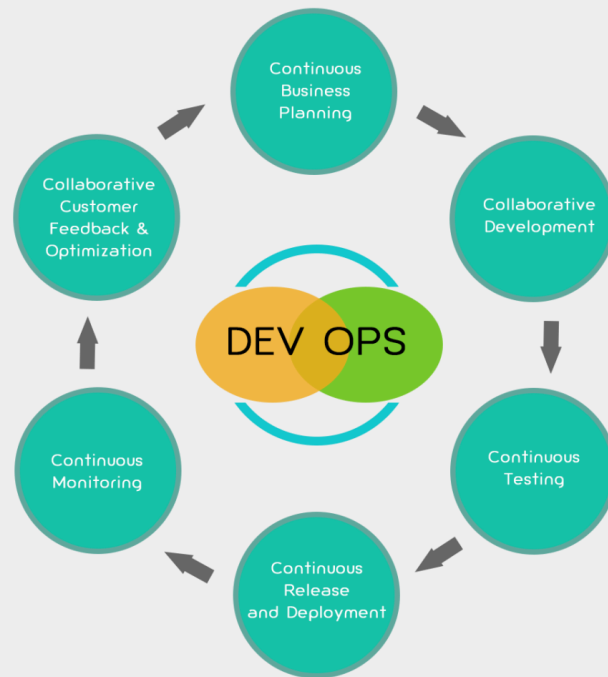


CD/CI



CD/CI - Virtualization

- **Virtualization enables** DevOps teams to develop and test within simulated environments that run the full gamut of devices available to consumers while also testing deployment on virtual live environments.
- This level of accuracy in testing makes for vastly reduced deployment times and increased stability



Automation



Configuration Management



Compute Virtualization



Data Virtualization



Vagrant

Vagrant

- **Vagrant is a tool** for working with virtual environments
- Provides a **simple and easy to use command-line** client for managing these environments
- An interpreter for the text-based definitions of what each environment looks like called **Vagrantfiles**
- It may be useful to a wide range of people working on different kinds of tasks



Vagrant

```

1 # Example 1
2 #
3 # Single box with some custom configuration.
4 #
5 # NOTE: Make sure you have the precise32 base box installed...
6 # vagrant box add precise32 http://files.vagrantup.com/precise32.box
7
8 Vagrant.configure("2") do |config|
9   config.vm.box = "precise32"
10  config.vm.hostname = "myprecise.box"
11  config.vm.network :private_network, ip: "192.168.0.42"
12 end

```

<https://github.com/patrickdlee/vagrant-examples>

```

1 # Example 5
2 #
3 # Separate Web and database servers serving up static/dynamic sites via Puppet.
4 #
5 # NOTE: Make sure you have the precise32 base box installed...
6 # vagrant box add precise32 http://files.vagrantup.com/precise32.box
7
8 nodes = [
9   { :hostname => 'ex5web', :ip => '192.168.0.42', :box => 'precise32' },
10  { :hostname => 'ex5db', :ip => '192.168.0.43', :box => 'precise32', :ram => 512 }
11 ]
12
13 Vagrant.configure("2") do |config|
14   nodes.each do |node|
15     config.vm.define node[:hostname] do |nodeconfig|
16       nodeconfig.vm.box = "precise32"
17       nodeconfig.vm.hostname = node[:hostname] + ".box"
18       nodeconfig.vm.network :private_network, ip: node[:ip]
19
20       memory = node[:ram] ? node[:ram] : 256;
21       nodeconfig.vm.provider :virtualbox do |vb|
22         vb.customize [
23           "modifyvm", :id,
24           "--cpucapexecutioncap", "50",
25           "--memory", memory.to_s,
26         ]
27       end
28     end
29   end
30
31   config.vm.provision :puppet do |puppet|
32     puppet.manifests_path = "puppet/manifests"
33     puppet.manifest_file = "site.pp"
34     puppet.module_path = "puppet/modules"
35   end
36 end

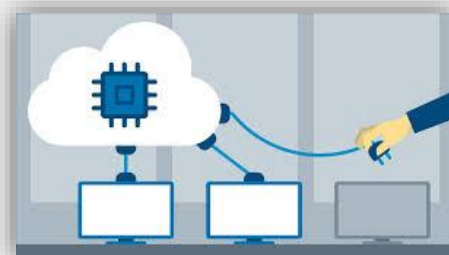
```

Virtualization in Cloud

Virtualization in Google Cloud

■ Google Compute Engine - GCE

- It's the Infrastructure as a Service (IaaS)
- Enables users to launch **Virtual Machines** on demand
- VMs can be launched from the standard images or custom images
 - An image is a persistent disk that contains the OS and root file system that is necessary for starting an instance



Virtualization in AWS

- **Amazon Elastic Compute Cloud – EC2**
 - Provides scalable computing capacity in AWS
 - Launch as many or as few virtual servers (instances) as you need
 - Preconfigurable templates for your instances
 - Amazon Machine Images (AMIs)



Virtualization in Azure

■ Azure Virtual machines

- Provides scalable computing capacity in Azure
- Deploying virtual machines featuring up to 416vCPUS and 12TB memory
- Templates for your instances

