@ More About Docker

Docker is a free piece of S/W that allows users to develop and manage apps in an environment that takes **advantage of Linux-based software containers**.
It's a great tool because it gives developers the ability to perform app management and development **without needing to use a virtual server or any extra hardware**. This means You can get **various IT infrastructure components to work together** with much less troubleshooting than other methods.

The **container technology** was originally developed with the aim of running several virtual operating systems in isolated environments on the same kernel. (known as **full-system containers**)

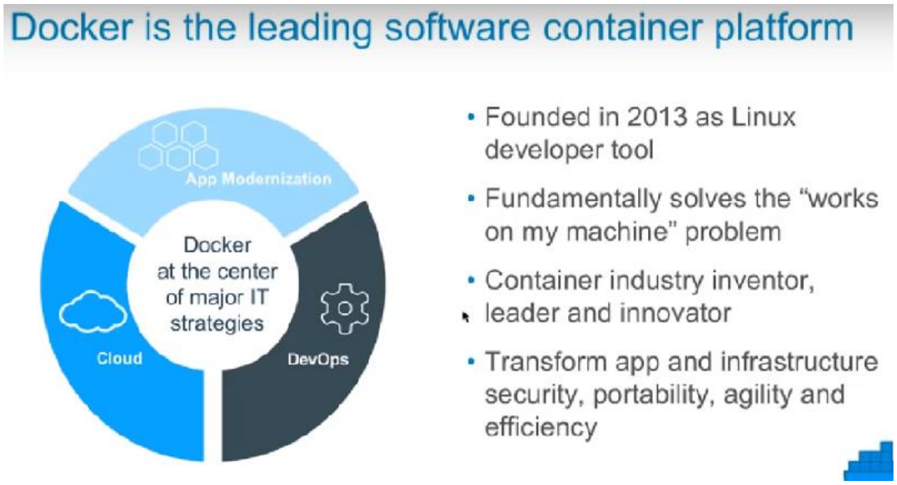
The **Docker container** platform, *on the other hand*, focuses on so-called **application containers**, in which each application runs in its own virtual environment.

While **full-system containers** are designed so that different processes can be executed in them, an **application container** always contains just one single process.

Extensive applications are therefore implemented as multi-container apps with Docker.

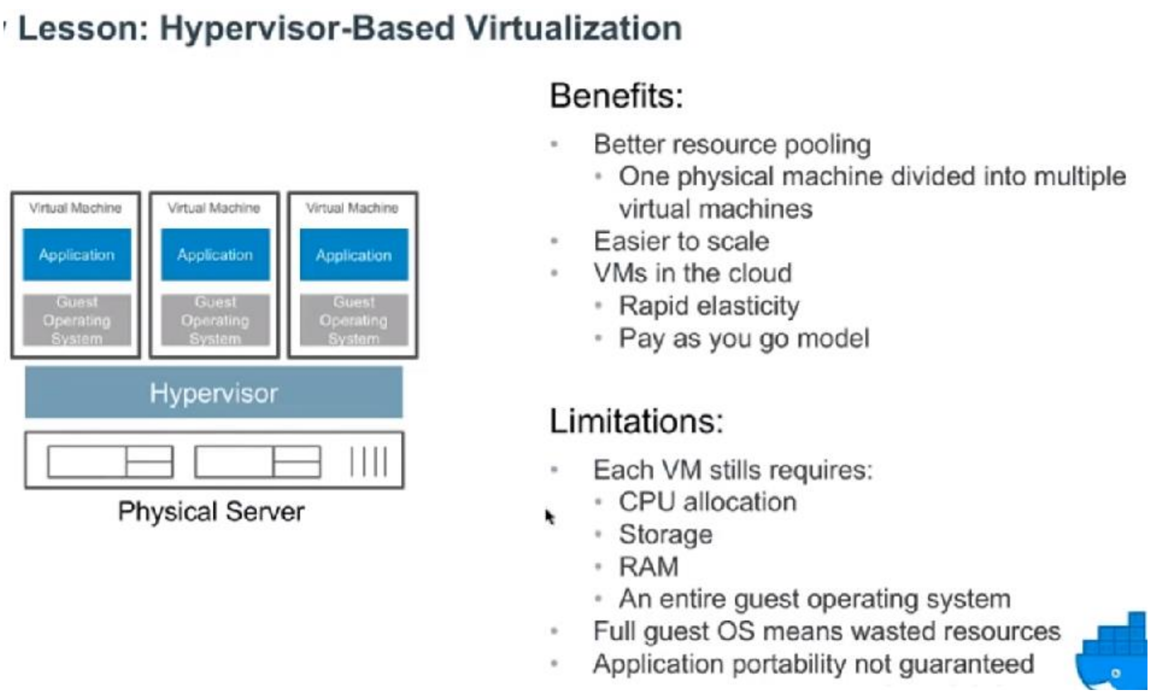
Pros : Efficient Tools, Affordable, Requires Limited System Resources, Open-Source

