Mastering Docker: Build, Deploy, and Run Your First Containerized App

Setting Up Your Development Environment

Mehul Patel
Independent OSS & Cloud Consultant

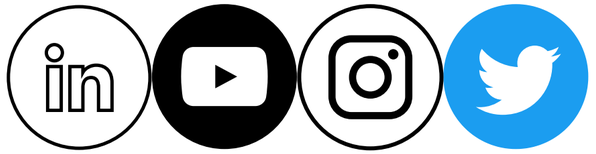
@NomadicMehul

$Whoami

~ Mehul Patel

OSS and **Cloud** Consultant | **Mozilla** Reps Council | **AWS** Community Builder | **Auth0** by **Okta** Ambassador | **AWS** and **GCP** Certified Solution Architect

@NomadicMehul



Aim of the Game

Setting Up Your Development Environment

Create an Application Image

Run an Application Container

Communicate Between Multiple Containers

Orchestrate Multiple Containers with Docker Compose

Code Samples

https://github.com/nomadicmehul/Docker-Ahmedabad-July-23


Gain the freedom to run
your applications anywhere!

The Case for Docker

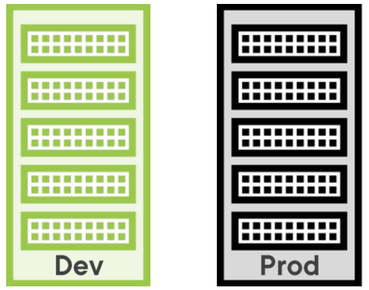
Shipping Applications the Traditional Way

