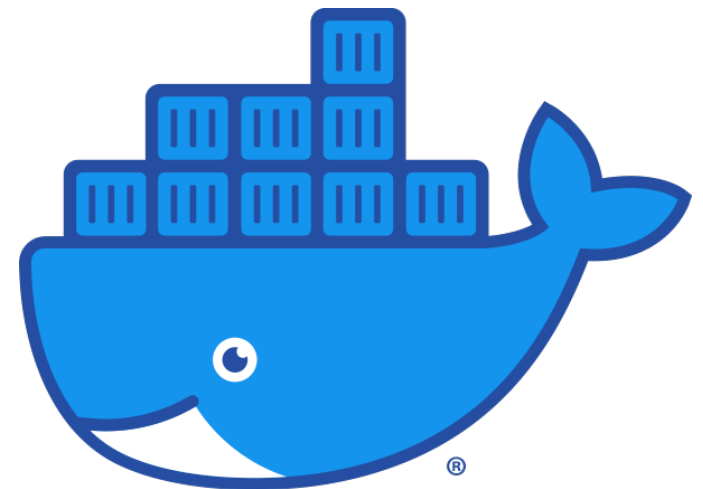


# [544] Docker Deployment

Tyler Caraza-Harter



# Learning Objectives

- use existing Docker images to launch containers
- define new Docker images using Dockerfiles
- troubleshoot common issues with running Docker containers

# Outline

## Virtualization

Images, Containers, and Dockerfiles

Demos...

# What is virtualization?

Definition: the illusion of **private** resources, provided by software

Contexts this semester

- Virtual Machines (hardware)
- Virtual Machines (languages)
- Virtual Operating System (container) *new today*
- Virtual Memory (covered later lecture...)

# What is virtualization?

Definition: the illusion of **private** resources, provided by software

Contexts this semester

- Virtual Machines (hardware)
- Virtual Machines (languages)
- Virtual Operating System (container)
- Virtual Memory (covered later lecture...)

virtualized resources include CPU, RAM, disks, network devices, etc

VMs rarely use all their allocated resources, so overbooking is possible

**VM:** 8 GB of RAM  
and 4 cores

**VM:** 6 GB of RAM  
and 3 cores

**VM:** 8 GB of RAM  
and 6 cores

virtual machines  
for rent (by you)

**Physical Machine:** 16 GB of RAM and 8 CPU cores

actual hardware bought by cloud provider (like Google GCP) for their cloud services

# What is virtualization?

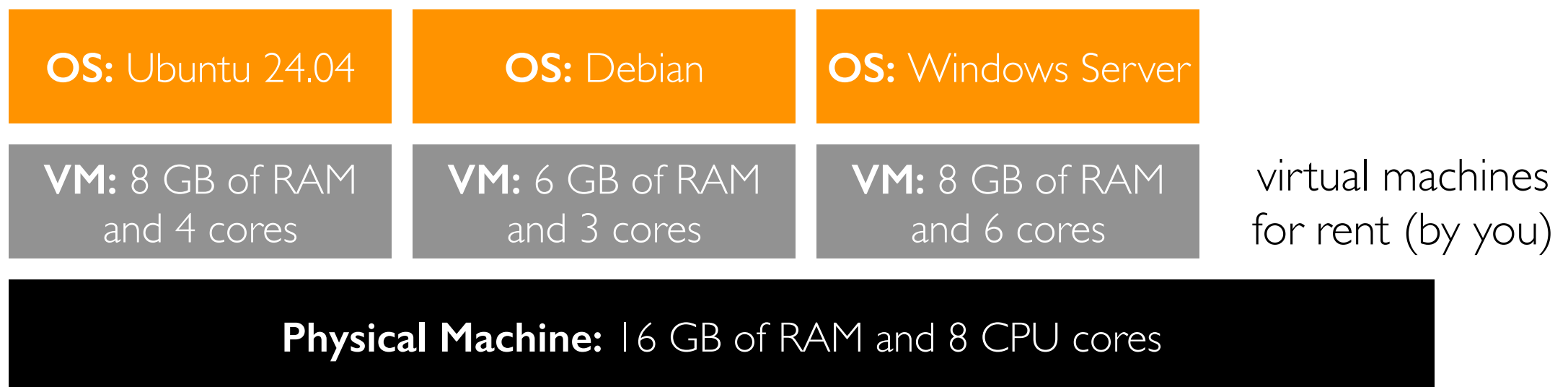
Definition: the illusion of **private** resources, provided by software

Contexts this semester

- Virtual Machines (hardware)
- Virtual Machines (languages)
- Virtual Operating System (container)
- Virtual Memory (covered later lecture...)

