University of California, Berkeley
Master of Information and Data Science (MIDS)
W205 – Fundamentals of Data Engineering

Week 4 – Containers and Container Images

Agenda for Today’s Class

- Attendance and Participation
- Announcements
- Schedule and Due Dates
- Work / Life / School Balance
- Asynch High Level Review in a Nutshell
- Breakouts
- Summary

Attendance and Participation

Please record your attendance and participation for today’s class:

GitHub => ucb_mids_w205_repo => README.md => Attendance and Participation

Announcements

- Upcoming holidays and/or breaks
- Makeup classes for holidays
- Upcoming events
- Student evaluations
- Etc.

Schedule and Due Dates

Take a quick look at the next couple of weeks’ due dates:

GitHub => ucb_mids_w205_repo => README.md => Schedule and Due Dates

Work / Life / School Balance
Open Discussion

Student feedback

- About 5 minutes
- How are things going related to work / life / school balance?
- How is w205 going? Difficulty? Time?
- Impact of any natural and/or man-made disasters
- Etc.

Asynch High Level Review in a Nutshell

Each week we will spend about 15 minutes reviewing the most important high level concepts from the asynch

Containers

- VM = emulation of a physical hardware computer
  - Multiple VM’s can run on the same physical hardware computer
- Container = emulation of an OS
  - Multiple Containers can run in the same VM
  - Should be the same everywhere:
    - In a VM (such as in a cloud)
    - On top of an OS (such as Windows or Mac on desktop)
    - In a VM running on top of an OS (such as a container running in a VM running on top of Windows)

Uses for Containers

- Learn a new product – spin up a container instead of complicated install and config
- Proof of concept – ditto
- Development – each software engineer gets their own container – experimental code doesn’t affect others
- QA – each QA tester gets their own container – doesn’t affect others
- Production – easier to “ship” products to production
- Microservices (more on next slide)

Microservices

- Smaller services than the traditional large, monolithic services
- Each microservice gets its own container
  - Create a container, run a microservice, destroy container
  - No propagation of memory corruption
  - Extra wall of separation between users or transactions
- Costs more, but worth the cost
- Banks, credit cards, brokerage, utility companies, insurance, online services, airlines, hotels, point of sale, etc.

Container Images

(Analogy to Object-oriented)

- Object-oriented
  - Object is an instance of a class
  - A class can have unlimited number of objects
- VMs
  - VM is an instance of a VM image
  - An image can have unlimited number of VMs
- Containers
  - Container is an instance of a container image
  - An image can have unlimited number of containers

Container Images
(Analogy to Class Hierarchies)

- Class hierarchy in object-oriented
  - Class can be the parent of other classes
  - Class can be the child of other classes
- VM image hierarchy
  - VM image can be the parent of other VM images
  - VM image can be the child of other VM images
- Container image hierarchy
  - Container image can be the parent of other container images
  - Container image can be the child of other container images

Container Image Building

- Scripted process
  - Start with a container image
  - Create a container from the image
  - Customize the container
    - Install more software
    - Change configuration
    - Add users, groups, directories, files, permission, etc.
  - Create a new image from our customizations
- Images of images of images… allowed
- Base images are pure OSs

Breakouts

GitHub => ucb_mids_w205_repo => breakouts

(time permitting, we may not get to all of them)

Summary

Instructor will give a brief (about 2 minute) summary of today’s class.