Shipping Applications with Docker

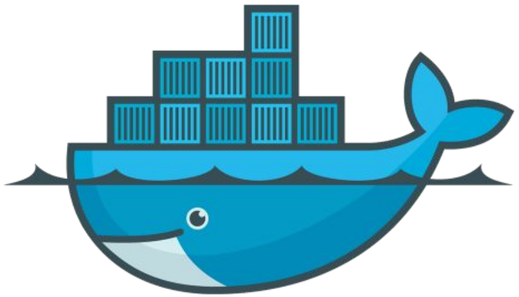
The Case for Docker

Accelerate Developer Onboarding

Eliminate App Conflicts

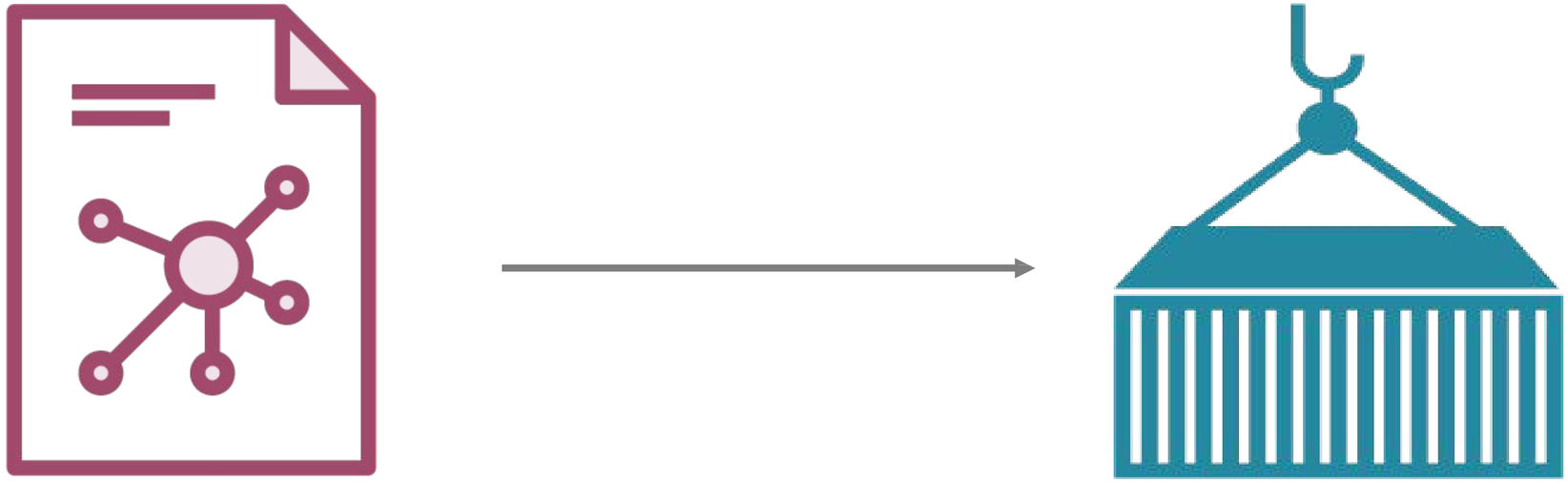


Environment Consistency



Ship Software Faster

Images and Containers

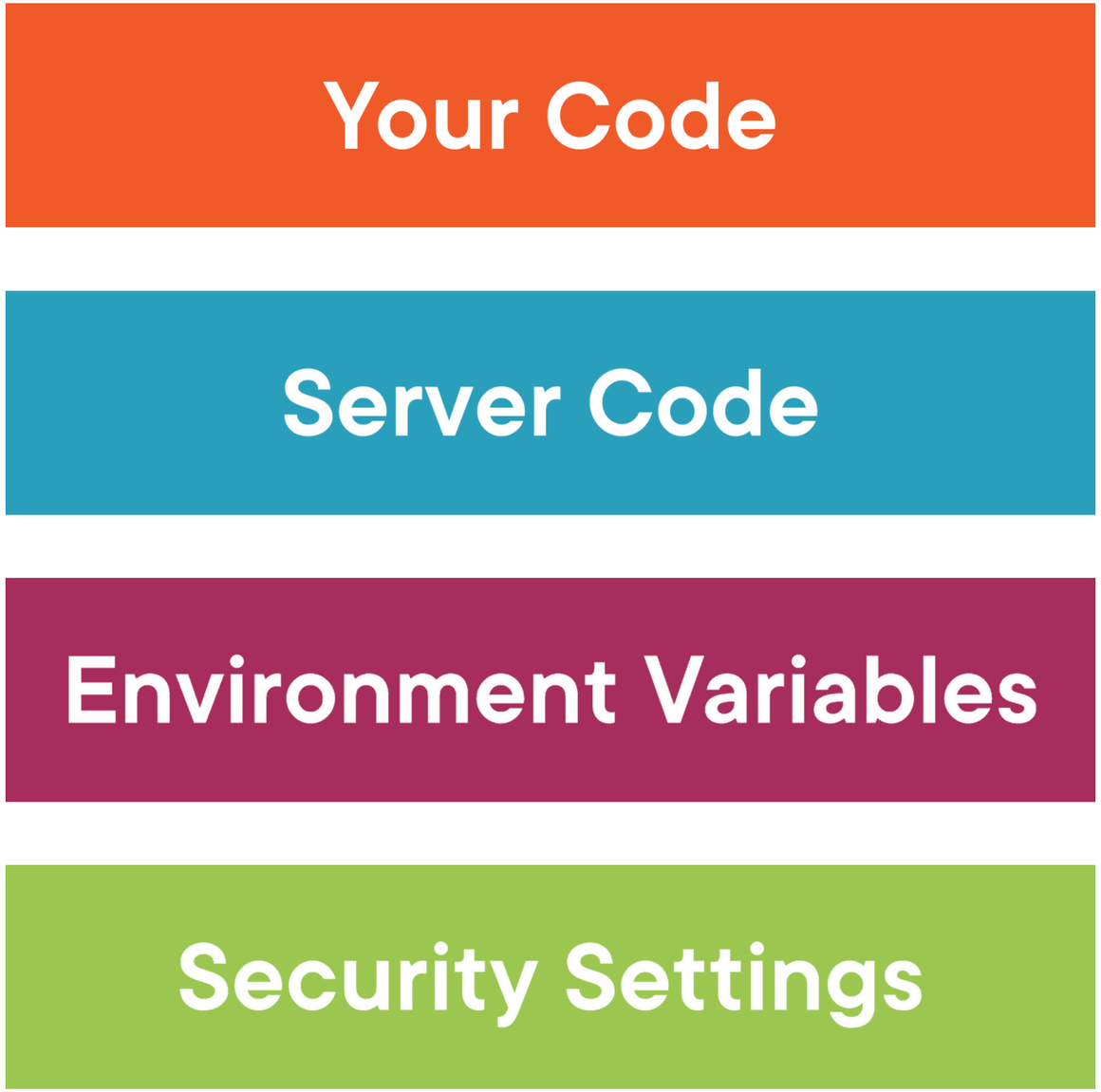


Docker Image

Define the contents that are needed to run a container

Docker Container

Runs your application


Your Code
Server Code
Environment Variables
Security Settings

Image

A read-only template composed of layered filesystems used to share common files and create Docker container instances.

Container

An isolated and secured shipping container created from an image that can be run, started, stopped, moved and deleted.

Software Installation

Docker Desktop

Docker Desktop (Windows or Mac)