Cons: Large Learning Curve, Requires Perfect Syntax



Docker is the leading software container platform

- Founded in 2013 as Linux developer tool
- Fundamentally solves the "works on my machine" problem
- Container industry inventor, leader and innovator
- Transform app and infrastructure security, portability, agility and efficiency



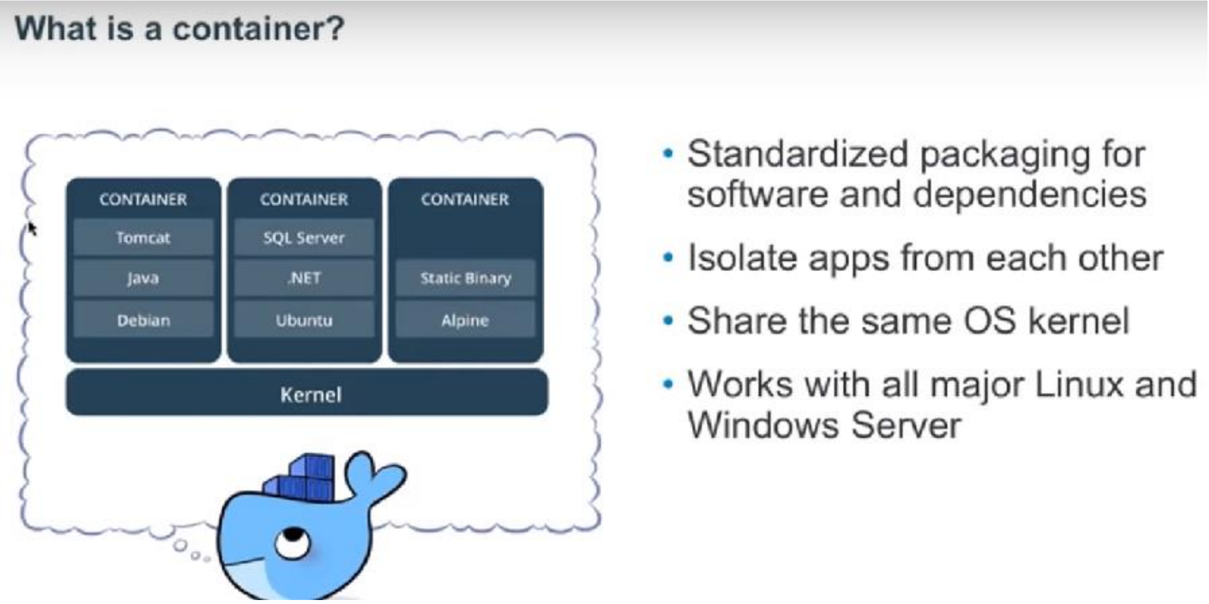
Lesson: Hypervisor-Based Virtualization

Benefits:

- Better resource pooling
  - One physical machine divided into multiple virtual machines
- Easier to scale
- VMs in the cloud
  - Rapid elasticity
  - Pay as you go model

