VERITAS

Introduction to Container Technology

Presented by Aayush Kumaria

How Does a Laptop Work?

A Typical Laptop

What Does a Laptop Contain?

NICs

Disks / SSD

Microprocessor

Memory

VERITAS

How does a Laptop Work?

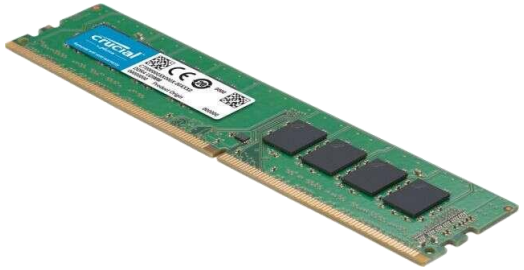
NICs

Operating System

Kernel

Disks / SSD

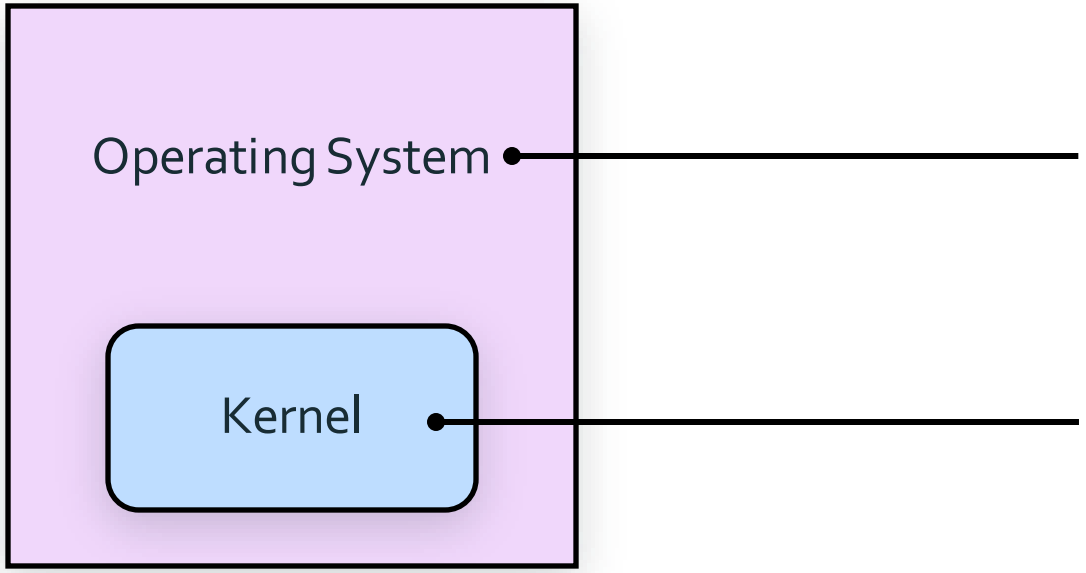
CPU



Memory

VERITAS

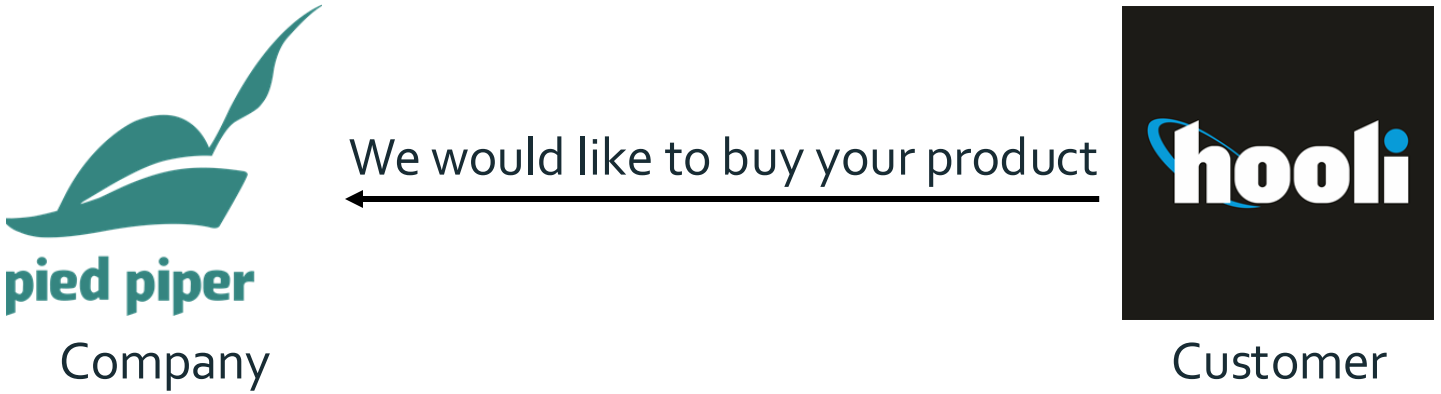
The OS and The Kernel



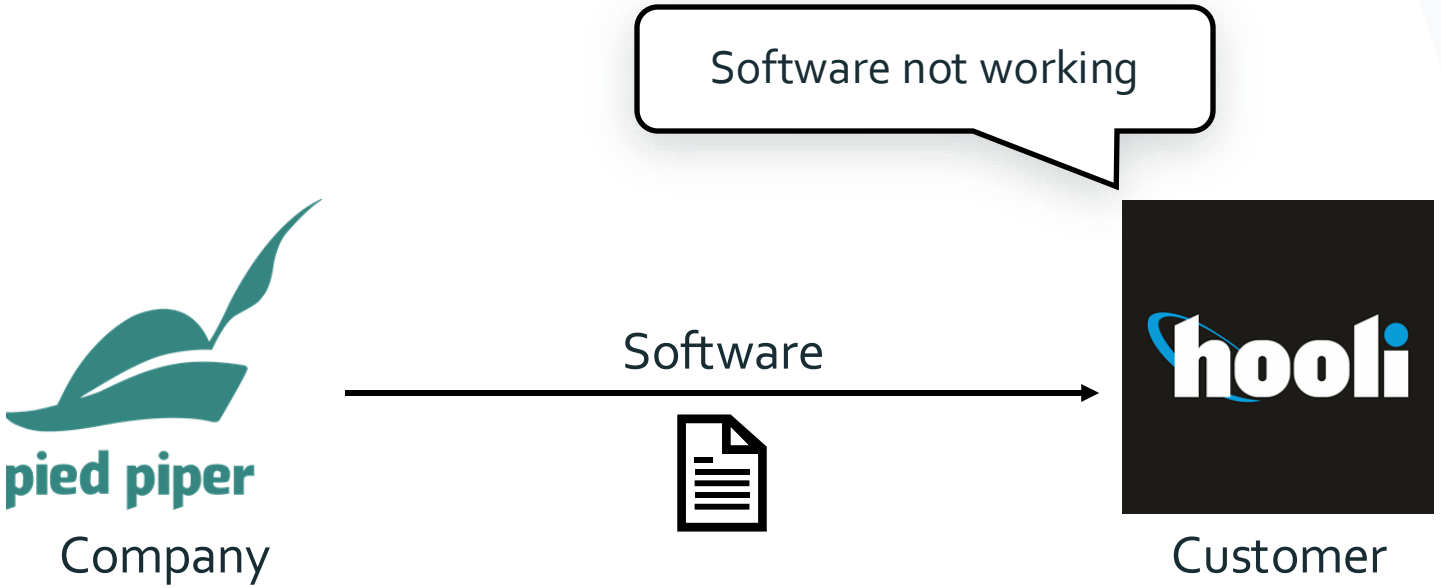
- First Program to Boot Up
- Manage User Space

- Schedule Programs
- Manage Devices
- Read / Write Address on Memory

Corporates Selling A Product



Corporates Selling A Product



What is Virtualization?

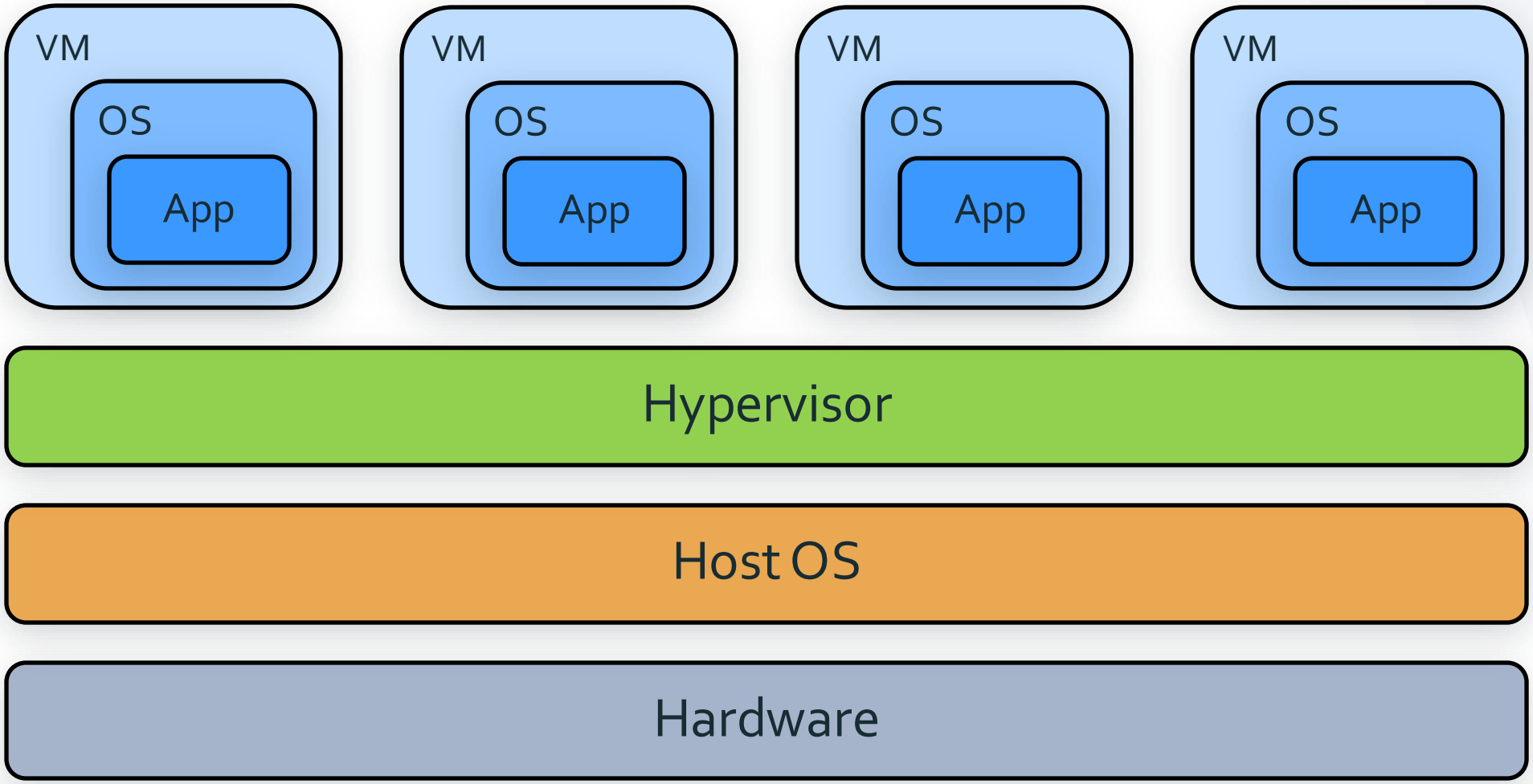
What is Proxy in College?