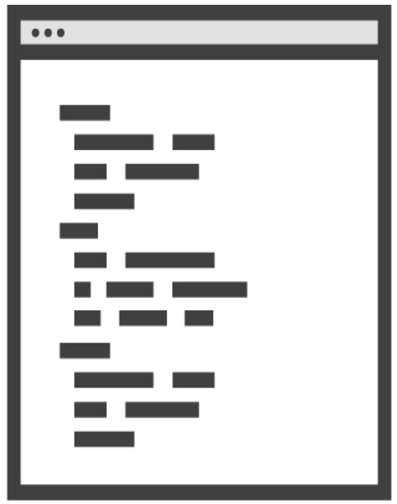
Provides image and container tools

Linux can run Docker Engine

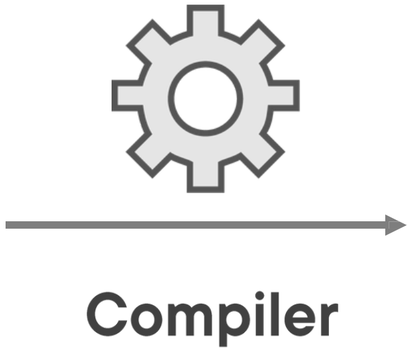
Windows and Mac use a hypervisor

https://www.docker.com/get-started

Understand Dockerfiles



Code



Compiler

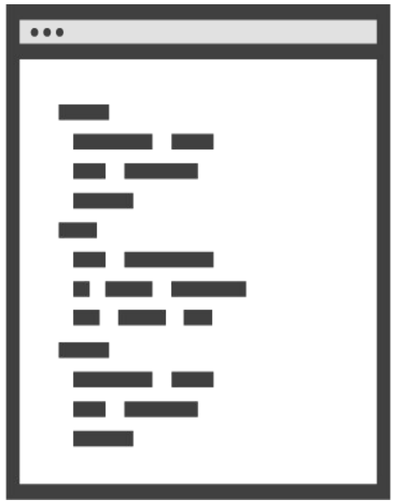
Binary

Dockerfile

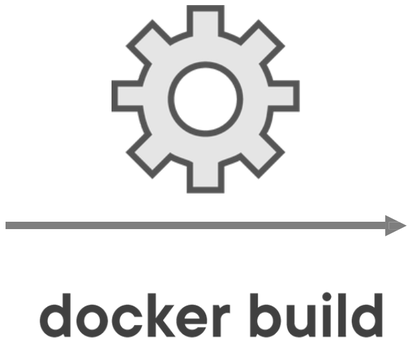
is a text document that contains all the commands a user could call on the command line to assemble an image.

https://docs.docker.com/

Dockerfile and Images

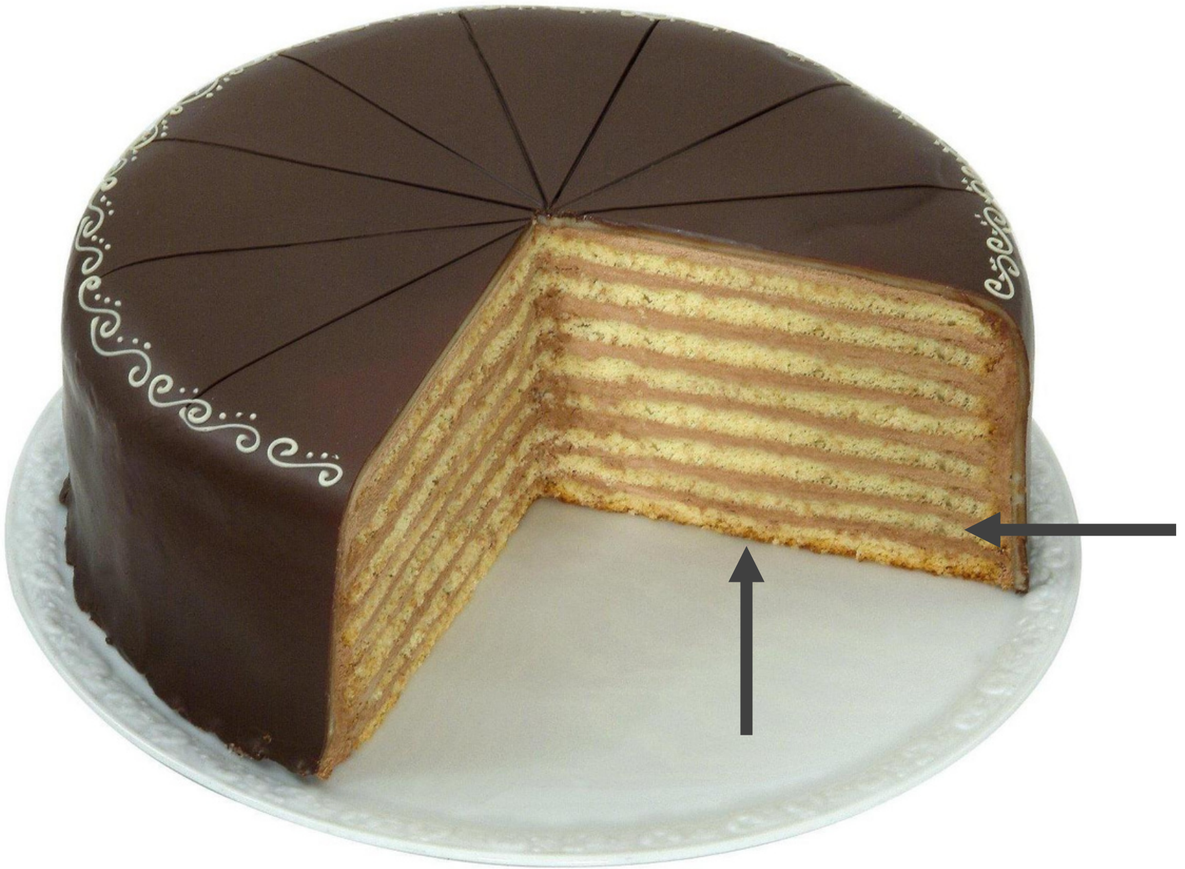


Dockerfile



docker build

Docker Image



Dockerfile Instructions

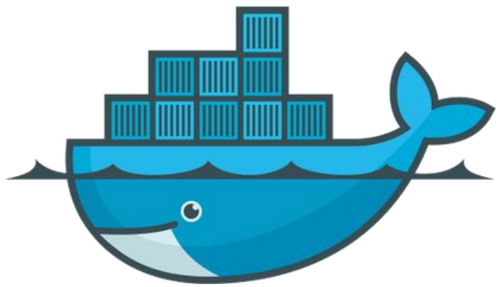
```
FROM        node:alpine
LABEL       author="Mehul Patel"
ENV         NODE_ENV=production
WORKDIR     /var/www
COPY        . .
EXPOSE      3000
ENTRYPOINT  ["node", "server.js"]
```

