DOCKER + SPARK

A glance at the data world, throughout a container

softserve

Daniel Restrepo Hincapié

Big Data Engineer Senior @ SoftServe.

softserve

Luis Fernando Vásquez

Data Architect Senior - Software Designer

softserve

Workshop Contents

01. Introduction

- About us
- Workshop abouts
- What are we going to see here

02. Source & Architecture

- Data source
- Flow & components

03. Core Concepts

- Understanding the traditional IT infrastructure
- Understanding the Hypervisor
- Understanding the container

04. What is Docker?

- Docker is ...
- Core Features
- Docker use cases in the world

05. Creating our Docker Image

- Walkthrough the Dockerfile
- Let’s build the image
- Let’s run the container

06. Demo ETL

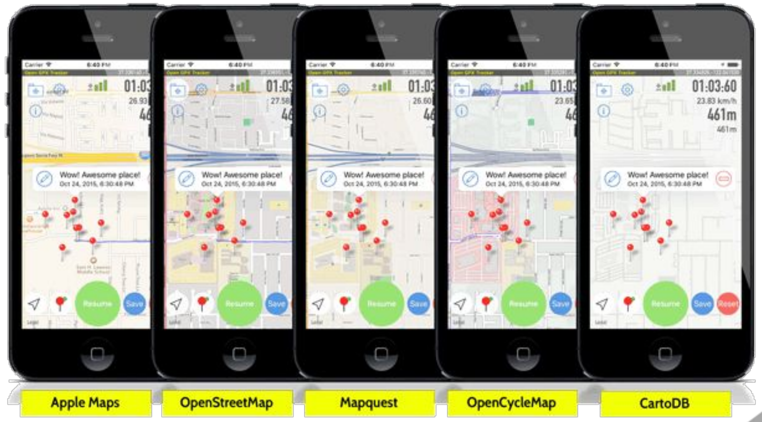
- Detailed example of a data pipeline
- Spark Code + UI + results review

