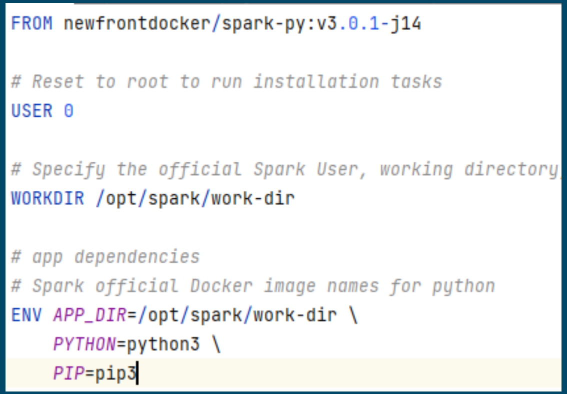
Docker introduction

Fundamentals for big data analysis

1-page Docker summary

- Run an app in a custom OS as a process

Dockerfile (text file)

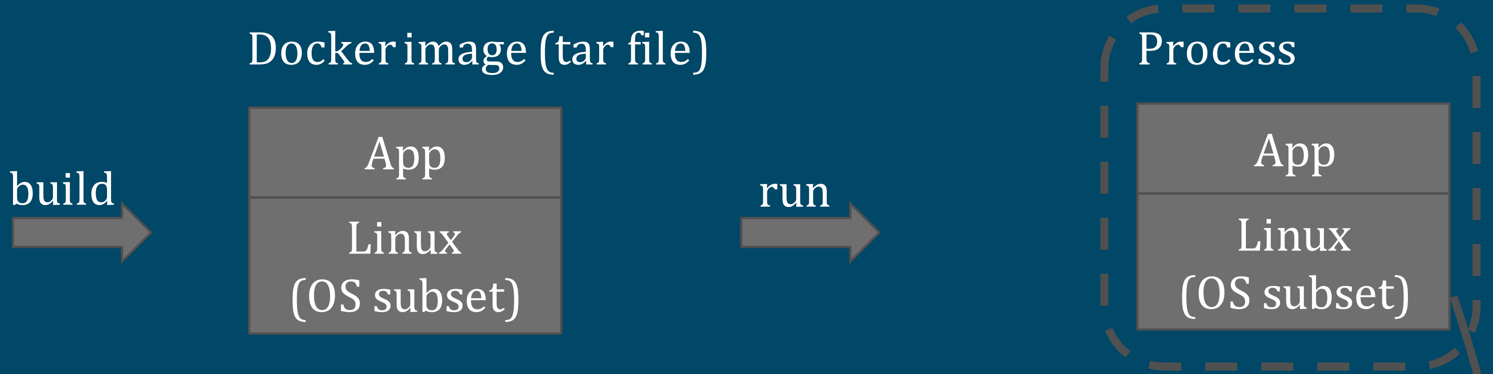


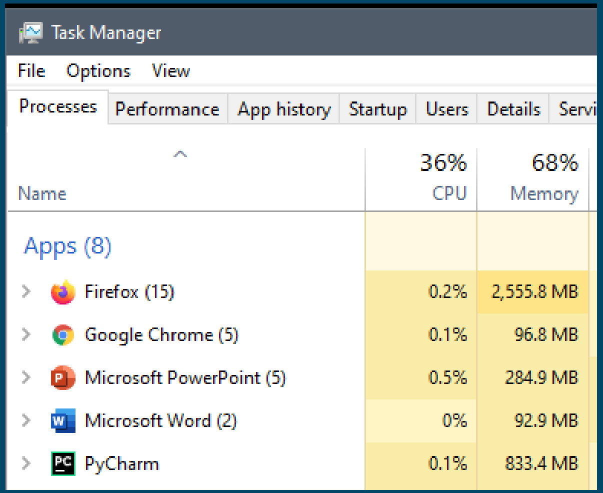
```
FROM newfrontdocker/spark-py:v3.0.1-j14

# Reset to root to run installation tasks
USER 0

# Specify the official Spark User, working directory
WORKDIR /opt/spark/work-dir

# app dependencies
# Spark official Docker image names for python
ENV APP_DIR=/opt/spark/work-dir \
    PYTHON=python3 \
    PIP=pip3
```



