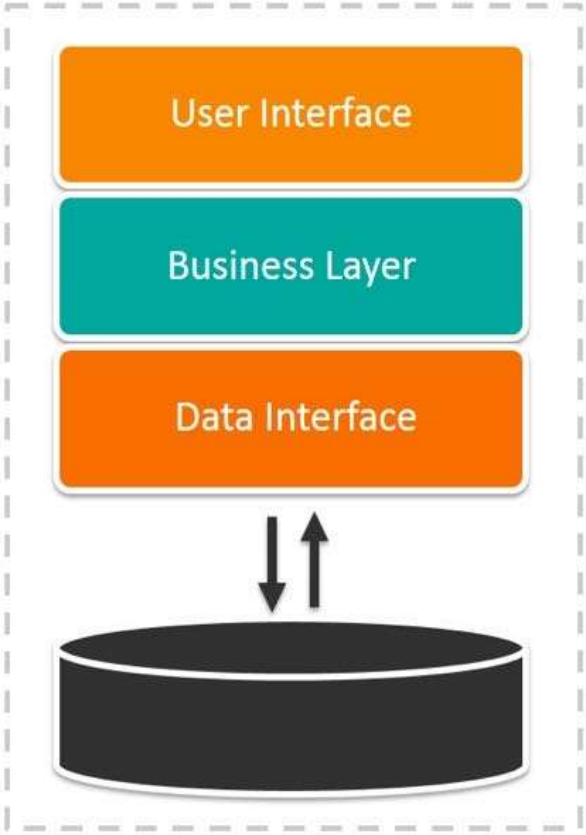
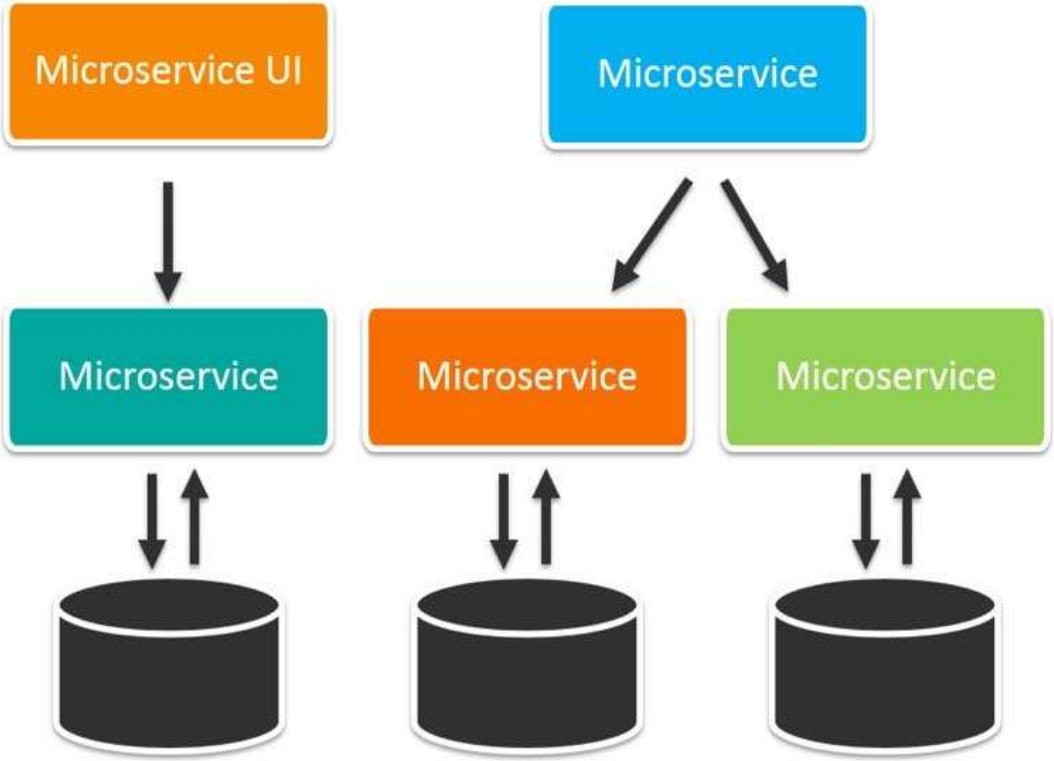
Docker & Kubernetes

