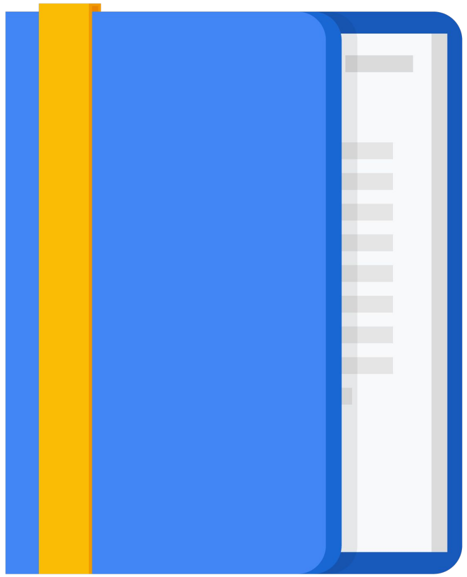
Introduction to Docker

Agenda

What is Docker

Why should I care

How does it work

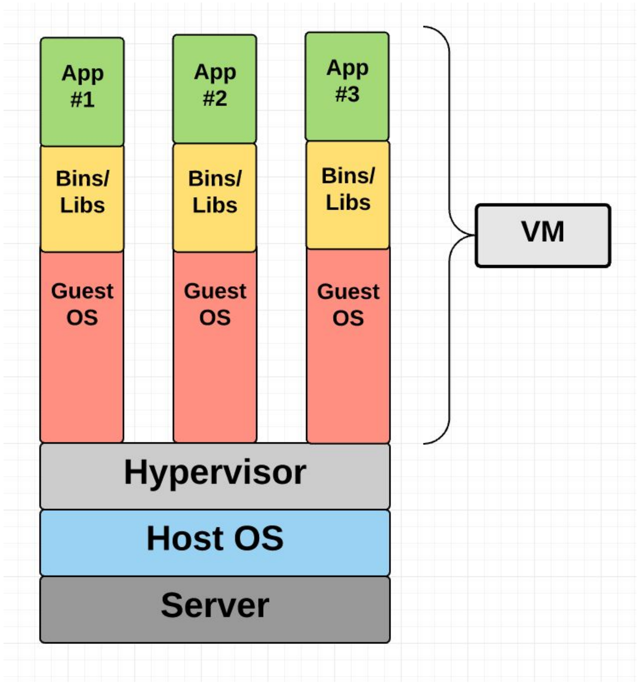


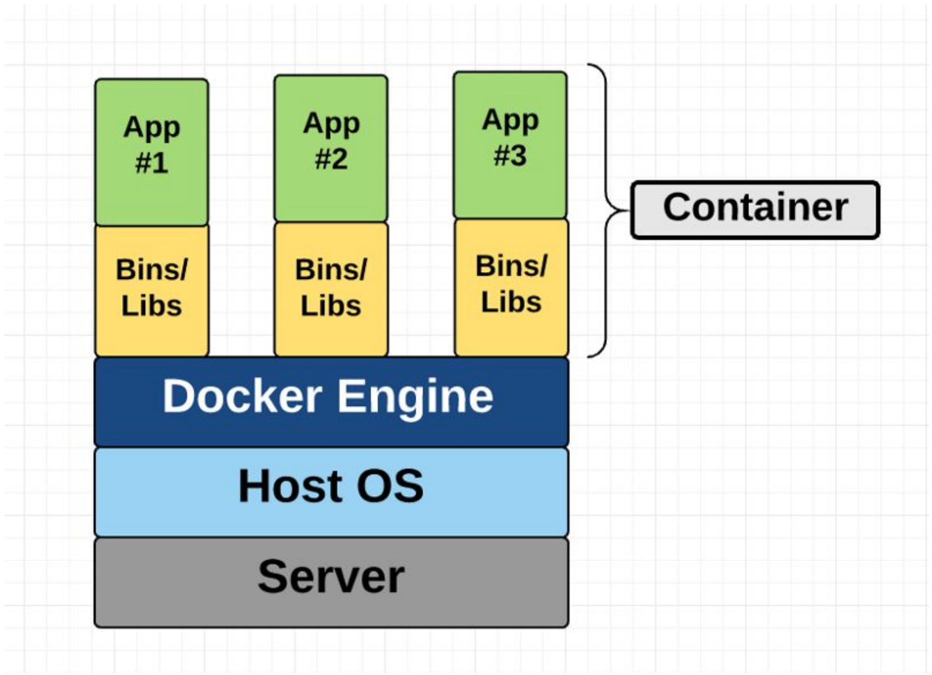
What is Docker?

