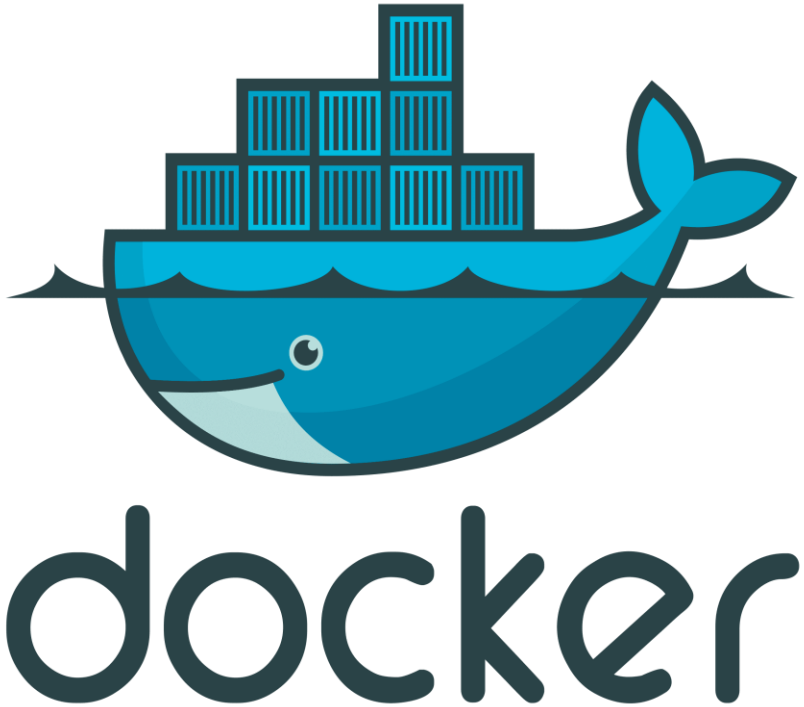


Docker Containerization

Hitesh Kumar Sharma
Technical Trainer

Session Agenda

- About Me
- Course Overview
- Course Features
- Course Delivery Mode
- Prerequisites
- Key Learning Outcomes
- IT Issues before Containerization
- Solution by Containerization
- Virtual Machines vs Containerization
- Problems with VMs
- Solution with Docker Containers
- Docker Container Architecture
- Docker Containers in Nutshell

HITESH KUMAR SHARMA

Technical Instructor & Consultant

ABOUT ME

- Industry Experience: **15 Years (5 Years in Mobile App development, 10 Years in DevOps and Data Science)**
- Worked as **IBM Instructor**
- Worked as **Microsoft Instructor**
- Core Technical Domains: **Android/iOS App Development**, **UiPath RPA, DevOps, Data Analytics**
- Academic Qualifications**: Ph.D. (CSE), M.Tech (CSE)**
- Certifications:
    - **UiPath RPA Certified Associate**
    - **Docker Certified Associate**
    - **Neo4J Certified Associate**
    - **Maven Certified Professional**
- 4 Books Published
- 30 Patents Published
- 02 Copyright Published

Program Overview

Docker Containerization Course will educate the learners the core Docker Technologies such as

- Docker Containers
- Docker Engine
- Docker Images
- Docker Network
- Docker Daemon
- Docker Hub
- Docker Compose
- Dockerfile and
- Docker Storage along with real-life case studies.

Course Features

- 04 Hours Daily of Instructor-Led Training
- Hands-On Lab Sessions
- Mid-Session Quiz
- Lesson-End Knowledge Checks
- Lesson-End Projects
- Course-End Assessment
- Daily Session Slides
- Online Self Learning Content

Course Delivery Mode

- Hands-on Live Demonstration
- Whiteboard/ Pen Tab based
- Slides Based Content

Prerequisites

- Basic Hands-on on Linux
- Basic Knowledge of Programming
- Basic Script Writing Skills

Table of Content for the Course

- Introduction of Virtual Machines and Containers
- Docker Architecture and its Ecosystem
- Docker CE on Linux Platform
- Docker Networking
- Docker Images
- Docker Storage and Volumes
- Docker Compose
- Universal Control Plane
- Docker Trusted Registry
- Docker Security

