IT 609
Network and System Administration

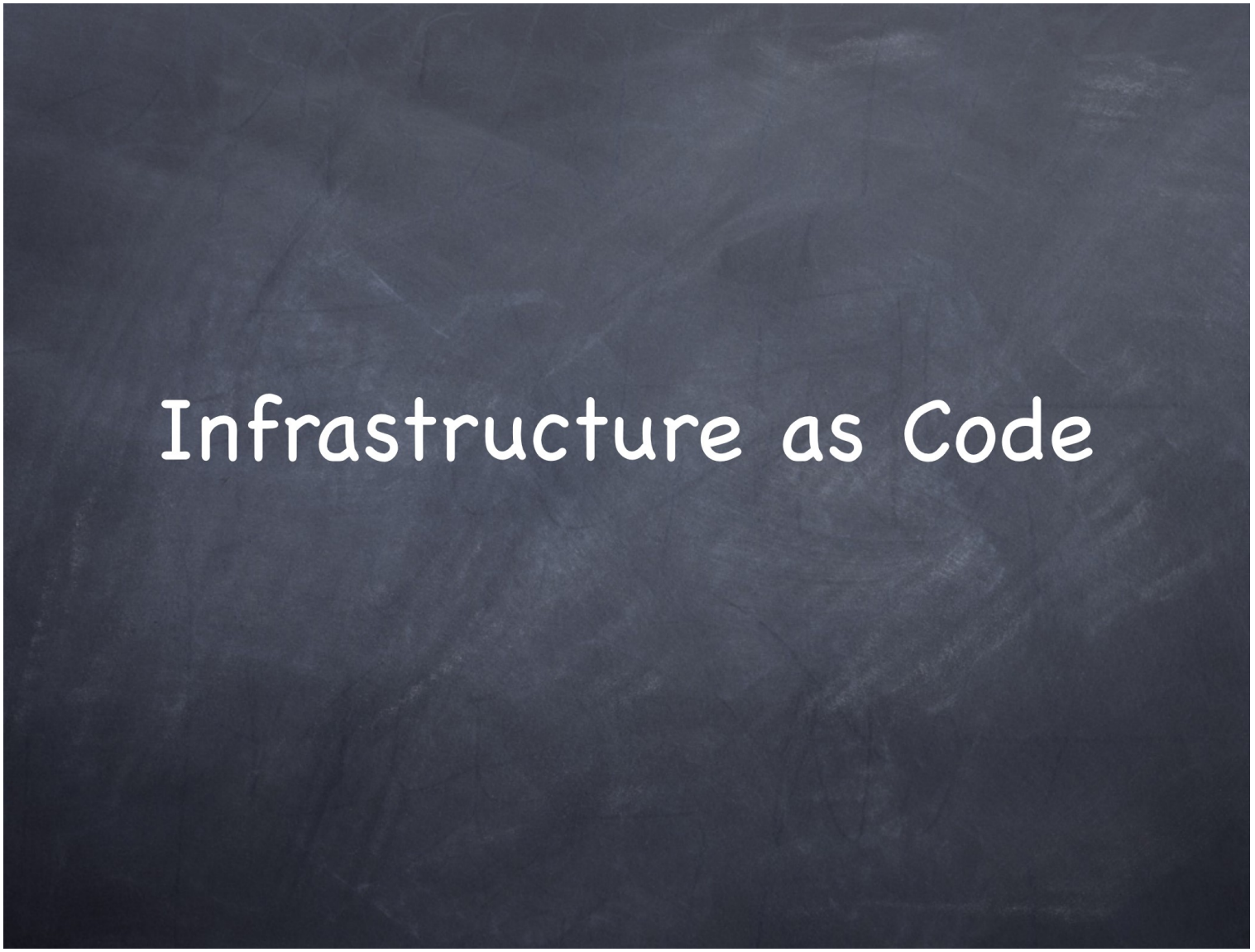
“The Cloud” & Containerization

Tuesday October 05, 2021

Section Overview

- “The Cloud”
- Containerization
- Assignment #02 - Current Event #01 - Due 05-Oct-2021
- Assignment #01 - Part 2 - Due 12-Oct-2021
- Exam #01 - 05-Oct-2021 - on material through 28-Nov-2021

Infrastructure as Code



Infrastructure as Code

What?