**problem:** if each program is deployed to a different VM, operating system overheads will dominate

these operating systems are mostly unaware that their on VMs instead of physical hardware



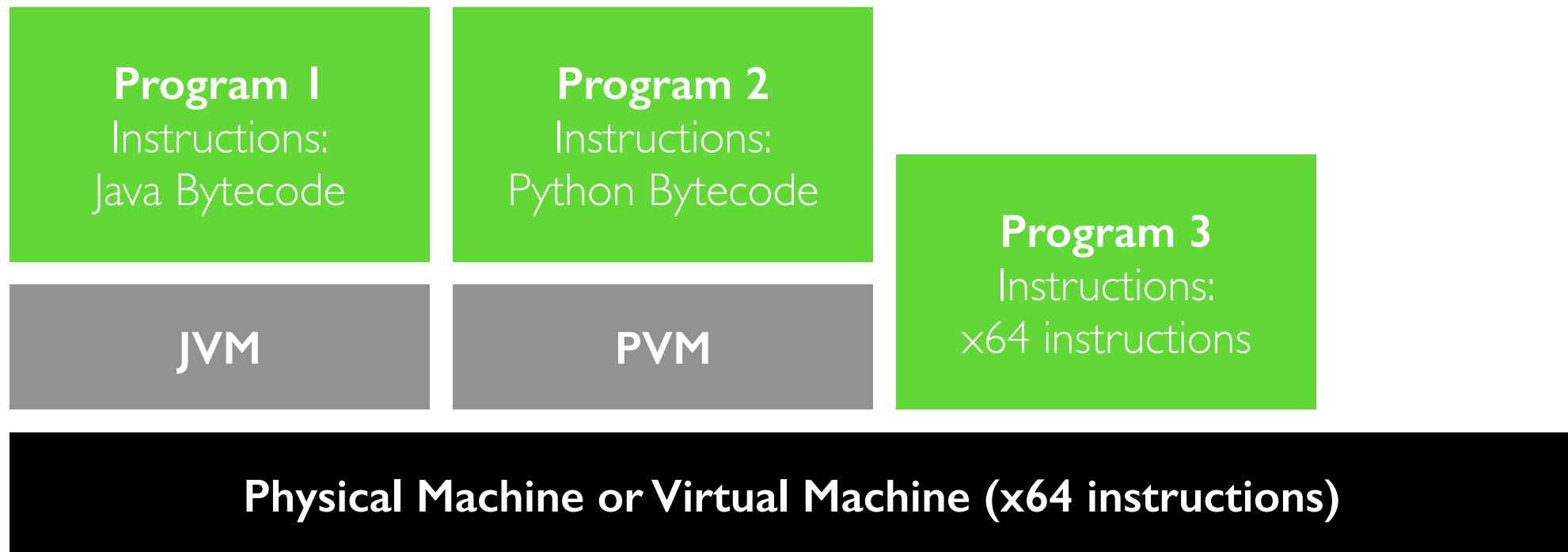
actual hardware bought by cloud provider (like Google GCP) for their cloud services

# What is virtualization?

Definition: the illusion of **private** resources, provided by software

Contexts this semester

- Virtual Machines (hardware)
- Virtual Machines (languages)
- Virtual Operating System (container)
- Virtual Memory (covered later lecture...)



# What is virtualization?

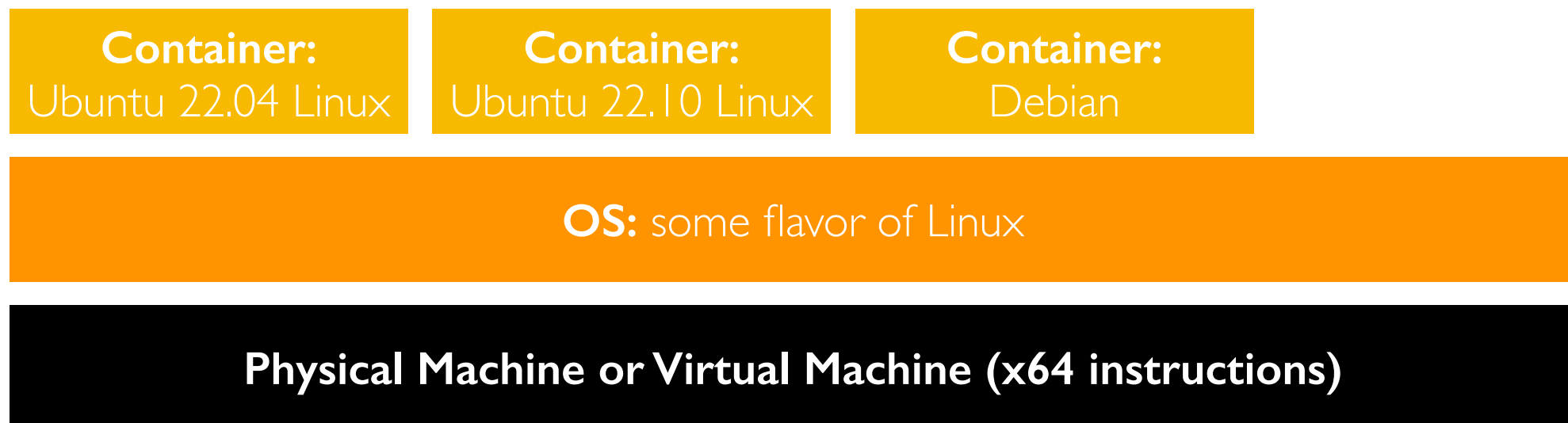
Definition: the illusion of **private** resources, provided by software

Contexts this semester

- Virtual Machines (hardware)
- Virtual Machines (languages)
- **Virtual Operating System (container)**
- Virtual Memory (covered later lecture...)