- Proxy is the process of marking the attendance of a person who is absent in a lecture or class.
- It involves having one physical person who can sign on behalf of multiple people hence showing that a lot of people attended a lecture
- Virtualization is very similar to this concept.

What is Virtualization?

- Virtualization is a technology that you can use to create virtual representations of servers, storage, networks, and other physical machines.
- This allows organizations to operate multiple OS, more than one virtual system and various applications on a single server.
- Virtual Machines are created and managed via a software component called a Hypervisor.

Virtualization Picturized

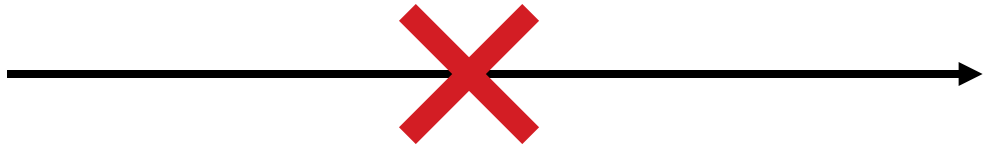


How does a Virtual Machine work?

VERITAS

Where would a VM be required?

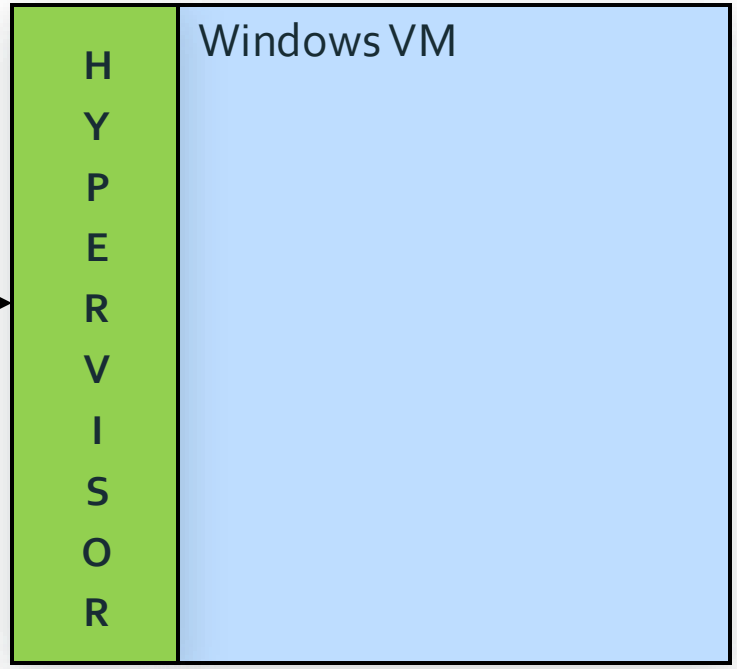
Your MacBook



The game you want to play

Where would a VM be required?

Your MacBook



The game you want to play

VERITAS

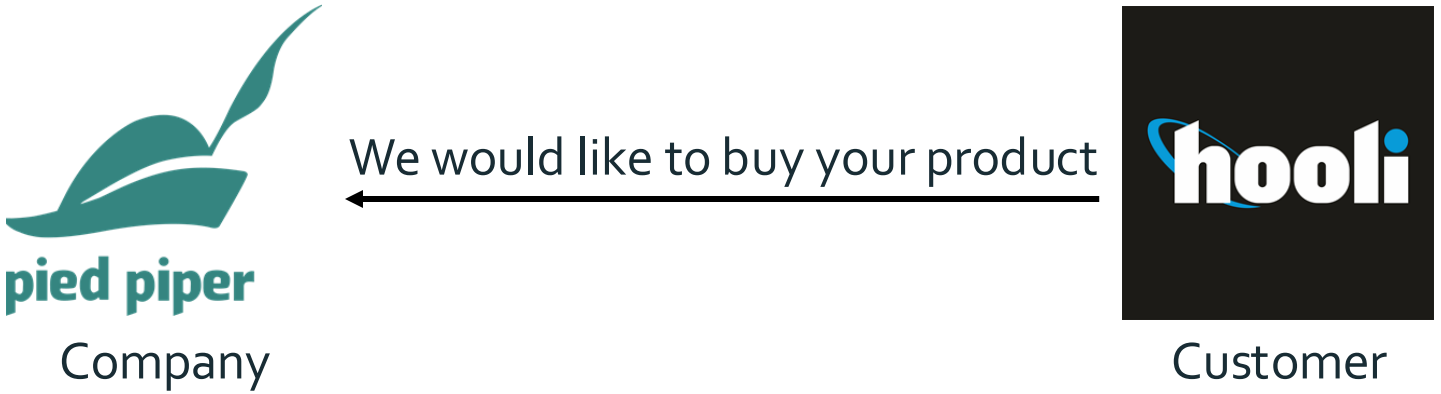
What is a Virtual Machine?

- Virtual Machines are software-defined machines that run inside a Physical Computer with its own Operating System and computing resources.

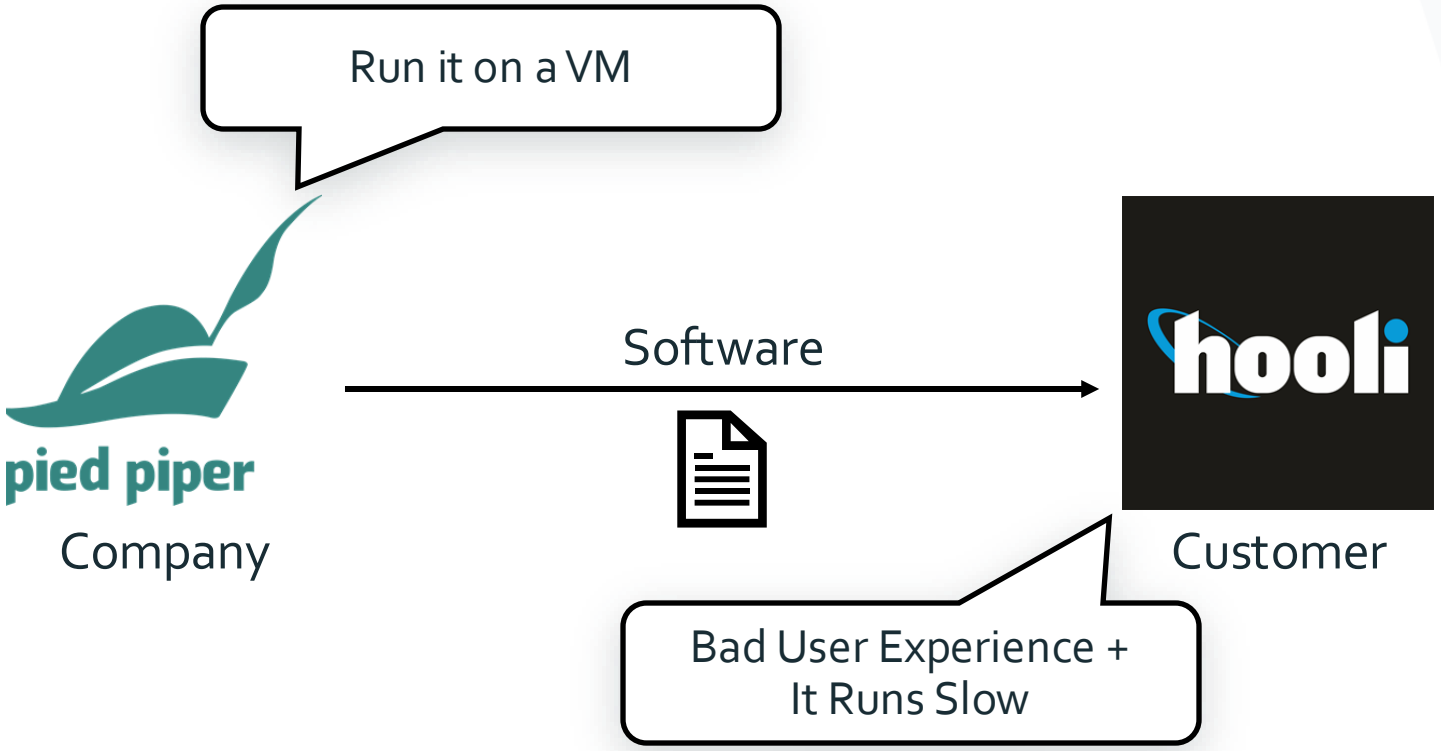
- It allows a business to run an OS that behaves like a separate computer in an app window on a desktop

- Multiple Virtual Machines can be created on a single Host Machine with each machine running a separate Operating System.

