Entrance Survey

http://bit.ly/199entrance

CS 199
Applied Cloud Computing

Prof. Robert J. Brunner

Ben Congdon
Tyler Kim
Bhuvan Venkatesh

Illinois Data Science Initiative

Google Cloud Platform

Course Intro / What is ACC?

This is **applied** cloud computing, not cloud computing

- Very little theory
- Focus on hands-on exercises

Who are we?

- Led by Professor Brunner
- Instructors
  - Ben Congdon
  - Tyler Kim
  - Bhuvan Venkatesh
- Course Staff
  - Osmar Coronel
  - Joshua Dunigan
  - Jake Trauger
  - Tyson Trauger

Contact: cs199-acc@lists.illinois.edu

Course Content

- What are we going to cover?
  - Containerization - Docker, VMs, Kubernetes
  - Hadoop / HDFS (Hadoop Distributed File System)
  - Spark
  - SQL/NoSQL Databases
  - Graph processing
  - Networking
  - Parallel Computing
  - Infrastructure Configuration

When & Where

Every Thursday, 7pm

Siebel Center 1131

Grading/Cutoffs

Attendance	10%
MPs	60%
Final Project (TBD)	30%

90%	A-
80%	B-
70%	C-
60%	D-

Communication

Gradebook: Moodle

Forum: Piazza

Course Website: bit.ly/cs199acc

MPs: Github + Nebula

Email: cs199-acc@lists.illinois.edu

Machine Problems (MPs)

- Released after lecture on **Thursday** evening
- Due the following **Wednesday** at 11:59pm
- **No late MPs** accepted, unless a valid excuse is provided

What is cloud computing?

What is cloud computing?

- It is NOT an actual cloud 

- Computing using many computers
- Layer of abstraction above physical hardware
- Typically made of up of a bunch of Virtual Machines
- Instead of one super powerful computer, use many weak computers
- Most of the time you *will not know* where your code is running physically

Virtual machines (VMs)

Virtual machines are simulated computers

- Allows us to simulate multiple computers on the same physical computer
- Each virtual computer shares the resources of the physical computer
- Lets you run an operating system on top of another operating system

Virtual machines (VMs)

Advantages?

Disadvantages?

Virtual machines (VMs)

Advantages

- Simulated computers are cheap to restart and destroy
- The environment of each simulated computer is identical!
  - If it runs on one VM, you can duplicate the VM and run hundreds of identical VMs
- They provide security for running untrusted code
- They can be used to scale horizontally by running duplicates of your code on many weak machines.

Disadvantages

- Computing is at lowest common denominator (of the hardware available, thus no special purpose GPUs, etc.).
- Slower than physical computers
  - One more layer of software between your code and the transistors

NCSA Nebula VMs

- You are going to run programs on part of the NCSA Nebula cluster
- NCSA is providing a compute cluster for us to run hadoop on.
- It will work similarly to EWS but most of the time your code will run on more than one computer in the cluster

Jake VanderPlas
@jakevdp

Following

CEO: we'll give them the cloud, but on their own desk!
Engineer: yeah but isn't that just a—
CEO: ship it.

1:20 PM - 16 Aug 2017

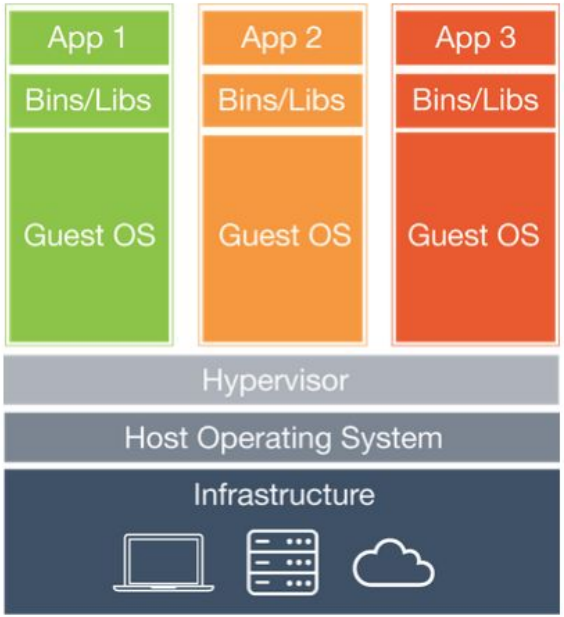
So... what is this course really about?

