# Docker Ex 1

Start a Docker machine from an image

Connect to it.

Start stop / explore

Then use Portainer:dmjrw

sudo docker run -d -p 9000:9000

-v /var/run/docker.sock:/var/run/docker.sock portainer/portainer

# Docker Ex 2

Create an empty directory

Create a dockerfile

FROM httpd:2.4

COPY ./public-html/ /usr/local/apache2/htdocs/

Add a /public-html dir to the image with an index.html file

Run the image – map port to 80:80

docker exec -it sad\_franklin /bin/bash

Explore

# Swagger

Create a service interface

swagger.io

Select tools / editor – online editor

Create a swagger specification for a Member service

Use the heoroku pet store as a template

Methods

get given and id returns an object representing a member

name

address

email

post adds a new member

put updates a member

delete deletes a member

# Mongo DB

docker run --name some-more-mongo -v /home/mjrw/Documents/BBC-Microservices/Resources:/mydata -d mongo

docker exec -it some-more-mongo bash

use testdb

foo = { name :"foo", location: { x: 3, y: 5 } }

bar = { name : "bar", location: { x: 1, y: 2, z: 4 } }

zep = { name : "zep", location: { x: -1, z: 4 } }

db.testdata.insert( foo )

db.testdata.insert( bar )

db.testdata.insert( zep )

db.testdata.find()

mongoimport --db dbsubs --collection subs --type csv --headerline --file /mydata/subscriptions.csv

db.subs.find({"Member\_Name" : /Michael.\*/});

Do some sample queries (see 4500)

Give them a query to figure out

\*\*\* Integrate with Node?

# CASSANDRA

docker run --name some-cassandra -d cassandra

docker exec -it some-cassandra bash

cqlsh

(May have to wait for this to work and then retry)

CREATE KEYSPACE subscriptions WITH REPLICATION = { 'class' :

'NetworkTopologyStrategy', 'datacenter1' : 2 };

(Network replicated)

use subscriptions;

CREATE TABLE subscriptions ( id varchar, name varchar, member\_id varchar, type varchar, renew boolean, startdate timestamp, enddate timestamp, PRIMARY KEY(id));

block

varchar, primary\_type varchar, description varchar, lat

float, lon float, PRIMARY KEY (id) );

Subscription\_Id,Member\_Name,Member\_Unique\_Id,Type,Renew,Start\_Date,End\_Date,Purchase\_Date,Paid\_Incl\_Tax

Docker Swarm

Connect Swarm to AWS

(Use pre-created AWS role).

Create Swarm – show resources being created

Connect to the swarm

Deploy the Composition onto the Swarm

\*\* Connect to a node and see what’s started?

db.testdata.find()