Linux containers

- Docker makes creation easy
- The "physical" OS is shared, which is very efficient
- Programs in different containers can use different flavors of Linux
- Cannot have a Windows container on Linux





# Containers and Virtual Machines are "Sandboxes"

A Computer

student submission:

```
import subprocess  
subprocess.check_output([  
    "rm", "-rf", "/"  
])
```

Sandbox



process

Sandbox



process

python3.12

Sandbox



process

Sandbox



process

python3.10



process

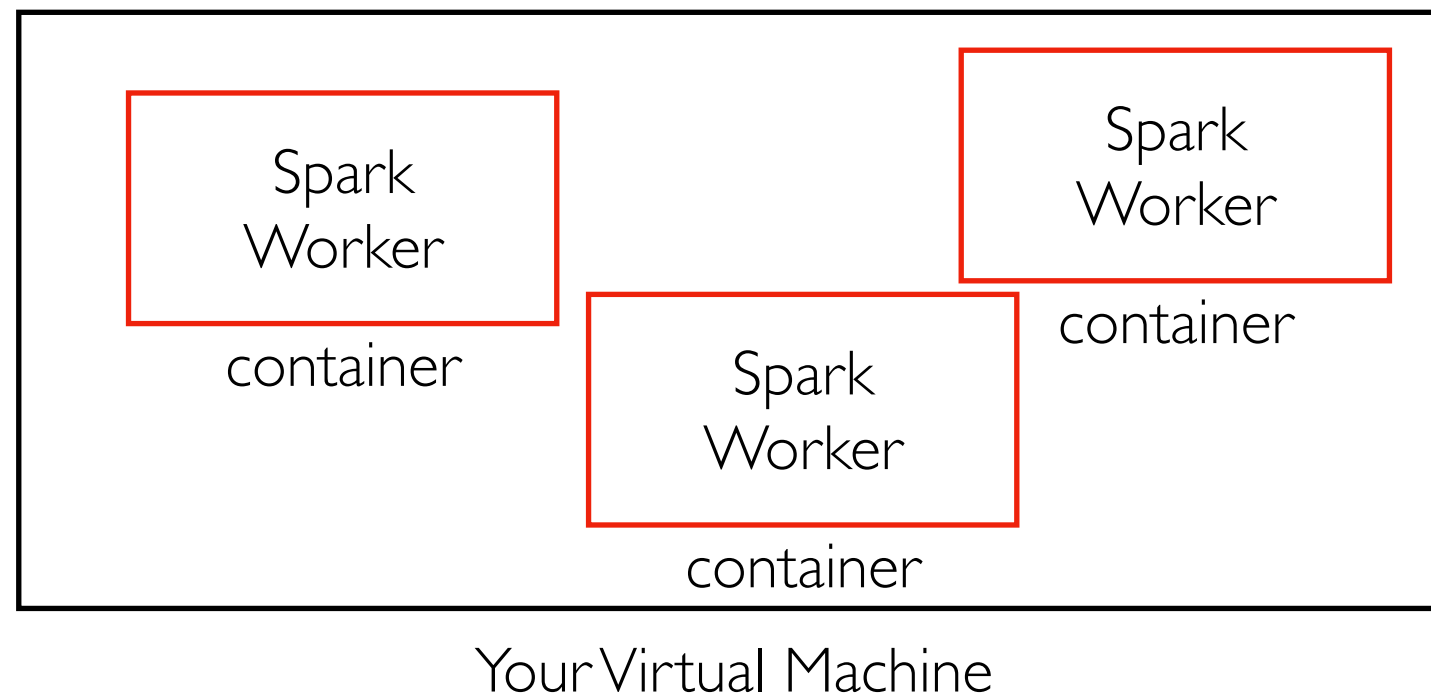


process

# Docker containers

Containers are a lightweight alternative to virtual machines.

You'll run Docker containers this semester to have your own "mini cluster"



Resources of the "cluster" are limited to those of a single VM, so we'll scale projects accordingly. But the techniques will apply to large clusters and datasets.