Limitations:

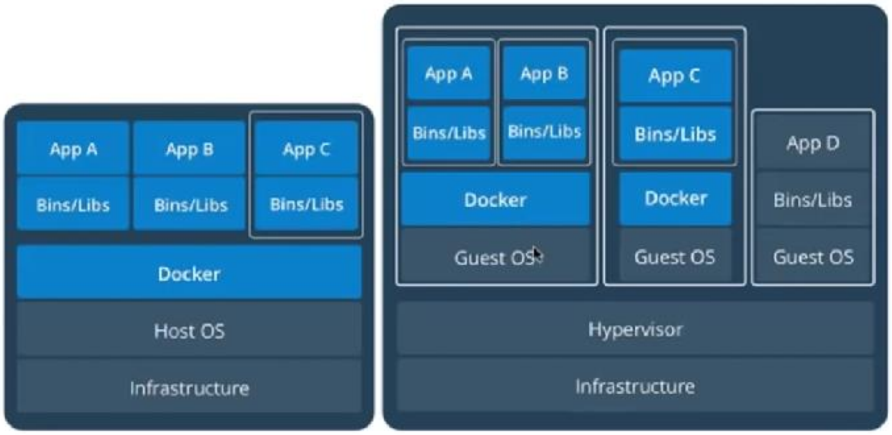
- Each VM stills requires:
  - CPU allocation
  - Storage
  - RAM
  - An entire guest operating system
- Full guest OS means wasted resources
- Application portability not guaranteed



What is a container?

- Standardized packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works with all major Linux and Windows Server

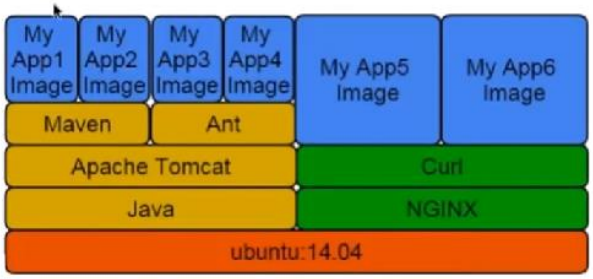
Containers and VMs Together

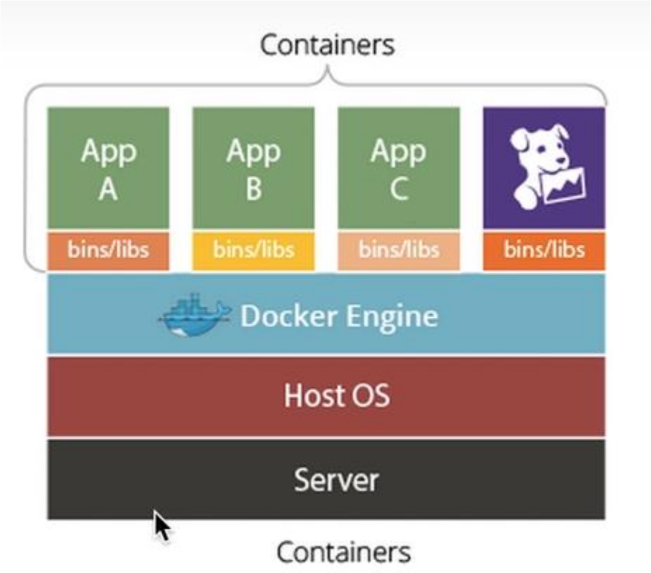


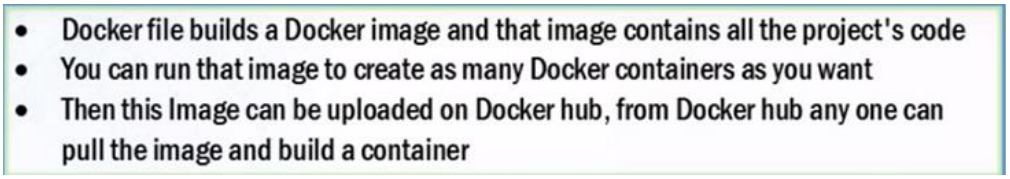
Containers and VMs together provide a tremendous amount of flexibility for IT to optimally deploy and manage apps.