Monolithic Architecture



Microservices Architecture



History of Docker

2004
Solaris Containers / Zones technology introduced

2008
Linux containers (LXC 1.0) introduced

2013
Solomon Hykes starts Docker as an internal project within dotCloud

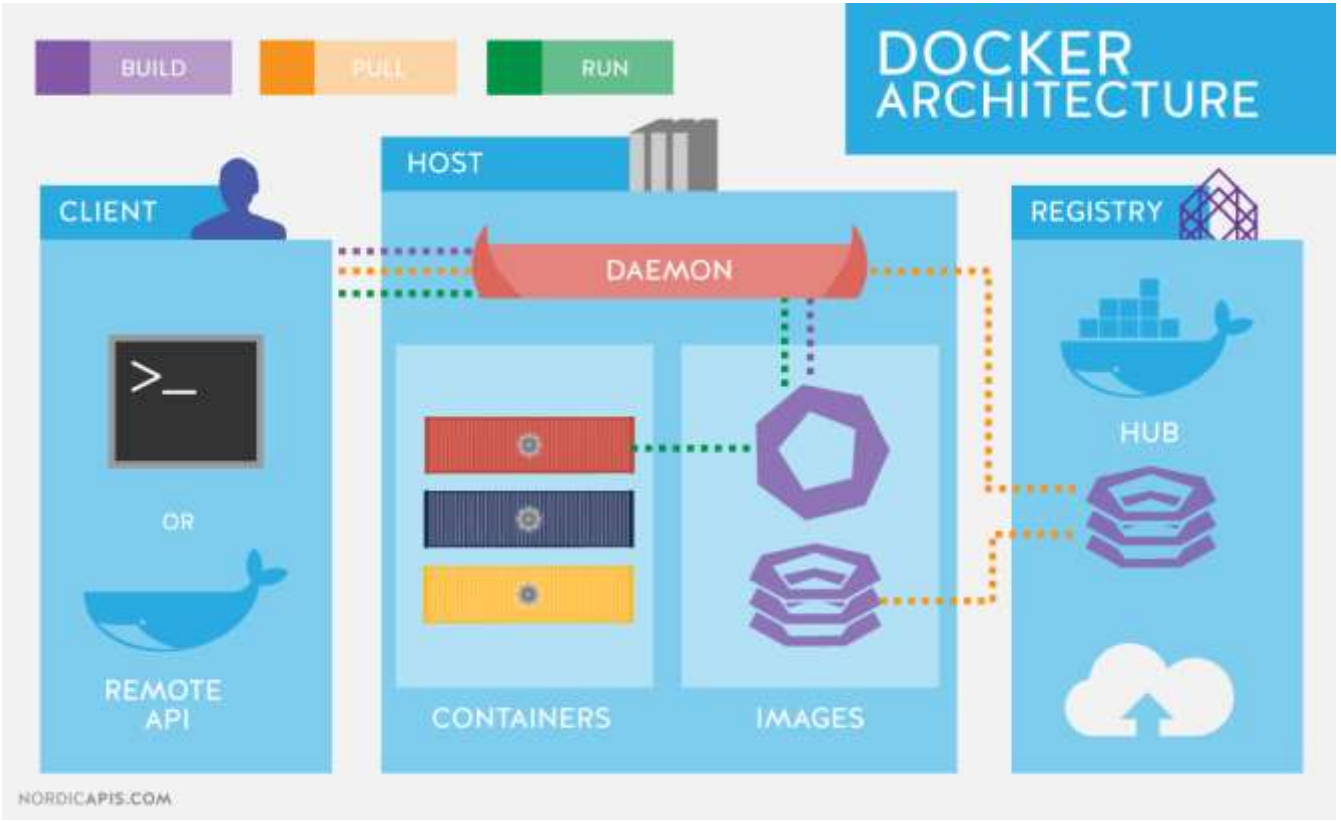
Mar 2013
Docker released to open source

Feb 2016
Docker introduces first commercial product – now called Docker Enterprise Edition