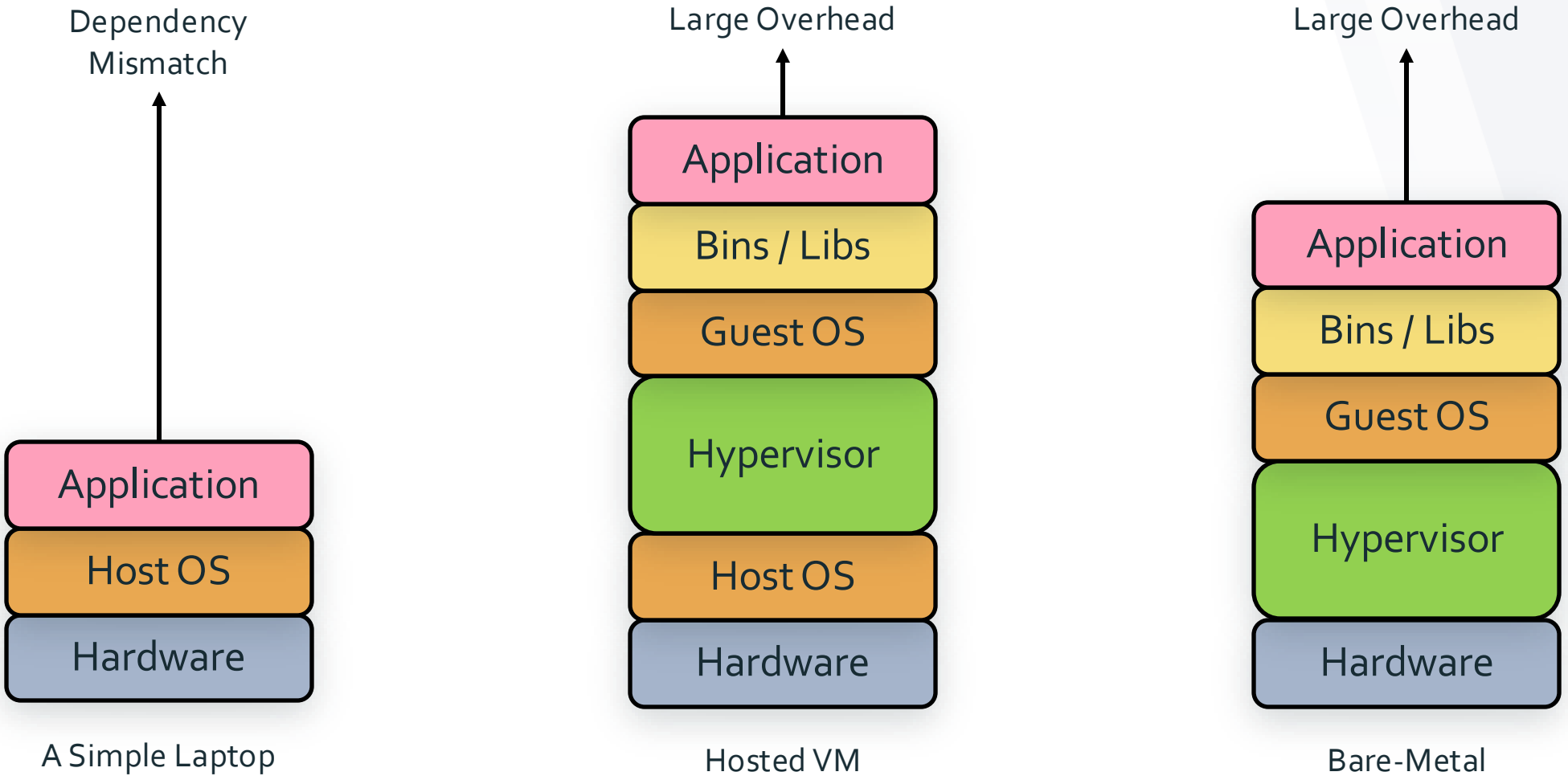
Corporates Selling A Product



Corporates Selling A Product



Why are VMs slow?



What is Containerization?

Picture a Wallet…

What does a Wallet Contain?

Cards

- To purchase stuff on Amazon with discounts
- To pay for food at fancy restaurants
- And much more…

VERITAS

What does a Wallet Contain?

Cash

- To pay Auto Rickshaw Drivers
- To pay for Parking
- And to do whatever Cards cannot do

VERITAS

What does a Wallet Contain?

Identification

- To show Traffic Police
- To let the college watchman let you enter the premise

VERITAS

What is a Wallet?

Cash

Cards

Identification

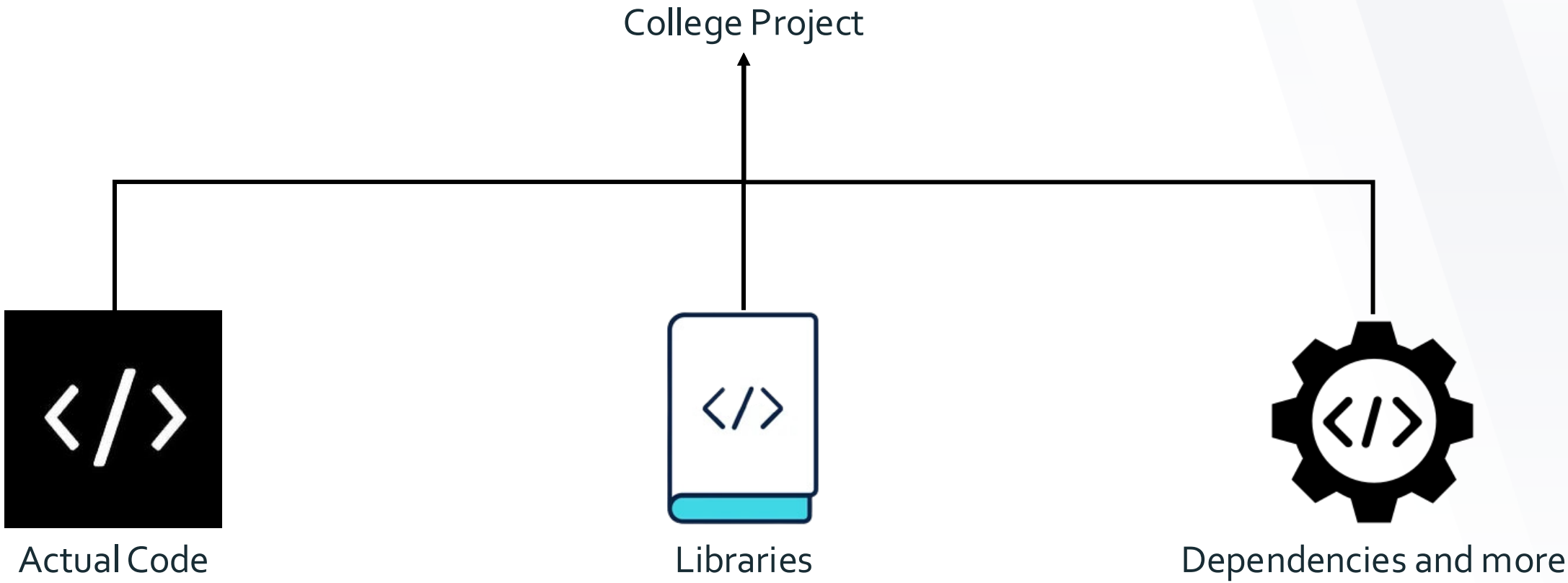
Bundled Up

What Is Containerization?

- It is the Software Deployment Process of packaging an application into a single lightweight executable called a Container.
- This Container includes the Application's Code, the OS Files, Libraries, and other Dependencies, bundled together.
- This ensures that the software can run on any infrastructure.

- Just like how a wallet is bundled with Cards, Cash, and Identification and can be used in various scenarios.

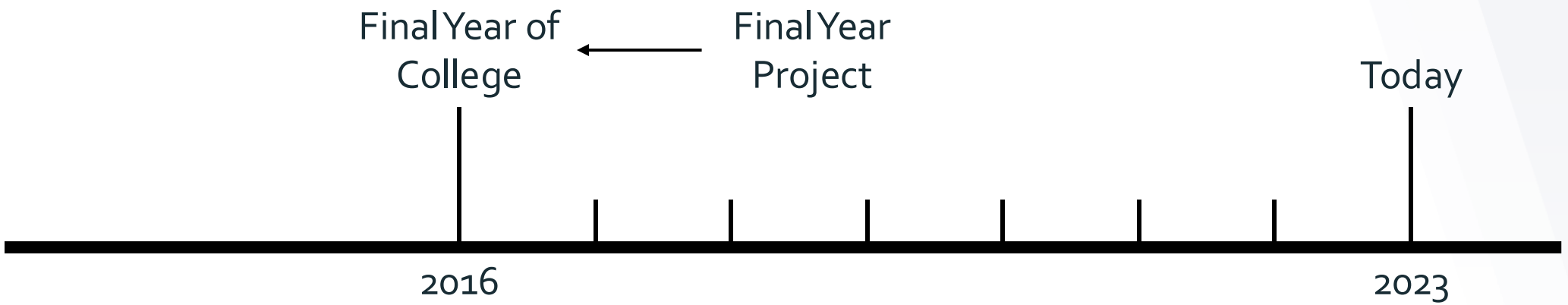
Containerization of Software



Why is Containerization Required?

