Week 9
Containers & Docker

Mastering Cloud Computing

Coleman Kane

(based on material by Paul Talaga)

UNIVERSITY OF Cincinnati

Shared Infrastructure Problems

Software version dependency issues
Resource contention impact
Access control management
Server migration / upgrade concerns

UNIVERSITY OF
Cincinnati

Version/Dependency Issues

If you have two web applications hosted:

- App A requires library version 2
- App B requires library version 3
- Cannot have v2 & v3 installed concurrently
- Must jump through hoops to maintain both apps on shared server

Resource Contention

Multiple applications installed on common infrastructure may compete for resources:

- Hardware (mem, disk, CPU, etc.)
- Network/bandwidth
- Software resources, like database users, other on-system services

Difficult to separate/partition usage of each application

Access Control Management

Each installed application shouldn’t be able to access sensitive info from the other

- Files on disk
- App-specific services
- User lists
- System devices

Server Migration & Upgrade

When doing infrastructure management, upgrading HW & OS may require reinstall of applications, and possibly impact compatibility

Additionally, it can be very time intensive and failure-prone to reinstall from scratch

UNIVERSITY OF Cincinnati

Container Approach

Already introduced earlier in the class

A form of process-based “userland” isolation that can be achieved without HW virtualization

Popular example: **Docker**

UNIVERSITY OF Cincinnati

Docker Features

Separation between containers

- Network
- Filesystem
- Memory

Execution/CPU control

Network traffic management

OS Kernel is shared among containers, while libs & binaries are separated

Docker Community

Docker is more than simply a hosting environment: It’s a whole community!

- Community members contribute their work (images) into the docker community
- Iterative development methodologies
- Images are managed as deltas of modifications to other images
- Like “GitHub” for application stacks
- https://hub.docker.com/explore/

UNIVERSITY OF Cincinnati

Using Docker

Pull down a community-hosted image

- docker pull **imagename***[:tag]*

Show locally-hosted/pulled images

- docker images

Start up a new docker instance from an image, run /bin/bash, and give control

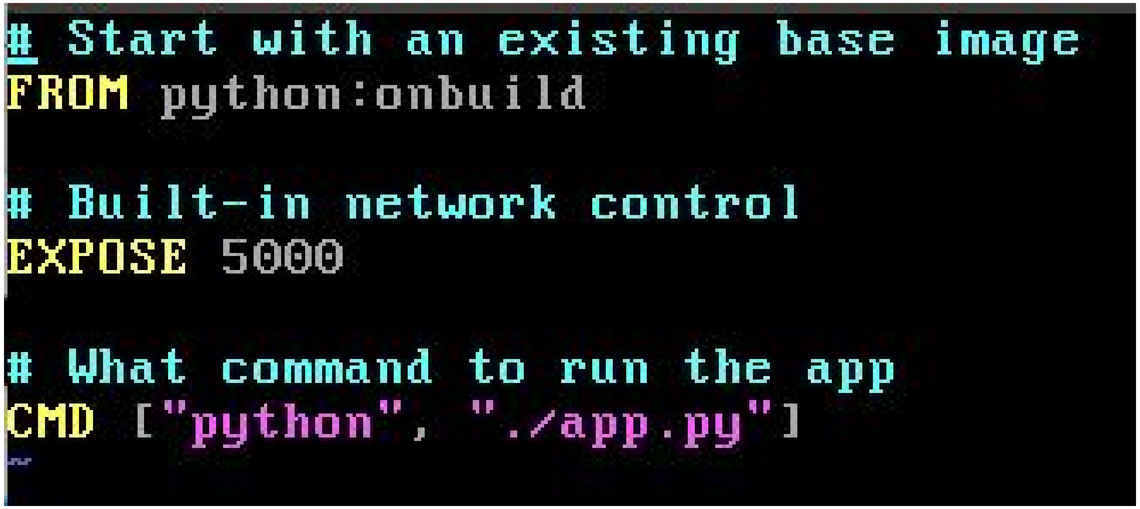
- docker run -it **imagename** /bin/bash

Developing on Docker

Docker images are built from a Dockerfile

- Instructions/recipe for building an image

Example:



```
# Start with an existing base image
FROM python:onbuild

# Built-in network control
EXPOSE 5000

# What command to run the app
CMD ["python", "./app.py"]
~
```