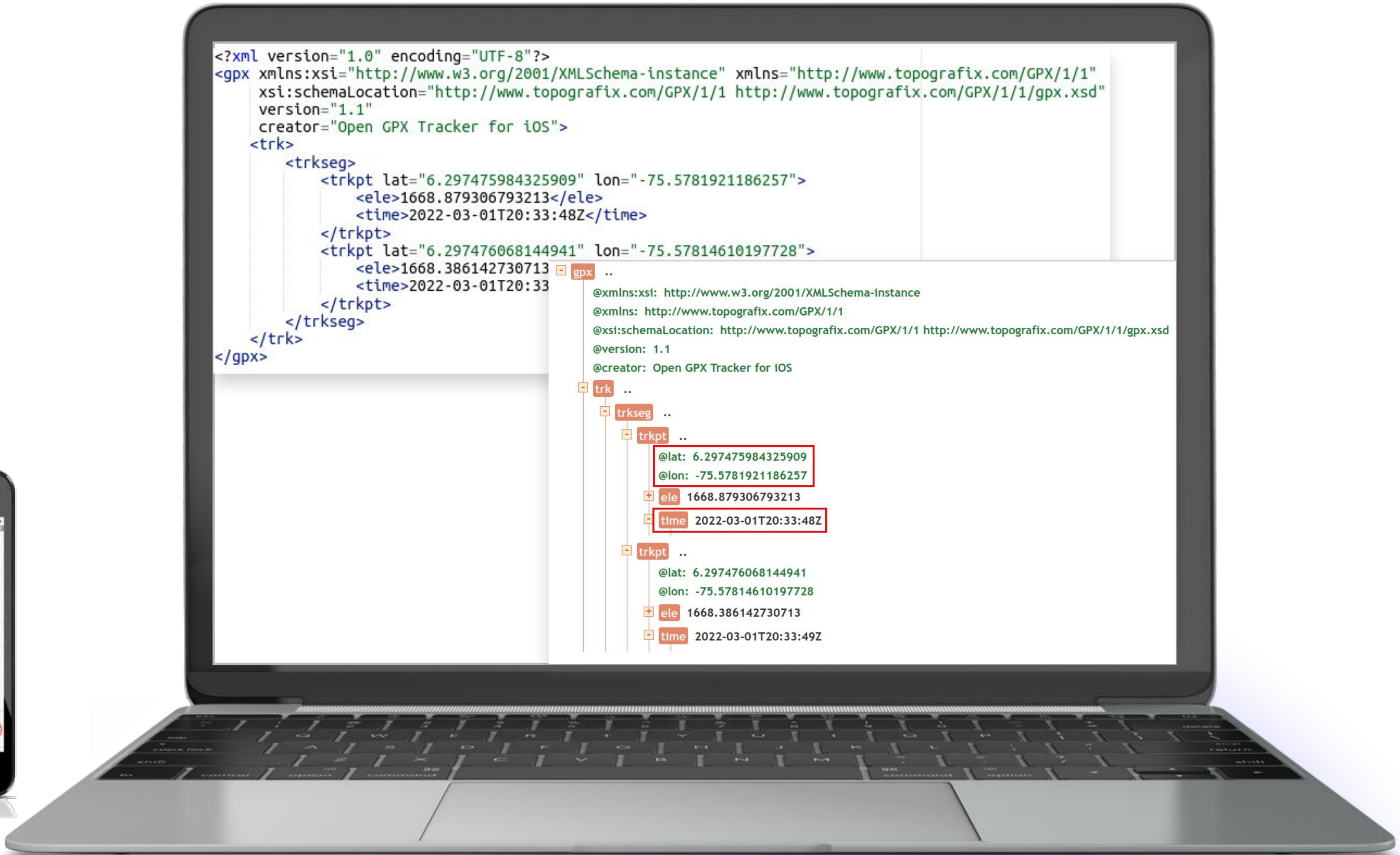
SOURCE & ARCHITECTURE

DATA SOURCE

Open GPX Tracker

Open GPX Tracker supports multiple map tile servers

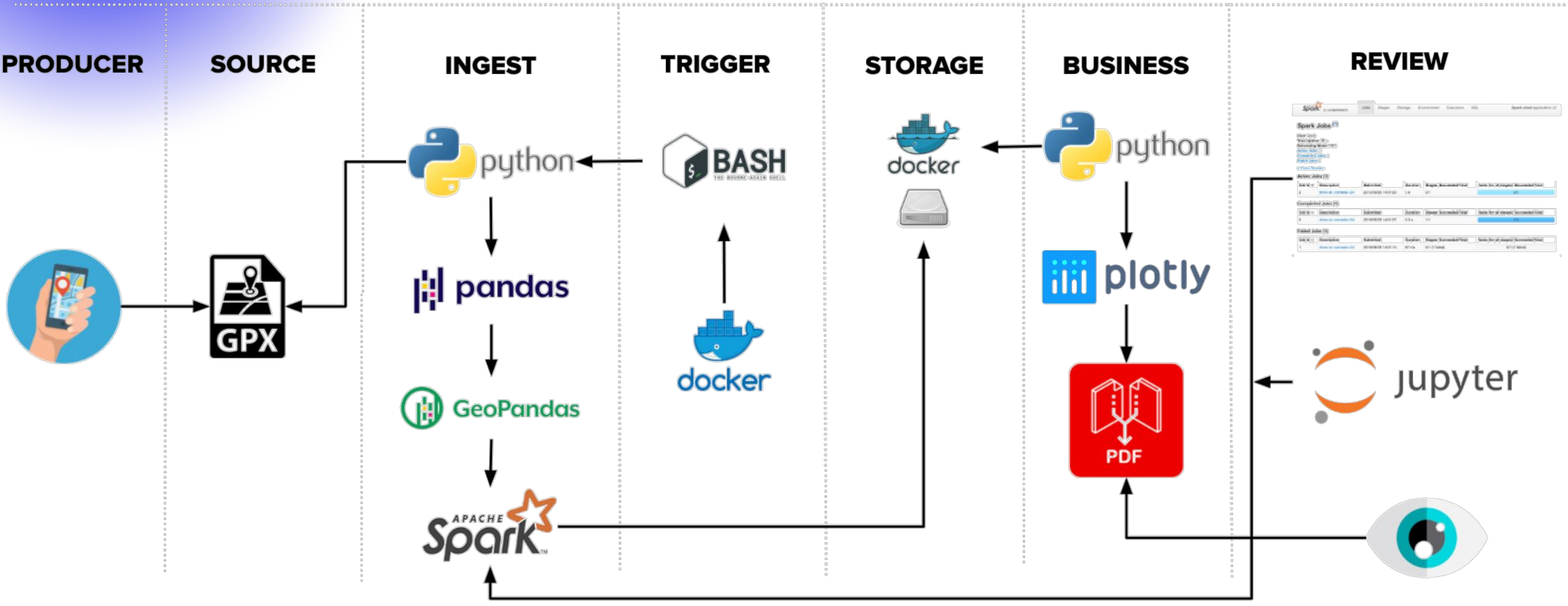




```
<?xml version="1.0" encoding="UTF-8"?>
<gpx xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.topografix.com/GPX/1/1"
    xsi:schemaLocation="http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd"
    version="1.1"
    creator="Open GPX Tracker for iOS">
  <trk>
    <trkseg>
      <trkpt lat="6.297475984325909" lon="-75.5781921186257">
        <ele>1668.879306793213</ele>
        <time>2022-03-01T20:33:48Z</time>
      </trkpt>
      <trkpt lat="6.297476068144941" lon="-75.57814610197728">
        <ele>1668.386142730713
        <time>2022-03-01T20:33
      </trkpt>
    </trkseg>
  </trk>
</gpx>
```

```
gpx ..
  @xmlns:xsi: http://www.w3.org/2001/XMLSchema-instance
  @xmlns: http://www.topografix.com/GPX/1/1
  @xsi:schemaLocation: http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd
  @version: 1.1
  @creator: Open GPX Tracker for iOS
  trk ..
    trkseg ..
      trkpt ..
        @lat: 6.297475984325909
        @lon: -75.5781921186257
        ele 1668.879306793213
        time 2022-03-01T20:33:48Z
      trkpt ..
        @lat: 6.297476068144941
        @lon: -75.57814610197728
        ele 1668.386142730713
        time 2022-03-01T20:33:49Z
```