Docker is a platform for developing, shipping and running applications

- separates applications from the infrastructure
- manages infrastructure the same way you manage code
- uses "containers" to isolate application dependencies

What is Docker?



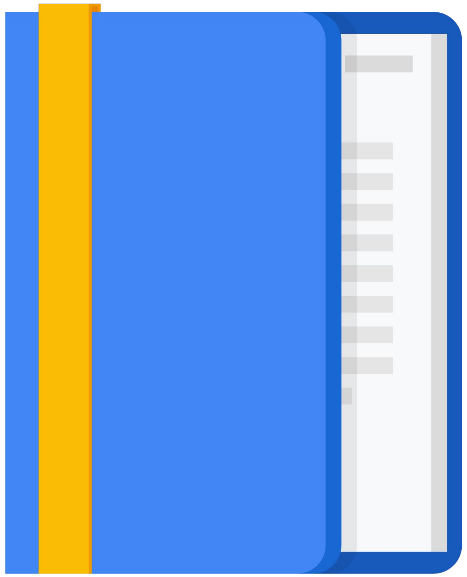


Agenda

What is Docker

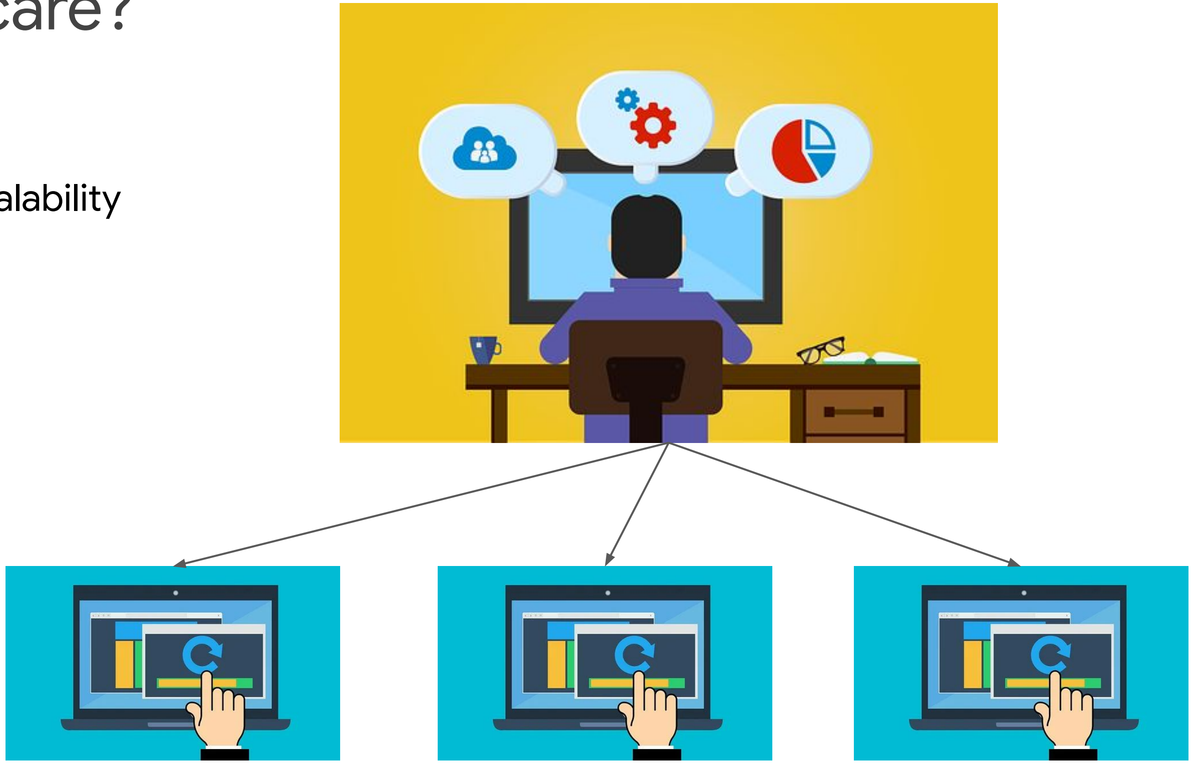
Why should I care

How does it work



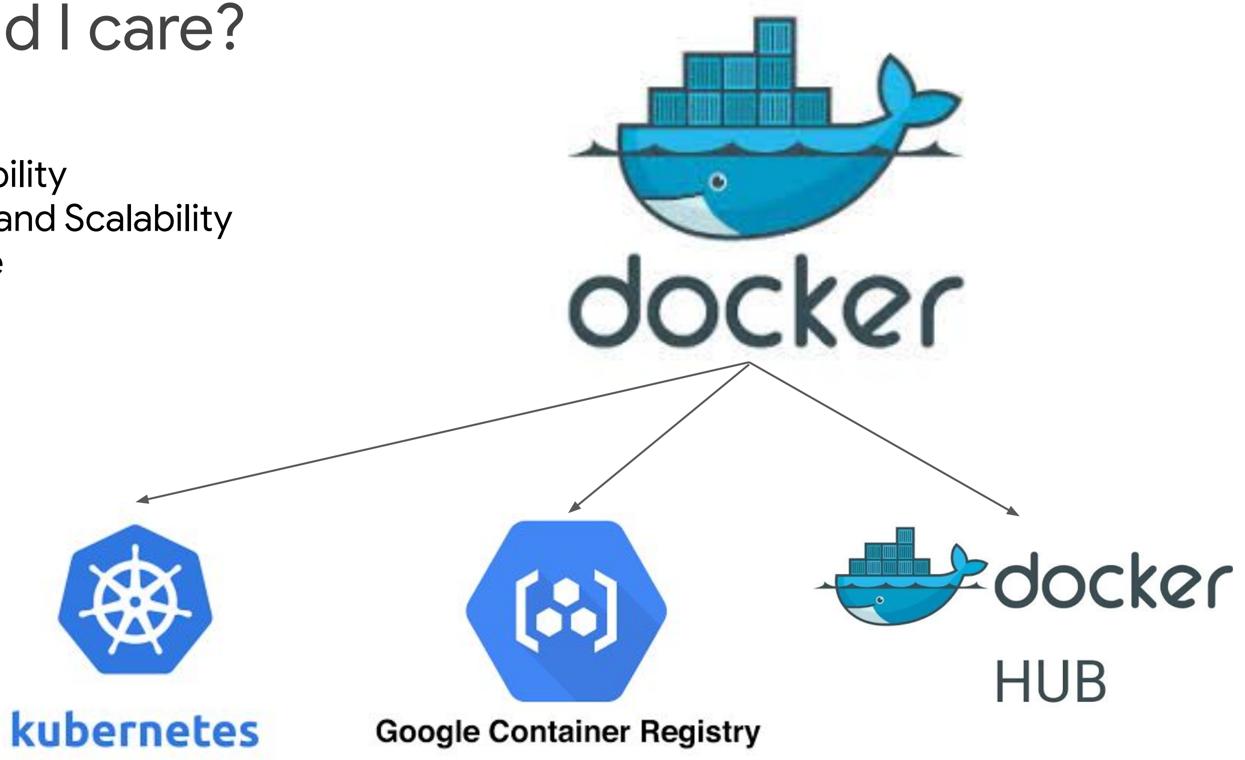
Why should I care?

- Reproducibility
- Portability and Scalability



Why should I care?