https://hub.docker.com/

Create a Custom Application Dockerfile

Using docker build

Docker Registries

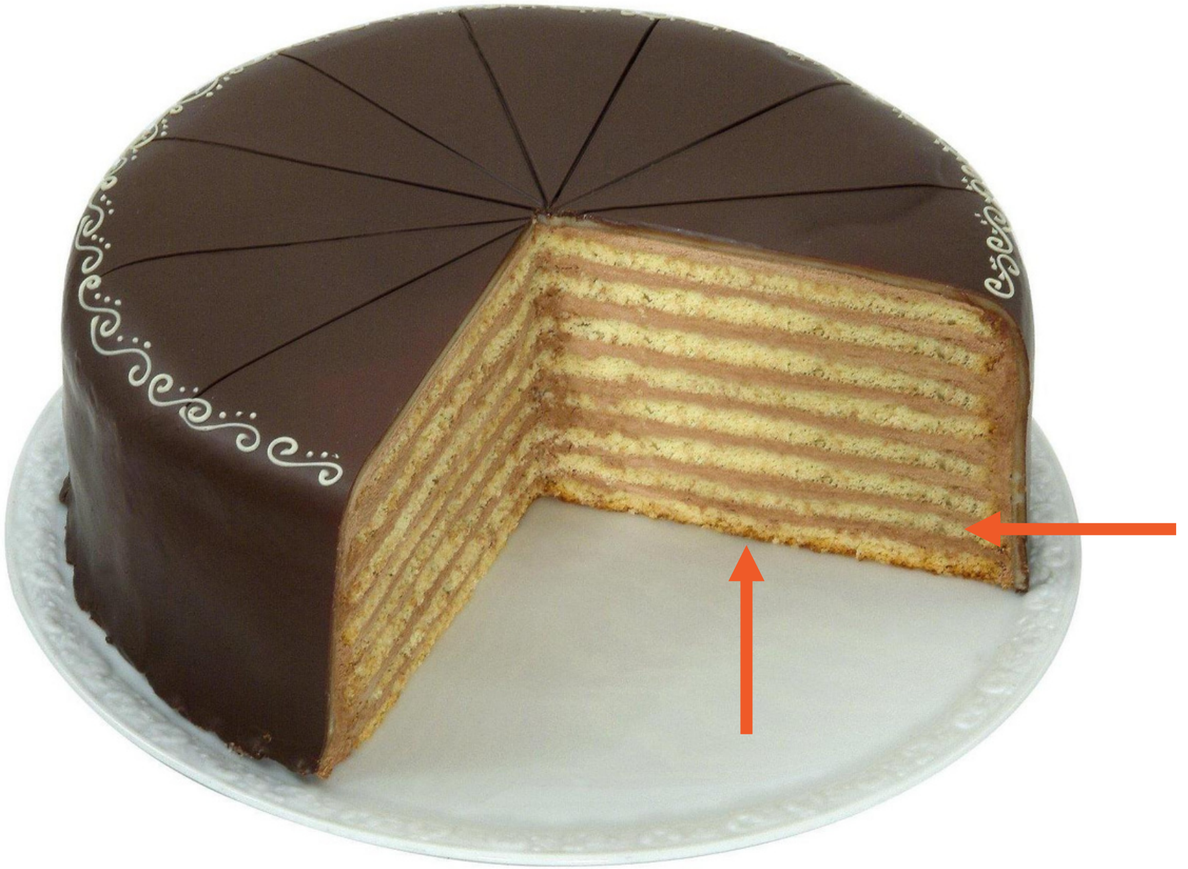


- Internal Registry
- Docker Hub
- Amazon ECR
- Azure Container Registry
- Google Container Registry
- More...

