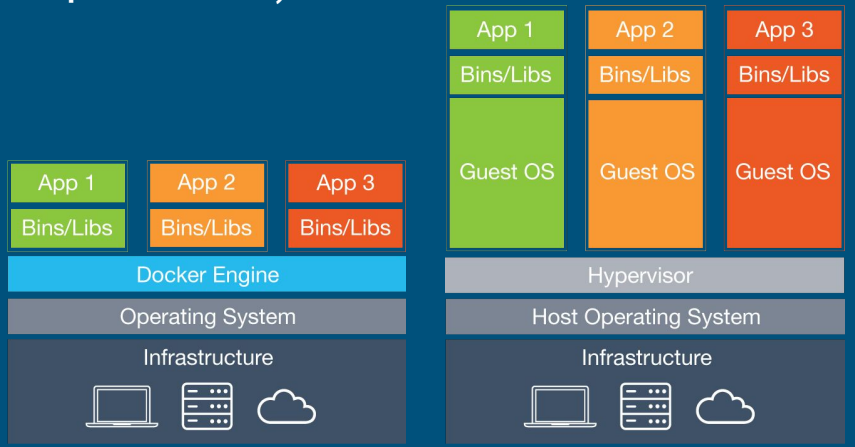
Introduction to Docker

AppDev SIG April 2016

Matthew Favetti

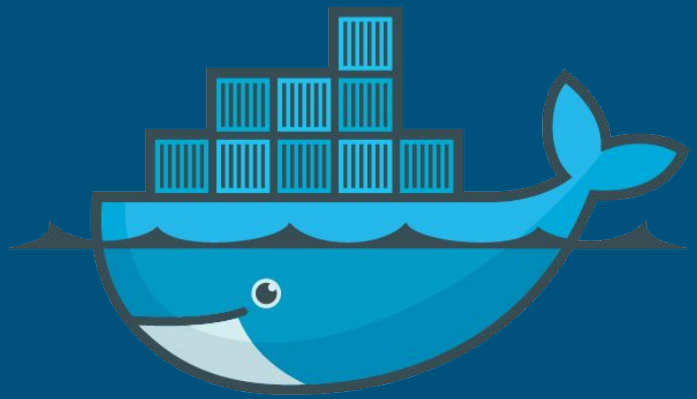
What is Docker?

- Packages an application into a container
- Containers share the host OS kernel
- Not a Virtual Machine (each VM has a separate OS)



Why Use Docker?

- Performance
- Optimization (Infrastructure)
- Encapsulation (CI/CD, DevOps)
- Portability (Cloud)



docker

Docker Hub

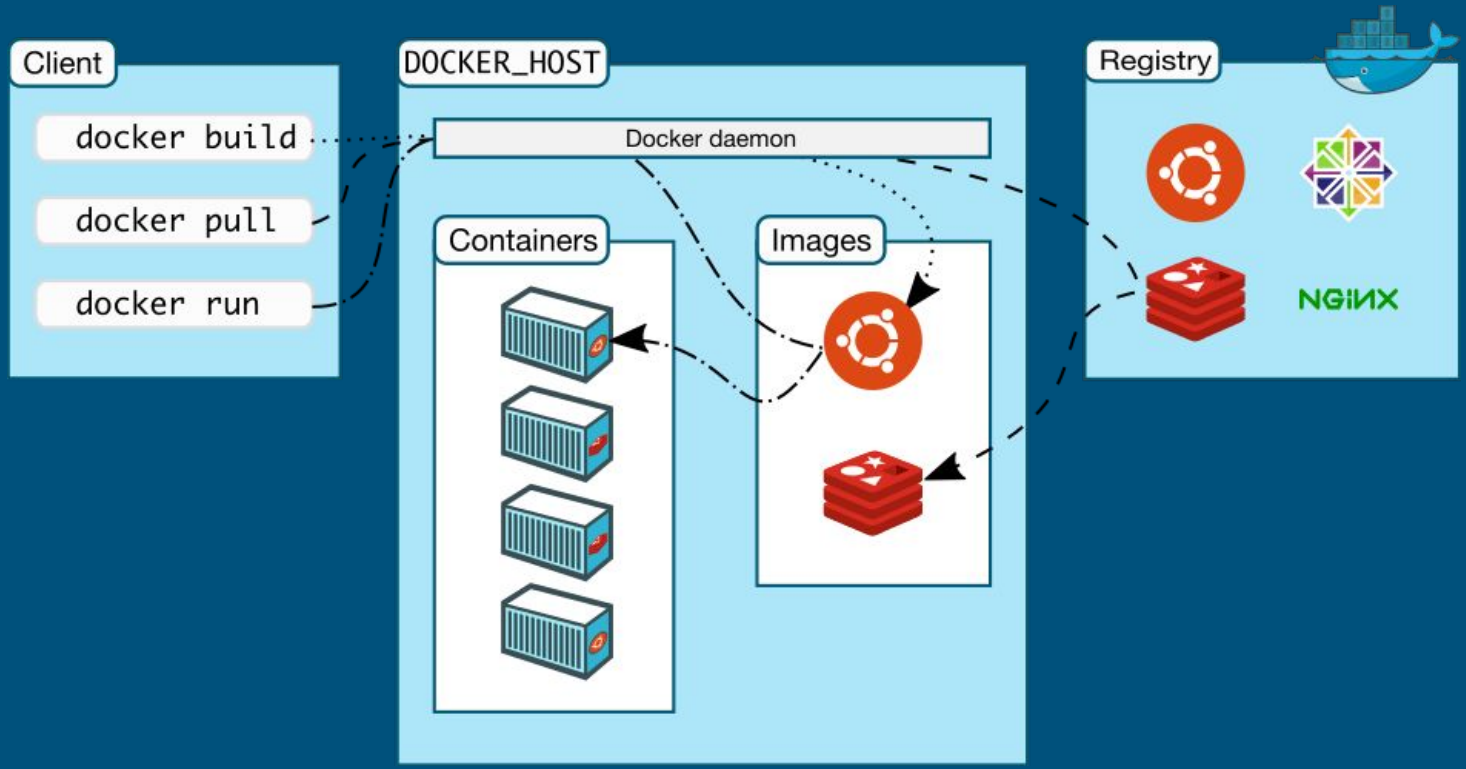
- Docker’s analog of GitHub for images instead of code
- Find images (ubuntu, redis, mysql, mongo, node, postgres, etc)
- Push / Pull your own images (public)
- >1 private image is paid

