

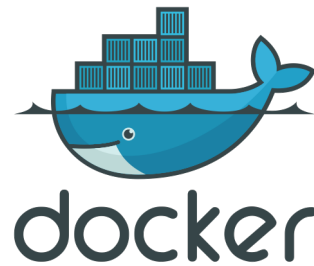
System Administration & Security



COMP 175 | Fall 2021 | University of the Pacific | Jeff Shafer

Containers (2)

Creating Custom Containers



Recap: Docker Images and Containers

DOCKER CLIENT

IMAGES &

CONTAINERS



Previously

- Lab 11 – Docker
 - Install Docker to manage containers on a *single* host Engine
 - *Run some previously-built images*

- Lab 12 – Kubernetes
 - Install Kubernetes to manage containers on *many* hosts
 - *Run some previously-built images*

- **How do I make my own containers?**

Image Creation

- Two methods:
 1. **Start with an existing Docker image**
 - Launch it
 - Manually modify it
(Add/remove files, programs, libraries)
 - Capture it as a new image
 2. **Use a Dockerfile to script image creation**
- Store final images in a repository

Container Repositories

➤ Docker Hub

- <https://hub.docker.com/>
- Free tier (*rate limited*)
- Public repository and private repository (\$\$)

➤ Docker Registry

- <https://docs.docker.com/registry/>
- Self-hosted, private
- Open Source

➤ Amazon Elastic Container Registry (ECR)

- Docker images
- Open Container Initiative (OCI) images
- Pay for data storage (per/GB)
- Pay for data transfer OUT of AWS (per/GB)

➤ Google Cloud Container Registry

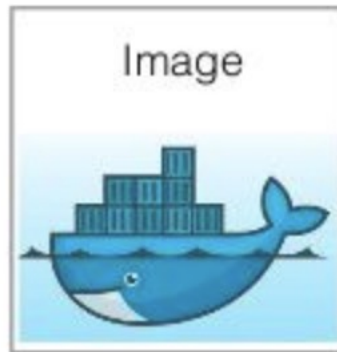
- <https://cloud.google.com/container-registry/>

Dockerfile

```
FROM ubuntu:14.04
MAINTAINER myname@mycompany.com
RUN apt-get update
RUN apt-get install -y python
RUN python --help
```

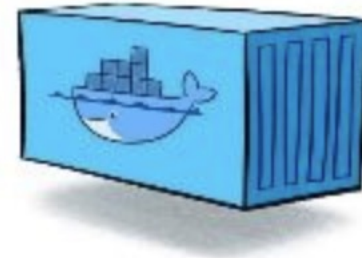
Dockerfile

build



Docker Image

run



Docker Container

Dockerfile Syntax

- **#** Comments
- **FROM** name/image:tag
 - *Specify the base image*
- **COPY** app.conf /etc/app.conf
 - *Copy a file into the image*
- **RUN** apt update && apt install -y curl
 - *Run a command in the image*
- **EXPOSE** 8080
 - *Expose a network port from the image*

Dockerfile

Create the Dockerfile with desired instructions

```
FROM ubuntu:latest
LABEL maintainer="Jeff Shafer <jshafer@pacific.edu>"
RUN apt update && apt install -y nginx
COPY website /etc/nginx/sites-available/website
EXPOSE 8080
```

Build it

```
$ docker build -f /path/to/Dockerfile -t myname/myapp:v1 .
```

Run it

```
$ docker run myname/myapp:v1
```

Push image to Docker Hub

```
$ docker push myname/myapp:v1 .
```


Dockerfile Instructions

➤ FROM

- Sets the base image for subsequent instructions

➤ RUN

- Executes a shell command while **building** the new image (i.e., “build time”)

➤ CMD

- Executes a shell command when creating a **container** out of the image (i.e., “run time”)

➤ ENV

- Set environment variables for both build time and run time

➤ EXPOSE

- Specifies ports that container listens on
- Use -p flag with docker run to publish all exposed ports