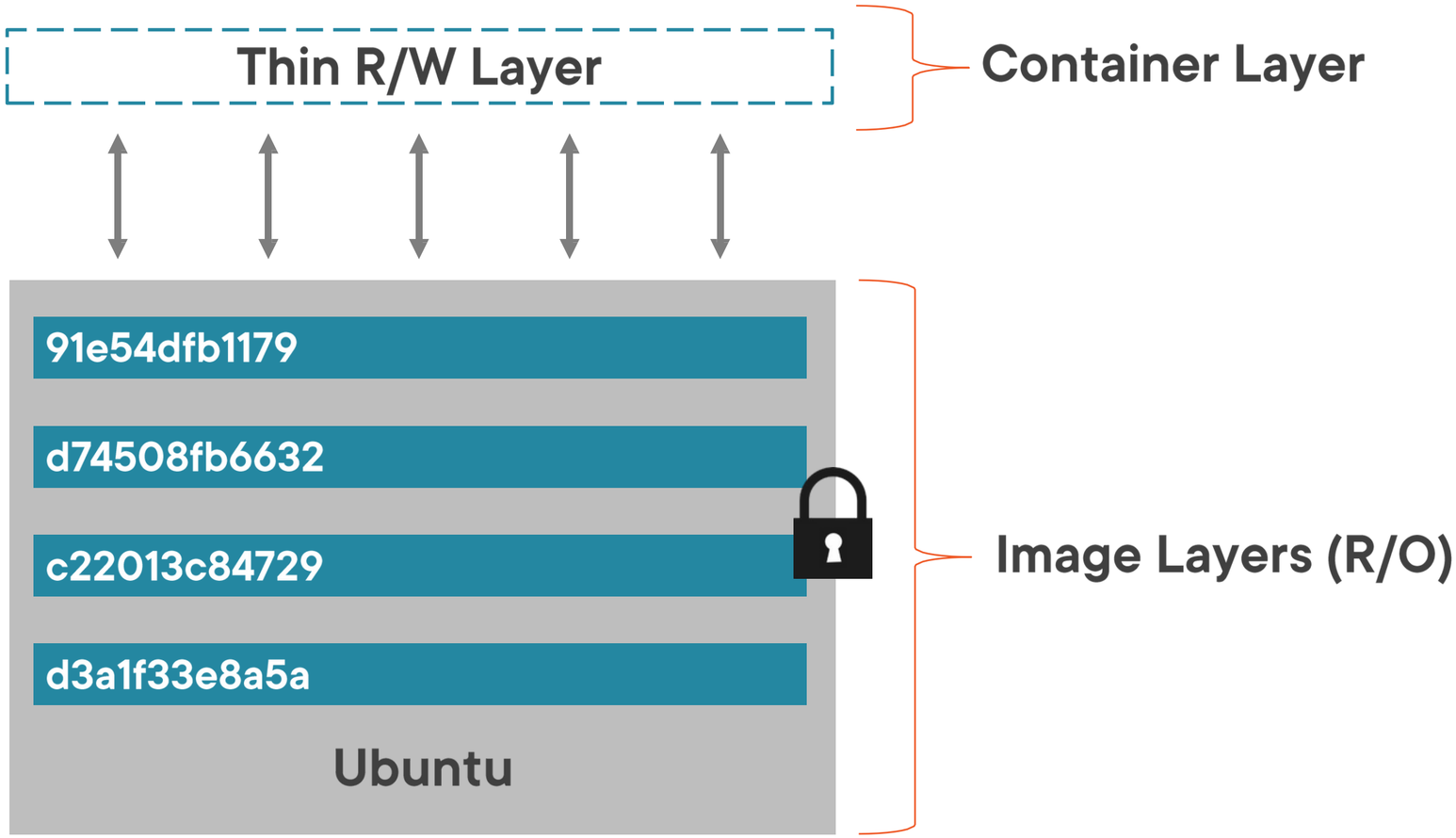
Deploy an Application Image to a Registry