Instead of model of building a server "by hand" to meet the needs of what it is hosting to building it in an automated way

Why?

Automation

Replication

Abstraction

Isolation

How?

Well…

VMs +...

Virtual machines can be created automatically

- Create from template
- APIs to make scripted calls to VMware, AWS, etc

Configuration management utilities

- Chef, Puppet, Ansible, SaltStack, Terraform, et al.
- Addresses the customization needed for a given application - application settings, resourcing,

Puppet Example

```
$doc_root = "/var/www/example"

exec { 'apt-get update':
 command => '/usr/bin/apt-get update'
}

package { 'apache2':
 ensure  => "installed",
 require => Exec['apt-get update']
}

file { $doc_root:
 ensure => "directory",
 owner => "www-data",
 group => "www-data",
 mode => 644
}

file { "$doc_root/index.html":
   ensure => "present",
   source => "puppet:///modules/main/index.html",
   require => File[$doc_root]
}

file { "/etc/apache2/sites-available/000-default.conf":
   ensure => "present",
   content => template("main/vhost.erb"),
   notify => Service['apache2'],
   require => Package['apache2']
}

service { 'apache2':
   ensure => running,
   enable => true
}
```

Set variable…

Install updates

Install Apache

Create directory for webpage, set ownership

Pull in webpage index.html

Configure Apache

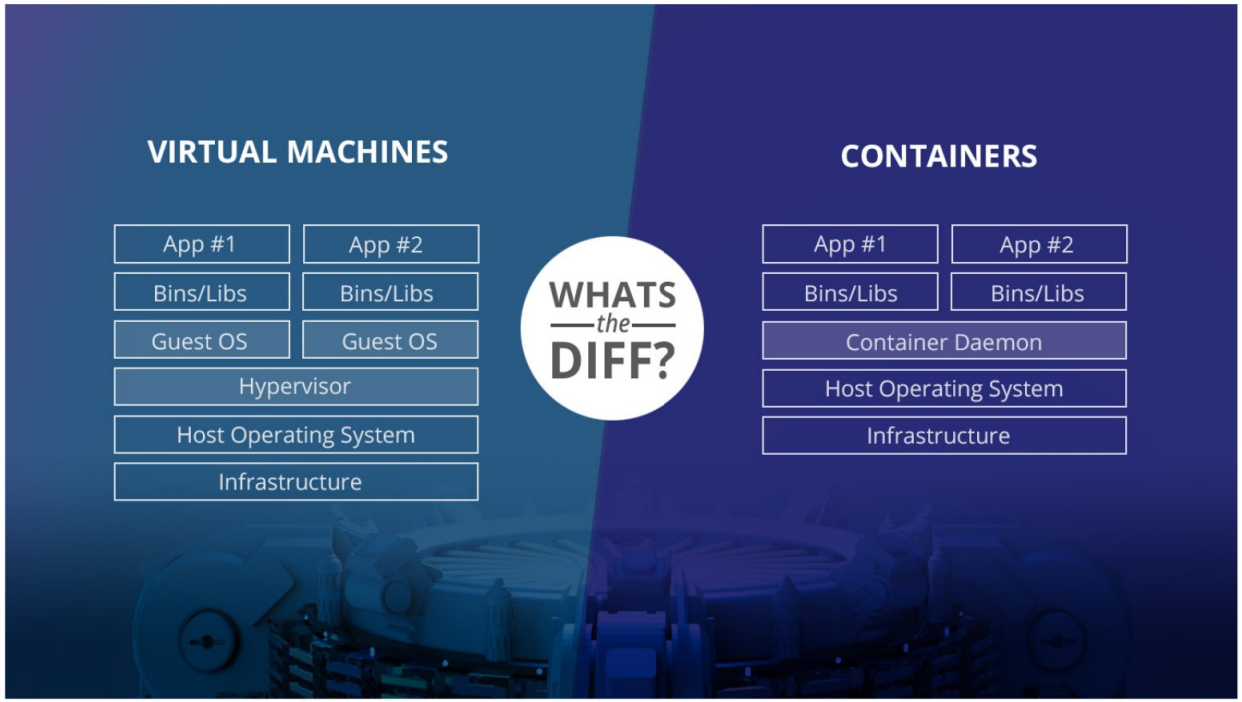
Define Apache service and set to be running

https://www.digitalocean.com/community/tutorials/configuration-management-101-writing-puppet-manifests

