Week 7 Day 2

Containerization Review

Virtual Machines

REVATURE

Simulates a physical machine/server

- Virtualize an entire OS
- Use hypervisors

Pros

- Near total isolation
- Virtualization
- App runs reliably regardless of host

Cons

- Considered bulky
- Resource expensive

Containers

REVATURE

Bundle together apps and their dependencies

- Share underlying OS kernel
- Lighter weight
- Provided by an engine running on the host

Pros

- Lightweight
- Layers of isolation
- Virtualized view
- Isolated environment
- Run reliably regardless of host

Cons

- Layers of isolation can cause issues

Containerization

REVATURE

The process of putting our apps in a container

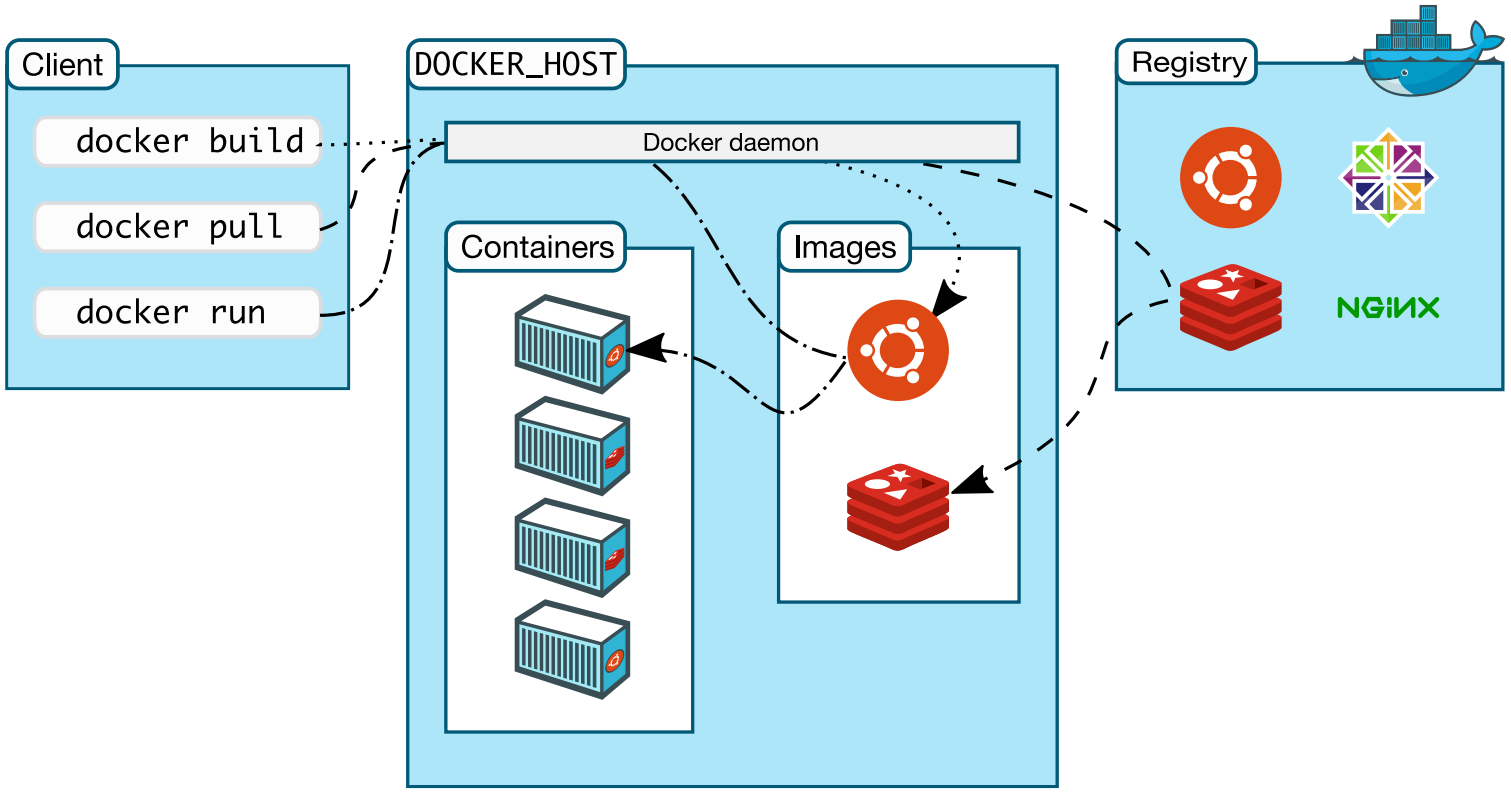
- Containers should be completely isolated from the outside world
- Linux is the foundation of most containers
- Containers are
  - Built from images
  - Run on an engine
  - Stateless
  - Virtualized and isolated

Docker

REVATURE

Open-source platform for developing, shipping, and running applications with containers

- Runs on a Client-Server Architecture



Docker: Architecture

REVATURE

CLI

- Docker Command Line Interface
- Interacts with the Daemon
- Uses Rest API

Daemon

- Long running process of Docker
  - Manages objects
  - Containers
  - Images

DockerHub

- Centralized place to store images

Docker: Objects

REVATURE

Building blocks that are managed by the daemon

- Images
  - Blueprint to construct a container
  - Created with a Dockerfile
- Containers
  - Runnable isolated instances of a set of processes and their dependencies
  - Built from an image
  - Managed by the daemon

Docker: Dockerfile

REVATURE

Defines everything needed for an image

- Step by step instructions to create an image

Important Dockerfile keywords

- FROM image name
- RUN <command>
- RUN [“executable”, “param”]
- ADD <src> <dest>
- COPY <scr> <dest>
- EXPOSE
- VOLUME [/dirname]
- WORKDIR
- CMD

Docker: Build an Image

REVATURE

1. `docker build anyflags PATH`
   - Creates an image with a dockerfile
2. `docker commit flags CONTAINER imagename`
   - Committing changes from the container specified to the image specified
   - Creating an image based off of an existing container

Docker: Build a Container

REVATURE

1. `docker create imagename`
   - Creates a container in created state
   - Configures it to be ready to run
   - Needs to run manually
2. `docker run flags imagename`
   - Pull an image from registry if it doesn’t exist locally
   - Creates and runs the container automatically

Docker: Container Management

REVATURE

- docker container ls
- docker ps –a
- docker container kill id
- docker container pause id
- docker container start id
- docker container rm flags id
- docker volume rm volname

Docker: Best Practices

REVATURE

- Ephemeral
- Lightweight
- Build context
- Multi-stage builds
- Single Responsibility
- Readible docker files
- Volumes to persist
- Secrets for sensitive data

REVATURE

Containerizing an API Demo