

Introduction to Docker



Announcements

- ▶ HW 2ab due *tonight*
 - ▶ Golang Syntax
 - ▶ Feedback form
- ▶ HW 3 releasing tonight/tomorrow morning
- ▶ AWS Educate accounts
 - ▶ Register + join asap

Last Time

- ▶ Golang
 - ▶ Finishing up web server programming
 - ▶ Cookies
 - ▶ Query Parameters
 - ▶ Request Bodies (JSON)

▶ Today

- ▶ Introduction Docker
 - ▶ Local vs Production Environments
 - ▶ VMs vs Containers
 - ▶ Hello World Program

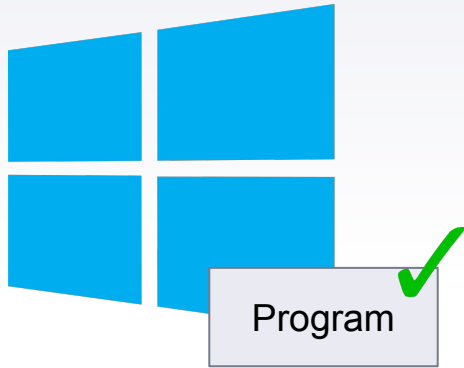
Local vs Production

- ▶ Programs
 - ▶ Program Code
 - ▶ .go, .py, .c, .cpp
 - ▶ Code Dependencies
 - ▶ gorilla/mux (Go), flask/Django (Python)
 - ▶ Environment Configuration Settings
 - ▶ environment variables, deployment configurations
 - ▶ .env

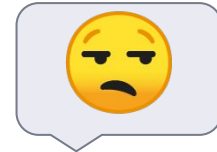
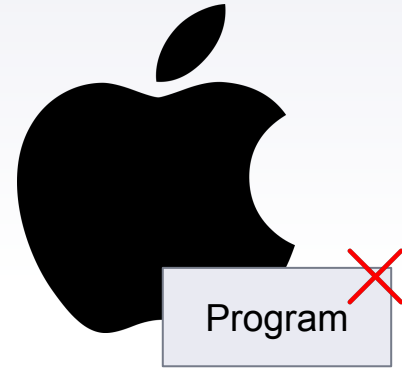
Local vs Production

- ▶ Currently
 - ▶ All code runs on our current operating system
 - ▶ All dependencies installed on our machine
 - ▶ All configs defined locally
- ▶ Issues?
 - ▶ Program should run the same regardless of where it's run
 - ▶ Reality: different operating systems, different configurations, different dependency versions, etc.

Compatibility Issues



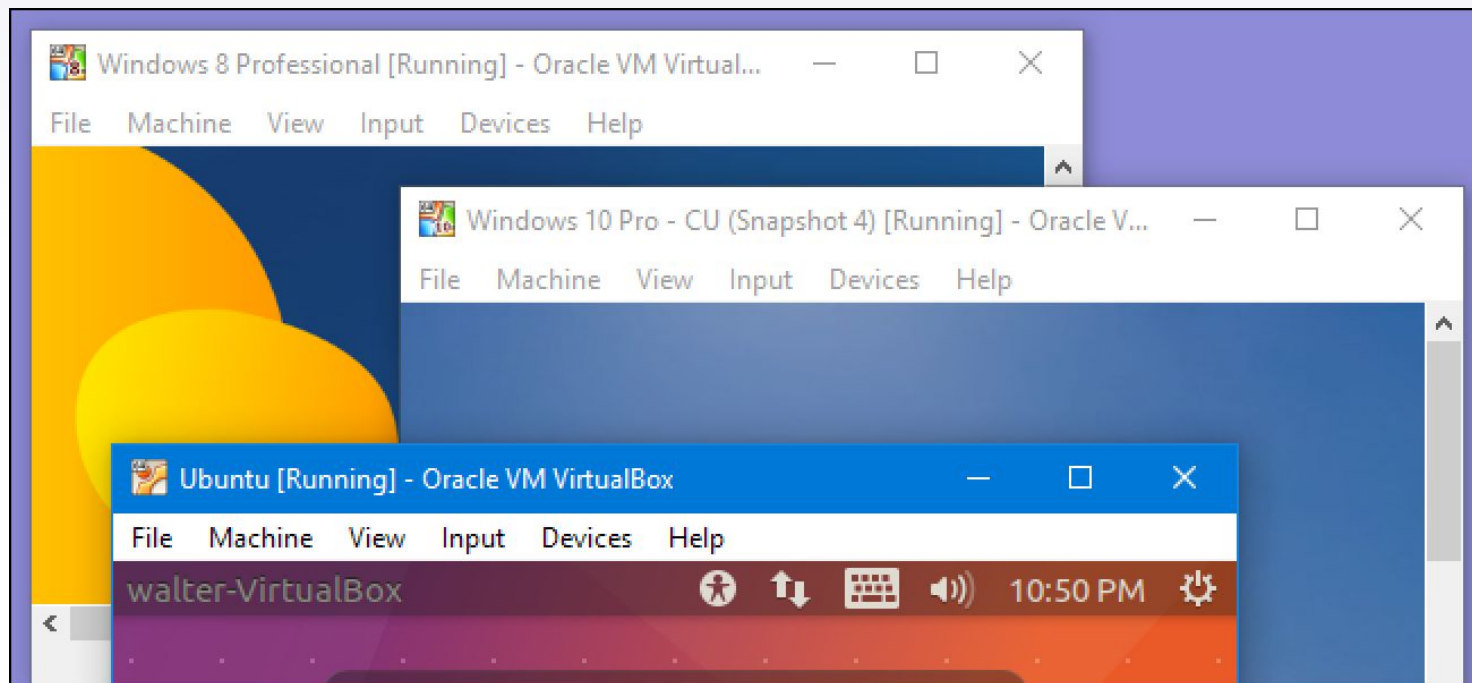
It worked
on my
machine!?