FLOW & COMPONENTS



PRODUCER

SOURCE

INGEST

TRIGGER

STORAGE

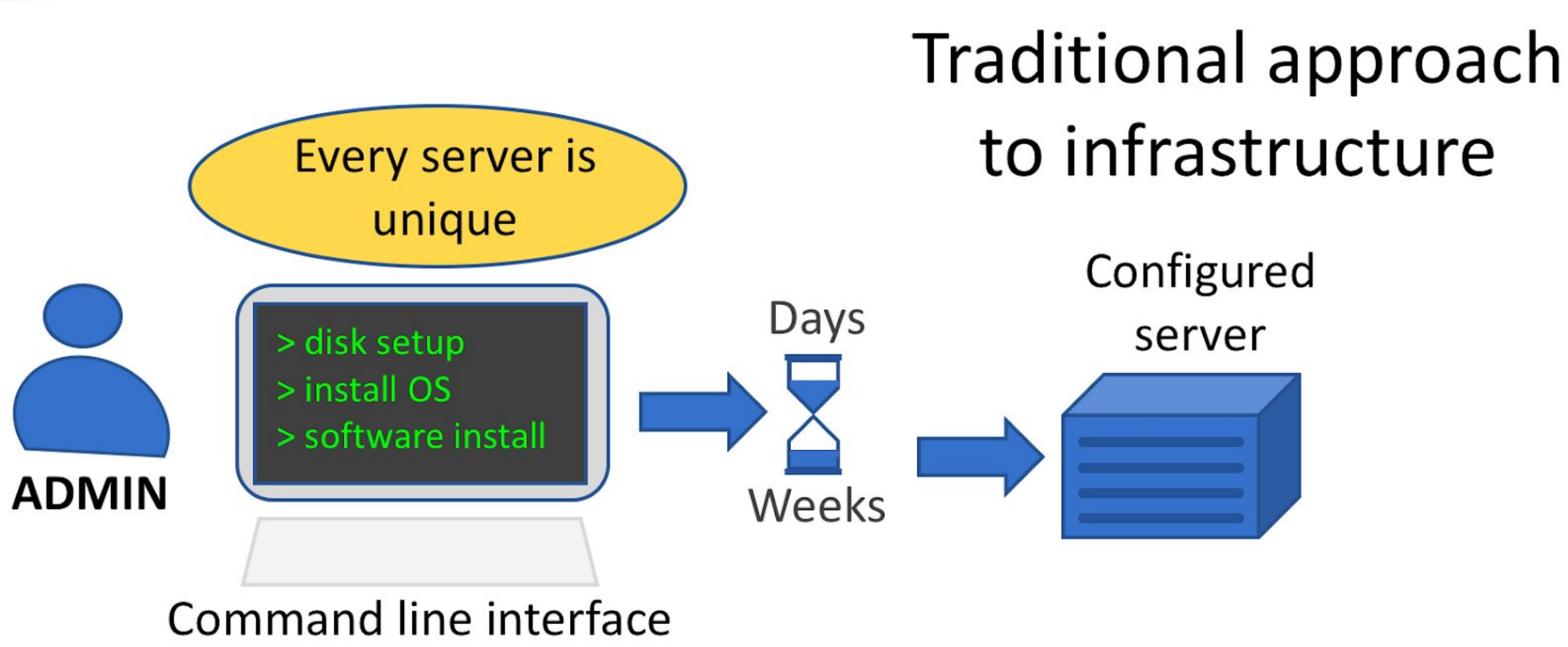
BUSINESS

REVIEW

OTHERS flake8 : my[py] GitLab GitHub WK<html>TOpdf HTML GNU Make Parquet

CORE CONCEPTS

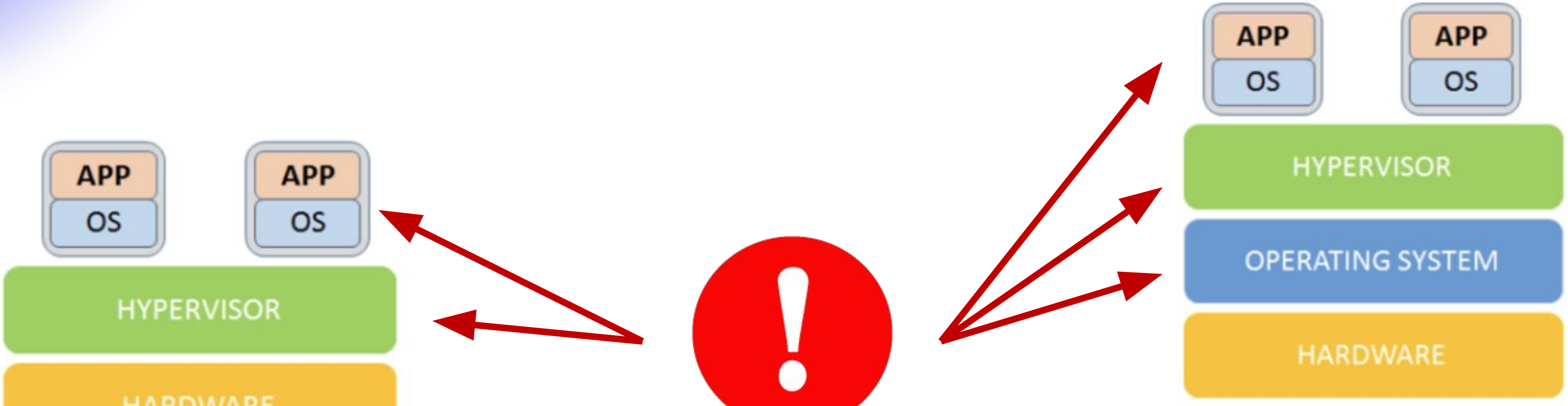
UNDERSTANDING THE TRADITIONAL IT INFRASTRUCTURE



Traditional approach to infrastructure

```
> disk setup
> install OS
> software install
```

