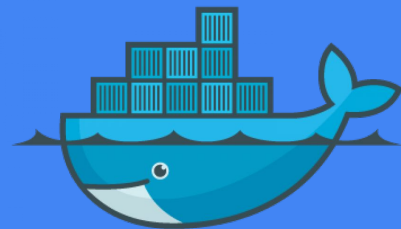# Beginning Docker

Enzo DeStephano

Preliminary Notes

Docker Overview

Defining Terminology

Docker Interaction

Building Images with Dockerfiles

Multiple Containers with Compose

Presentation Subtopics

Preliminary Notes

Containerization and Virtualization

Not a full OS	Self-contained OS
Runs off a Separate Container Engine	Uses a Hypervisor
Smaller Overhead	Larger Overhead
Create a Segmented Environment	Create a Segmented Environment

Docker Overview

What is Docker?

Docker is a vastly popular application containerization platform

Pro-open source toolkit

Works on multiple operating systems

Written in Golang

Why use Docker?

Docker is a way to package software so that it can run on any hardware

Prevent applications from reaching primary OS

Extremely lightweight

Commonly used with CI/CD

Defining Terminology

Instructions

↓

Image

↓

Container

Docker Terminology

Instructions Commands and arguments used to build an image

Image Base state of an instance when a container is spun up

Container Containerized running application

Volume File mount point for multi-environment interaction

Engine/Daemon Service that manages building and running docker containers

CLI Tool used to interact with the Docker daemon

Hub Registry of all available Docker images

Docker Interaction

CLI Usage

Generally, CLI usage can be boiled down to:

```
docker <subcommand> [options]
```

There exists a number of basic subcommands that are used consistently

Common Arguments

```
-it          = Interactive

--rm         = Remove image when finished

-d           = Daemonize the container

-v           = Mount a volume between the container and host system

-t           = Tag an image with a name

-p <H:C>     = Specify Host:Container networking ports
```

Pulling an Image

SYNTAX

docker pull <image:tag>

```
❯ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
59bf1c3509f3: Pull complete
Digest: sha256:21a3deaa0d32a8057914f36584b5288d2e5ecc984380bc0118285c70fa8c9300
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
```

Starting a Docker Container

SYNTAX

```
docker run [options] <image:tag>
```

```
❯ docker run -it alpine
/ # whoami
root
/ # ls -al
total 64
drwxr-xr-x    1 root     root          4096 Feb 14 20:52 .
drwxr-xr-x    1 root     root          4096 Feb 14 20:52 ..
-rwxr-xr-x    1 root     root             0 Feb 14 20:52 .dockerenv
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 bin
drwxr-xr-x    5 root     root           360 Feb 14 20:52 dev
drwxr-xr-x    1 root     root          4096 Feb 14 20:52 etc
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 home
drwxr-xr-x    7 root     root          4096 Nov 24 09:20 lib
drwxr-xr-x    5 root     root          4096 Nov 24 09:20 media
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 mnt
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 opt
dr-xr-xr-x  221 root     root             0 Feb 14 20:52 proc
drwx------    1 root     root          4096 Feb 14 20:52 root
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 run
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 sbin
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 srv
dr-xr-xr-x   13 root     root             0 Feb 14 20:52 sys
drwxrwxrwt    2 root     root          4096 Nov 24 09:20 tmp
drwxr-xr-x    7 root     root          4096 Nov 24 09:20 usr
drwxr-xr-x   12 root     root          4096 Nov 24 09:20 var
/ #
```

Querying Images

```
docker image[s] [options]
```

```
> docker images
REPOSITORY      TAG       IMAGE ID        CREATED        SIZE
alpine          latest    c059bfaa849c    2 months ago   5.59MB
```

Querying Running Containers

SYNTAX

docker ps [options]

OR

docker container ls [options]

```
❯ docker ps
CONTAINER ID    IMAGE     COMMAND               CREATED          STATUS
7ed5e722242b    httpd     "httpd-foreground"    4 seconds ago    Up 3 seconds
```

Stopping Running Containers

SYNTAX

```
docker kill <container_id>
```

```
❭ docker ps
CONTAINER ID    IMAGE      COMMAND               CREATED         STATUS
fbe14e551ea2    c2_c2      "python3 -m flask ru…"  23 hours ago    Up 6 hours
❭ docker kill fb
fb
```

Building Images with Dockerfiles

What is a Dockerfile?

Set of Docker-specific instructions

Used for creating a custom image

Builds upon an existing image, whether local sourced or from DockerHub

Dockerfile Syntax

FROM   Specify a base image to utilize

WORKDIR   Sets the default working directory for Dockerfile commands

RUN   Execute a specified command while building the image

ADD/COPY   Move files into the docker image, either from a local or remote location

EXPOSE   Listen on a specific port during container runtime

ENV   Set an environment variable

ENTRYPOINT Run an executable on container startup. Can take arguments

CMD   Run an executable on container startup. Ignored if arguments are specified

Building an Image

SYNTAX

```
docker build [options] <context>
```



```
FROM alpine:latest

RUN echo hello > file.txt

ENTRYPOINT cat file.txt
```



```
❯ docker build -t alpine-dfile .
[+] Building 0.4s (6/6) FINISHED
 ⇒ [internal] load build definition from Dockerfile
 ⇒ ⇒ transferring dockerfile: 113B
 ⇒ [internal] load .dockerignore
 ⇒ ⇒ transferring context: 2B
 ⇒ [internal] load metadata for docker.io/library/alpine:latest
 ⇒ CACHED [1/2] FROM docker.io/library/alpine:latest
 ⇒ [2/2] RUN echo hello > file.txt
 ⇒ exporting to image
 ⇒ ⇒ exporting layers
 ⇒ ⇒ writing image sha256:adaa50db2cd61662fcdf24638cacbf0dec631fa9b8e4cd25c93bf4da84aca843
 ⇒ ⇒ naming to docker.io/library/alpine-dfile

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
❯ docker run alpine-dfile
hello
```

Dockerfile Demos

Multiple Containers with Compose

Compose

Compose is a way to compile all Docker operations into one command

Can utilize all three states - Dockerfile → Image → Containers

Often used for spinning up multiple containers at once

File necessary: `docker-compose.yml`

Command: `docker-compose up`

Compose Syntax

Utilizes similar syntax to that of a Dockerfile

```
version: "3.9"
services:
  web:
    build: .
    ports:
      - "5000:5000"
  redis:
    image: "redis:alpine"
```

[Compose Reference](#)

Compose Demos

Thank you!

Questions?