Virtual Machines

- ▶ Emulation of an Operating System
- ▶ “A computer within a computer”
- ▶ Host -> Computer running the VM
- ▶ VM is completely isolated from the host
 - ▶ Might as well be a different computer altogether
- ▶ Software Applications?

Virtual Machines



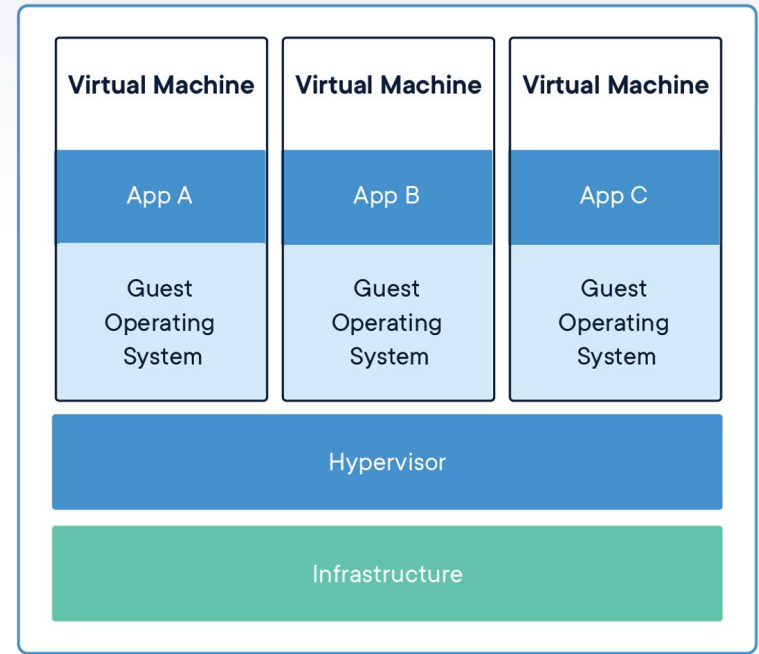


Virtual Machines

Should we use virtual machines to package our applications?

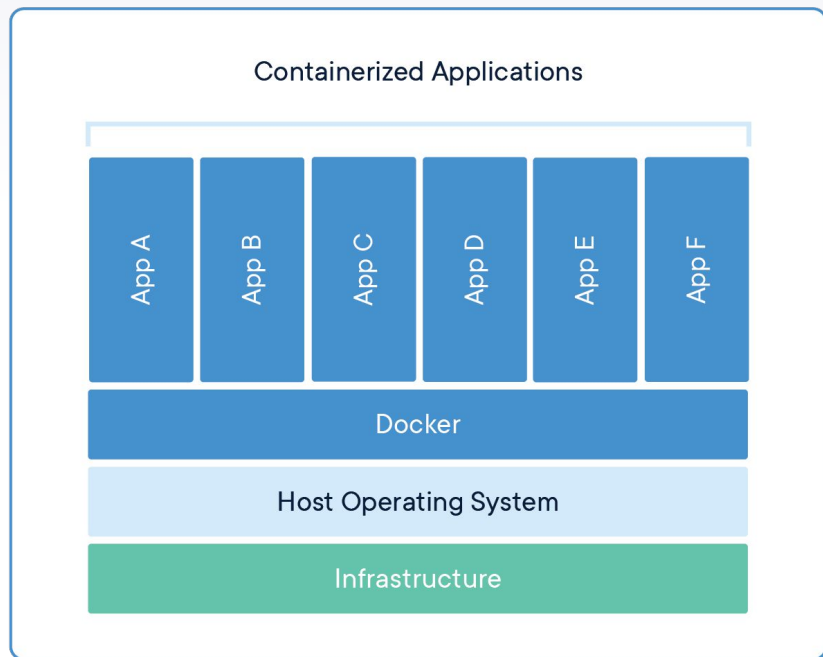
Virtual Machines

- ▶ Extremely computation heavy
 - ▷ Using resources for running two computers simultaneously
- ▶ Inefficient
 - ▷ Indirect access to hardware
- ▶ Difficult to develop with
 - ▷ Need to be