Containers

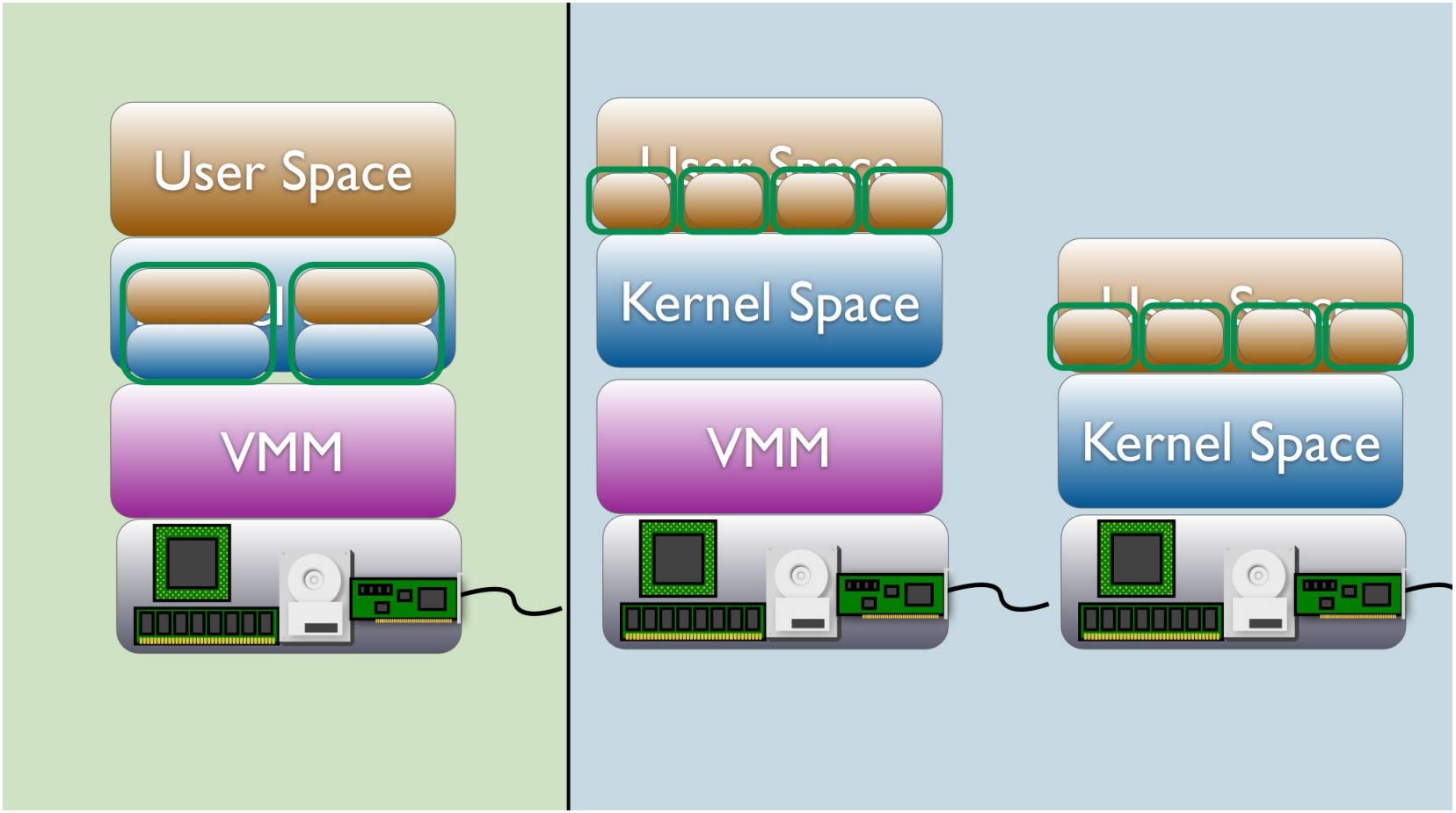
Think VMs, but more lightweight

Shared underlying OS, but isolation between running containers



https://www.backblaze.com/blog/vm-vs-containers/

VMs vs. Containers



Docker

The most prevalent container technology today

Open-source and paid versions

Docker containers can run on most ANY underlying infrastructure

Public repository of images for many standard services

Can define your own environment

Simple text definition files

Dockerfile Example

```
FROM debian:stretch
MAINTAINER Andras Mitzki <andras.mitzki@balabit.com>

RUN apt-get update -qq && apt-get install -y \
    wget \
    gnupg2

RUN wget -qO - https://download.opensuse.org/repositories/
home:/laszlo_budai:/syslog-ng/Debian_9.0/Release.key | apt-
key add -
RUN echo 'deb http://download.opensuse.org/repositories/
home:/laszlo_budai:/syslog-ng/Debian_9.0 ./' | tee --
append /etc/apt/sources.list.d/syslog-ng-obs.list

RUN apt-get update -qq && apt-get install -y \
    syslog-ng

ADD syslog-ng.conf /etc/syslog-ng/syslog-ng.conf

RUN find /usr/lib/ -name 'libjvm.so*' | xargs dirname | tee
--append /etc/ld.so.conf.d/openjdk-libjvm.conf
RUN ldconfig

EXPOSE 514/udp
EXPOSE 601/tcp
EXPOSE 6514/tcp

ENTRYPOINT ["/usr/sbin/syslog-ng", "-F"]
```

Starting point image

Install updates & wget and gnupg2

Download release key syslog-ng

Add source for syslog-ng

Install syslog-ng

Configure syslog-ng

Open needed network ports

Command to run to start syslog-ng

https://hub.docker.com/r/balabit/syslog-ng/dockerfile

Microservices

Breaking down a large application into small, interconnected segments

Example - Web shopping cart - each can be a separate element — a separate server/service/container — instead of a monolithic application

- Select item