Today
Open source community includes:

- 3,300+ contributors
- 43,000+ stars
- 12,000+ forks

Docker Architecture

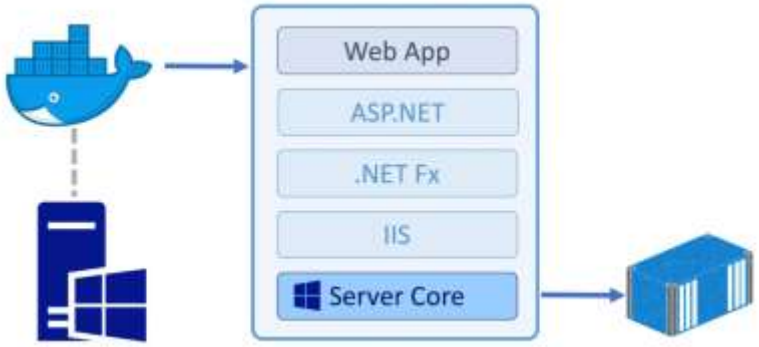


Docker Images

Docker images have intermediate layers that increase reusability, decrease disk usage, and speed up docker build by allowing each step to be cached. These intermediate layers are not shown by default.

The SIZE is the cumulative space taken up by the image and all its parent images. This is also the disk space used by the contents of the Tar file created when you docker save an image.

An image will be listed more than once if it has multiple repository names or tags. This single image (identifiable by its matching IMAGE ID) uses up the SIZE listed only once.