VERITAS

Project Fiasco…



Project Fiasco…

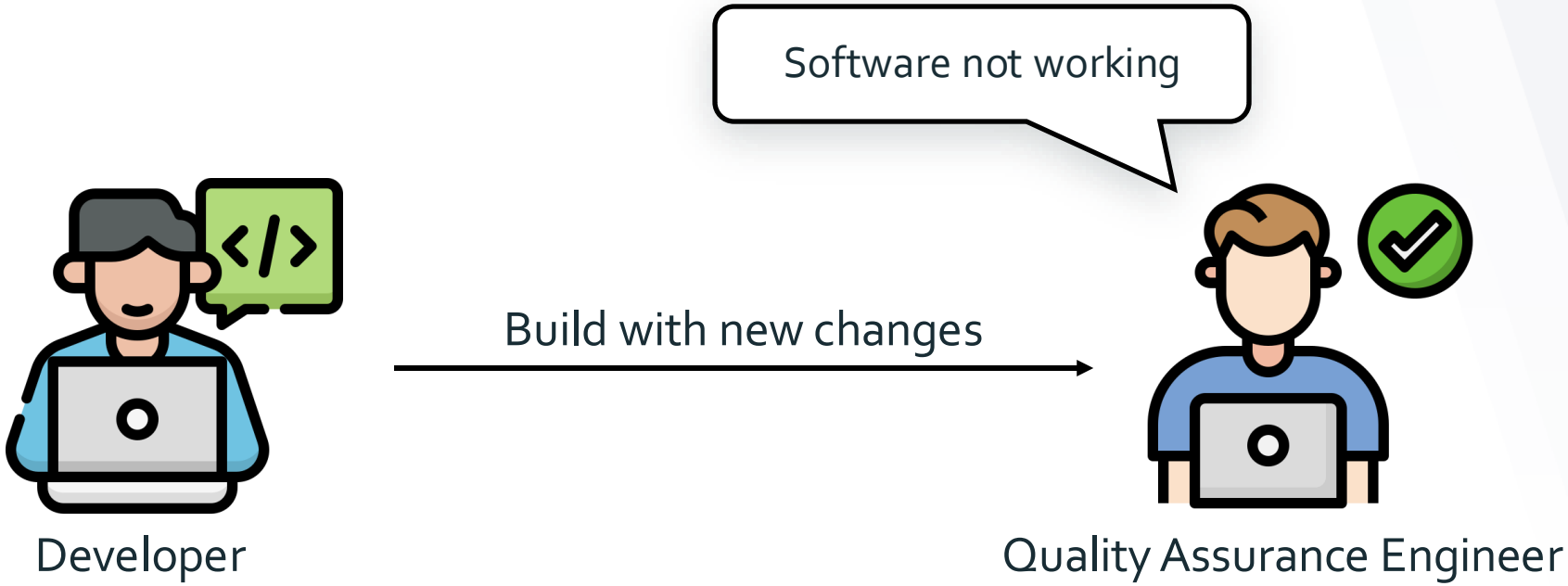
Project Fiasco…

Laptop
Destroyed

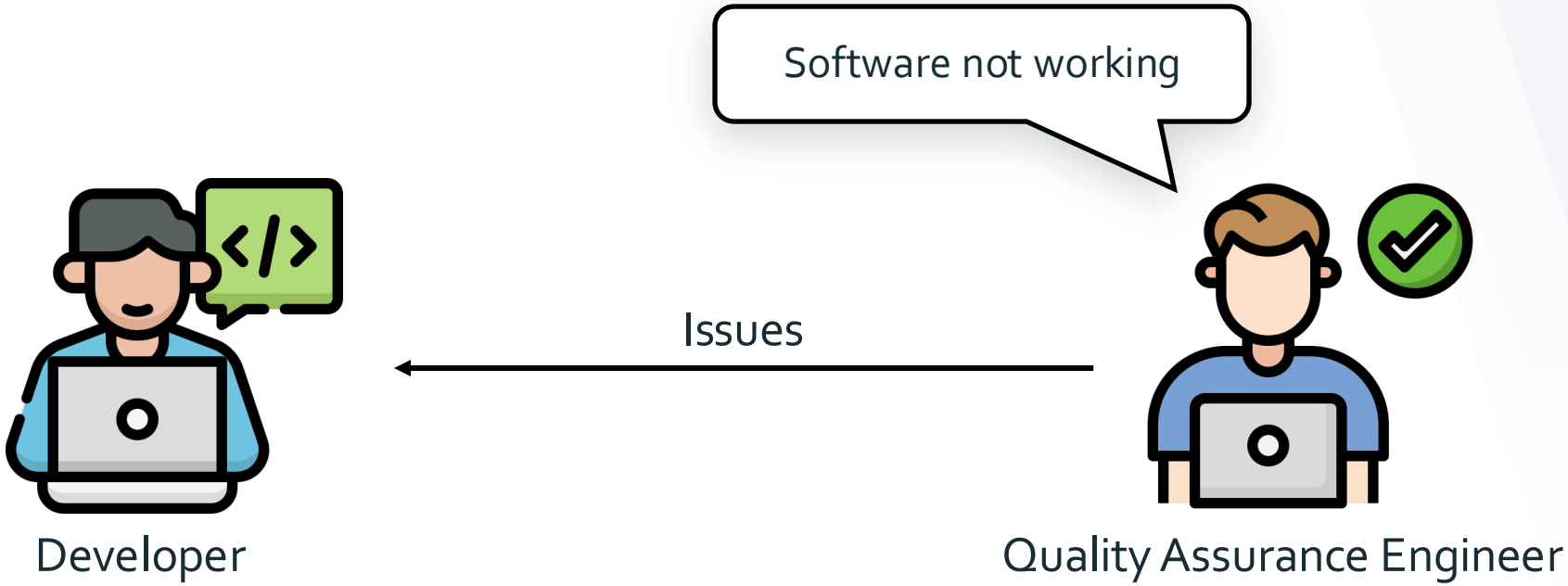
Dependency
Mismatch

OS Mismatch

QA – Dev Scenario



QA – Dev Scenario



Advantages of Containerization - 1

- Portability
  - Creating an executable software package abstracted away from the host OS, hence, removing dependency on the OS.
- Speed
  - Containers share the host machine's OS kernel and aren't subject to extra overhead hence allowing fast boot-up.
- Scalability
  - An application deployed on a container can handle increasing workloads by reconfiguring the existing architecture.

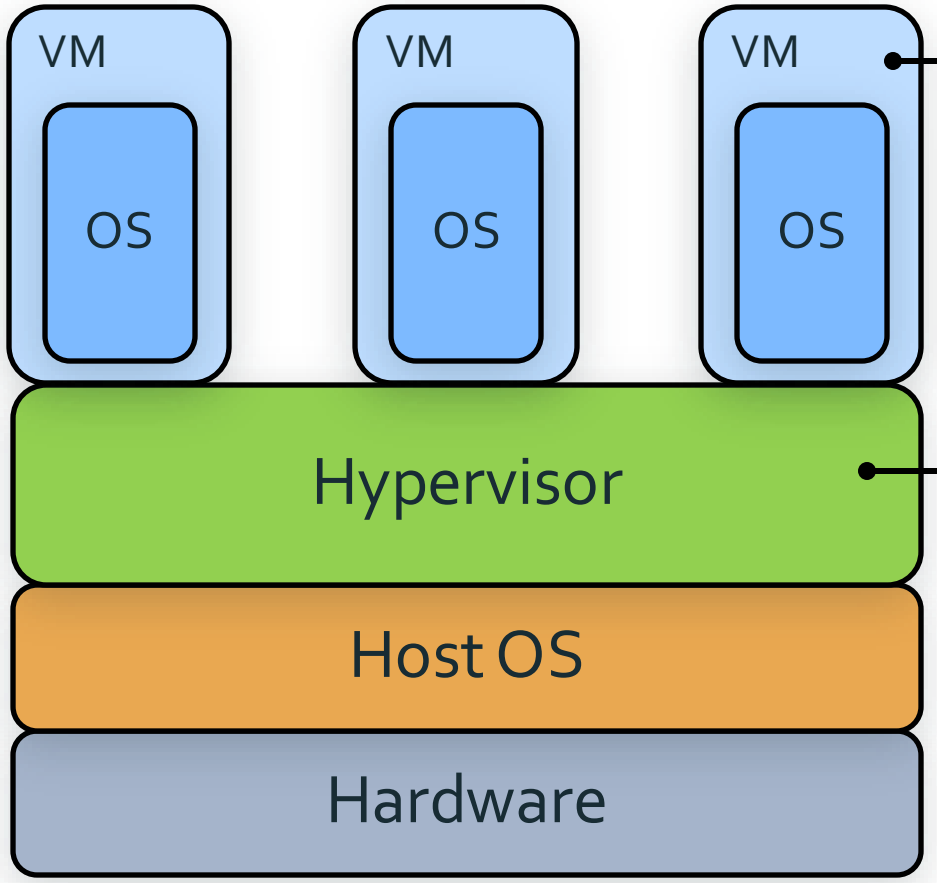
Advantages of Containerization - 2

- Fault Isolation
  - Each Containerized Application works independent of the others and hence a fault in one would not affect the other
- Ease Of Management
  - Using certain platforms you can automate the installation, management, and scaling of containerized applications.
- Developer Friendly
  - One Common Container for Development and Production

And Much More…..

What is a Container?

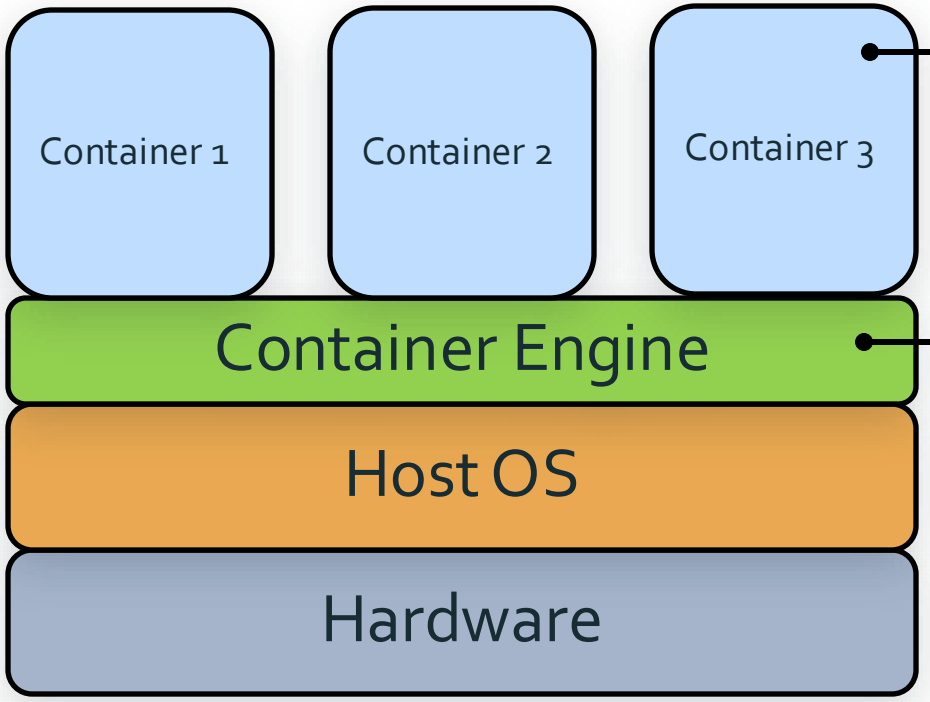
The Difference between a VM and a Container



Hosted VM

- Each VM has its own OS
- The Hypervisor (which is heavy) Virtualizes the Hardware