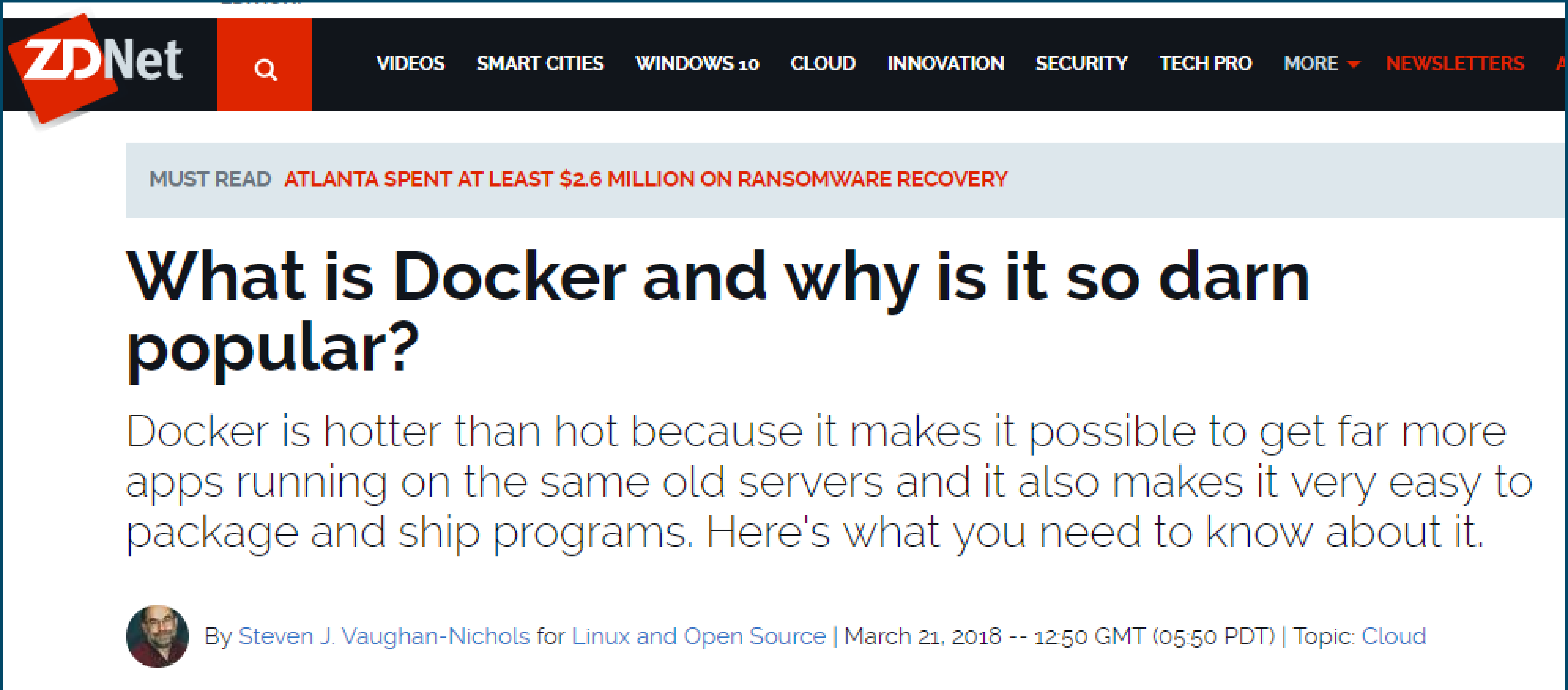


Name	36% CPU	68% Memory
Apps (8)		
Firefox (15)	0.2%	2,555.8 MB
Google Chrome (5)	0.1%	96.8 MB
Microsoft PowerPoint (5)	0.5%	284.9 MB
Microsoft Word (2)	0%	92.9 MB
PyCharm	0.1%	833.4 MB

Docker Concepts

Summary of Topics

- Why Docker containers
  - Simplifies hardware and operating system management
- Virtualization
  - Clear separation of hardware from software
    - E.g., Allows running Windows on a Mac computer
- Container defined
  - Run an app and its operating system in a process on any computer
- Docker terminology
  - Image, container, dockerfile



MUST READ ATLANTA SPENT AT LEAST $2.6 MILLION ON RANSOMWARE RECOVERY

What is Docker and why is it so darn popular?