1.3 Shipping Industry Challenges

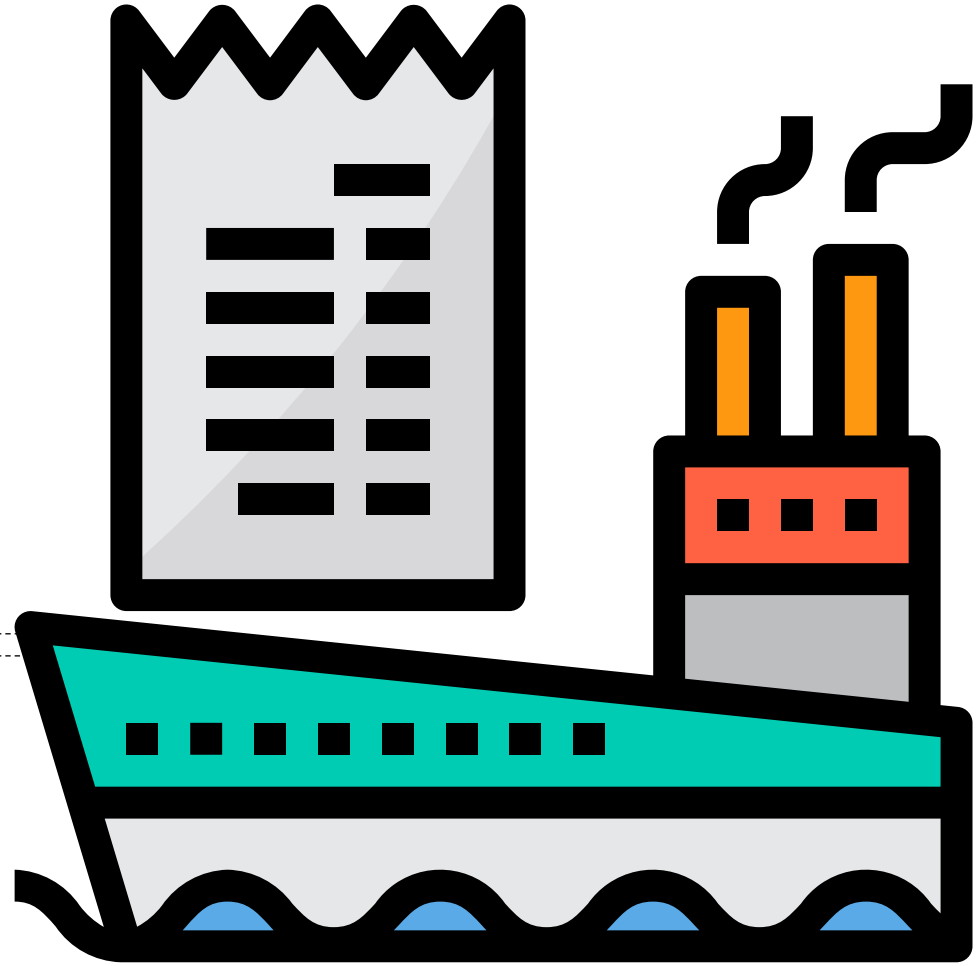
The various challenges faced by the shipping industry.

Multiplicity of goods

Do I worry about how goods interact? (e.g., coffee beans next to spices)

Multiplicity of methods for transporting/storing

Can I transport quickly & smoothly? (e.g., from boat to train to truck)



1.4 Container: The Saviour

How did the container become the saviour?

Multiplicity of goods

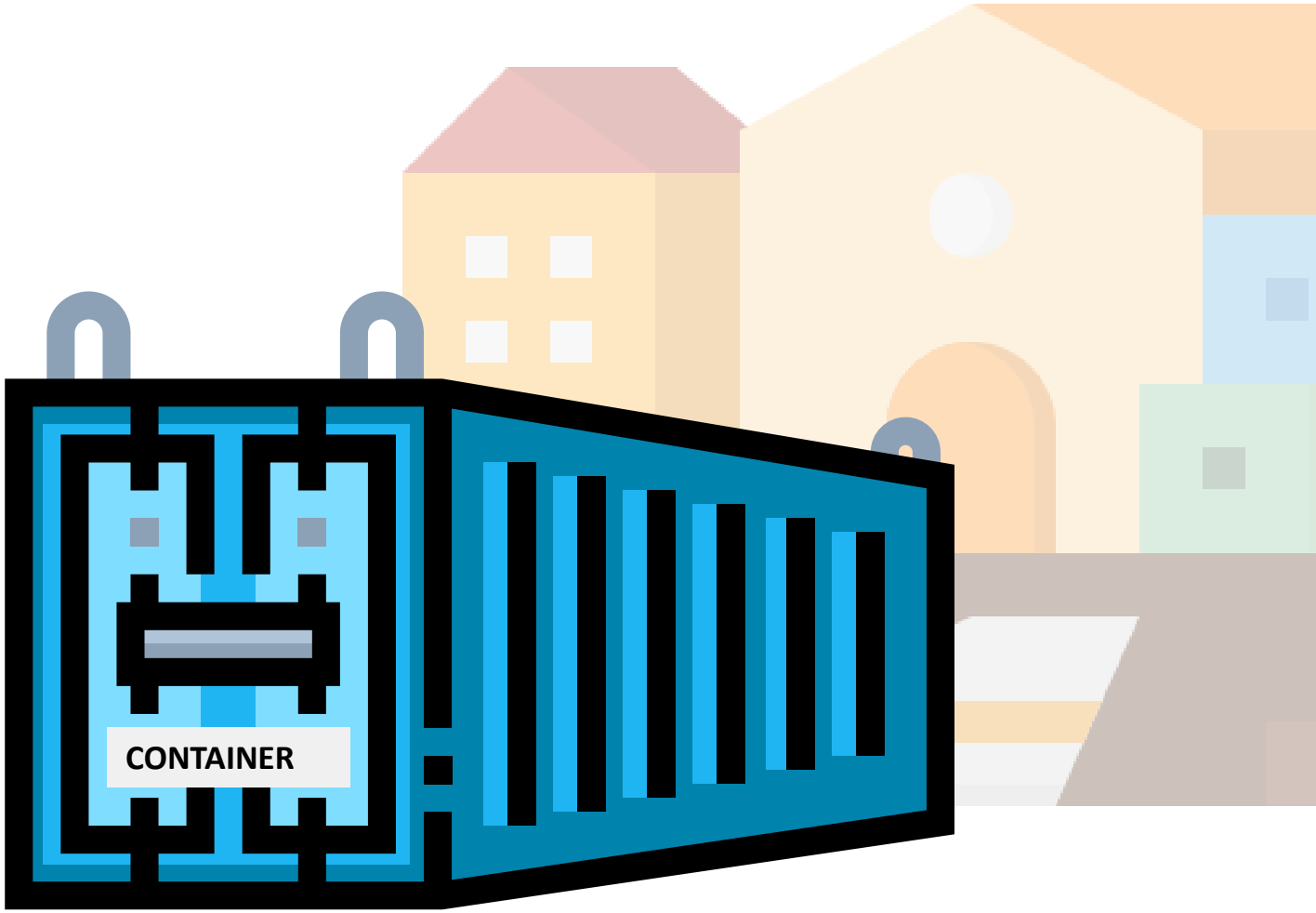
Do I worry about how goods interact? (e.g., coffee beans next to spices)