UNDERSTANDING THE HYPERVISOR



APP

OS

APP

OS

HYPERVISOR

HARDWARE

Type 1 Hypervisor

APP

OS

APP

OS

HYPERVISOR

OPERATING SYSTEM

HARDWARE

Type 2 Hypervisor

Microsoft Hyper-V

ORACLE VM

vmware ESXi

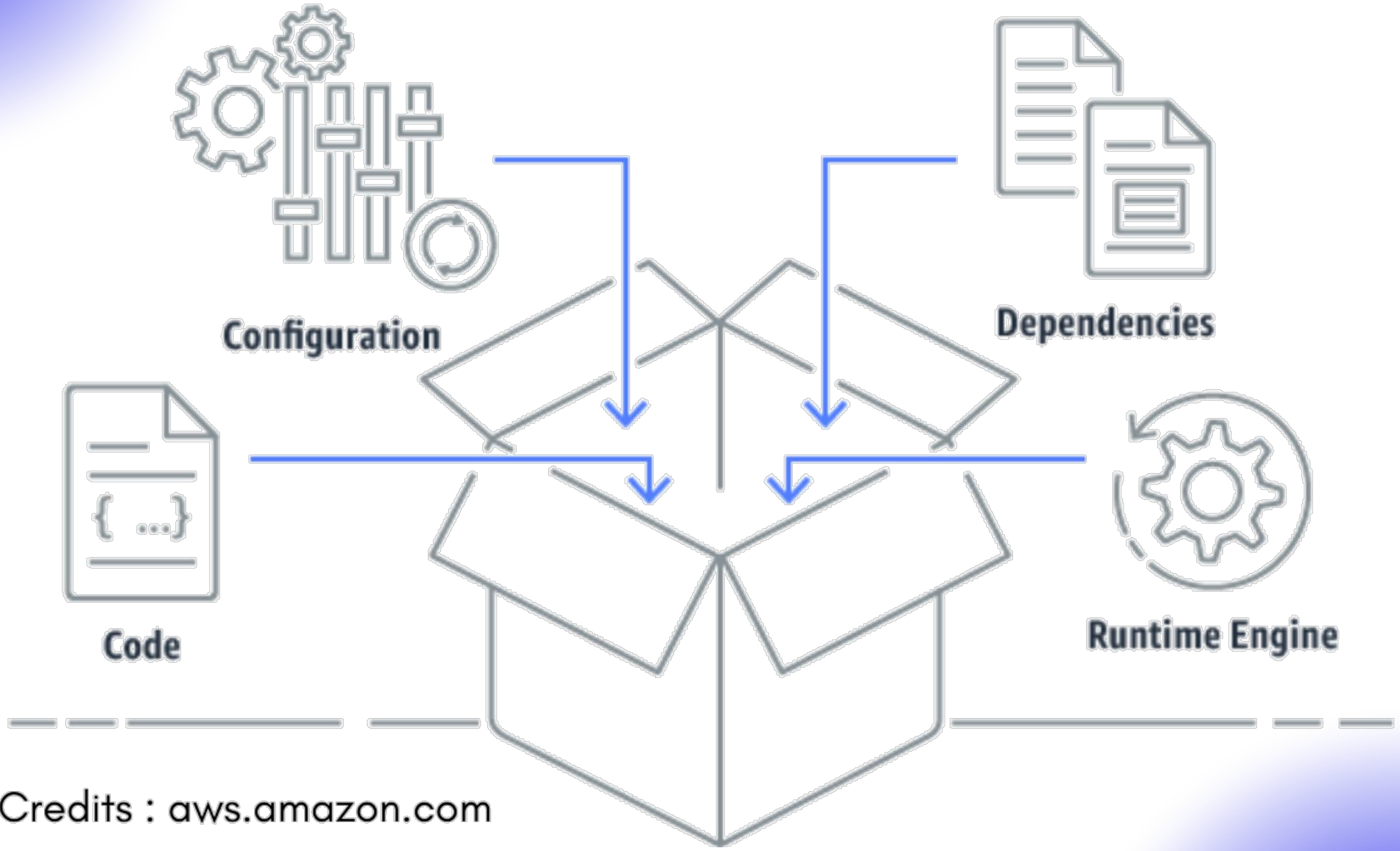
CiTRIX Hypervisor

VirtualBox

vmware

QEMU

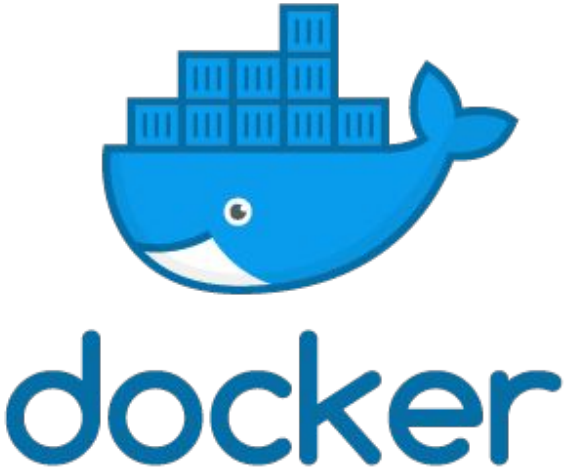
UNDERSTANDING THE CONTAINER

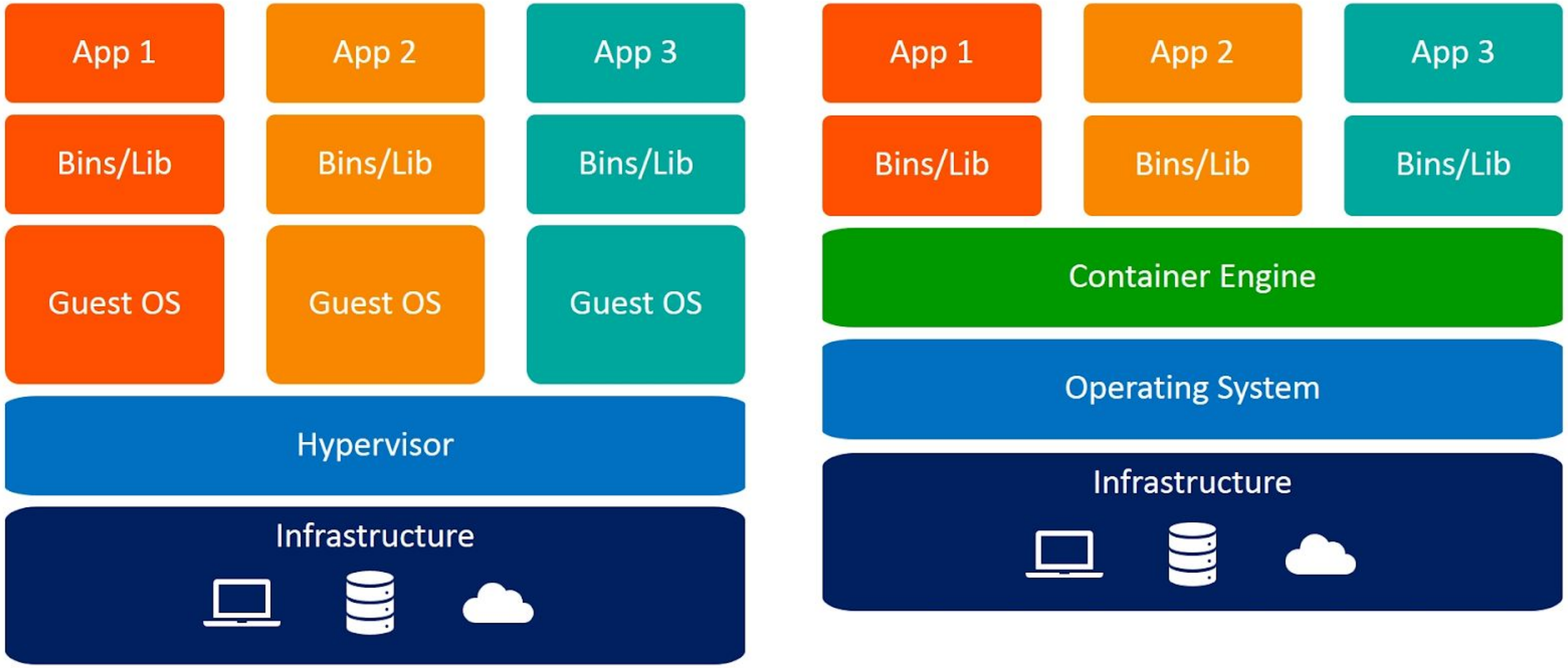


Pic Credits : aws.amazon.com

WHAT IS DOCKER?