- Reproducibility
- Portability and Scalability
- Easy to use
- Speed

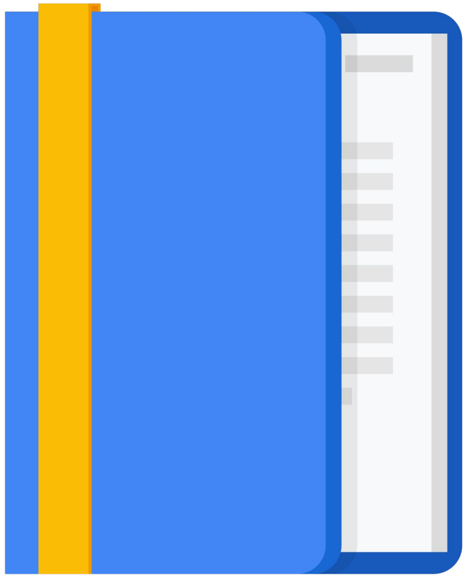


Agenda

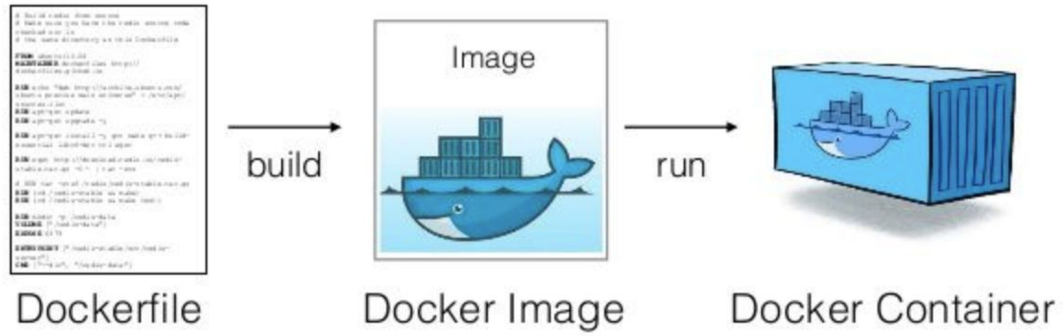
What is Docker

Why should I care

How does it work



How does it work?



A Dockerfile is a recipe for creating an Image

A Docker Image is a serialized package containing an OS, the software dependencies, and the application code to be run.

A Docker Container is a process running an image

How does it work?

Dockerfile

```
# Dockerfile
FROM python:2.7.15-jessie
WORKDIR /root

# Installs pytorch and torchvision.
RUN pip install torch==1.0.0 torchvision==0.2.1

# Installs cloudml-hypertune for hyperparameter tuning.
RUN pip install cloudml-hypertune

# Path configuration
ENV PATH $PATH:/root/tools/google-cloud-sdk/bin

# Make sure gsutil will use the default service account
RUN echo '[GoogleCompute]\nservice_account = default' >
/etc/boto.cfg

# Copies the trainer code
RUN mkdir /root/trainer
COPY trainer/mnist.py /root/trainer/mnist.py

# Sets up the entry point to invoke the trainer.
ENTRYPOINT ["python", "trainer/mnist.py"]
```

How does it work?

base image you want to build on top of

```
# Dockerfile
FROM python:2.7.15-jessie
WORKDIR /root

# Installs pytorch and torchvision.
RUN pip install torch==1.0.0 torchvision==0.2.1

# Installs cloudml-hypertune for hyperparameter tuning.
RUN pip install cloudml-hypertune

# Path configuration
ENV PATH $PATH:/root/tools/google-cloud-sdk/bin

# Make sure gsutil will use the default service account
RUN echo '[GoogleCompute]\nservice_account = default' >
/etc/boto.cfg

# Copies the trainer code
RUN mkdir /root/trainer
COPY trainer/mnist.py /root/trainer/mnist.py

# Sets up the entry point to invoke the trainer.
ENTRYPOINT ["python", "trainer/mnist.py"]
```

How does it work?

run arbitrary shell commands to install packages and dependencies

```
# Dockerfile
FROM python:2.7.15-jessie
WORKDIR /root

# Installs pytorch and torchvision.
RUN pip install torch==1.0.0 torchvision==0.2.1

# Installs cloudml-hypertune for hyperparameter tuning.
RUN pip install cloudml-hypertune

# Path configuration
ENV PATH $PATH:/root/tools/google-cloud-sdk/bin

# Make sure gsutil will use the default service account
RUN echo '[GoogleCompute]\nservice_account = default' >
/etc/boto.cfg

# Copies the trainer code
RUN mkdir /root/trainer
COPY trainer/mnist.py /root/trainer/mnist.py

# Sets up the entry point to invoke the trainer.
ENTRYPOINT ["python", "trainer/mnist.py"]
```