➤ LABEL

- Metadata

➤ COPY

- Copy file/directory from host into Docker image

➤ VOLUME

- Creates mount point

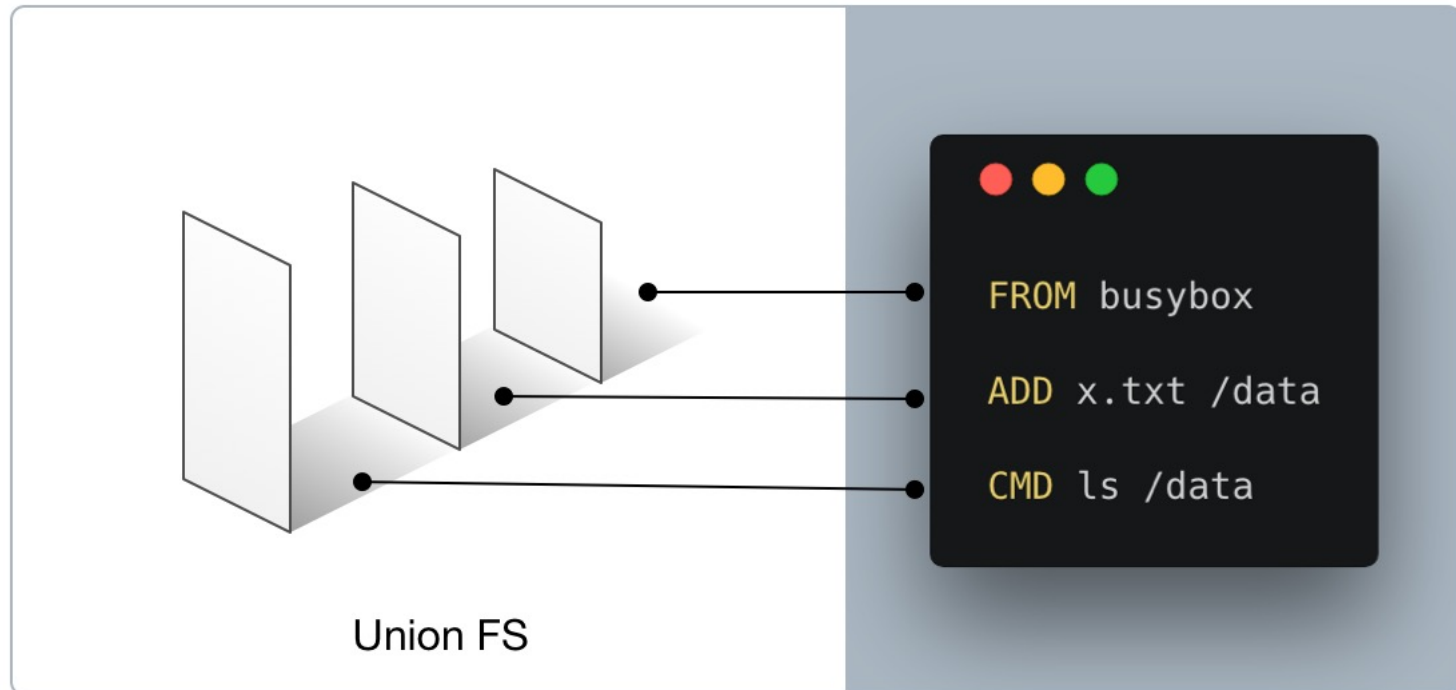
Dockerfiles



Container Storage

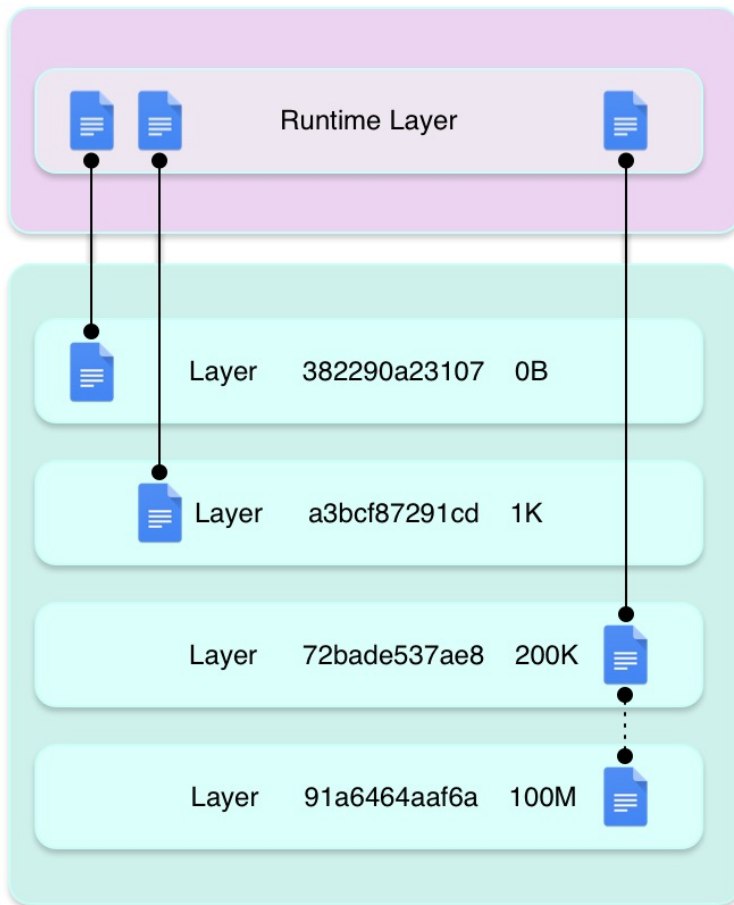
- Docker images are read-only templates that containers are created from
- Docker images consist of *layers* using a *union file system*
 - The union file system presents a single coherent (unified) file system to the application

Container Storage



- Each line in Dockerfile is a *layer* in the final image
- Changing a line **rebuilds** *all subsequent layers* in the final image

Container Storage



Read / Write
Container Layers

➔ When container
is created, a
runtime layer is
added

➔ Readable and
writable

Read-Only
Image Layers

Wrap-Up

➤ Questions?

➤ Concerns?

➤ **This Week**

➤ **Lab 13** – Custom Containers

➤ **Next Week**

➤ IT Automation

➤ **Lab 14** – Ansible