# Outline

Virtualization

Images, Containers, and Dockerfiles

Demos

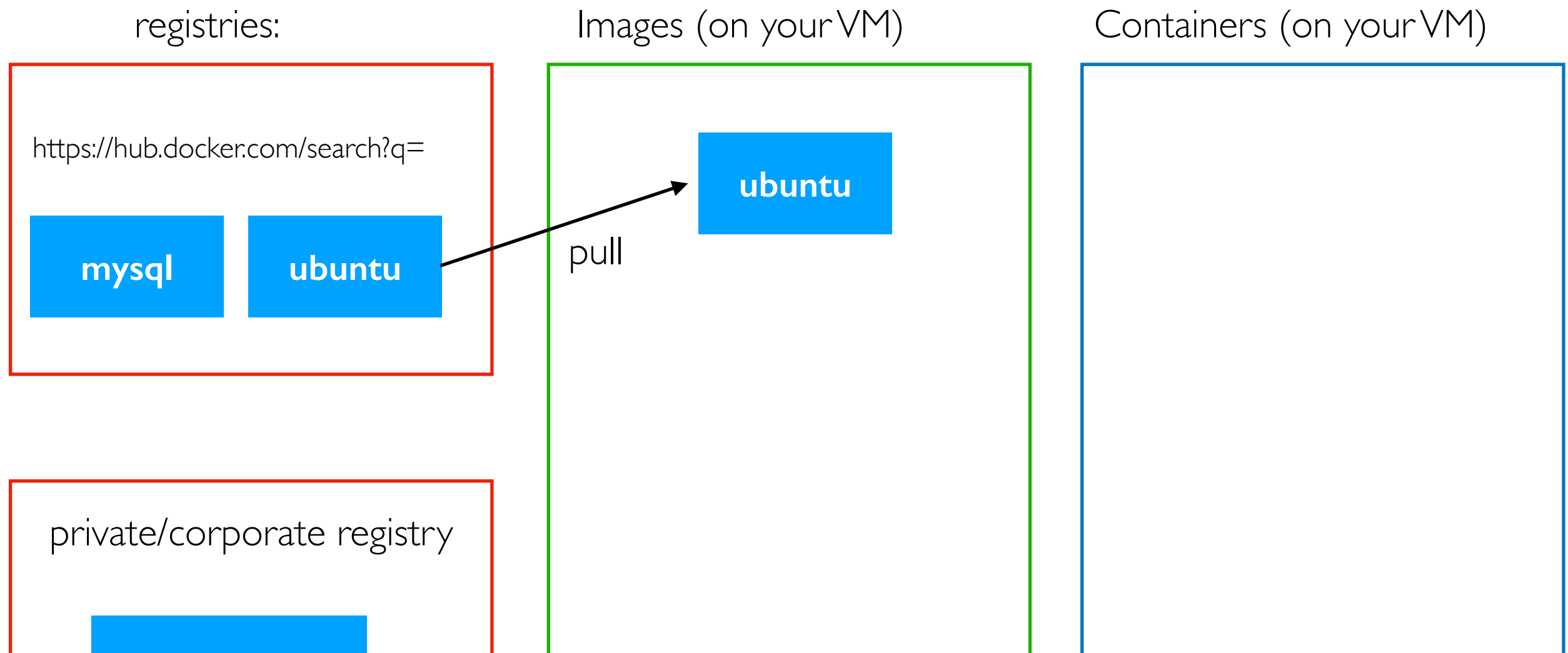
TIP: make notes of docker commands

docker **SOME-COMMAND** arg1, arg2, ...

# Docker Install

See Project I specification...

# Registries, Images, Containers, and Dockerfiles

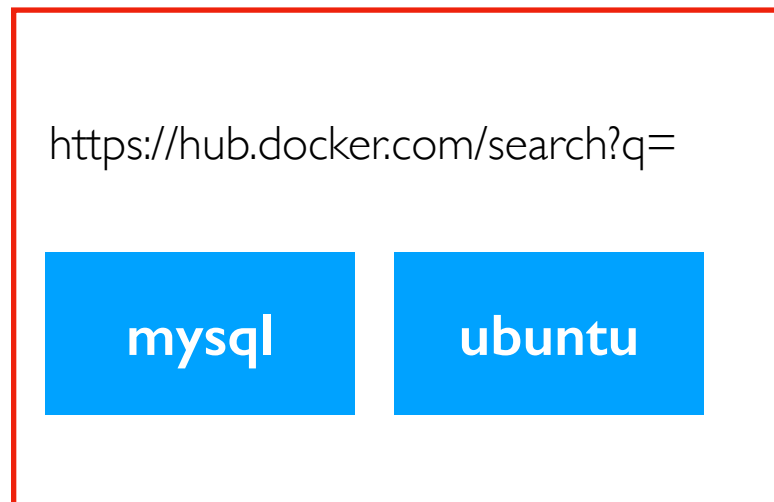


## Images

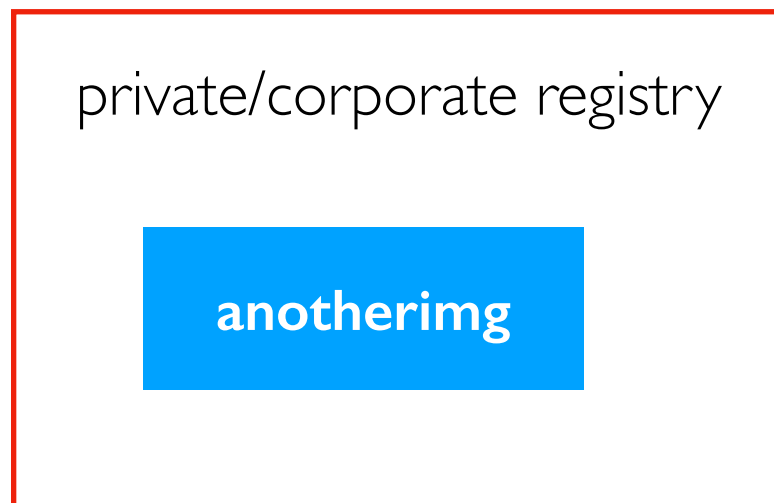
- snapshot of installed software from which to create a container
- `docker pull ubuntu:22.04`