Run an Application Container

Images and the Layered File System



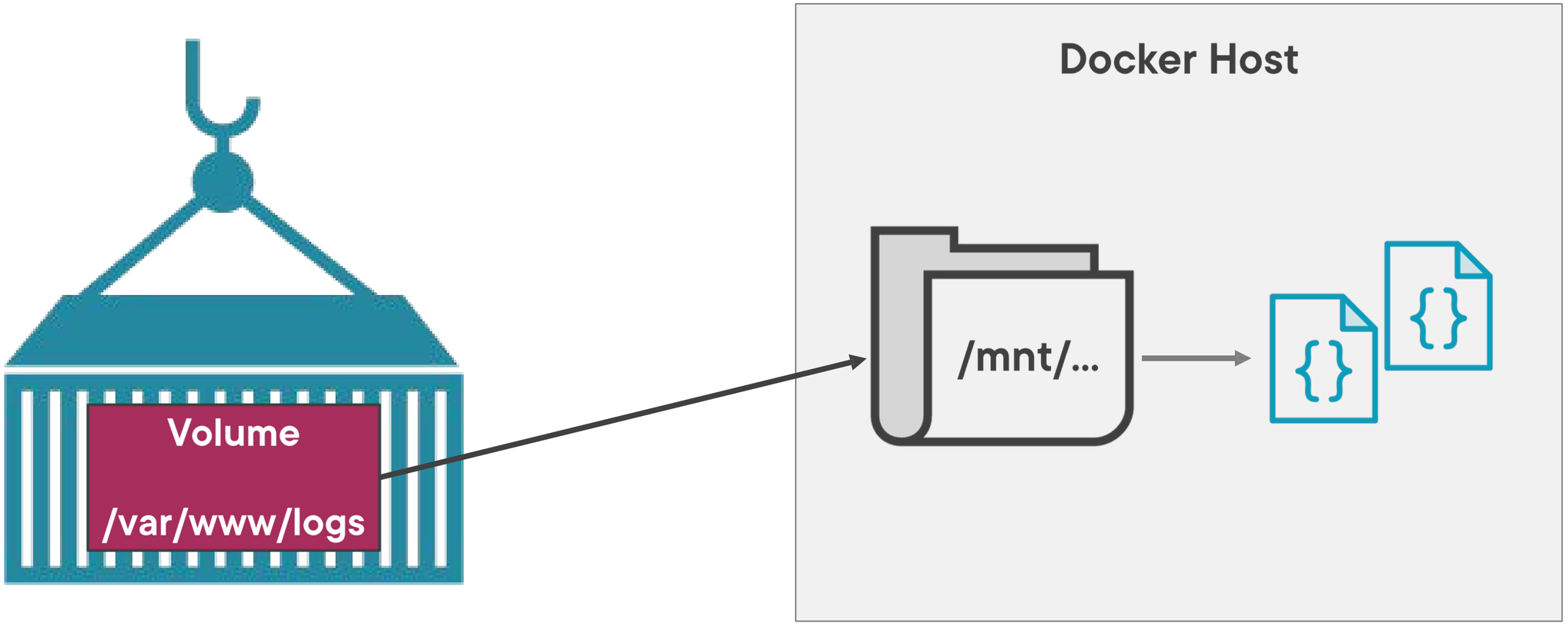
Images, Containers and File Layers



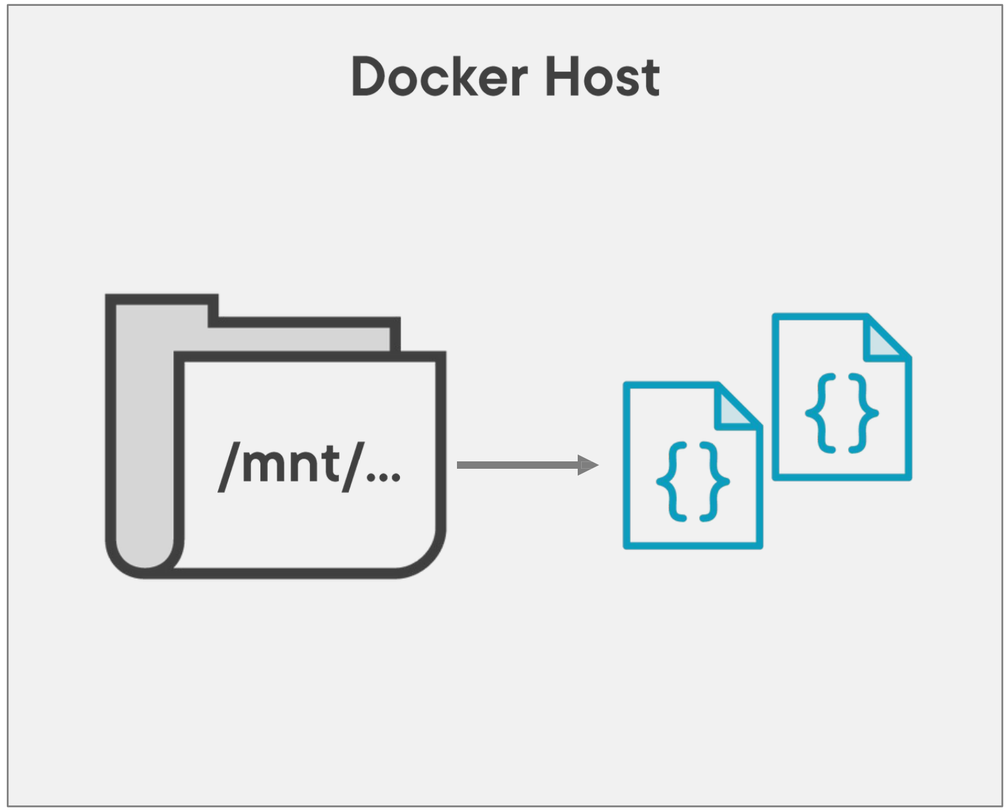
Using Container Volumes