Sharing Layers

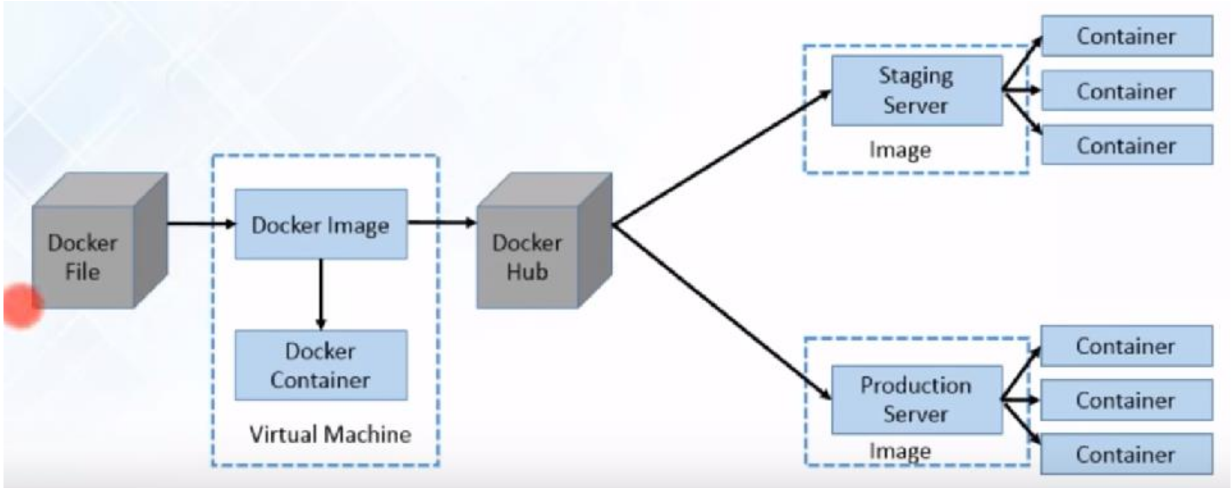
- Images can share layers in order to speed up transfer times and optimize disk and memory usage
- Parent images that already exists on the host do not have to be downloaded