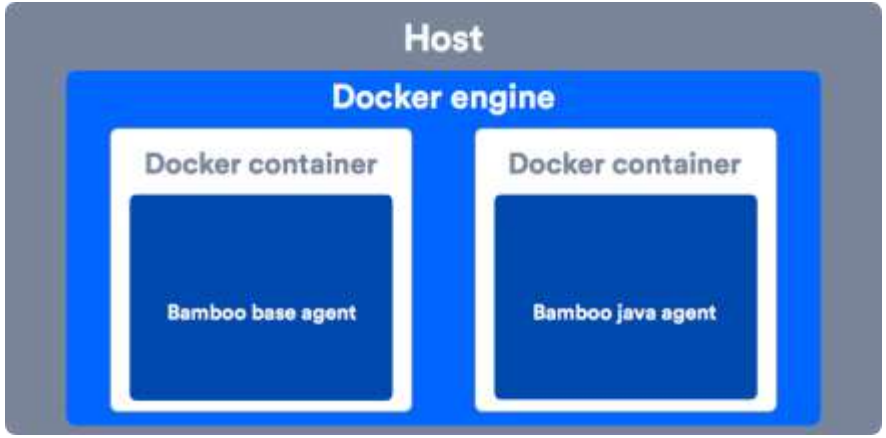


Docker Containers

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

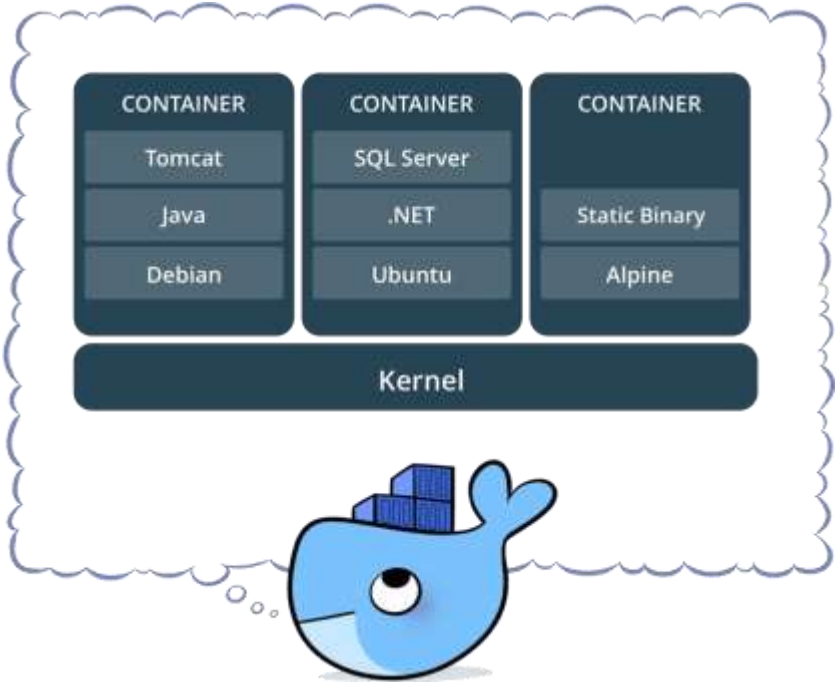
A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.



Source: Atlassian

What is a container?



- Standardized packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works for all major Linux distributions
- Containers native to Windows Server 2016

Tug of War Between Developers and Ops

Developers

- Freedom to create and deploy apps fast