Docker is hotter than hot because it makes it possible to get far more apps running on the same old servers and it also makes it very easy to package and ship programs. Here's what you need to know about it.

By Steven J. Vaughan-Nichols for Linux and Open Source | March 21, 2018 -- 12:50 GMT (05:50 PDT) | Topic: Cloud

Why Containers?

Why containers?

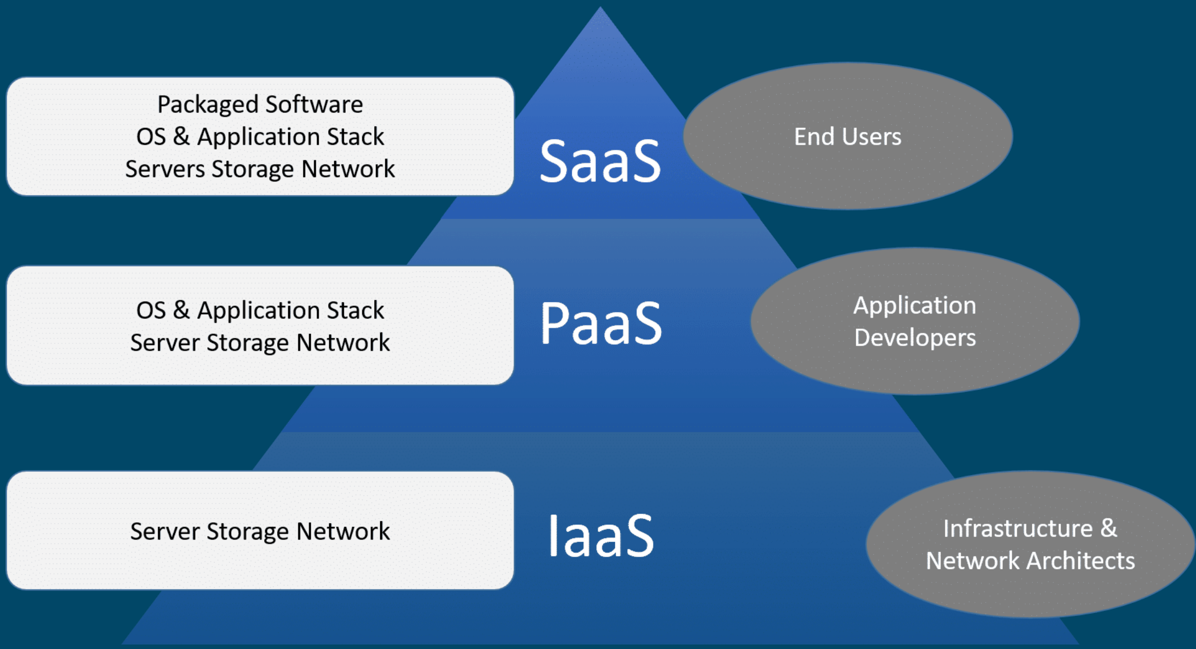
- Software will work wherever Docker is installed
- More software runs on less hardware
- Increased security
- Automates the build and deploy of software and hardware
- Infrastructure as code (IaC)
  - cost (reduction)
  - speed (faster execution)
  - risk (remove errors and security violations)