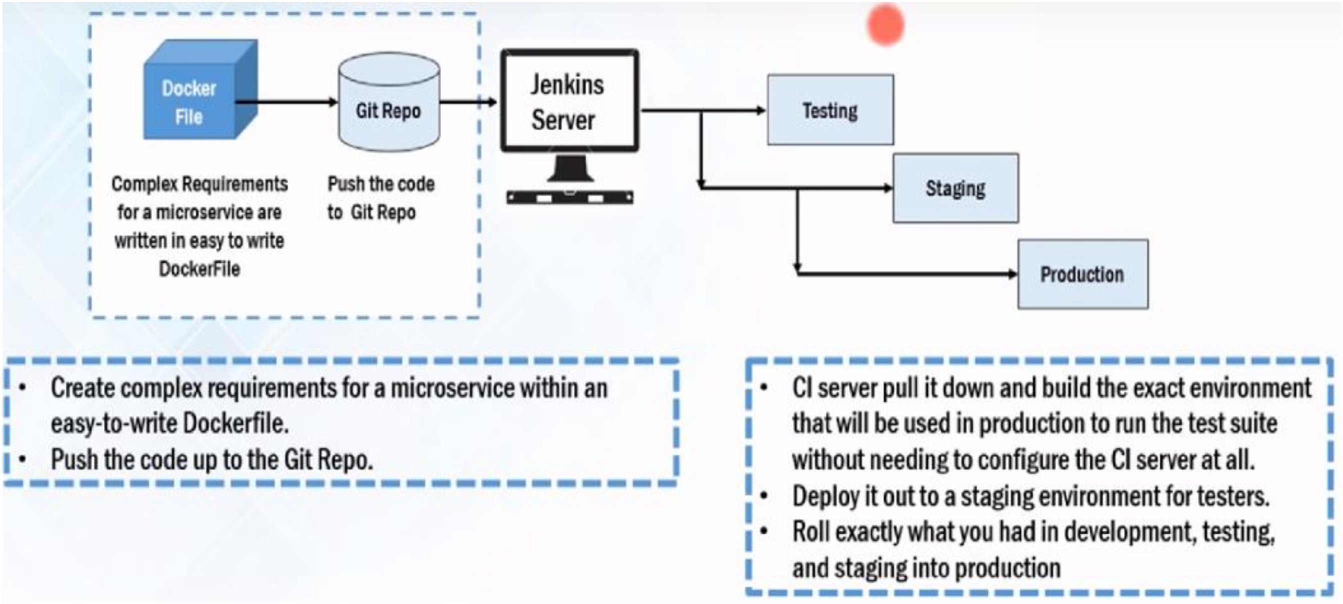






- Docker file builds a Docker image and that image contains all the project's code
- You can run that image to create as many Docker containers as you want
- Then this Image can be uploaded on Docker hub, from Docker hub any one can pull the image and build a container