DOCKER





App 1 | App 2 | App 3

Bins/Lib | Bins/Lib | Bins/Lib

Guest OS | Guest OS | Guest OS

Hypervisor

Infrastructure

Virtual Machines

App 1 | App 2 | App 3

Bins/Lib | Bins/Lib | Bins/Lib

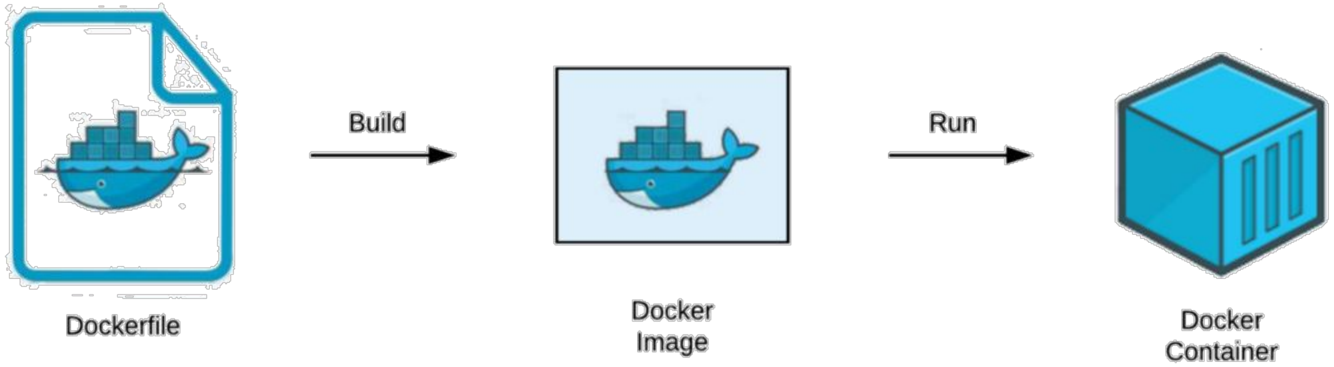
Container Engine

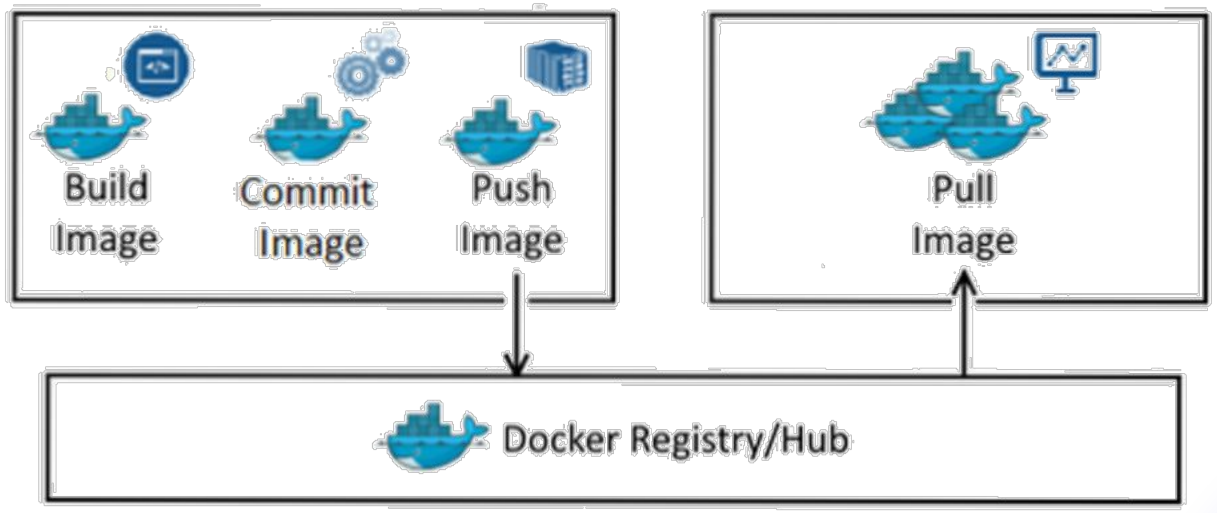
Operating System

Infrastructure

Containers

CORE FEATURES





DOCKER USE CASES IN THE WORLD

- SIMPLIFIED CONFIGURATION
- SERVER CONSOLIDATION
- PRODUCTIVITY
- PIPELINES
- AUTOMATION
- EASY CONFIGURATION
- APP ISOLATION
- CODE VALIDATION
- DEBUG CAPABILITIES

13 million + developers

7 million + applications

13 billion + monthly image downloads

GitHub

GitLab

ATLASSIAN Bitbucket

aws

Google Cloud

Azure

AT&T

NETFLIX

PayPal

CREATING OUR DOCKER IMAGE

Walkthrough the Dockerfile

We're going to defining the context of our Docker container step by step.

Let's build the Image

After defining the context within the Dockerfile, let's buid the Docker Image.

Let's build the Container

Right after the Image has built, let's start the cointainer and play with it.

DEMO ETL

Detailed example of a data pipeline

We've buit a local represantation of how a data pipeline would work from end to end.

Spark Code + UI + Results Review

Once the data pipeline has finished, let's review its results and also let's dive into the Spark code and the History of operation performed thorughout the UI.

Break Section

01:00pm ~ 01:30pm

QUESTIONS ?

THANKS !