https://en.wikipedia.org/wiki/Infrastructure_as_code

Software will work wherever Docker is installed

Cloud service layers

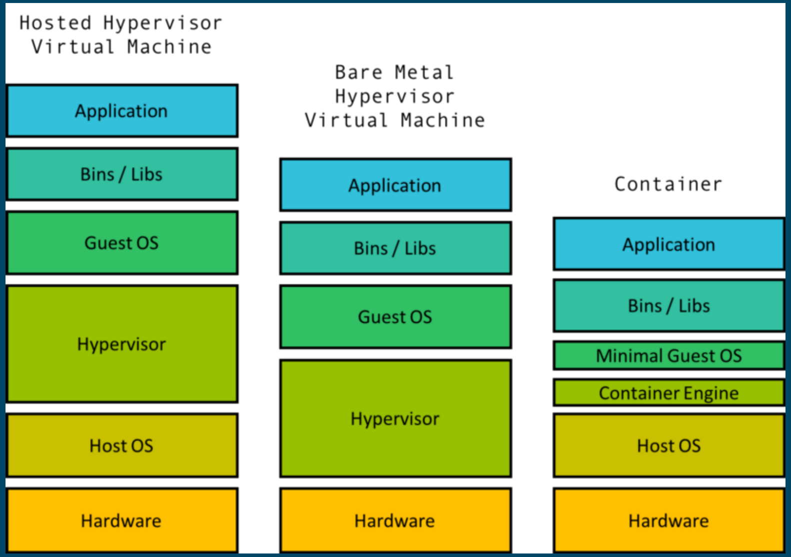
Cloud Service Models



- Infrastructure as a Service (IaaS)
  - Online services that provide high-level APIs for low-level details of networked computer infrastructure, which supports data partitioning, scaling, security, backup etc.
  - Supported by pools of hypervisors that support large numbers of virtual machines
- Platform as a Service (PaaS)
  - Online service to provision, instantiate, run, and manage a modular bundle comprising a computing platform and one or more applications
- Providers
  - AWS, Digital Ocean, Microsoft Azure, Rackspace Open Cloud, Google Compute Engine

More software runs on less hardware

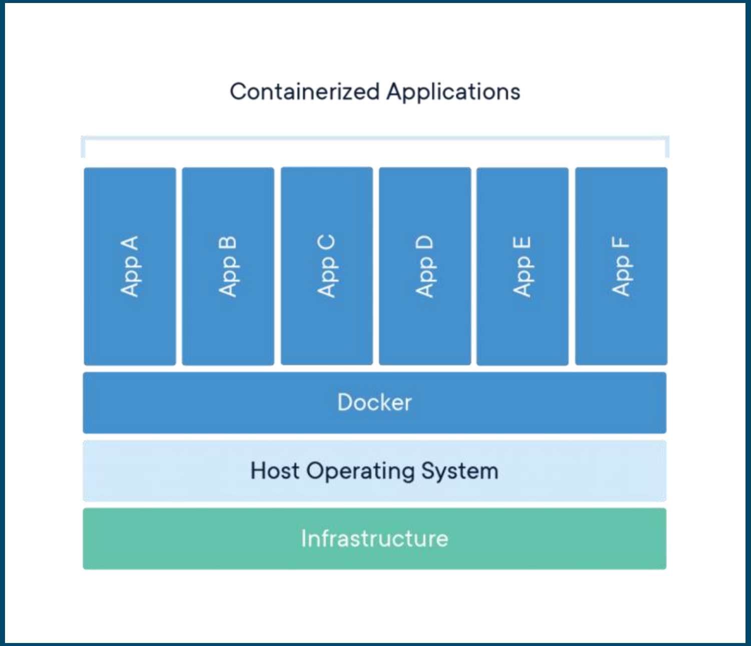
- Uses fewer resources than other virtual environments



- Used in managed clusters, where idle services can have minimal resources until they are scaled up