Docker File

Git Repo

Jenkins Server

Testing

Staging

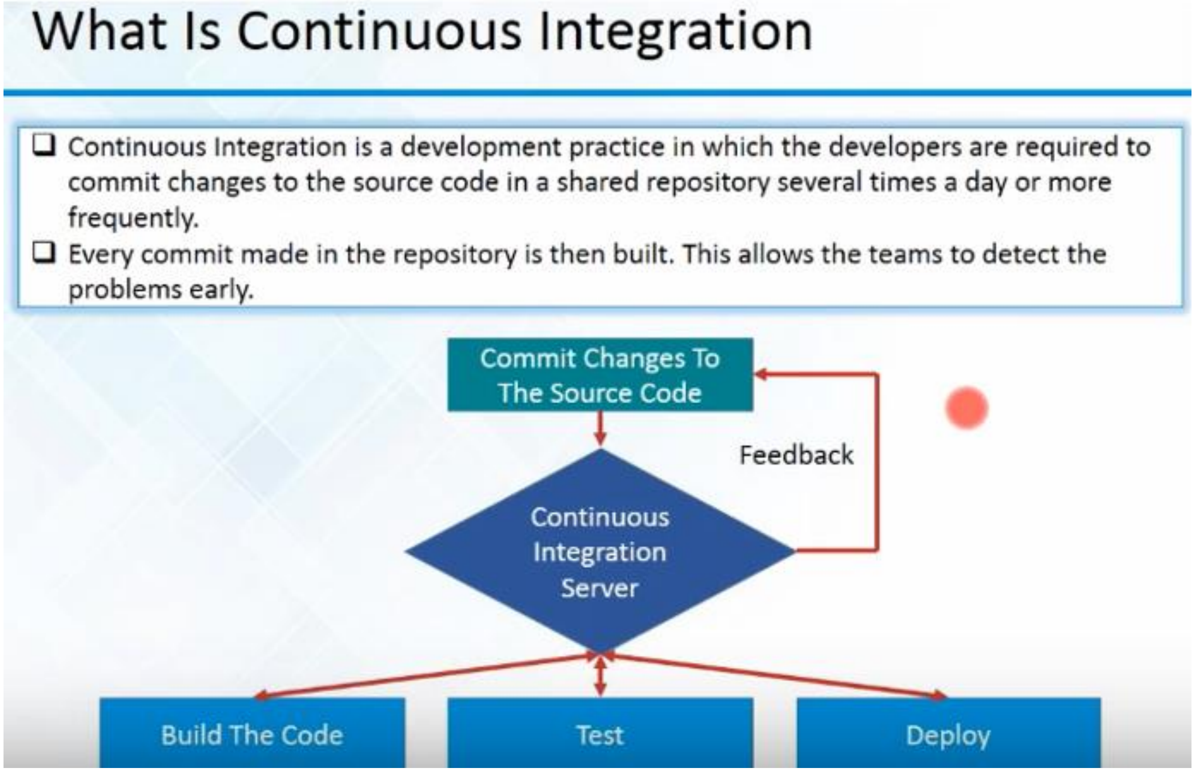
Production

Complex Requirements for a microservice are written in easy to write DockerFile

Push the code to Git Repo

- Create complex requirements for a microservice within an easy-to-write Dockerfile.
- Push the code up to the Git Repo.

- CI server pull it down and build the exact environment that will be used in production to run the test suite without needing to configure the CI server at all.
- Deploy it out to a staging environment for testers.
- Roll exactly what you had in development, testing, and staging into production



What Is Continuous Integration

❑ Continuous Integration is a development practice in which the developers are required to commit changes to the source code in a shared repository several times a day or more frequently.

❑ Every commit made in the repository is then built. This allows the teams to detect the problems early.