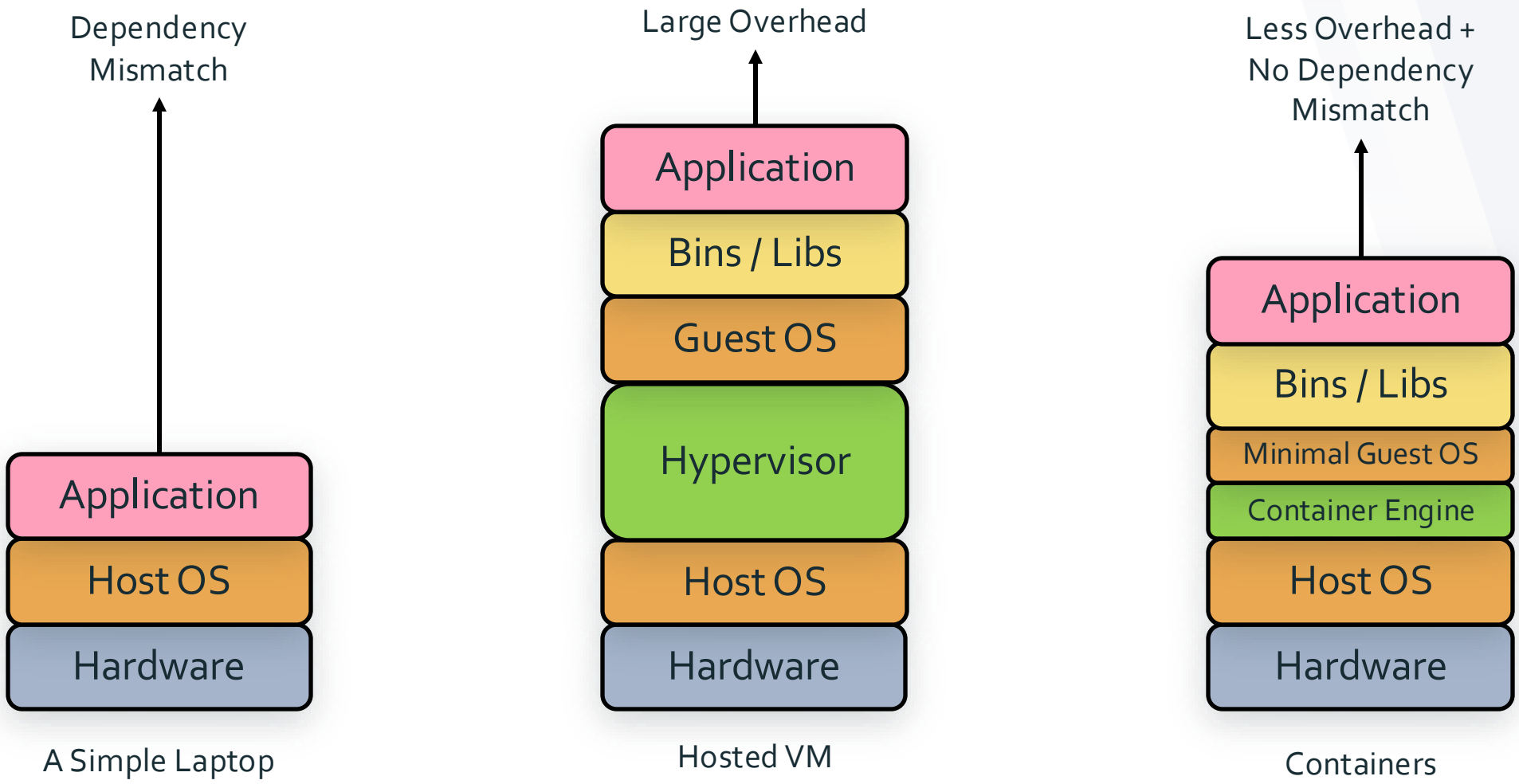
The Difference between a VM and a Container



Containers

- Has Minimal Guest OS files
- The Container Engine (which is light) creates isolated environment by virtualizing the OS
- The Kernel is shared

The Difference between a VM and a Container



VERITAS

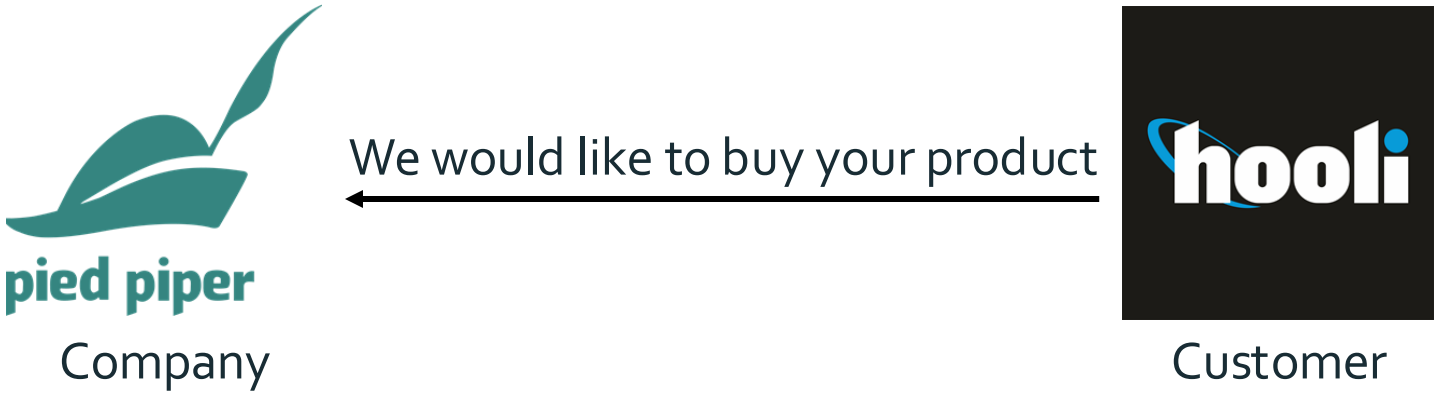
What is a Container?

- A Container is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings

- It contains less overhead compared to a VM and removes the dependency mismatch issue

- Containers work just like a VM but contain bare minimum OS Files unlike a Virtual Machine, which houses the entire OS.