- Define and package application needs

IT Operations

- Quickly and flexibly respond to changing needs
- Standardize, secure, and manage

The Role of Images and Containers

Docker Image

Docker Container

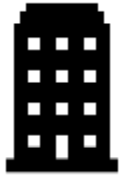
Example: Ubuntu with Node.js and Application Code

Created by using an image. Runs your application.

Docker containers are NOT VMs

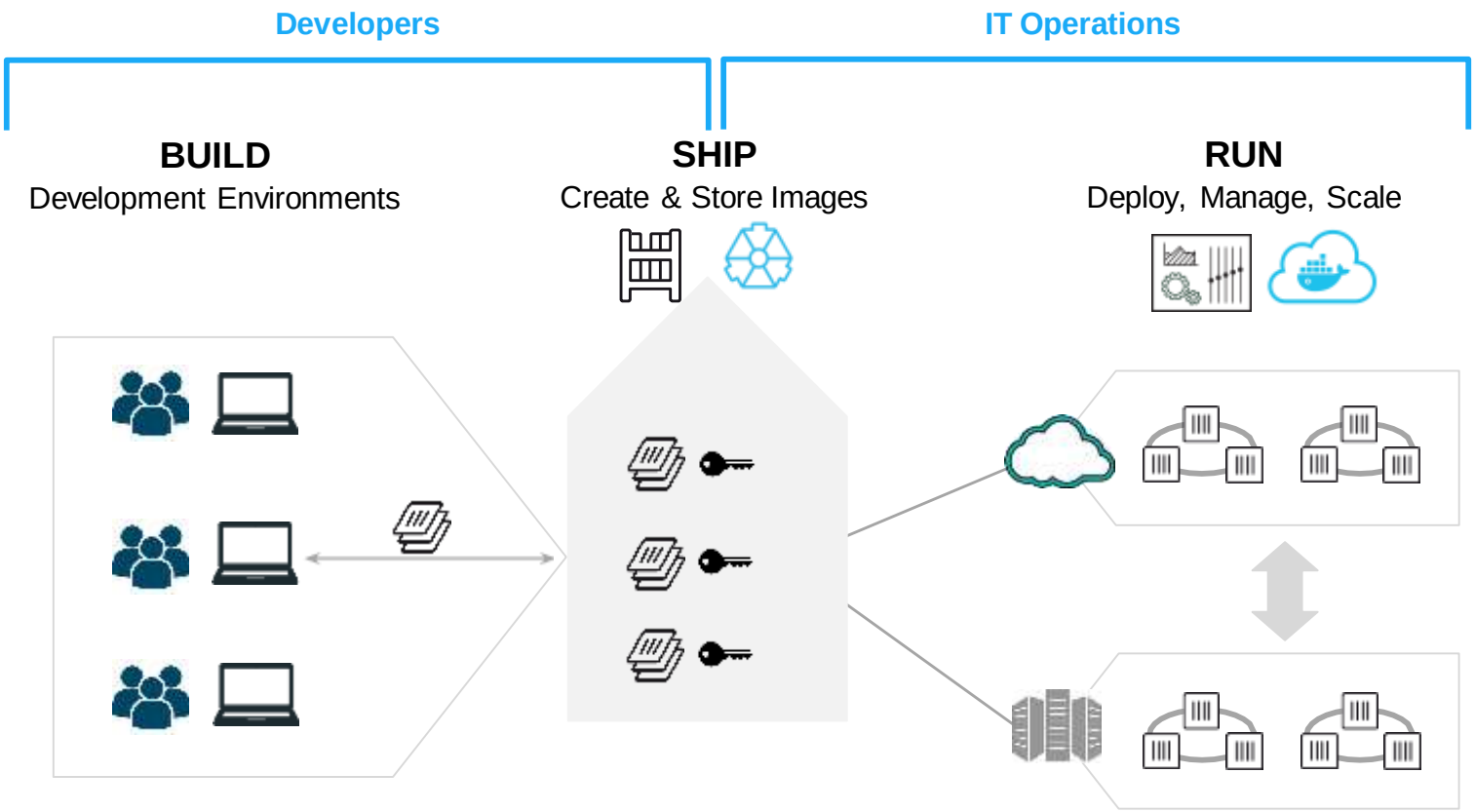
- Easy connection to make
- Fundamentally different architectures
- Fundamentally different benefits