A standard container that is loaded with virtually any goods, & stays sealed until it reaches final delivery.

Multiplicity of methods for transporting/storing

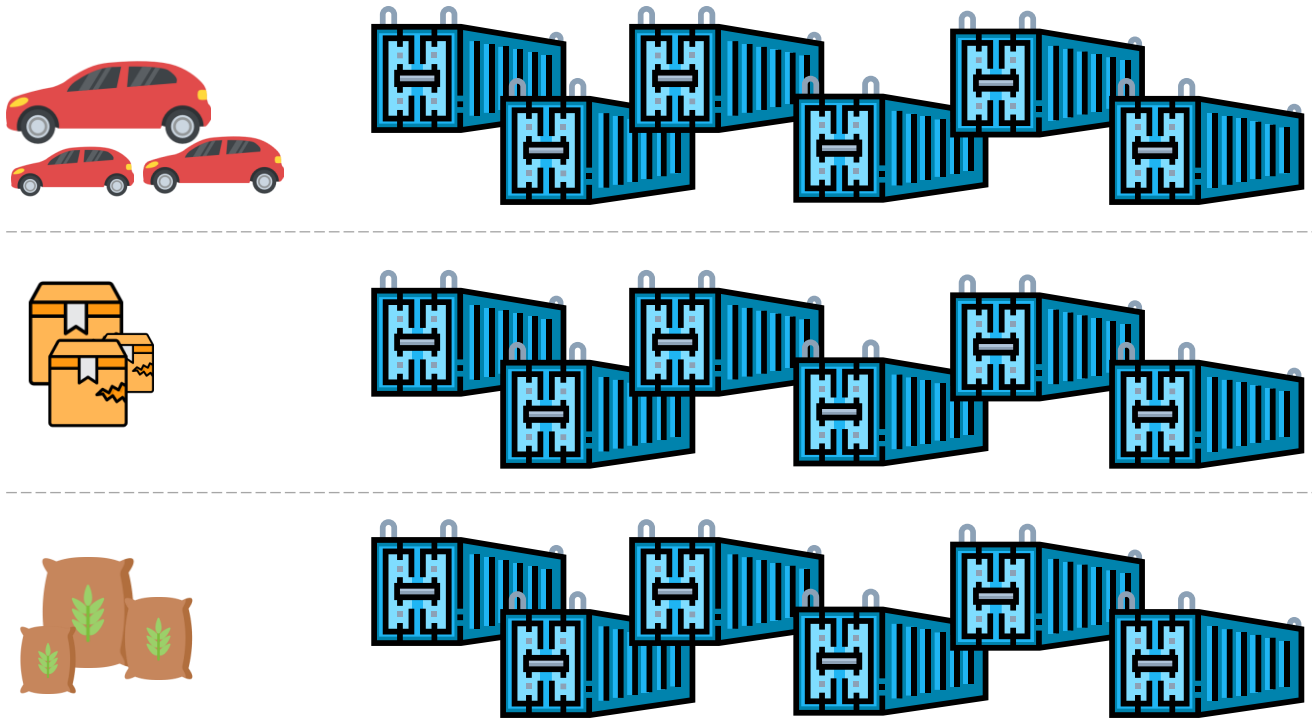
Can I transport quickly & smoothly? (e.g., from boat to train to truck)