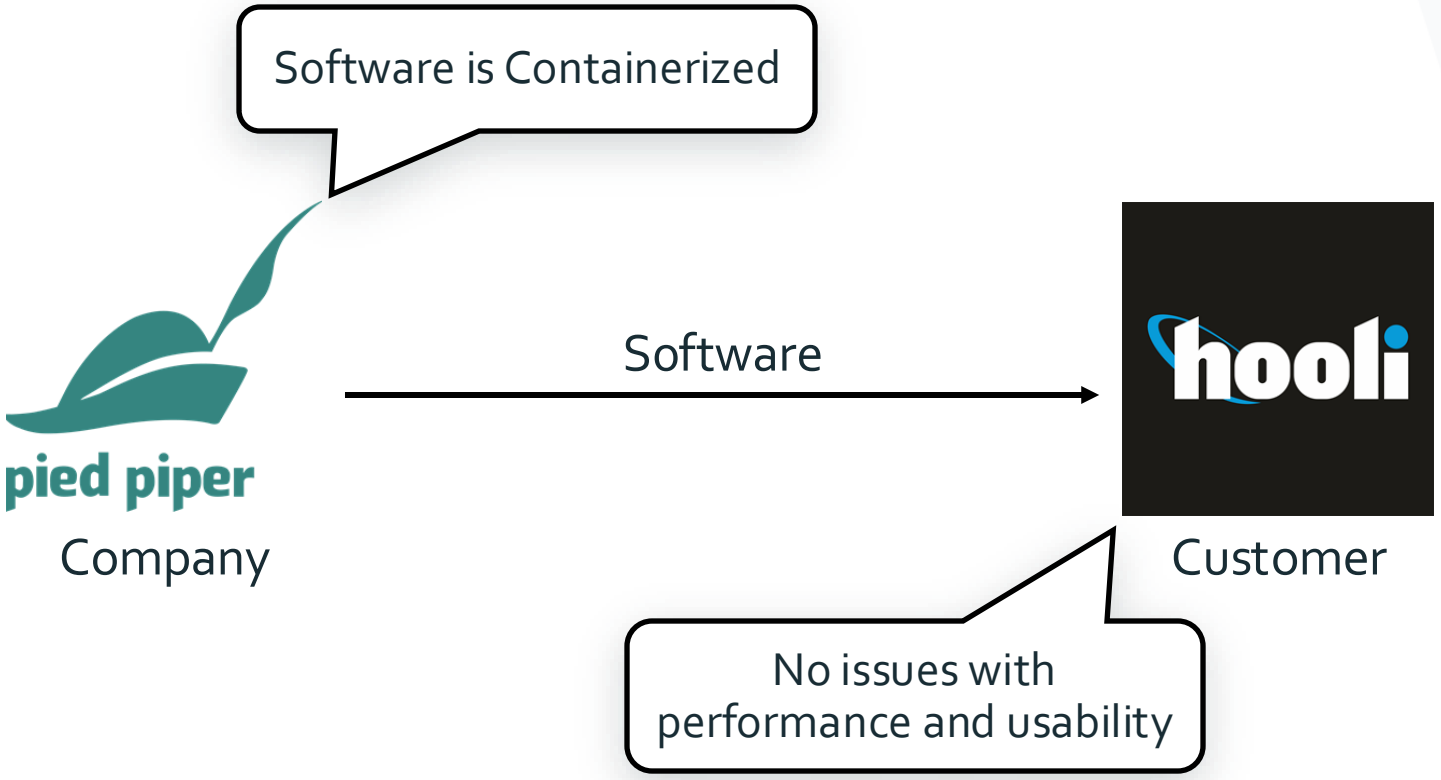
Again, Back to the Use-Case



Company

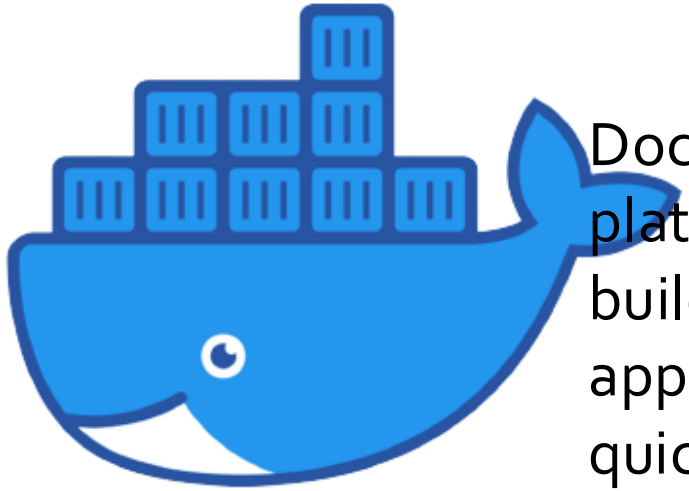
Customer

Back to the Use-Case



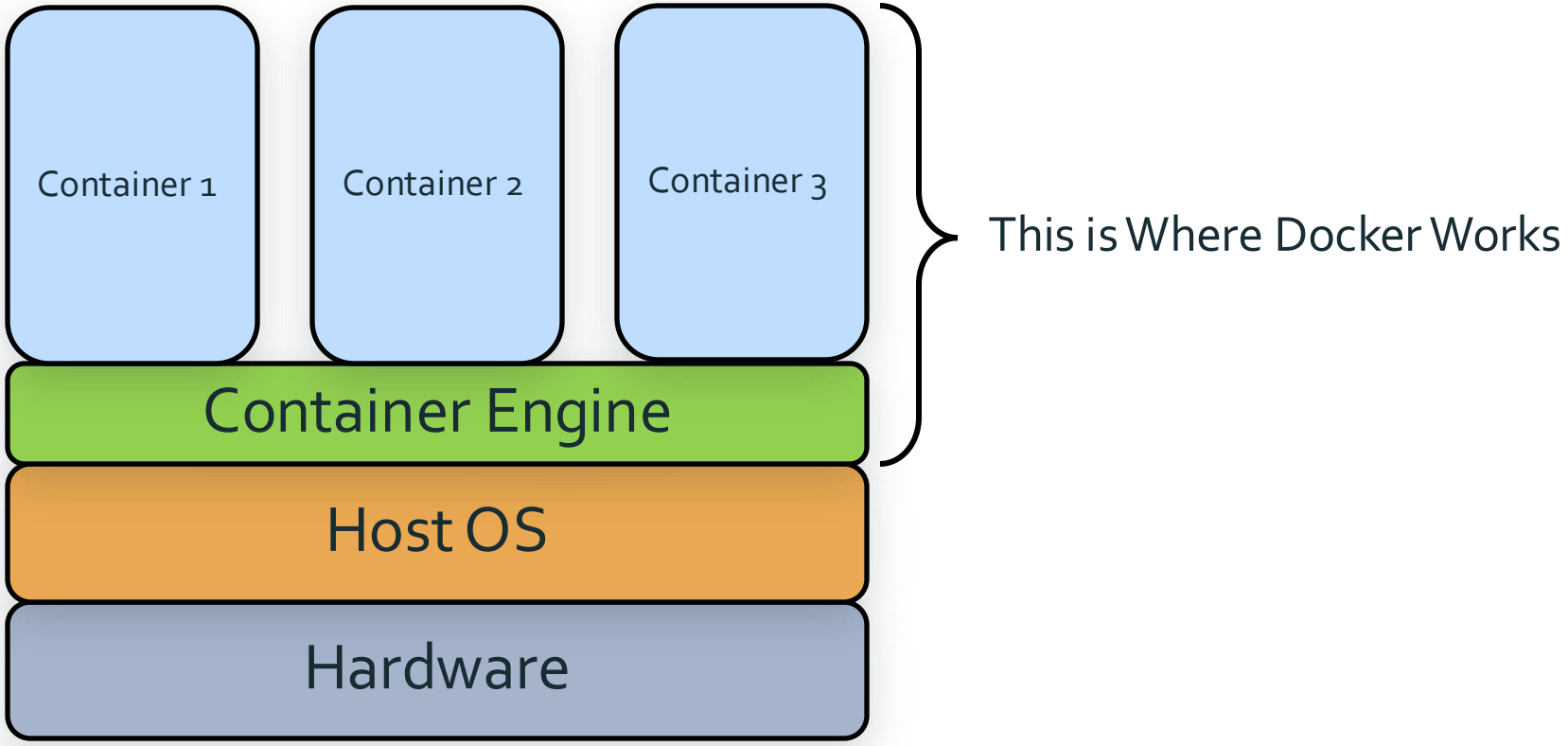
What is Docker?

This Whale represents Docker

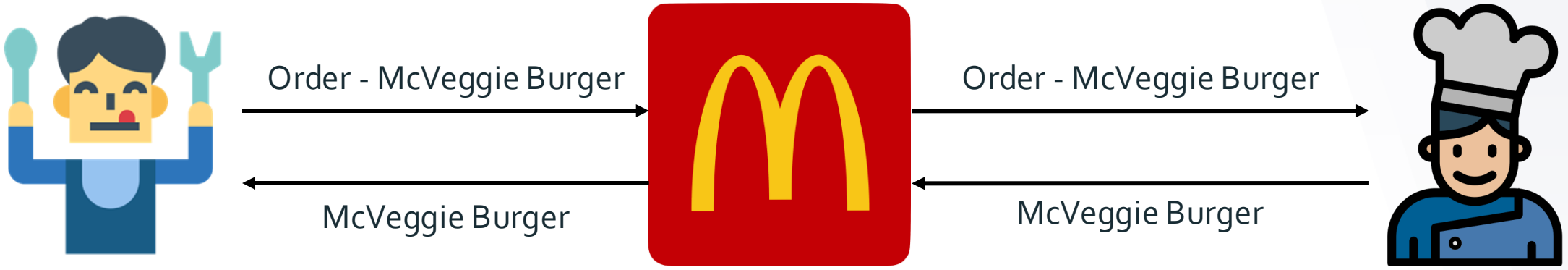


Docker is a software platform that allows you to build, test, and deploy applications on containers quickly

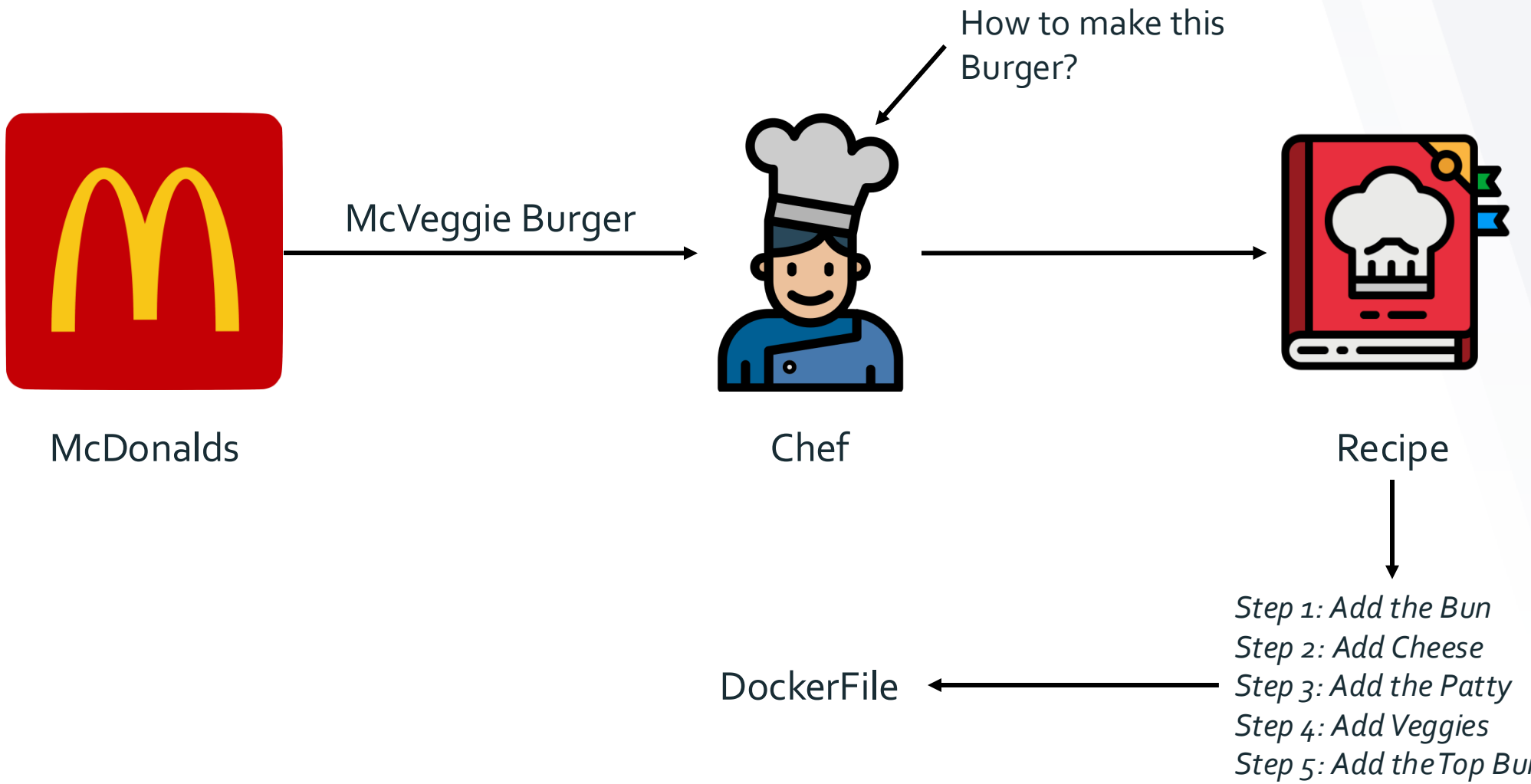
Where Is The Ocean of this Whale?