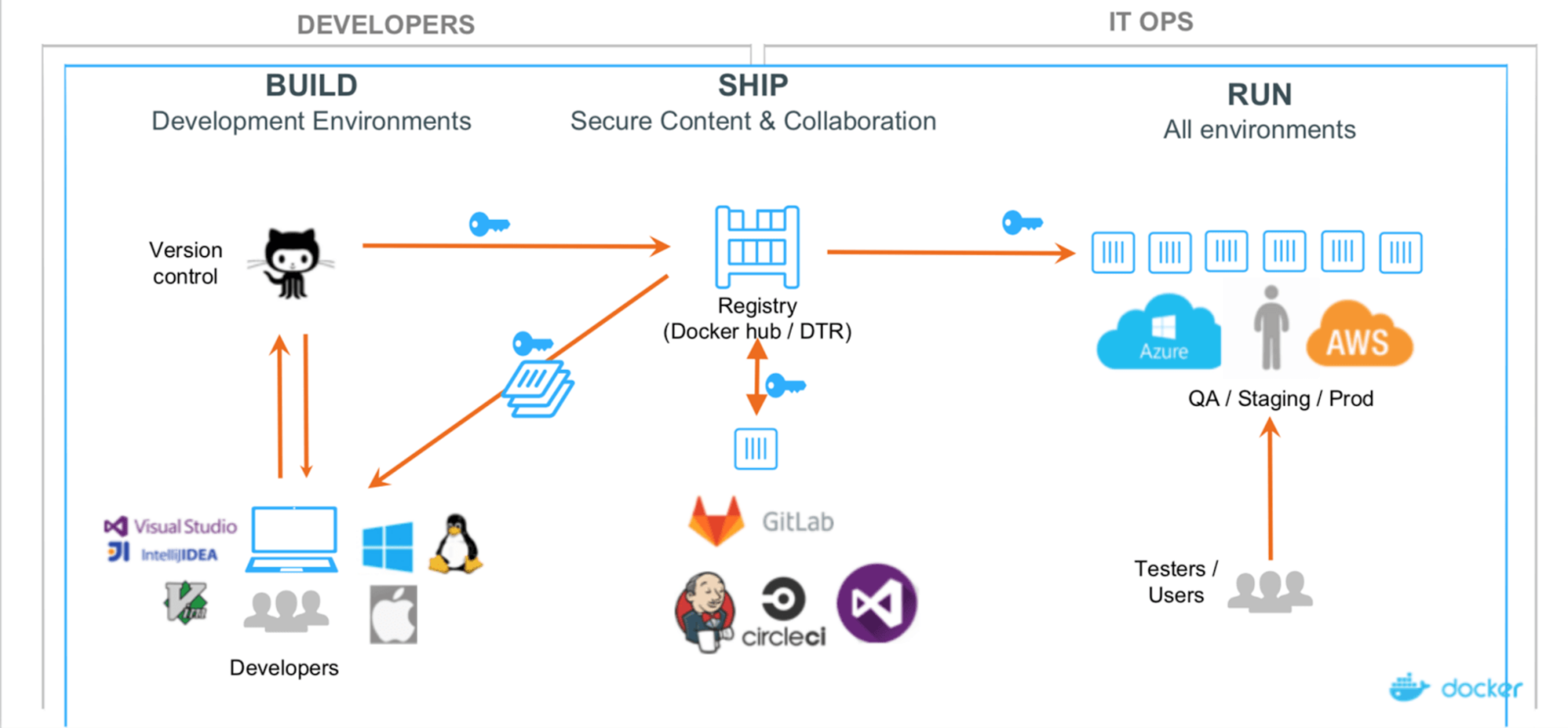
Container isolation

- Each container is isolated from each other using Linux primitives
  - Kernel namespaces
    - Each container in its own network stack
  - Control groups
    - Each container has a limited share of memory, CPU, disk I/O
- Containers run services, which communicate
  - Easier to develop with services
  - Can mix and match kinds of apps, frameworks, environments
    - Windows and Linux containers