# Registries, Images, Containers, and Dockerfiles

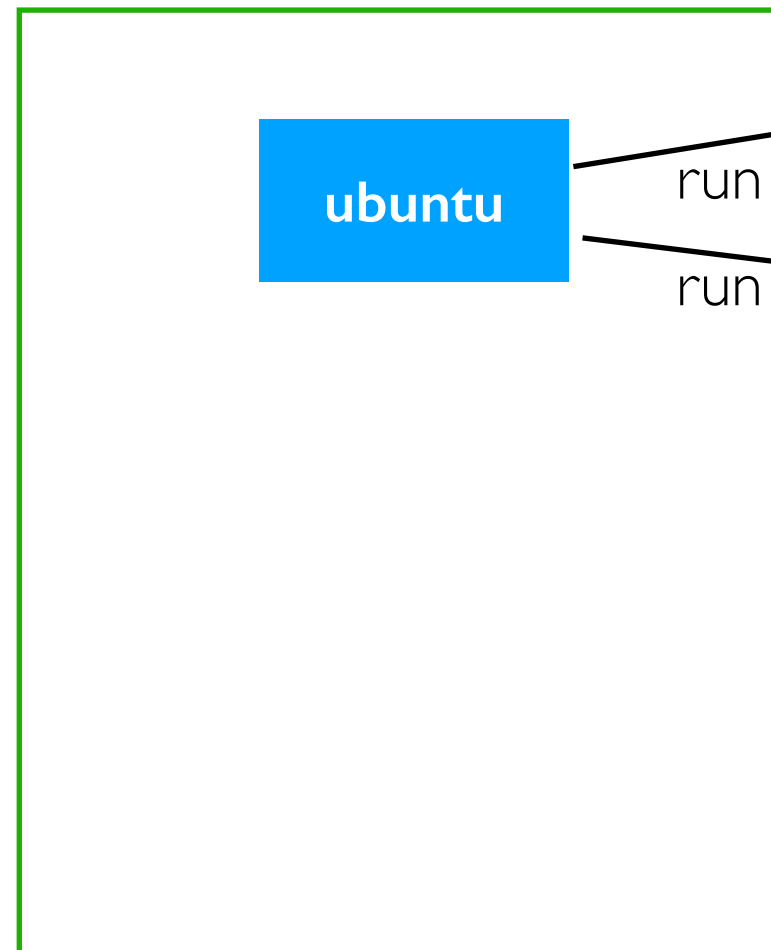
registries:



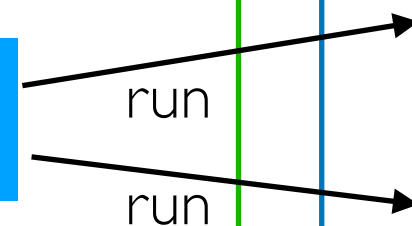
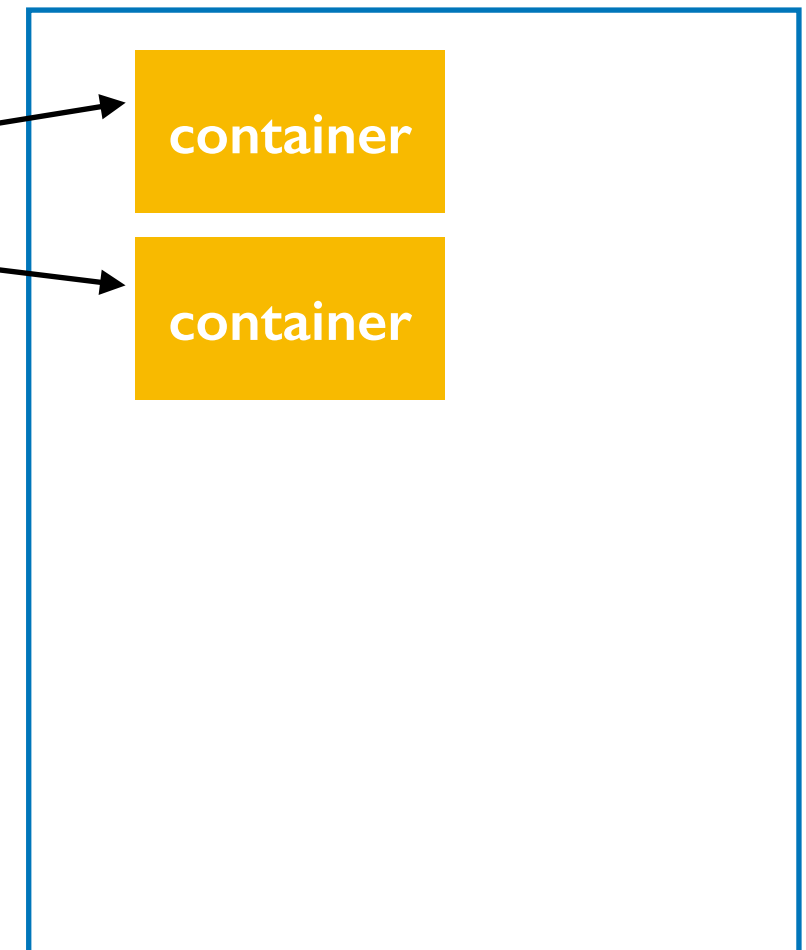
private/corporate registry



Images (on your VM)



Containers (on your VM)