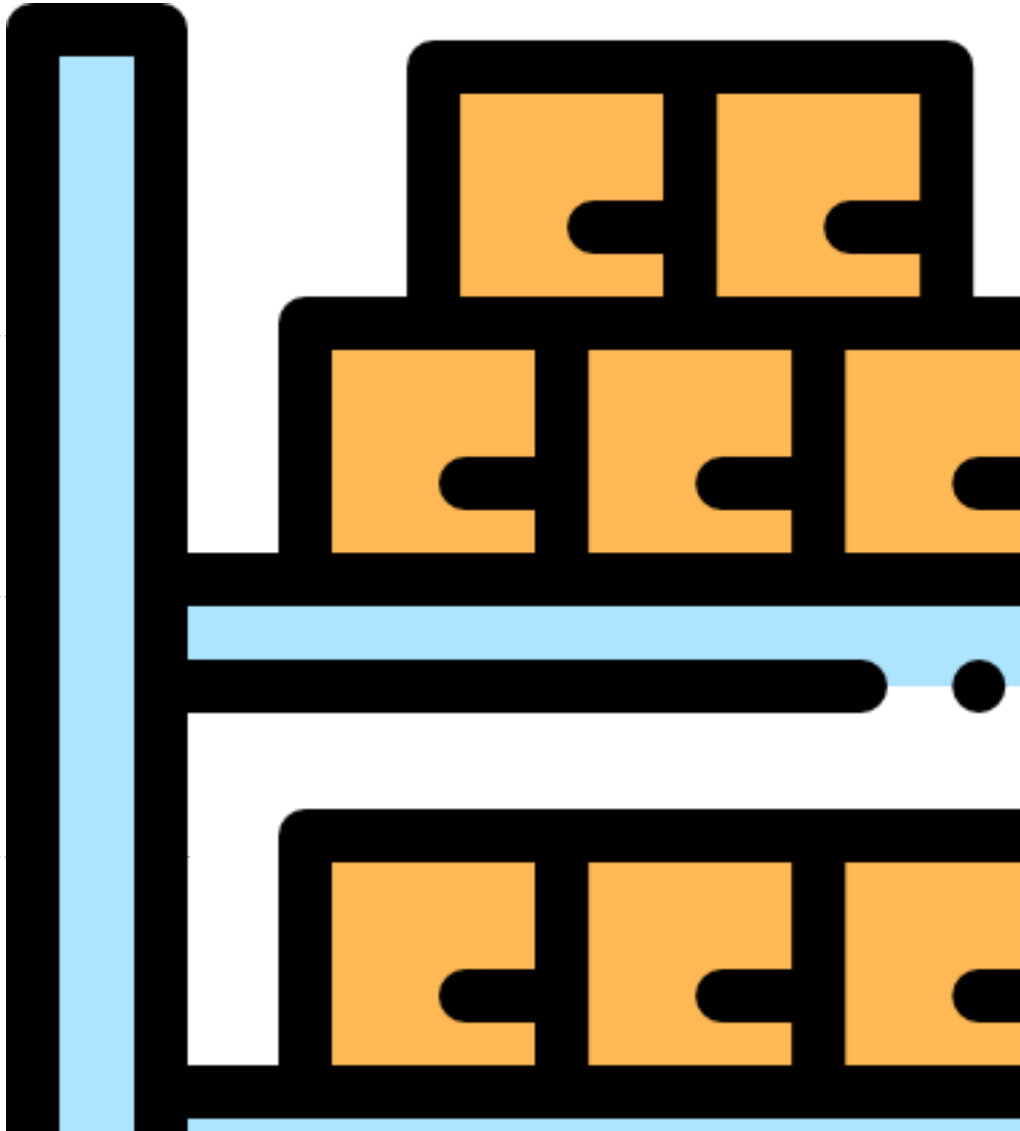
In between, can be loaded & unloaded, stacked, transported efficiently over long distances, & transferred from one mode of transport to the other.



1.5 Solution by Containers in the Shipping Industry

Everything falls into place with the help of containers.





1.6 Challenges in the Software Industry (Contd.)

The various challenges in the software industry are as follows:

Do services & apps interact appropriately?

Multiplicity of Stacks

Static Website
nginx 1.5, modsecurity, openssl, bootstrap2

User DB
postgresql, pgv8, v8

Web frontend
Ruby, Rails, sass, Unicorn

Background workers
Python 3.0, celery, pyredis, libcurl, ffmpeg, libopencv,nodejs, phantomjs

API Endpoint
Python 2.7, Flask, pyredis, celery, psycopg, postgresql-client

Queue
Redis, redis-sentinel

Can I migrate smoothly & quickly?

Multiplicity of hardware environments

Development VM

Customer Data Center

Contributor's laptop

Public Cloud

QA server

Production cluster

Data recovery

Production Servers

1.7 Problems in Software Industry Before Containers (Contd.)

The chaos in the software industry while managing diverse stack in different environments:

	Developmen VM	QA Server	Single Prod Sever	Onsite Cluster	Public Cloud	Contributor Laptop	Customer Servers
Static Website	?	?	?	?	?	?	?
Background Workers	?	?	?	?	?	?	?
Web Front End	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

1.8 Put that in Container!

Multiplicity of stacks

Developer: Build once, run anywhere(finally)