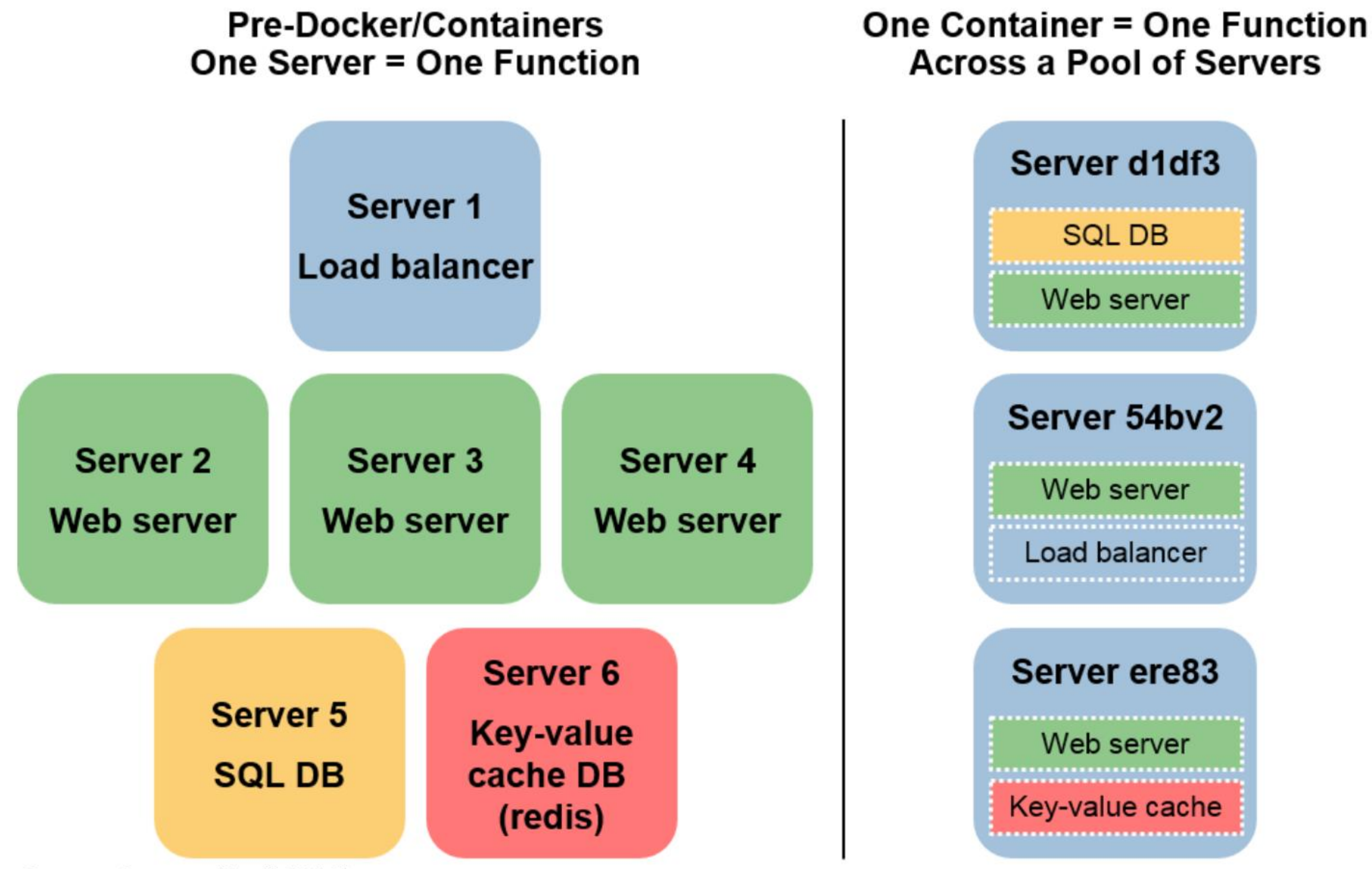


Automates the build and deploy of software and hardware

- When a developer checks in code from their laptop, then steps to deployment on AWS are automated



Containers for micro-services

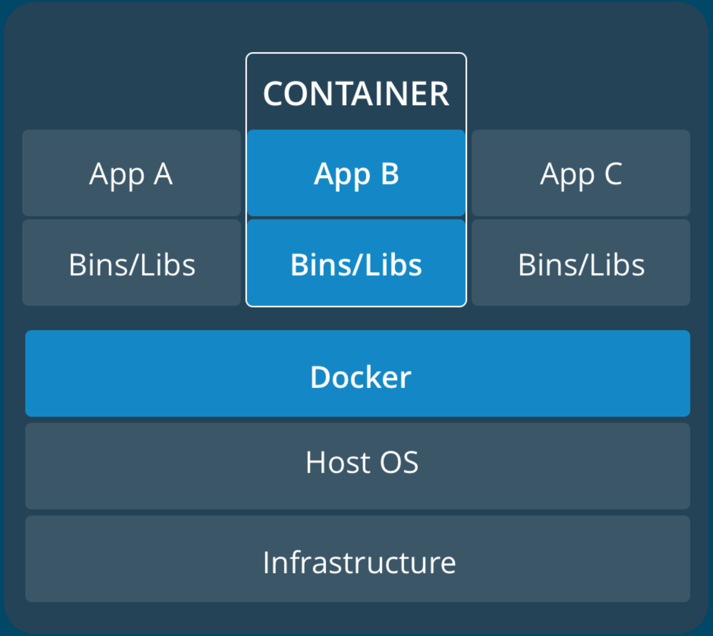


Source: Gartner (April 2016)