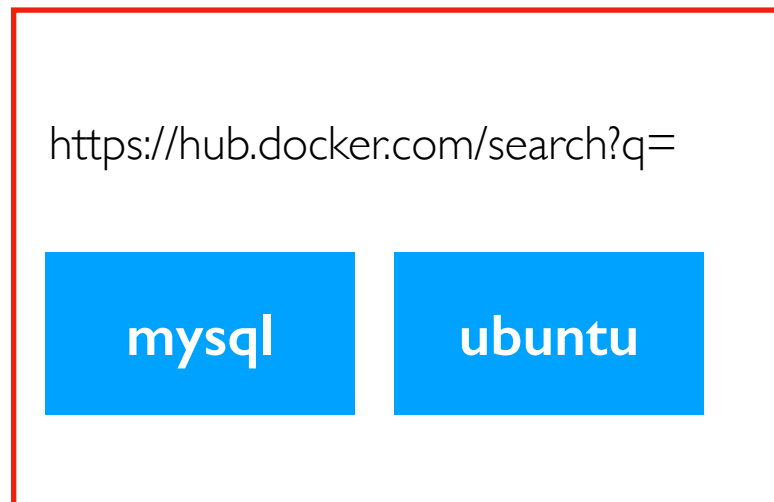


Containers

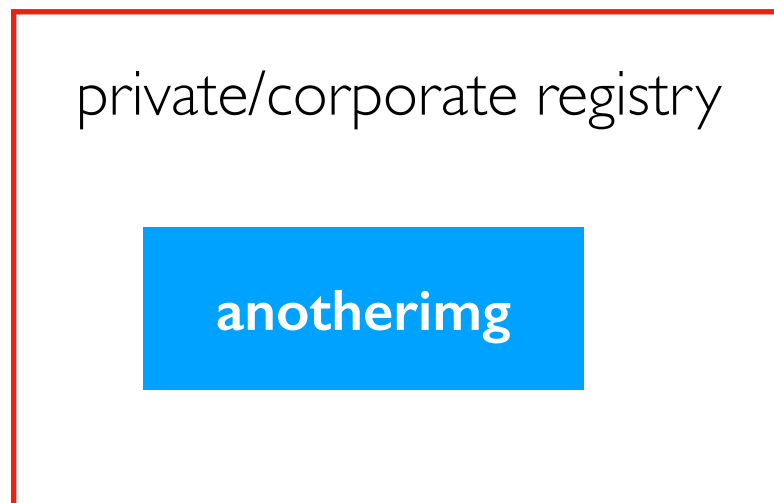
- Linux sandbox in which to run processes
- `docker run ubuntu`

# Registries, Images, Containers, and Dockerfiles

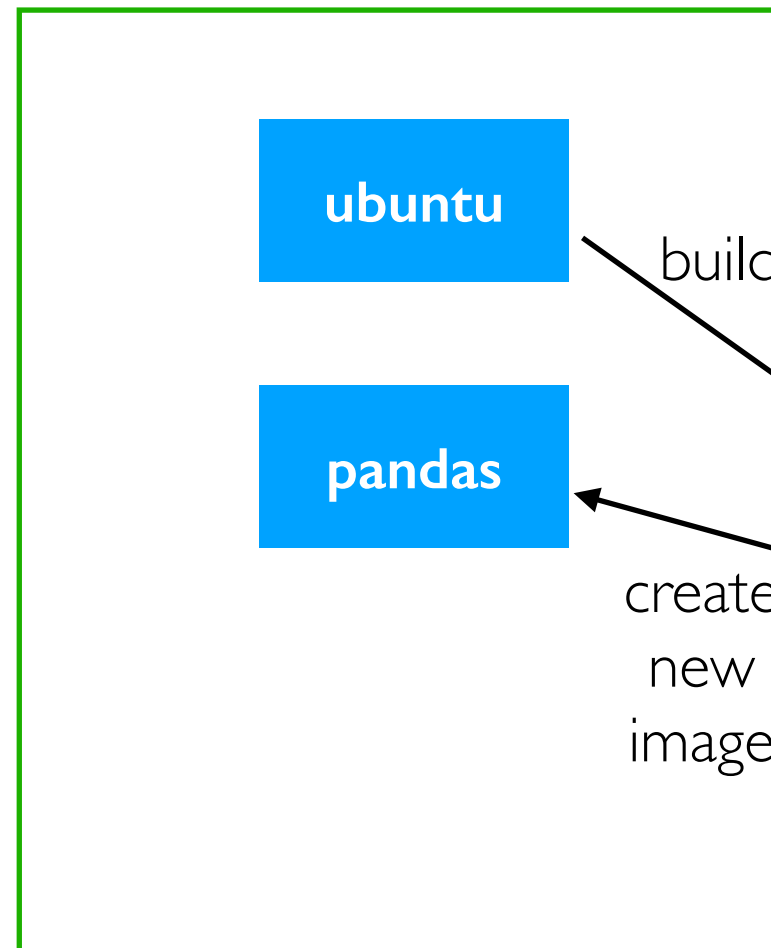
registries:



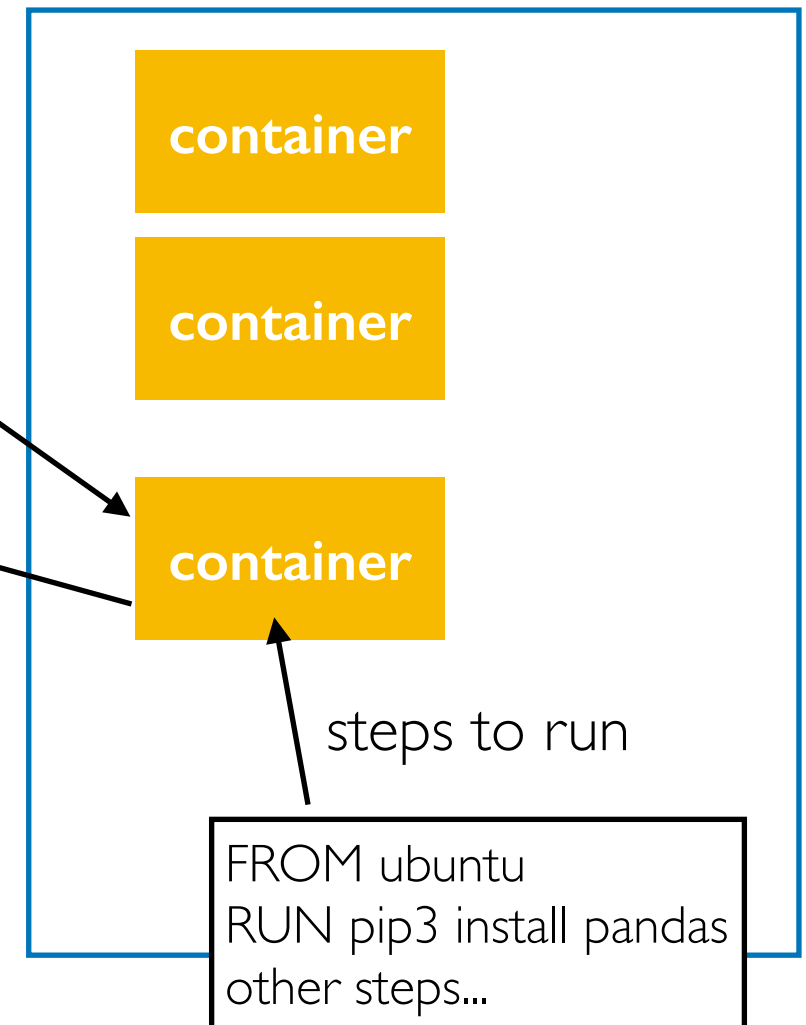
private/corporate registry



Images (on your VM)



Containers (on your VM)



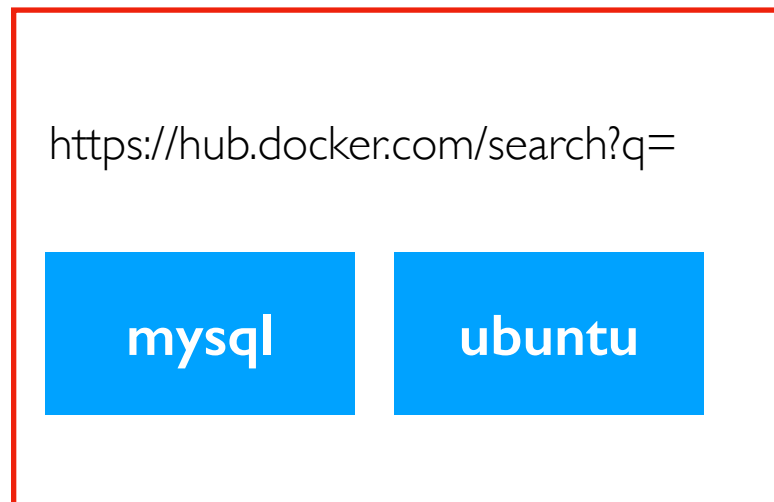
Dockerfiles

Dockerfile

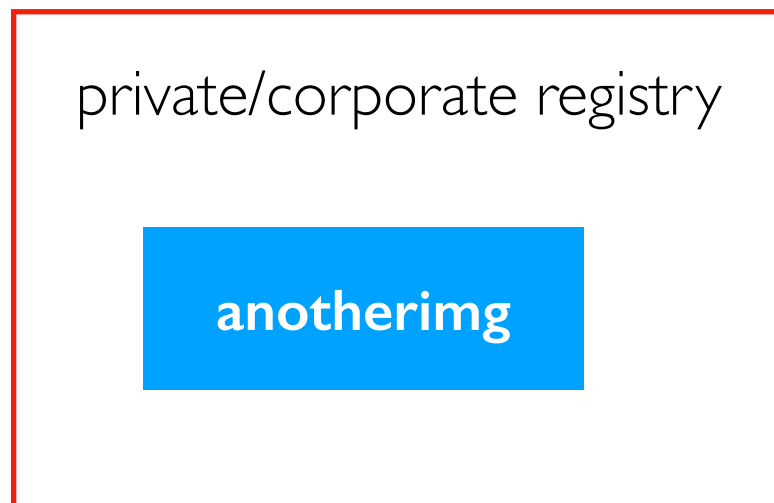
- steps to run in a container (like installs)
- creates a new image
- docker **build** myimg -t pandas