Static Website
nginx 1.5, modsecurity, openssl, bootstrap2

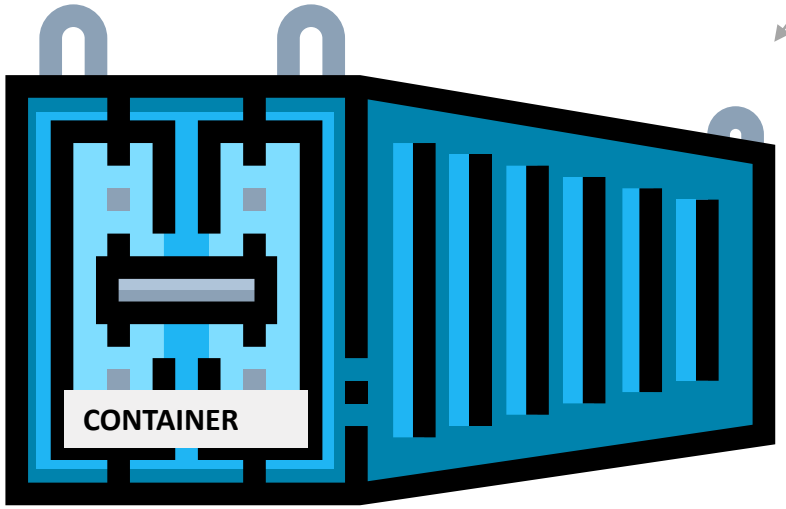
User DB
postgresql, pgv8, v8

Web frontend
Ruby, Rails, sass, Unicorn

Background workers
Python 3.0, celery, pyredis, libcurl, ffmpeg, libopencv,nodejs, phantomjs