Commit Changes To The Source Code

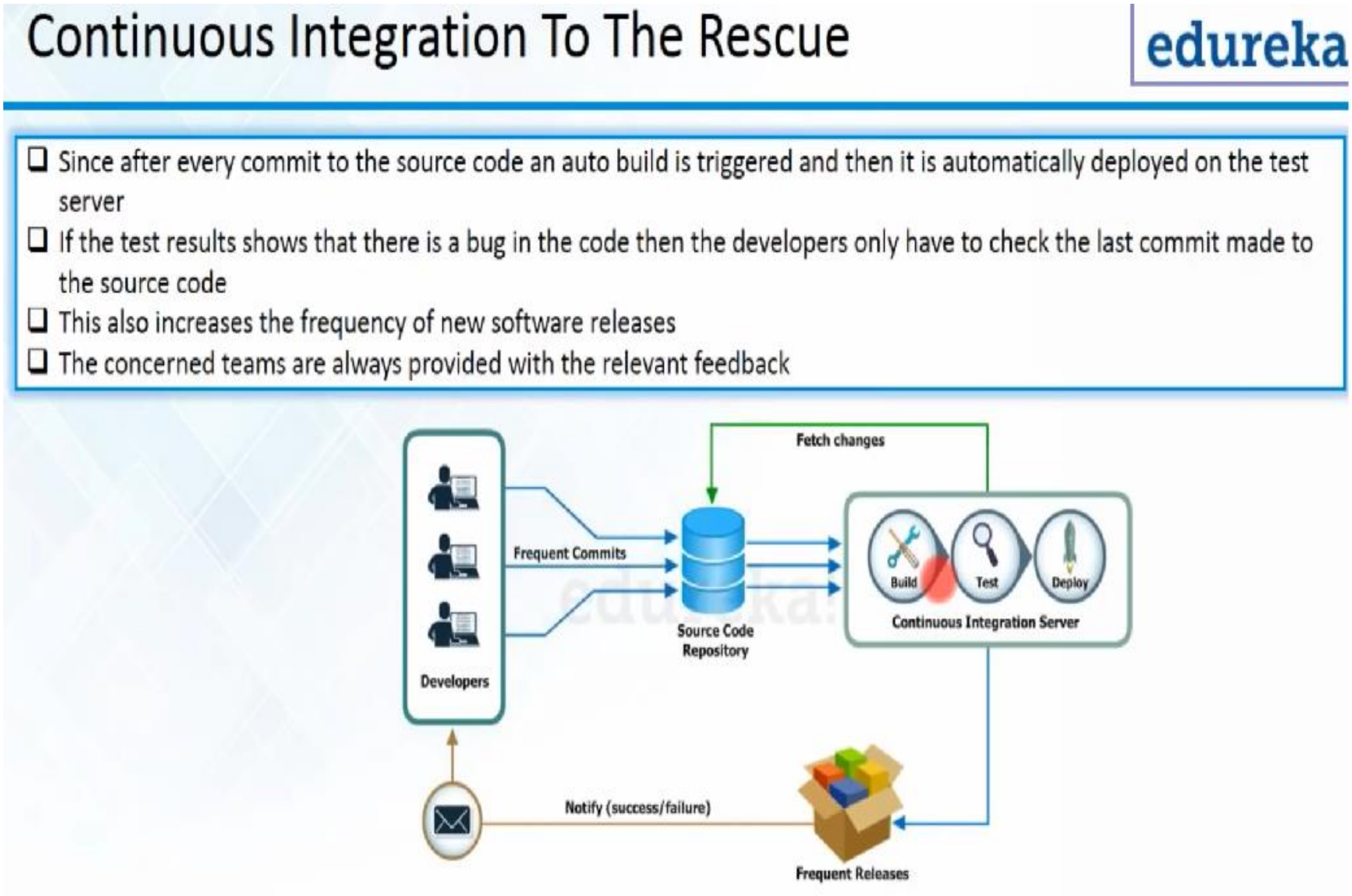
Feedback

Continuous Integration Server

Build The Code

Test

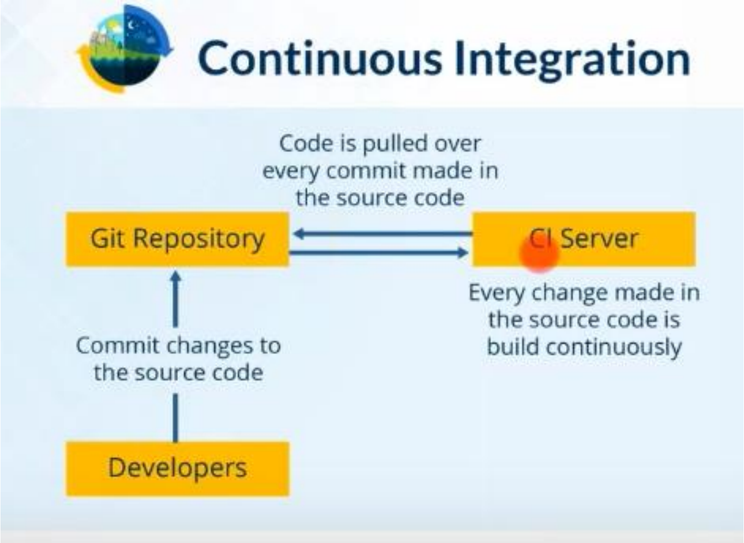
Deploy



Continuous Integration To The Rescue

edureka

- ❑ Since after every commit to the source code an auto build is triggered and then it is automatically deployed on the test server
- ❑ If the test results shows that there is a bug in the code then the developers only have to check the last commit made to the source code
- ❑ This also increases the frequency of new software releases
- ❑ The concerned teams are always provided with the relevant feedback