How does it work?

```
# Dockerfile
FROM python:2.7.15-jessie
WORKDIR /root

# Installs pytorch and torchvision.
RUN pip install torch==1.0.0 torchvision==0.2.1

# Installs cloudml-hypertune for hyperparameter tuning.
RUN pip install cloudml-hypertune

# Path configuration
ENV PATH $PATH:/root/tools/google-cloud-sdk/bin

# Make sure gsutil will use the default service account
RUN echo '[GoogleCompute]\nservice_account = default' >
/etc/boto.cfg

# Copies the trainer code
RUN mkdir /root/trainer
COPY trainer/mnist.py /root/trainer/mnist.py

# Sets up the entry point to invoke the trainer.
ENTRYPOINT ["python", "trainer/mnist.py"]
```

change environments

How does it work?

```
# Dockerfile
FROM python:2.7.15-jessie
WORKDIR /root

# Installs pytorch and torchvision.
RUN pip install torch==1.0.0 torchvision==0.2.1

# Installs cloudml-hypertune for hyperparameter tuning.
RUN pip install cloudml-hypertune

# Path configuration
ENV PATH $PATH:/root/tools/google-cloud-sdk/bin

# Make sure gsutil will use the default service account
RUN echo '[GoogleCompute]\nservice_account = default' >
/etc/boto.cfg

# Copies the trainer code
RUN mkdir /root/trainer
COPY trainer/mnist.py /root/trainer/mnist.py

# Sets up the entry point to invoke the trainer.
ENTRYPOINT ["python", "trainer/mnist.py"]
```

copy code from training application to container

How does it work?

```
# Dockerfile
FROM python:2.7.15-jessie
WORKDIR /root

# Installs pytorch and torchvision.
RUN pip install torch==1.0.0 torchvision==0.2.1

# Installs cloudml-hypertune for hyperparameter tuning.
RUN pip install cloudml-hypertune

# Path configuration
ENV PATH $PATH:/root/tools/google-cloud-sdk/bin

# Make sure gsutil will use the default service account
RUN echo '[GoogleCompute]\nservice_account = default' >
/etc/boto.cfg

# Copies the trainer code
RUN mkdir /root/trainer
COPY trainer/mnist.py /root/trainer/mnist.py

# Sets up the entry point to invoke the trainer.
ENTRYPOINT ["python", "trainer/mnist.py"]
```

configures entrypoint

Building, Pushing, and Running a Docker image

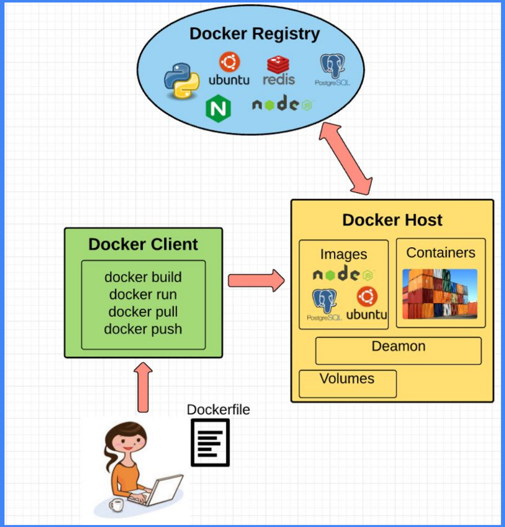
```
IMAGE_NAME=gcr.io/${PROJECT_ID}/node-app:0.2

docker build -t $IMAGE_NAME .

docker push $IMAGE_NAME

docker run -p 4000:80  $IMAGE_NAME
```

Supposes the Dockerfile is in the CWD (“.”)

Lab

In this lab, you will learn how to:

- How to build, run, and debug Docker containers.
- How to pull Docker images from Docker Hub and Google Container Registry.
- How to push Docker images to Google Container



docker_and_kubernetes/labs/1_intro_docker.ipynb cloud.google.com