API Endpoint
Python 2.7, Flask, pyredis, celery, psycopg, postgresql-client

Queue
Redis, redis-sentinel



Multiplicity of hardware environments

Operator: Configure once, run anything

Development VM

Production Servers

Customer Data Center

Contributor's laptop

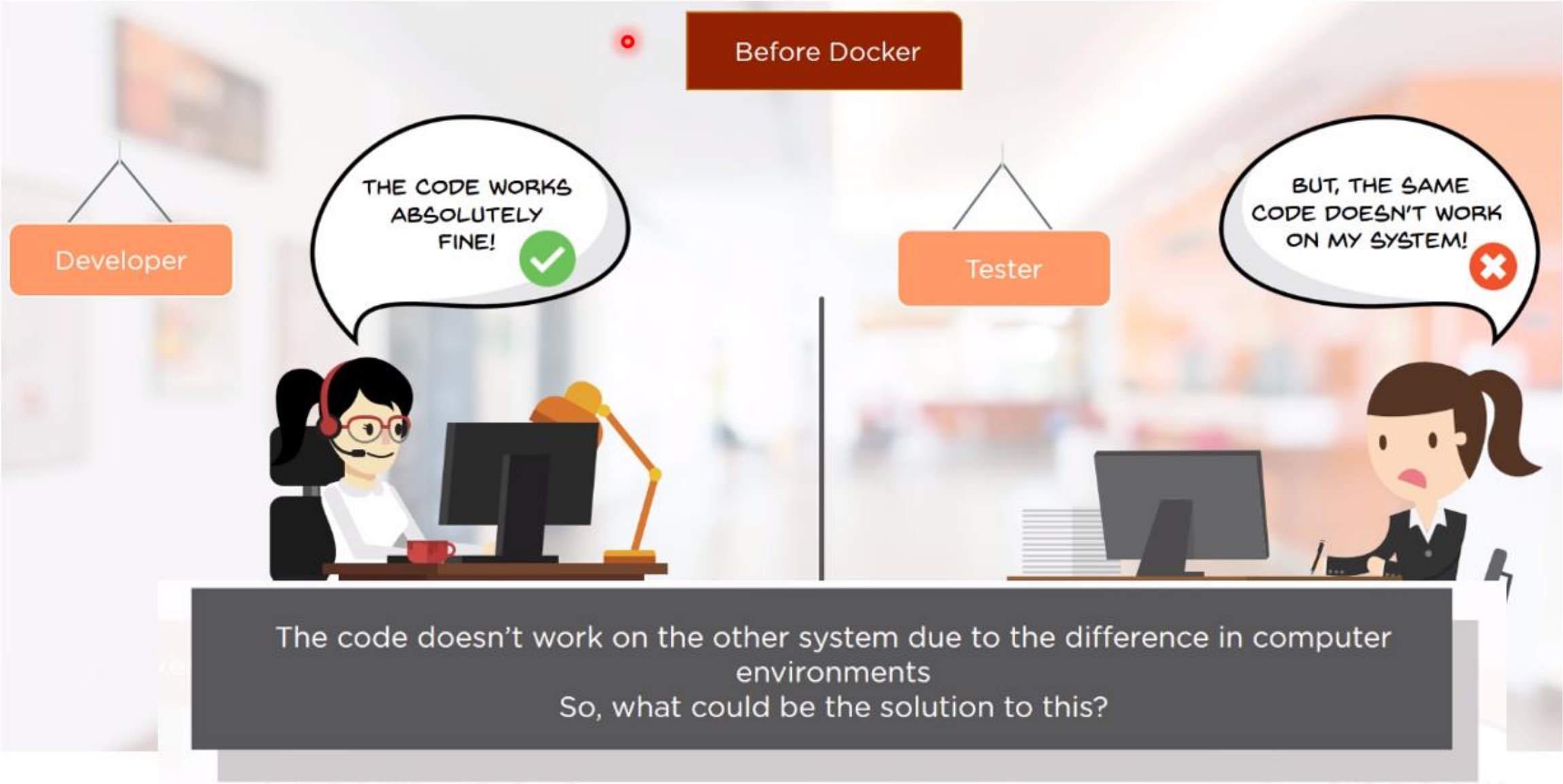
QA server

Production cluster

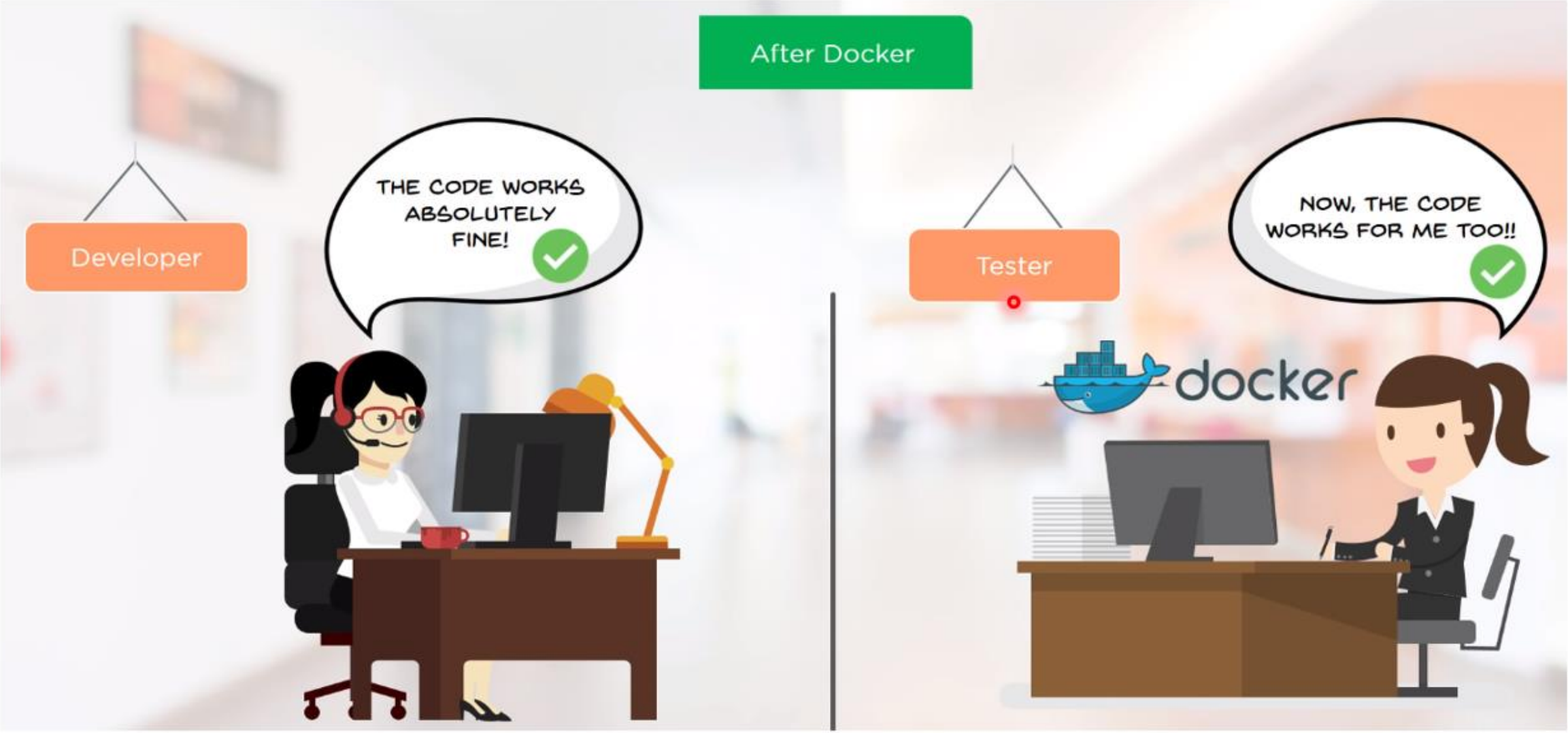
Data recovery

Source: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRWHcB-4JmViuSdW3tEtckQ2HLOrIhv-AnO9_6jzUohl1NmPyOV&s