# Registries, Images, Containers, and Dockerfiles

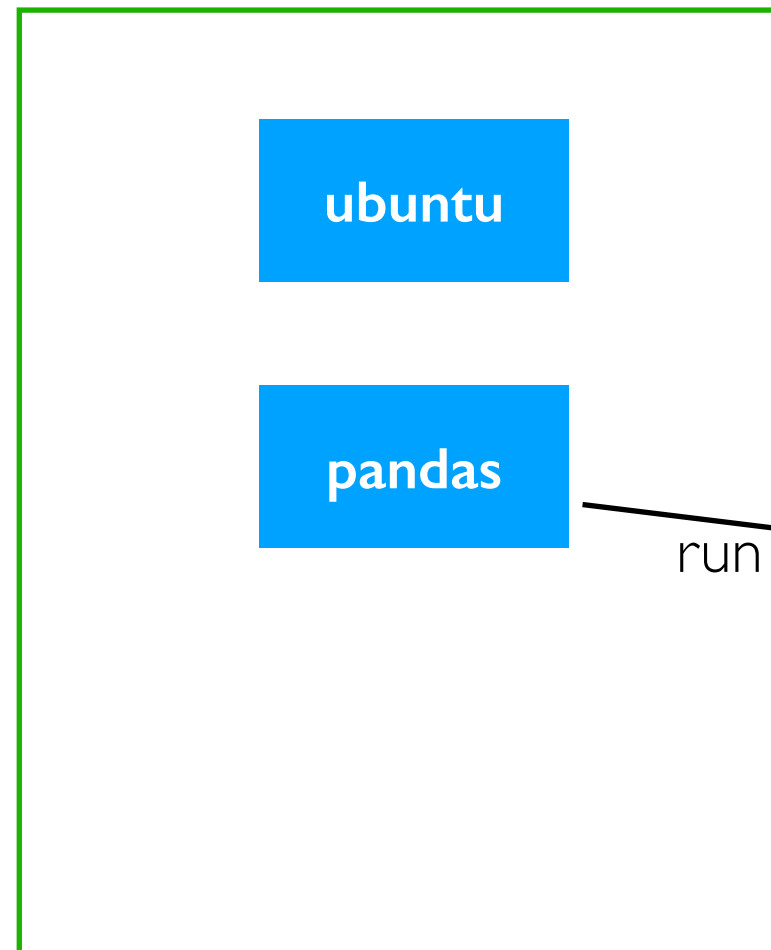
registries:



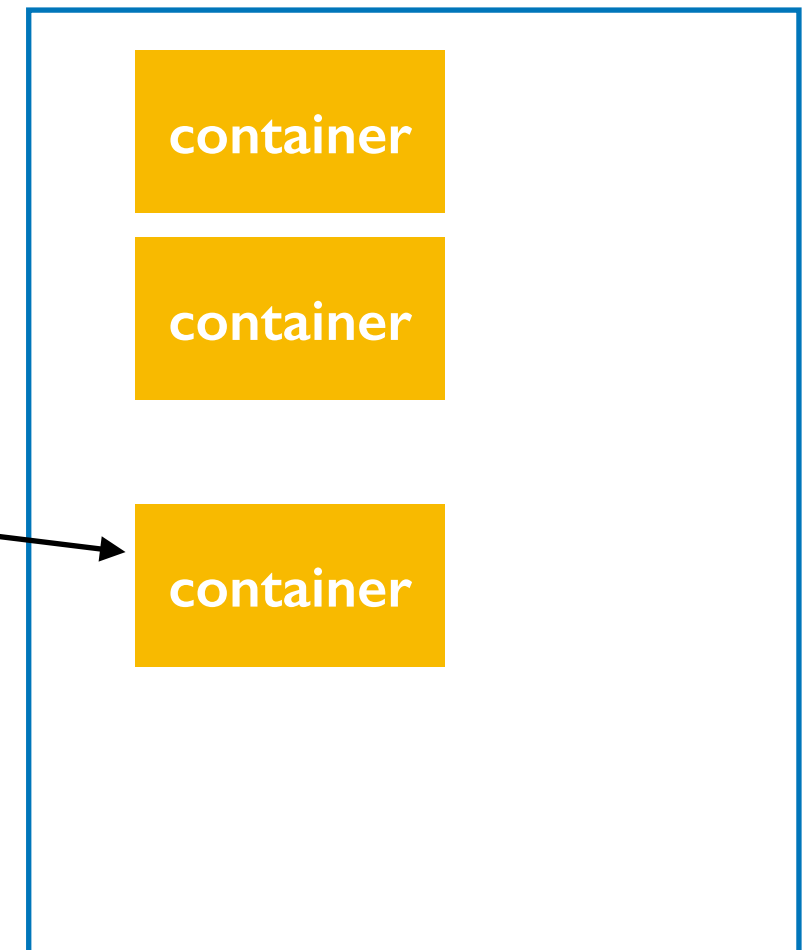
private/corporate registry



Images (on your VM)



Containers (on your VM)



run

Reproducibility

- Docker files unambiguously describe the setup
- Others can get all the same version numbers



TopHat, Demos...