Container

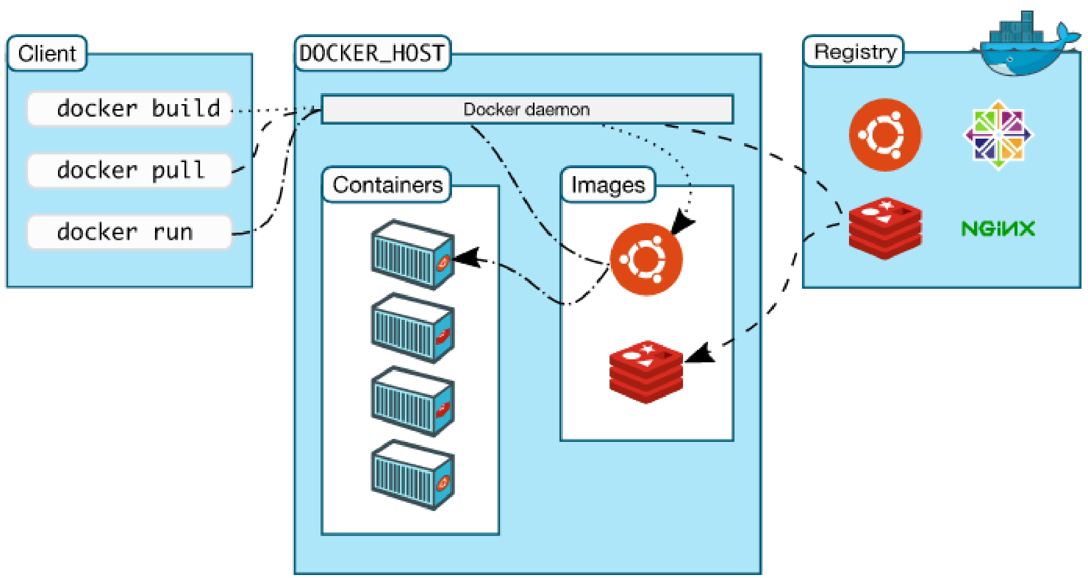
Container

- 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



Docker Architecture

- Docker client – Command Line Interface (CLI) for interfacing with the Docker
- Dockerfile – Text file of Docker instructions used to assemble a Docker Image
- Image – Hierarchies of files built from a Dockerfile, the file used as input to the docker build command
- Container – Running instance of an Image using the docker run command
- Registry – Image repository



Source: Docker docs and https://docs.docker.com/glossary/

Docker terminology

Run Hello World in a container

- docker run ubuntu echo "Hello World"
- View Docker information
  - docker images [-a]
  - docker ps –a