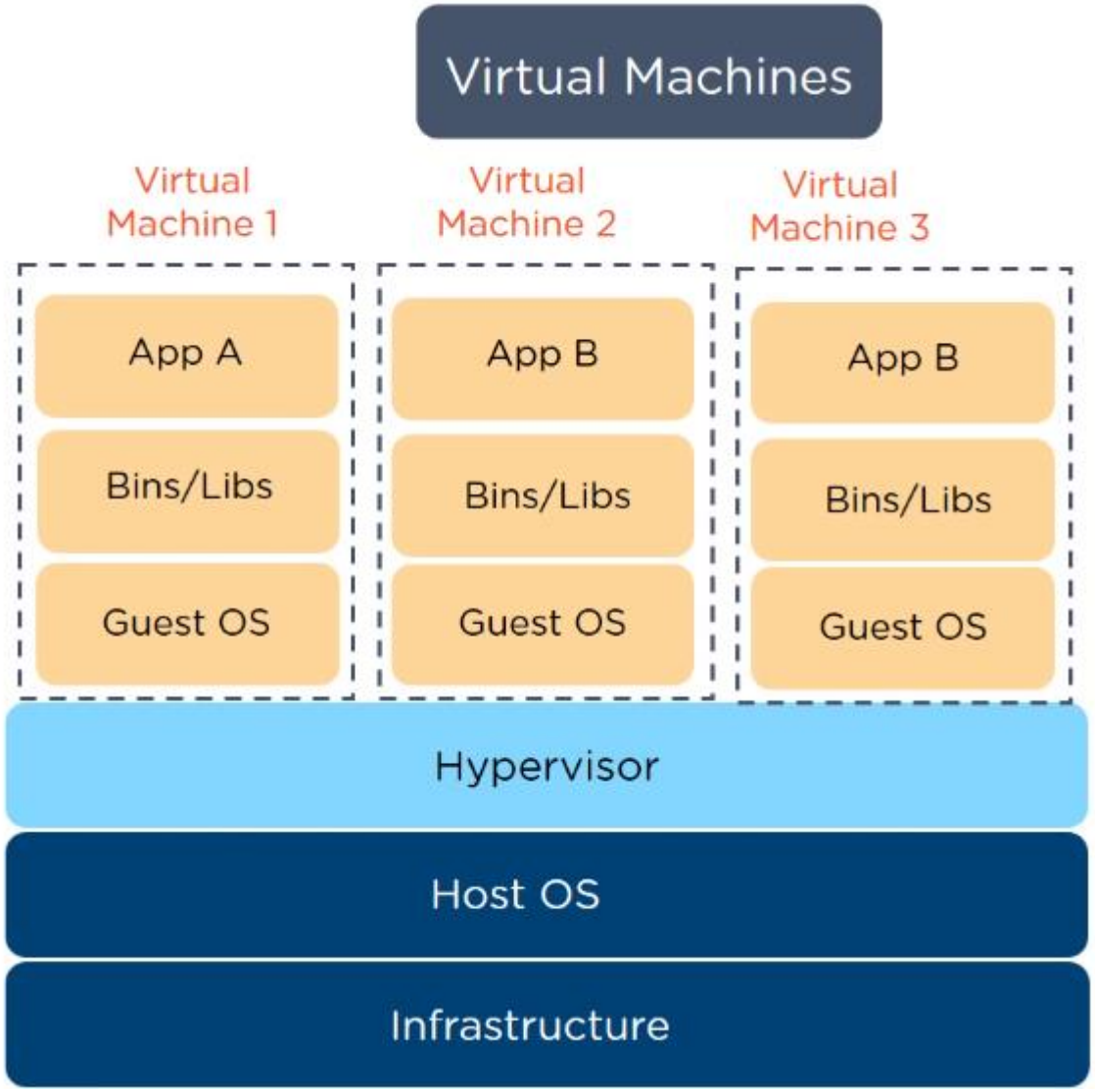
IT Issues Before Docker Containerization