Storing Data Outside of a Container

Volume Mounts



Volume Mounts



Key Volume Scenarios

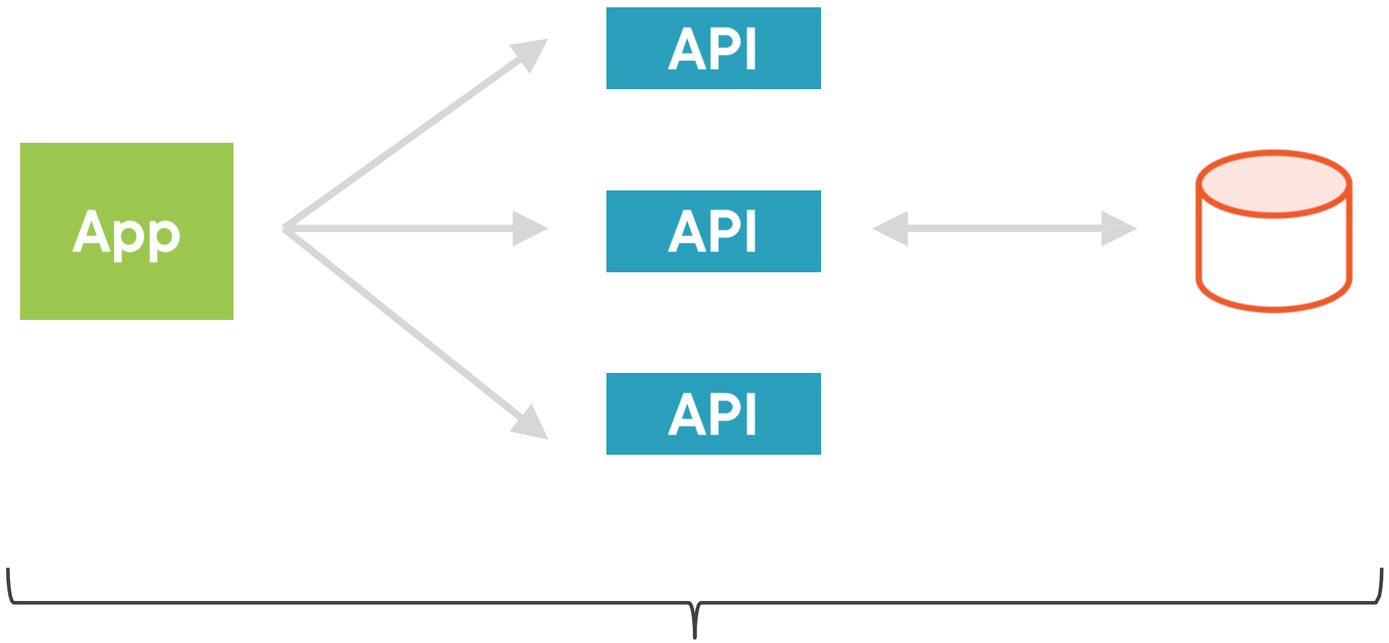
Production Scenarios
(logs, database files and more)

Development Scenarios
(develop using containers)