Maquina Virtual

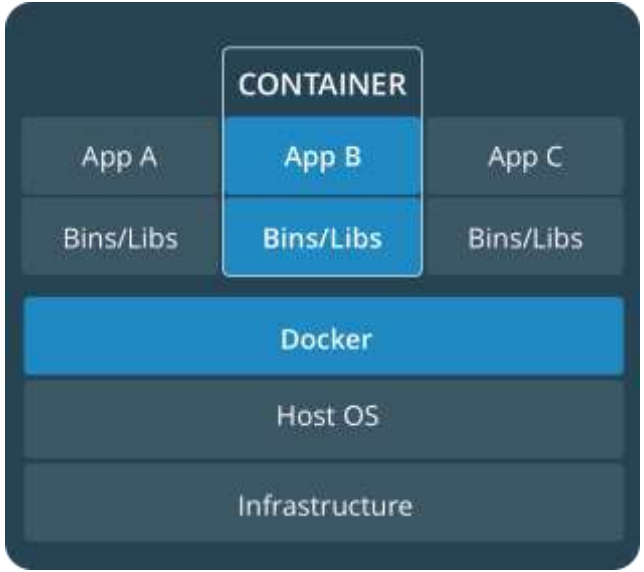


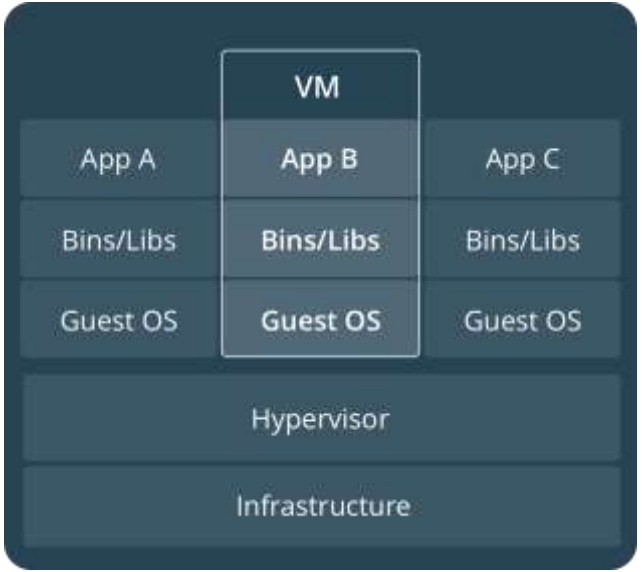
Contenedores

Using Docker: Build, Ship, Run Workflow



Comparing Containers and VMs

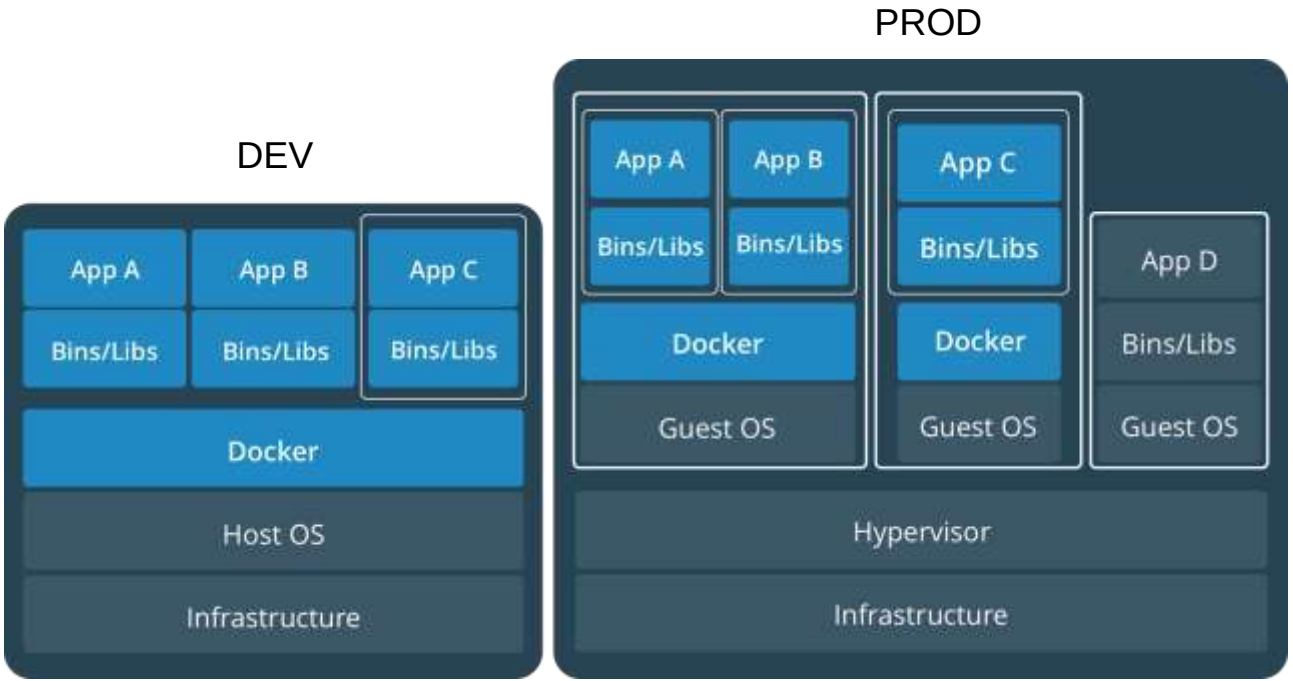




Containers are an app level construct

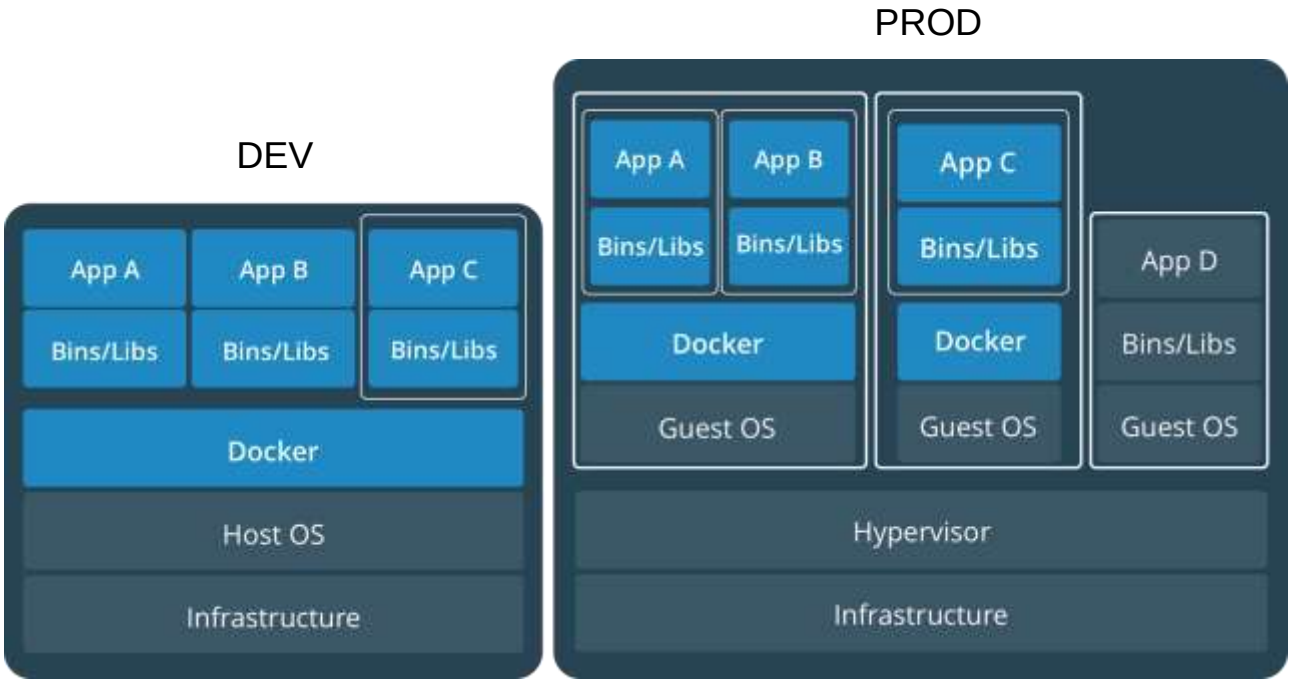
VMs are an infrastructure level construct to turn one machine into many servers

Containers and VMs together



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

Containers and VMs together



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

Docker Is in the Enterprise

Service Provider

- AT&T
- Sprint
- verizon

Healthcare & Science

- AMGEN
- HudsonAlpha Institute for Biotechnology
- MERCK
- OPTUM

Financial Services

- Goldman Sachs
- Northern Trust
- Societe Generale
- VISA
- ADP

Tech

- Juniper Networks
- RSA
- TIBCO
- splunk>
- GE

Insurance

- Anthem
- Liberty Mutual Insurance
- MetLife
- Mutual of Omaha

Public Sector

- CDC
- Cornell University
- GSA
- Indiana University

Dockerfile

- Dockerfile is used to build custom Docker images. You can add the application source or configure and install any application software in Docker image.
- There are a few attributes that can be used in Dockerfile:
  - FROM → Used to define base image information
  - SHELL → Used to set the default shell
  - COPY → Used to transfer the local file to Docker container

Dockerfile (Contd.)

- ADD → This is similar to COPY, but you can use URL instead of local file. Archived files get unachieved automatically once transferred inside container to a specific folder
- RUN → Run command used to execute command inside Docker image
- ENV → Used to create environment variables inside Docker image
- COMMAND → Command defines the executable to be execute while running Docker image
- WORKDIR → Used to configure working DIR during executing Docker build command
- EXPOSE → Exposes port on which application will be hosted