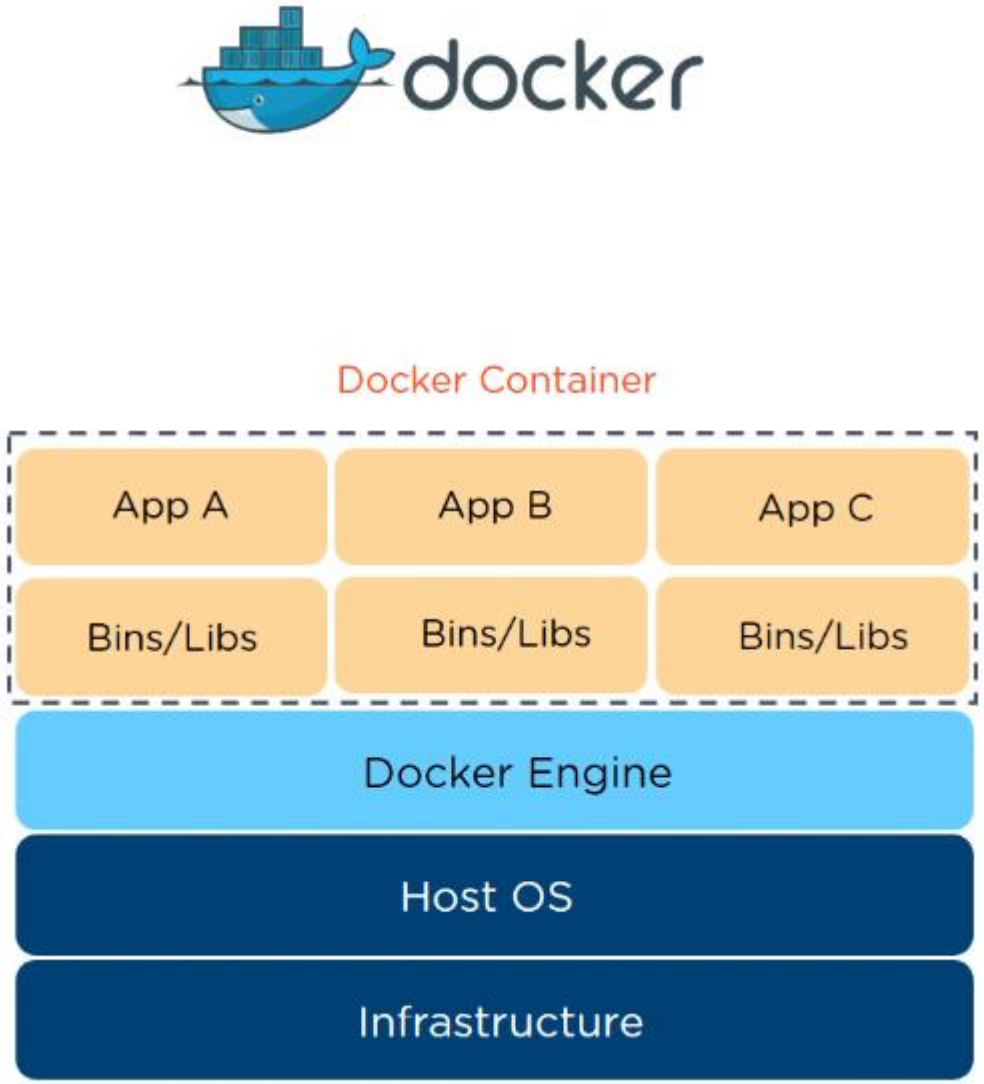


After Docker Containerization



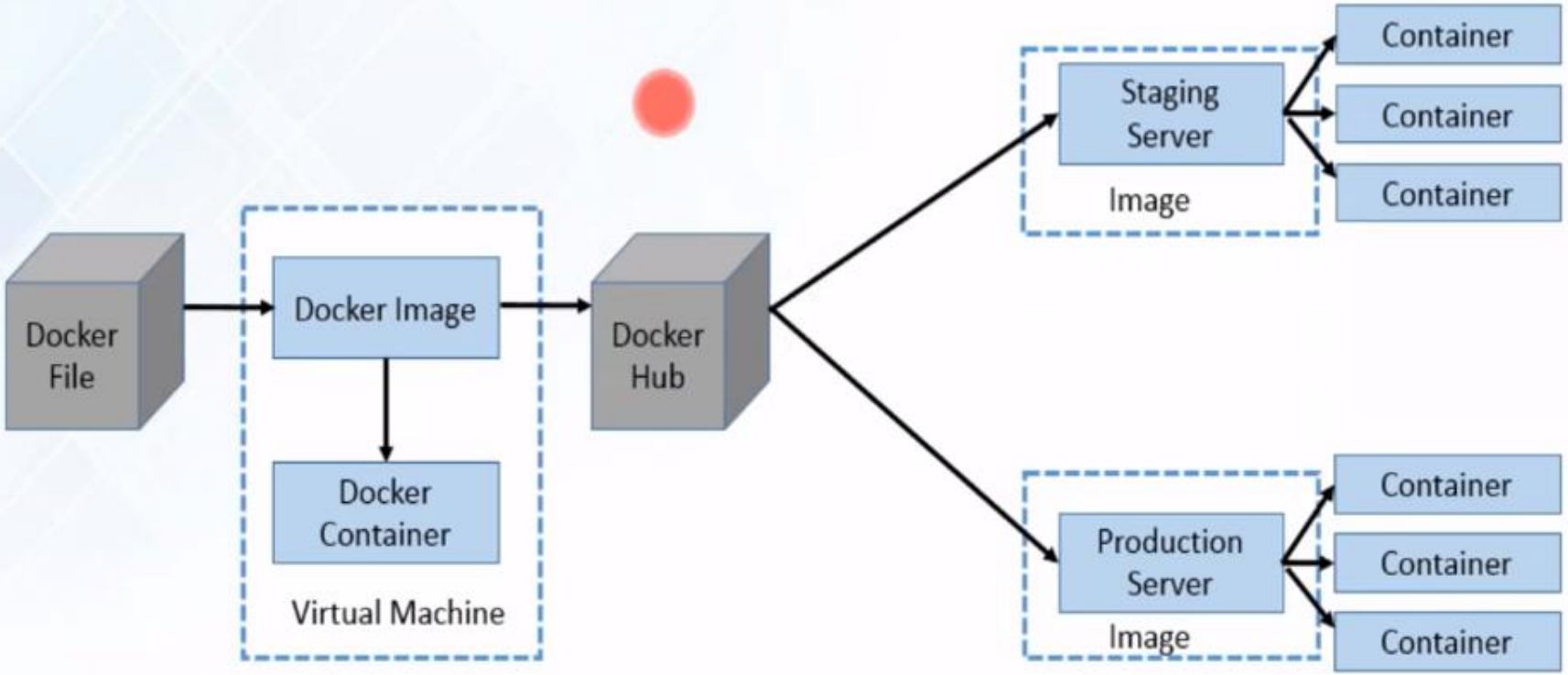
Virtual Machines vs Docker Containers





Docker in Nutshell

- 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 Commands (Hands-On)

- **docker –version**
- **docker pull**
- **docker run**
- **docker ps**
- **docker ps -a**
- **docker run -it**
- **docker stop**
- **docker login**
- **docker push**
- **docker images**
- **docker rm**
- **docker rmi**
- **docker build**

Thank You