Docker image

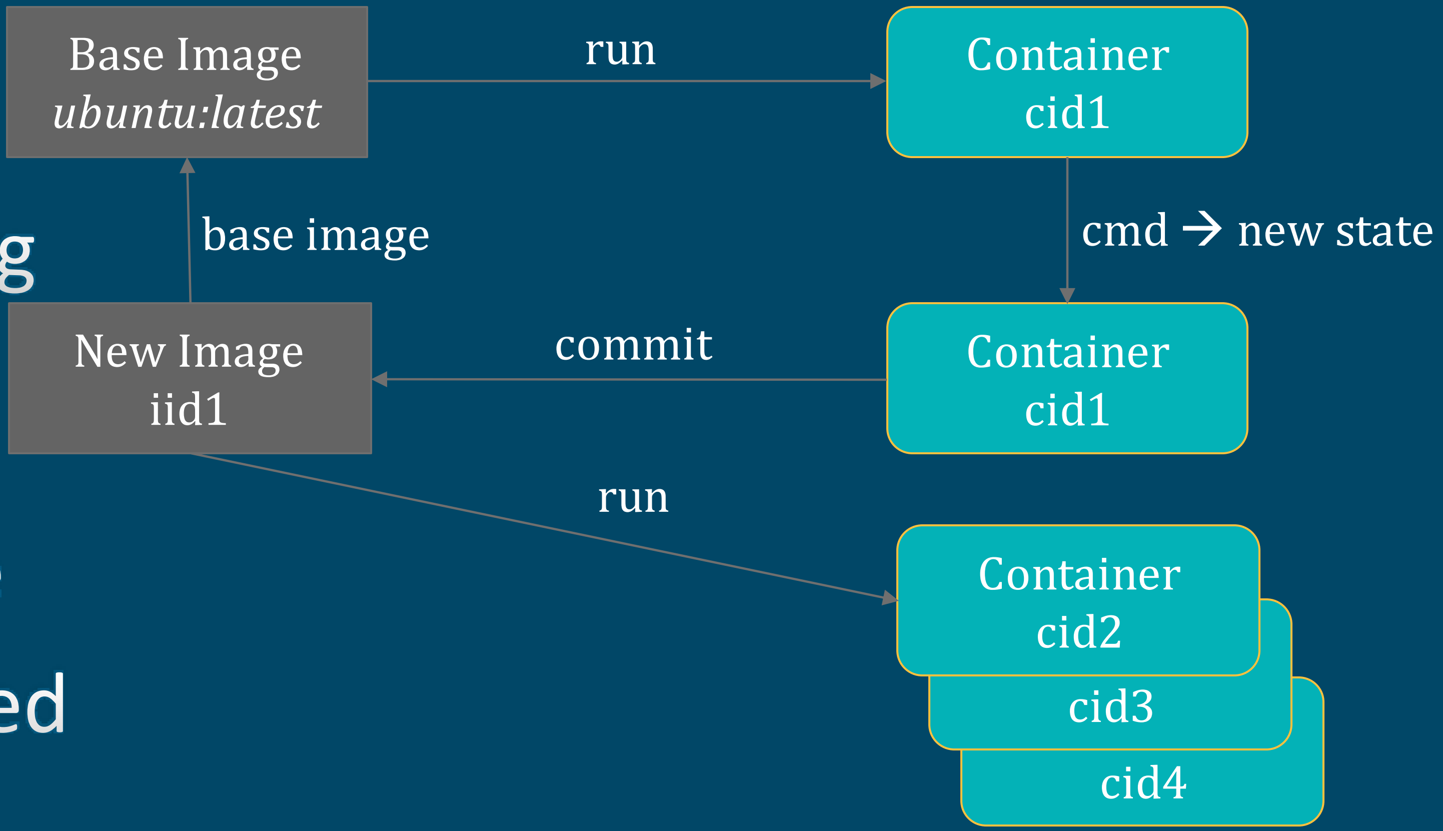
- Persisted snapshot that can be run
- docker image commands
  - images: List all local images
  - run: Create a container from an image and execute a command in it
  - tag: Tag an image
  - pull: Download image from repository
  - rmi: Delete a local image
    - This will also remove intermediate images if no longer used

Container

- Runnable instance of an image
- docker container commands
  - ps: List all running containers
  - ps –a: List all containers (incl. stopped)
  - top: Display processes of a container
  - start: Start a stopped container
  - stop: Stop a running container
  - pause: Pause all processes within a container
  - rm: Delete a container
  - commit: Create an image from a container

Image vs. Container

- Image is the specification and container is the running instance
- Container is created as the result of run image
- An image can be created by committing a container



I ♡ DOCKER

Dockerfile

- Simple text file
- Using docker build on the Dockerfile, an image is created
- Can be versioned e.g., Git or SVN
- Docker Hub can automatically build images based on dockerfiles on Github

Dockerfile Example

- Dockerfile:

  -
    ```
    FROM ubuntu
    ADD dir /files
    CMD ["bash", "someScript"]
    ```
- docker build [DockerFileDir]
- docker inspect [imageId]

Mount Volumes

- Host files can be mounted (included) in the running container
- docker run –ti –v /hostLog:/log ubuntu
- Run second container: Volume can be shared
  - docker run –ti --volumes-from firstContainerName ubuntu
- Volumes are used to shared and persisted data, which can be accessed by many Docker containers

Docker Hub

- Public repository of Docker images
  - https://hub.docker.com/
  - docker search [term]
- Automated: Has been automatically built from Dockerfile
  - Source for build is available on GitHub

Important to remember

- Why Docker containers
  - Simplifies hardware and operating system management
- Container defined
  - Run an app and its operating system in a process on any computer
- Docker terminology
  - Image, container, dockerfile