- Update quantity
- Remove item
- Post status updates

- Checkout
  - Get identity
  - Shipping address
  - Payments
    - Credit card
    - PayPal

Containers support this approach

APIs provide connections between sub-services

Microservices - App Stack

Multiple microservices can be combined into an application stack that can be deployed at once, for example:

- Load balancer
- Multiple application servers
- Caching layer
- Database

Docker Compose - tool for deploying multi-container applications

- Uses standard YAML-based definition file

Docker Compose Example

```
services:
  app-service:
    image: universityofnewhampshire/apache-php56-mc:v3
    volumes:
      - ../../../www/:/var/www/html
      - ../../../src/:/var/www/src
      - ../../../vendor/:/var/www/vendor
    ports:
      - 80:80
    networks:
      - app-net
      - db-net
      - cache-net

  db-service:
    image: mariadb
    volumes:
      - ../../../dbdata:/rdbms/data
    environment:
      - MYSQL_ROOT_PASSWORD=asdfasdfasdfasdf
    command: docker-entrypoint.sh --datadir=/rdbms/data
    networks:
      - db-net
    expose:
      - 3306

  cache-service:
    image: memcached
    networks:
      - cache-net
    expose:
      - 11211

networks:
 app-net:
    driver_opts:
      subnet: 10.0.0.0/24
  db-net:
    driver_opts:
      subnet: 10.1.0.0/24
  cache-net:
    driver_opts:
      subnet: 10.2.0.0/24
```

Apache/PHP service

Exposed on network port 80

Uses these 3 "networks"

MariaDB service and configuration

Uses these "db-net" network on port 3306

Memcached service, config, and "cache-net" use

Defining internal "app", "db", and "cache" networks

Homework Assignments

- Assignment #02 - Current Event #01 - Due 05-Oct-2021
- Assignment #01 - Part 2 - Due 12-Oct-2021
- Readings (see myCourses)