Create a Container Volume

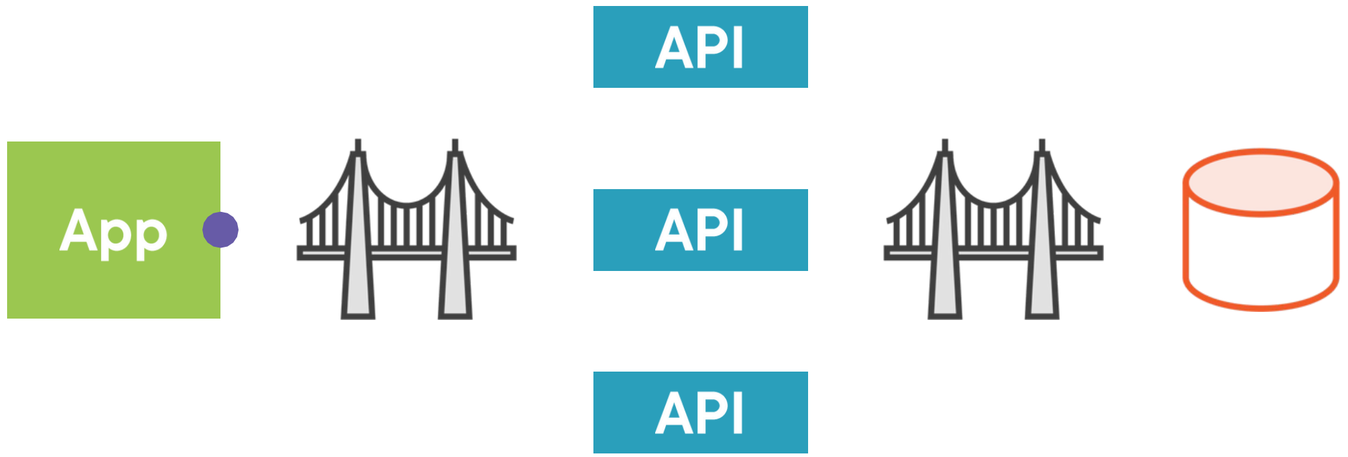
Create a Bridge Network

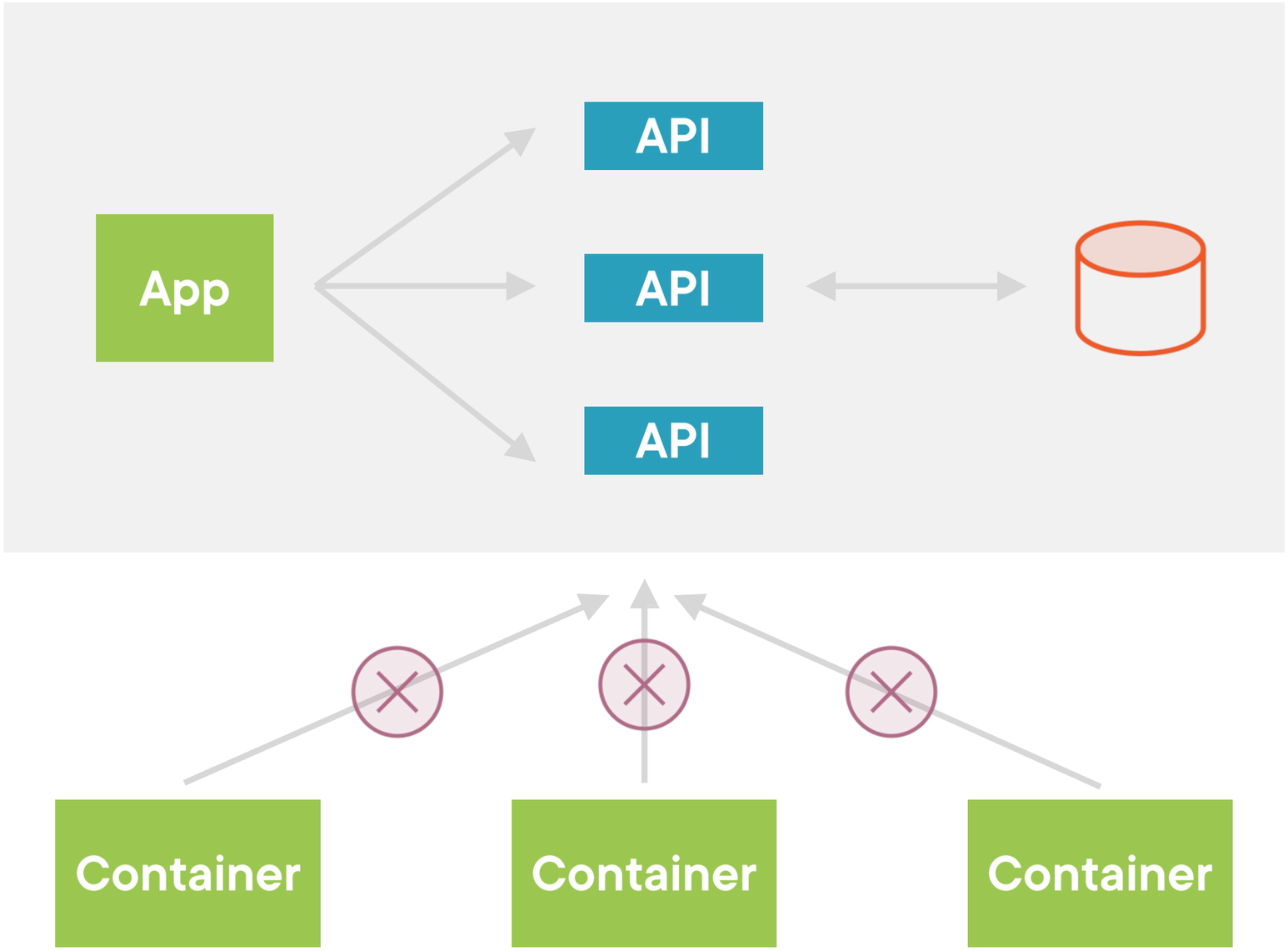
Communicating Between Containers



How do you communicate between containers?

Using a Bridge Network to Communicate




App
API
API
API
Container
Container
Container

Key docker network Commands

docker network create

docker network ls

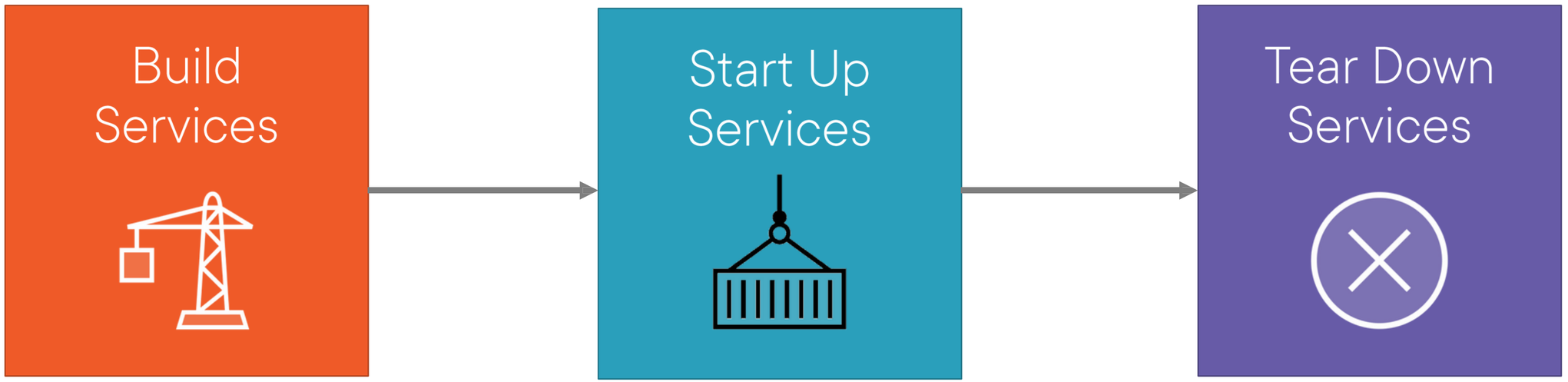
docker network rm [network]

Building and Running Multiple Containers with Docker Compose

Docker Compose Features

- Define services using a YAML configuration file
- Build one or more images
- Start and stop services
- View the status of running services
- Stream the log output of running services

Docker Compose Workflow



Key Docker Compose Commands

docker-compose build

docker-compose up

docker-compose down

@NomadicMehul

Looking for one-stop-shop for all things cloud-related?

https://github.com/nomadicmehul/CloudCaptain

Thank You!

@NomadicMehul