Also, Docker Trusted Registry is available if you must maintain control over your own code.

Getting Started

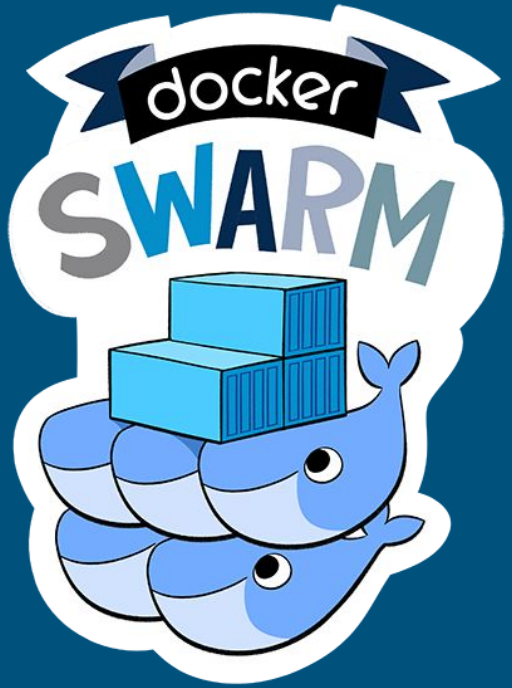
- Build/push/pull images
- Run an image to create a container
- The container runs a command and stops when it exits
- You can start/stop/restart, attach to the console, view logs, inspect configuration, etc
- Make changes in the container, commit the changes back to an image
- Tag images for easy reference

Process Overview



Beyond the Basics

- Run multiple commands utilizing supervisor (or similar)
- Networking (map host ports to the container, etc)
- Data Volumes / Data Volume Containers
  - Persist data
  - Share data among containers
- Docker swarm (native clustering)



Configuration as Code

- Dockerfile
  - All the steps necessary to build your container
  - “Document” the exact requirements / environment / dependencies of your app
- Docker-compose
  - Multiple (or single) container apps
  - Abstract command line arguments

Docker in the Cloud

- Provision manually
  - Install Docker
  - Pull the image from Docker Hub (or private registry)
- Provision using management tools
  - Docker-machine (this is also used to run docker on Windows/OS X)
  - Docker Cloud (formerly Tutum)
  - Universal Control Plane (on premises)
- Cloud service specific tools
  - AWS EC2 Container Service
  - Azure Container Service
  - Google Container Service (Kubernetes)

Google Cloud Platform

Quick Demo

Run a Simple node.js webserver in a Docker container

1. Create the app
2. Write the Dockerfile
3. Simply by using the official repositories from Docker Hub
4. Write a docker-compose file
5. Add a database