Continuous Integration

Code is pulled over every commit made in the source code

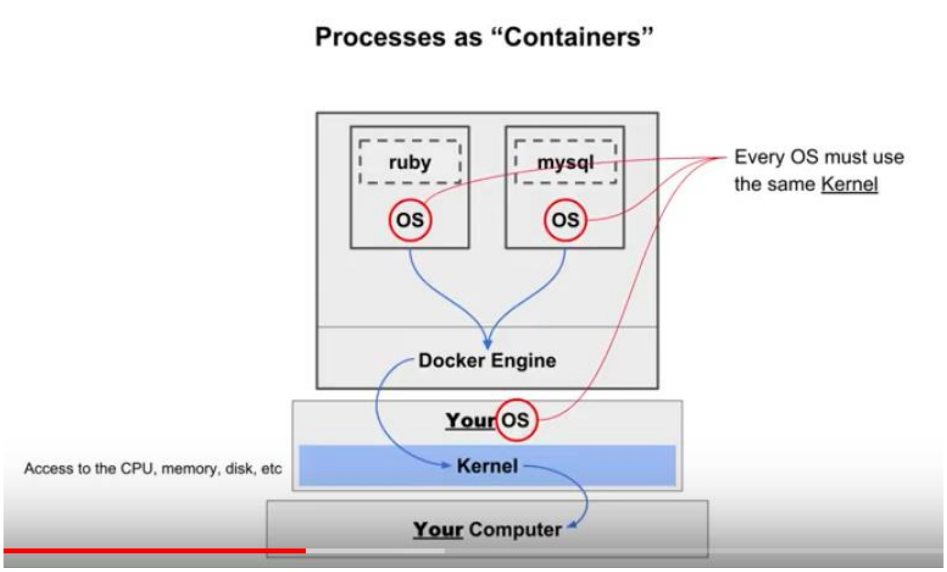
Git Repository

CI Server

Every change made in the source code is build continuously

Commit changes to the source code

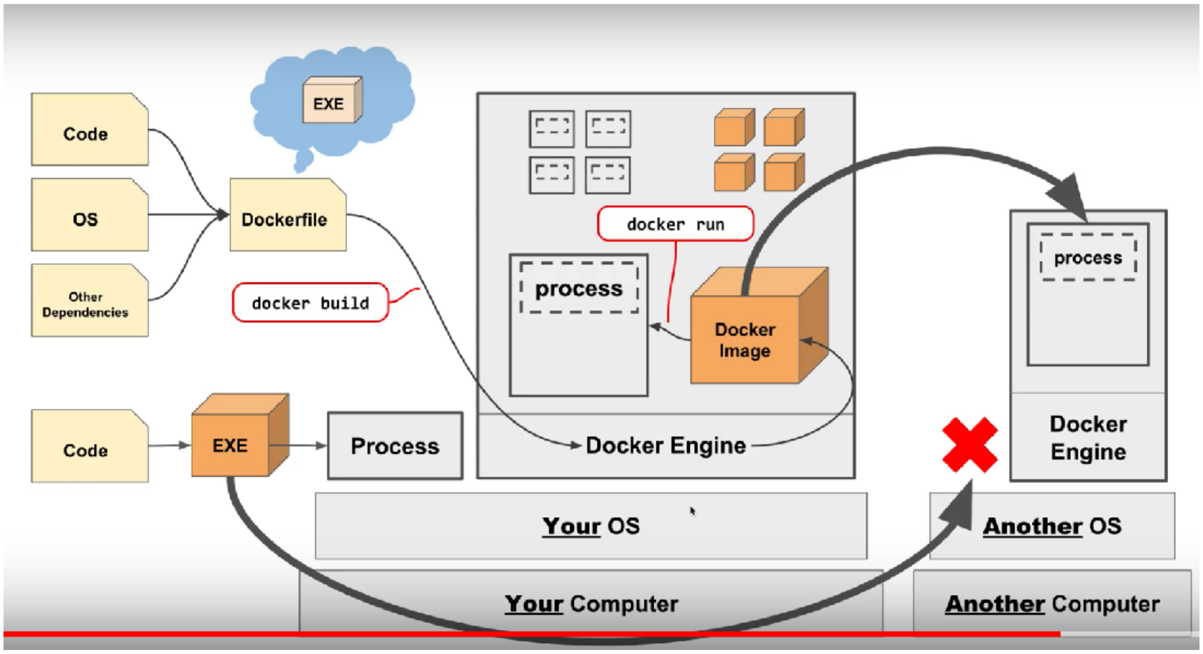
Developers



Processes as “Containers”

Every OS must use the same Kernel

Access to the CPU, memory, disk, etc



EXE
Code
OS
Other Dependencies
Dockerfile
docker build
docker run
process
Docker Image
Docker Engine
Code
EXE
Process
process
Docker Engine
Your OS
Another OS
Your Computer
Another Computer