> “The cloud is just using someone else’s dumb computer instead of using your own dumb computer”
>
> - Professor Brunner

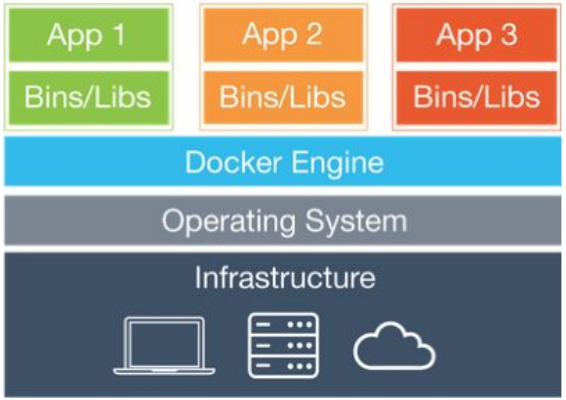
Motivation for Containerization

- VMs are great, but have high runtime overhead
- What if we could sandbox VMs, but share the OS kernel?

Big Idea: Containers have less OS overhead



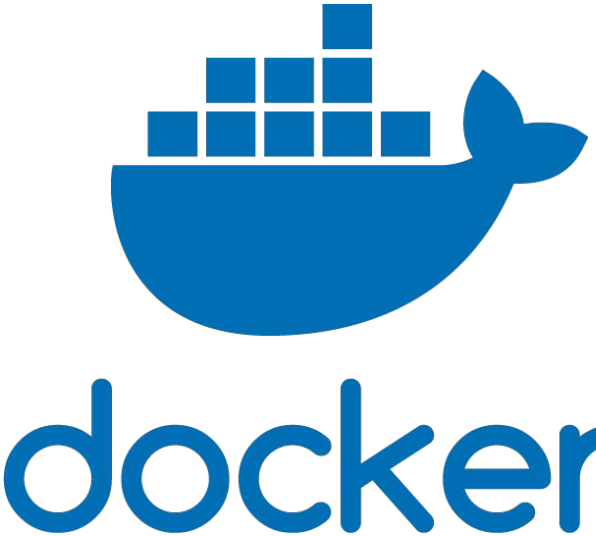
Traditional VM-based Infrastructure



Container Infrastructure

https://newsroom.netapp.com/blogs/containers-vs-vms/

Popular Containerization Platforms



Docker Concepts

- Image
  - Frozen description of an environment
- Container
  - Executed instantiation of an image
- Volume
  - Persistent data container

The Dockerfile

Hitchhiker's Guide to Running Code in Docker:

1. Inherit from a parent OS/platform container
2. Install any packages / libraries you need
3. Add any source code you need
4. Attach any volumes you need for data persistence
5. Set a command/script to be run at startup

The Dockerfile

Essential Commands:

- FROM - Inherit from a parent container
  - i.e. “FROM ubuntu”
- RUN - Runs a command *during* the build process
  - i.e. “RUN apt-get install python3”
- ADD - Copies files from the build directory into the image
  - i.e. “ADD hello_world.py /usr”
- EXPOSE - Register a port that the image will listen on
  - i.e. “EXPOSE 80”
- CMD - Set the default command to be executed on startup
  - i.e. “CMD python /usr/hello_world.py”

The Dockerfile

- Each command in a Dockerfile creates an intermediate image
  - Useful for caching!
- Structure your Dockerfiles to take advantage of caching
  - Install packages first, then add source code
  - Within reason, “Funnel down” from most general to most specific

Brief Docker Demo

Where to go from here

- Docker Swarm
  - Pools multiple Docker engines into a combined virtual host
  - Allows multiple VMs to collaborate to host clustered Docker containers
- Docker Compose
  - Orchestrate multiple-container applications
  - Declarative format for configuring volumes, container networking, and scaling
- Kubernetes
  - “Planet-Scale” container orchestration created by Google
  - Uses Docker images, but has separate container deployment system

MP 0

MP0 is released on the course website under MP.

It is due next Wednesday midnight - file submission on Moodle.