The McDonald's Burger Use-Case



The McDonald's Burger Use-Case



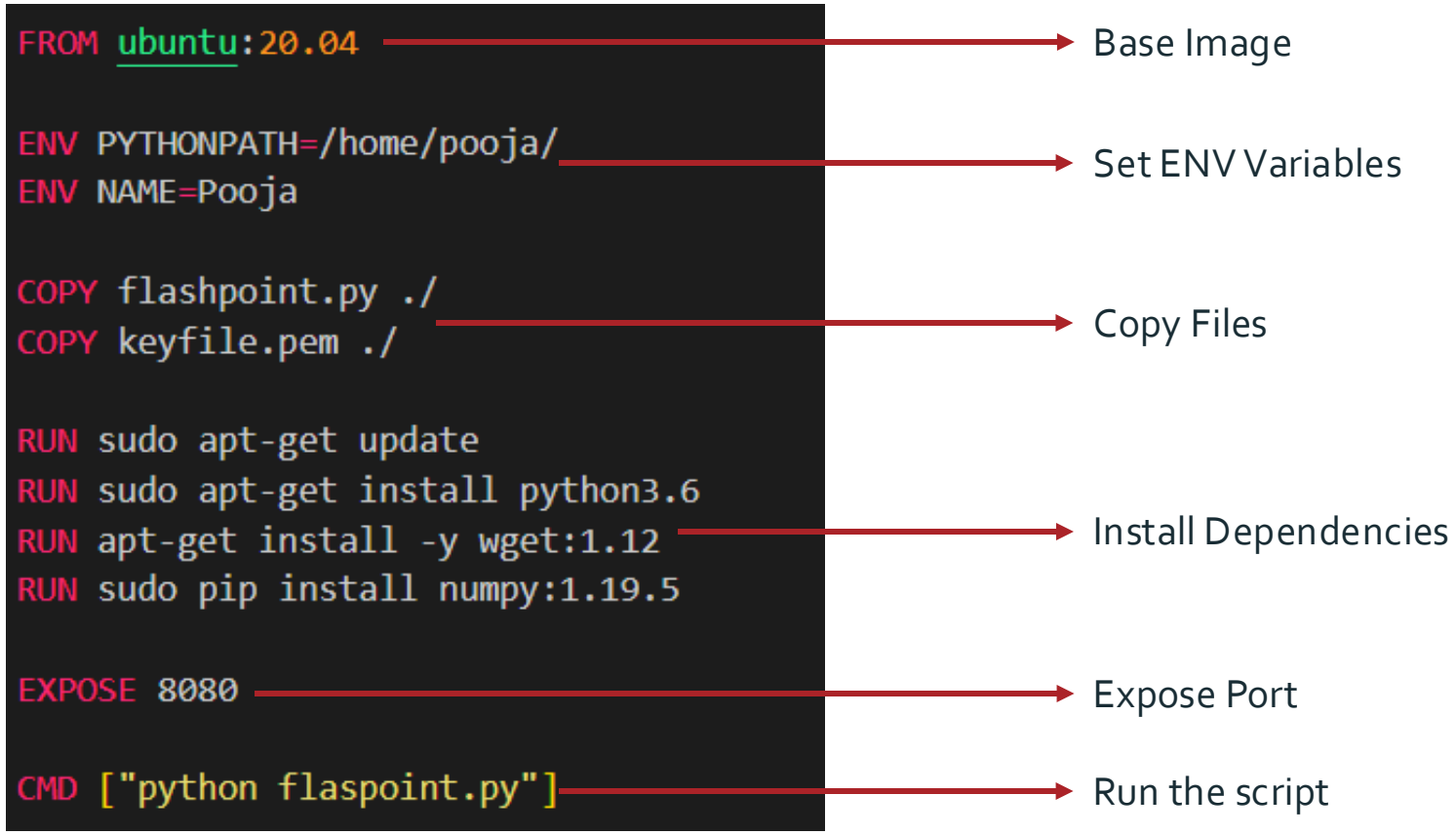
What is a Dockerfile

- A Dockerfile is basically a text document that contains all the commands a user would perform, from start to finish, to create the environment.

- Dockerfile is essentially the build instructions to build the Docker Image (In this case the McVeggie Burger)

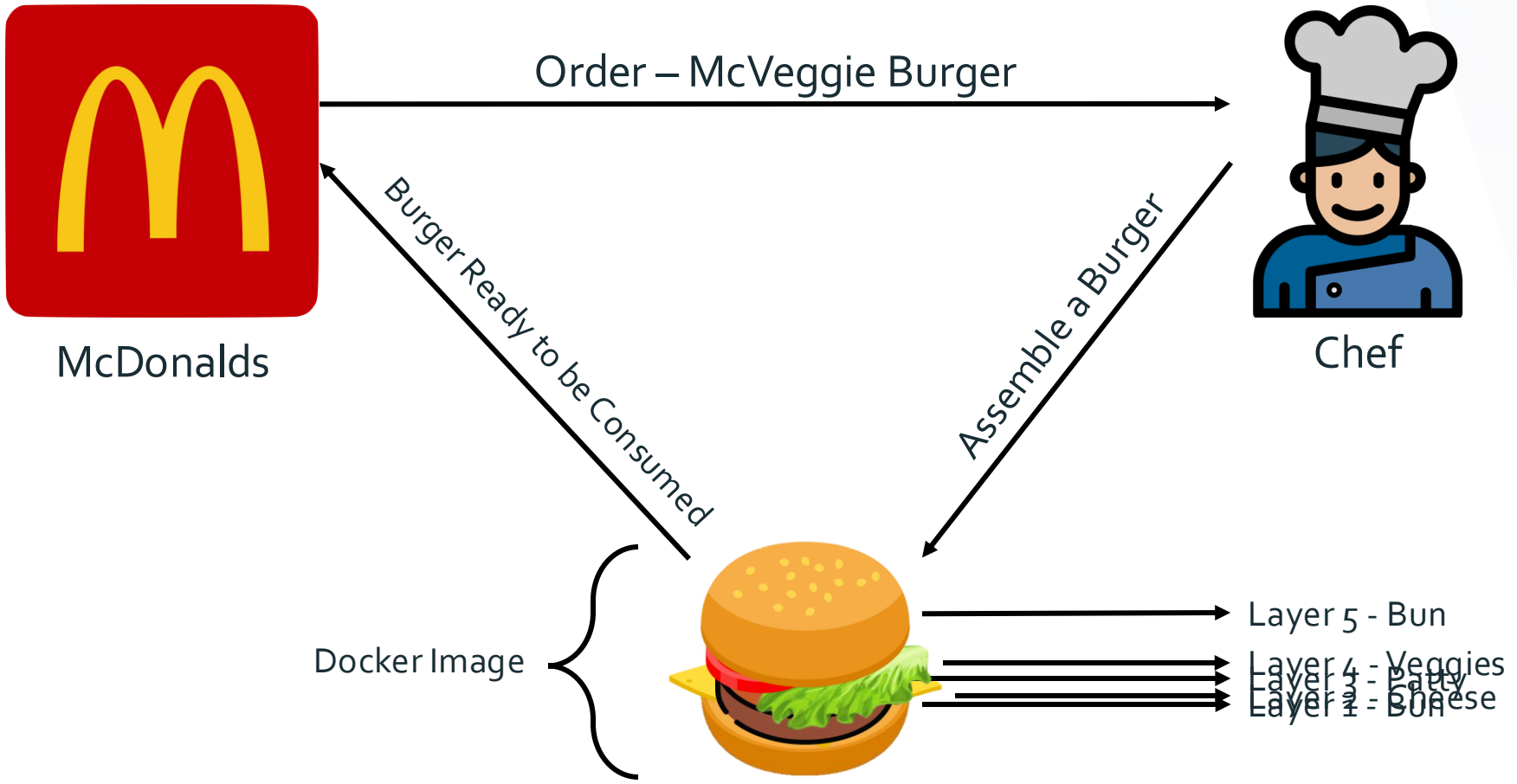
- The commands or instructions in the Dockerfile are executed successively to perform actions on the base image

Example of a Dockerfile



Dockerfile

The McDonald's Burger Use-Case



What is a Docker Image

- A Docker image is a read-only template containing instructions for creating a container.

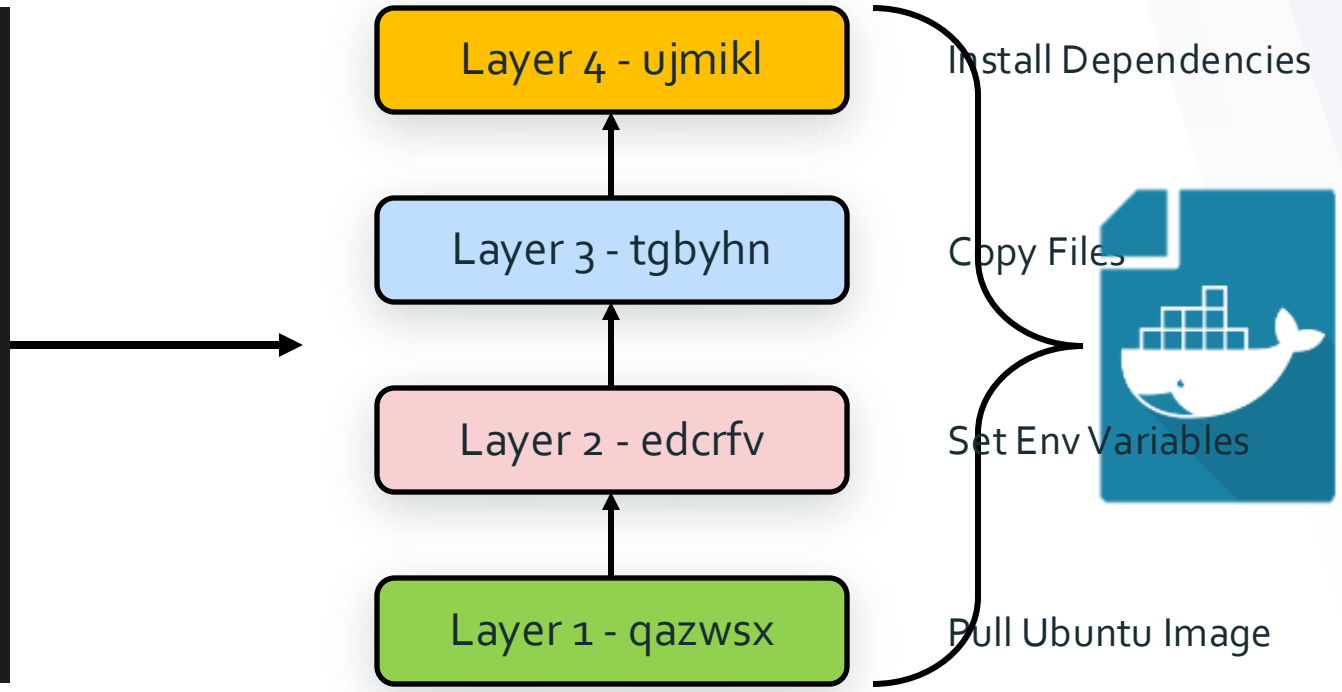
- An Image is like a Blueprint of the Container when it runs. These Images contain the code or binary, runtimes, dependencies, and other filesystem objects to run an application.

- An Image is composed of multiple stacked layers, with each layer changing something in the environment.