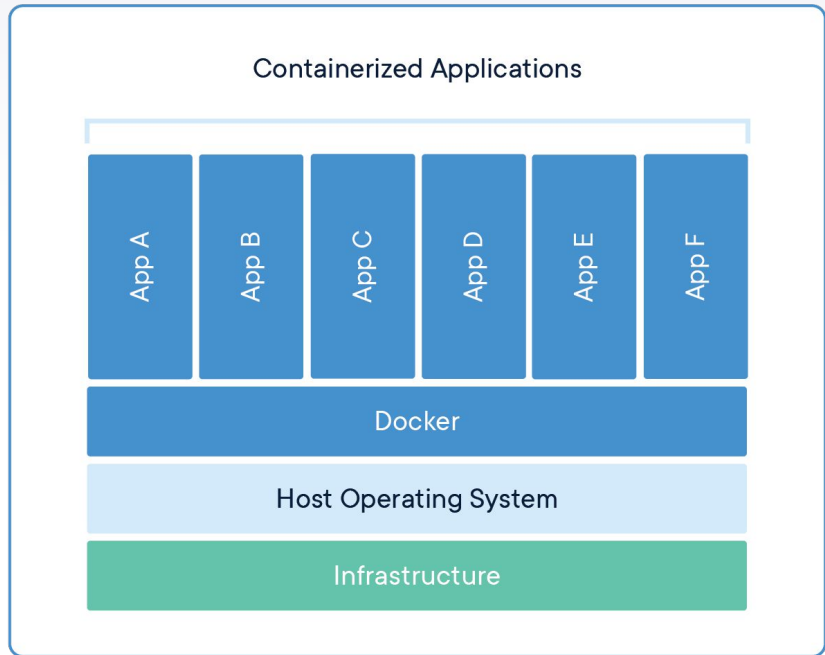
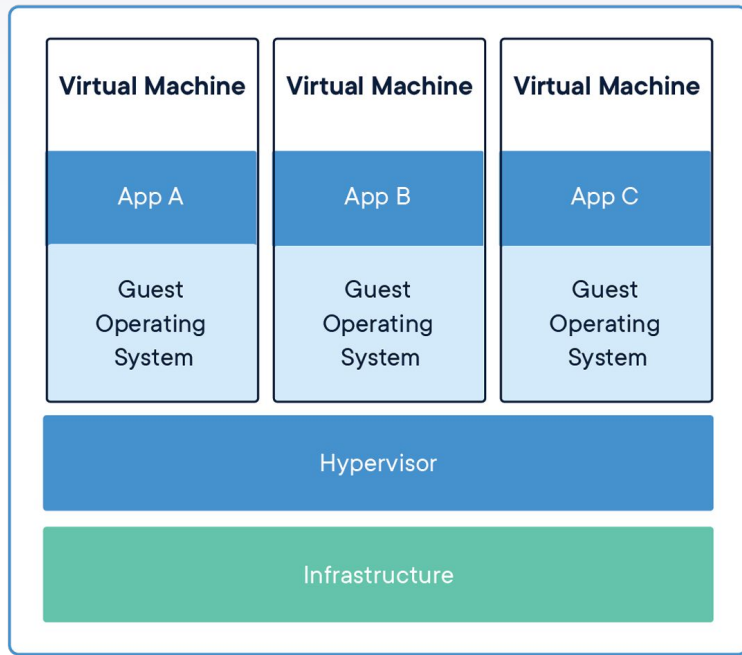


Containers

- ▶ Packaging mechanism abstracted from the running environment
- ▶ VMs without the overhead
- ▶ Isolated environment
 - Very close to host OS
 - Can be thought of an application running locally on host OS
- ▶ Easily configurable, shareable, and deployable



VMs vs Containers





VMs vs Containers

Should we use virtual machines or containers
to package our applications?

Docker

- ▶ Build and run containers
- ▶ Building
 - ▶ Packages an application with all of its dependencies into one unit
 - ▶ *"Image"*
- ▶ Running
 - ▶ Create an *"instance"* of an *"image"*
 - ▶ Can have multiple instances of the same image



Dockerfiles

- ▶ Text document that describes the *image* you want to build
 - What language/OS to optimize for?
- ▶ Copies files from local system to your image
- ▶ Allows you to pass in environment variables
- ▶ Etc.
 - Will get into more once project is assigned!

Docker Demo