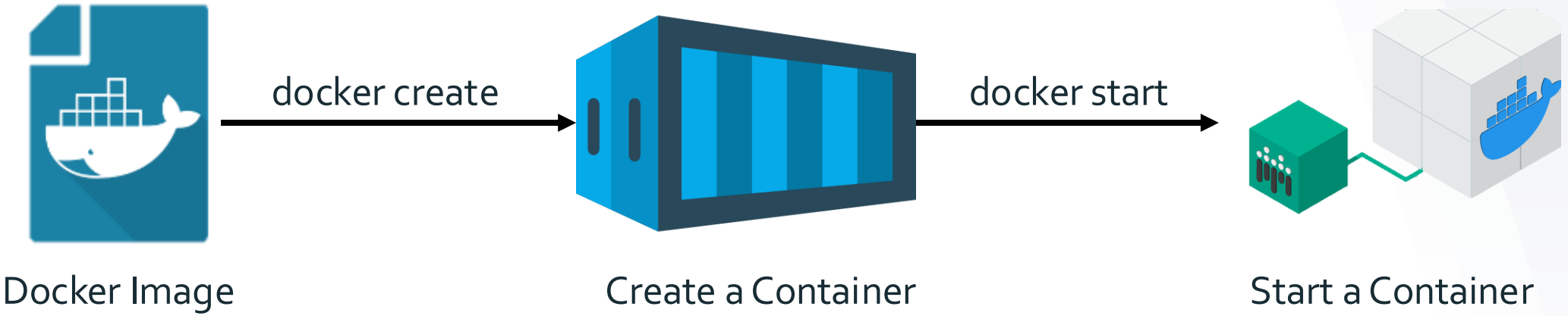
Example of a Docker Image

```
FROM ubuntu:20.04

ENV PYTHONPATH=/home/pooja/
ENV NAME=Pooja

COPY flashpoint.py ./
COPY keyfile.pem ./

RUN sudo apt-get update
RUN sudo apt-get install python3.6
RUN apt-get install -y wget:1.12
RUN sudo pip install numpy:1.19.5

EXPOSE 8080

CMD ["python flaspoint.py"]
```

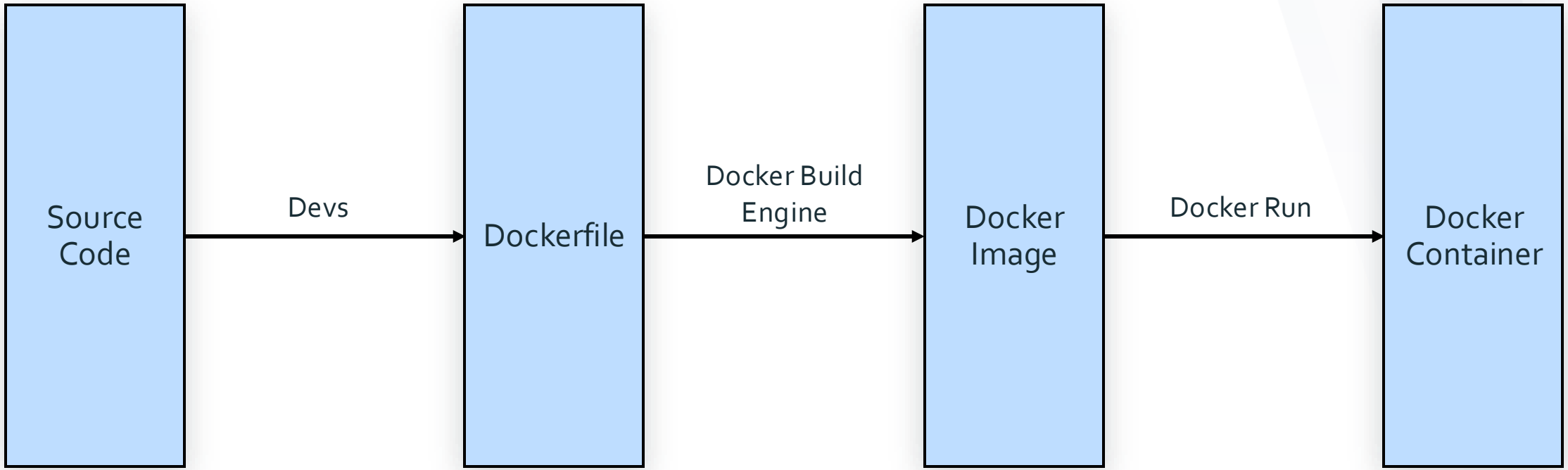
Dockerfile



Docker Image to Container



Start to Finish with Docker

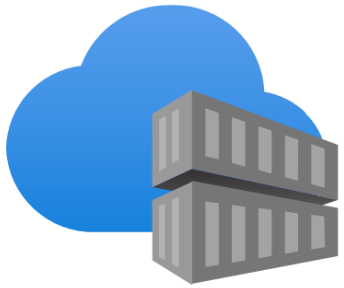


What are Container Registries?

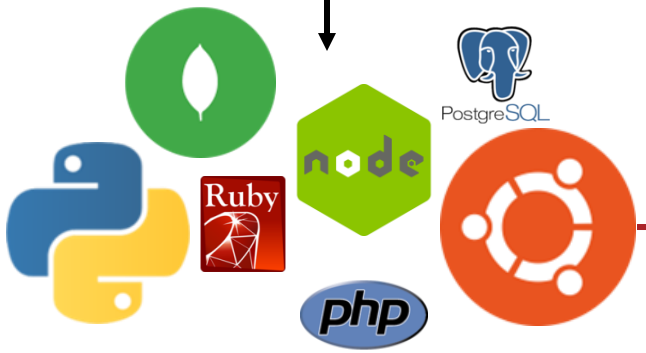
VERITAS

Container Registry

Where do we get this from?



Container Registry




```
FROM ubuntu:20.04

ENV PYTHONPATH=/home/pooja/
ENV NAME=Pooja

COPY flashpoint.py ./
COPY keyfile.pem ./

RUN sudo apt-get update
RUN sudo apt-get install python3.6
RUN apt-get install -y wget:1.12
RUN sudo pip install numpy:1.19.5

EXPOSE 8080

CMD ["python flaspoint.py"]
```

Types of Container Registries

Docker Hub

Third Party Registries Services

docker login

What is a Container Registry?

- A container registry is essentially acts as a place for developers to store container images and share them out via a process of uploading (pushing) to the registry and downloading (pulling) into another system.

- A Container Registry can have multiple images. Each image can have multiple versions. Each image is identified by the tag (version) or its own unique hash.

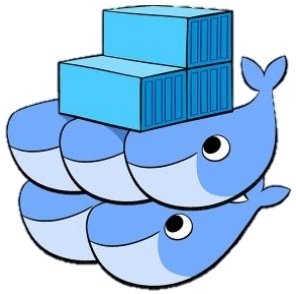
- Container registries make efficient use of storage space by sharing any layers that are common to more than one image.

Trends with Containers

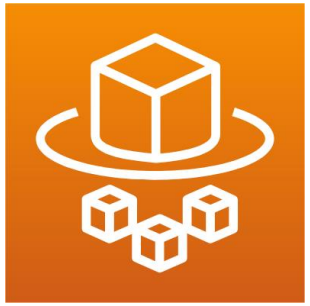
VERITAS

Current Trends Of Container Technology

Kubernetes



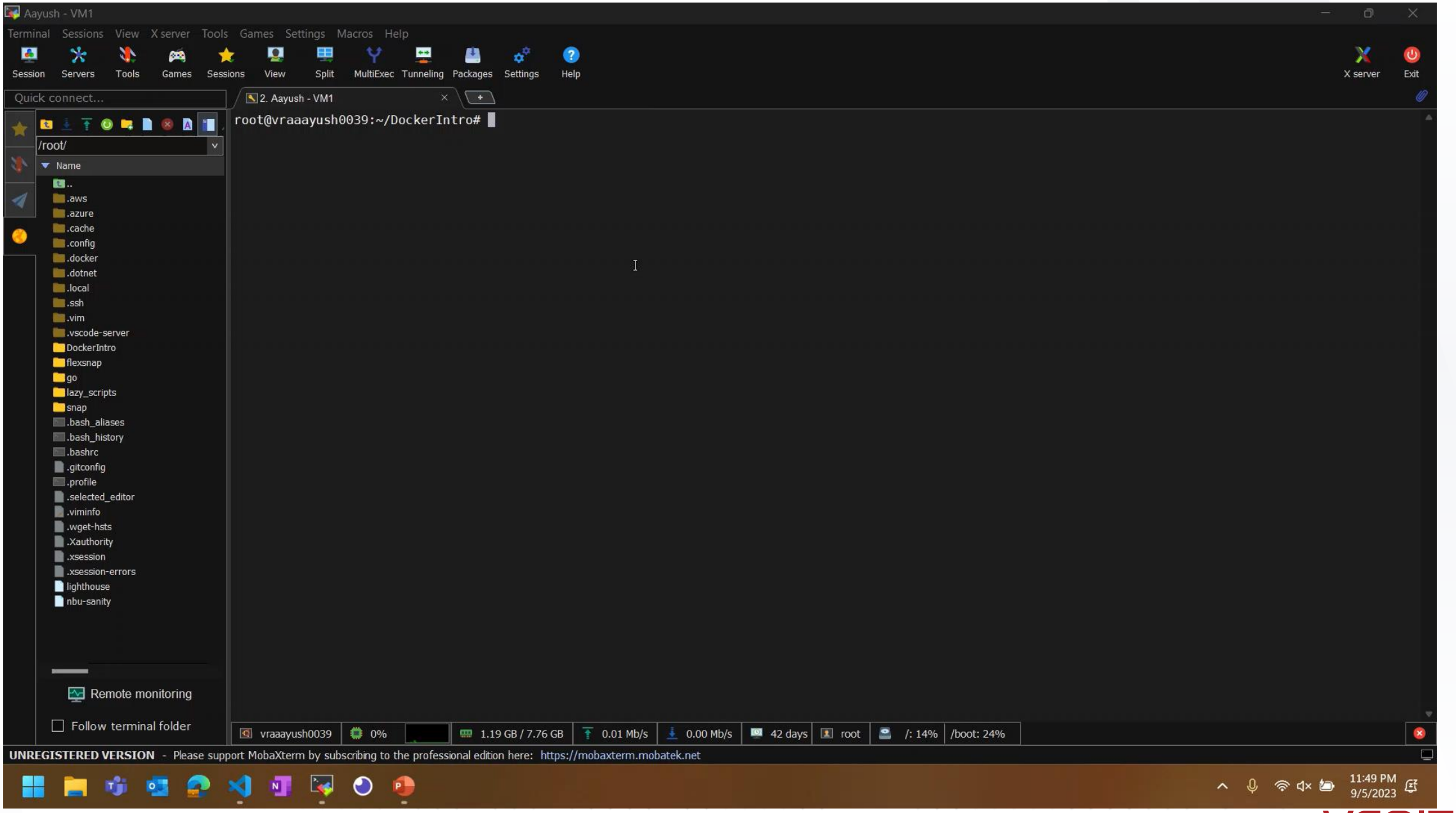
Docker Swarm



Amazon Fargate

OpenShift

Demo


Aayush - VM1
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. Aayush - VM1
root@vraaayush0039:~/DockerIntro#
/root/
Name
..
.aws
.azure
.cache
.config
.docker
.dotnet
.local
.ssh
.vim
.vscode-server
DockerIntro
flexsnap
go
lazy_scripts
snap
.bash_aliases
.bash_history
.bashrc
.gitconfig
.profile
.selected_editor
.viminfo
.wget-hsts
.Xauthority
.xsession
.xsession-errors
lighthouse
nbu-sanity
Remote monitoring
Follow terminal folder
vraaayush0039 0% 1.19 GB / 7.76 GB 0.01 Mb/s 0.00 Mb/s 42 days root /: 14% /boot: 24%
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
11:49 PM
9/5/2023


VERITAS
Thank You