CS 218
Topics in Cloud Computing

DOCKER

Dept of Computer Science, San Jose State University

Lab #07

Written By
Dr. Melody Moh & Nathan Kong

Purpose:

Learn about Docker
Learn how to create an image
Learn how to create a container from an image
Learn how to transfer a container

Requirements:

Learn about Docker and answer questions
Install Docker in two environments
Create an image in a container
Run an image in two environments

Environment:

You will need to install Docker on your native environment
You will need to install Docker on Linux using a Virtual Machine

What to turn in?

Turn in the questions and answers to the lab in addition to the screenshots. The questions are in red. For the screenshots it must include your name in the terminal (or command prompt).

Background:

Watch the video about Docker:
https://www.youtube.com/watch?v=pGYAg7TMmp0

Answer the following questions:

1. Draw a picture and explain in your own words the software development to production cycle using Docker.
2. What does the Docker image contain?
3. Name at least three differences between Docker and virtual machines.
4. How many containers can you have?

(Optional readings/videos) Real World Examples of Docker:
https://www.youtube.com/watch?v=4ywzwCmML2Y
http://engineeringblog.yelp.com/2015/08/docker-in-the-real-world-at-yelp.html
http://blogs.gartner.com/richard-watson/ok-get-dockers-great/
https://www.docker.com/products/resources

LAB:

Download and install Docker in your native environment

1) Download Docker for your environment
   a) For Windows https://docs.docker.com/engine/installation/windows/
   b) For Mac https://docs.docker.com/engine/installation/mac/
   c) For Linux https://docs.docker.com/engine/installation/linux/
   d) Binaries https://docs.docker.com/engine/installation/binaries/
2) **Describe/Explain what the following items are:**
   a) Docker Machine
   b) Docker Engine
   c) Docker Compose
   d) Kitematic
3) Install docker and check the version
   a) Install docker
   b) Go to Docker and open it
      i) For Windows: Open the Docker Quickstart Terminal
   c) Find the versions. In the Docker terminal type:
      i) `$ docker --version`
      ii) `$ docker-compose --version`
      iii) `$ docker-machine --version`
      iv) `$ docker version`
   d) **Take a screenshot of all four versions listed above**

Create a HelloWorld

1) In the Docker terminal type "`docker ps`"
   a) This lists the containers that are running
2) Type "`docker run hello-world`"
   a) What did docker do (in your own words)?
3) Show all your containers
   a) Flag references: https://docs.docker.com/engine/reference/commandline
4) Run the hello world container again
5) Show all your containers
6) Type "`docker images`"
   a) This command shows images you have
   b) You should notice that you have one image and two containers
   c) Remove one of the containers
7) **Take a screenshot of your terminal. It should show one image and two containers (6b) as well as the removal of one of the containers (6c).**

Create a Node.js web application

1) Create the Node files
   a) Create a new directory where you want to keep your files
      i) In the directory create a file **package.json**
      ii) In the **package.json** add:

```
{
 "name": "docker_web_app",
 "version": "1.0.0",
 "description": "Node.js on Docker",
 "author": "First Last <first.last@example.com>",
 "main": "server.js",
 "scripts": { "start": "node server.js"},
 "dependencies": {"express": "^4.13.3" }
}
```

   b) Create a **server.js** file
      i) In the file add:

```
'use strict';

const express = require('express');

// Constants
const PORT = 8080;

// App
const app = express();
app.get('/', function (req, res) {
 res.send('Hello World\n');
});

app.listen(PORT);
console.log('Running on http://localhost:' + PORT);
```

2) Create a Docker file
   a) Create a file called **dockerfile** (this has no extension)
   b) Add to the **dockerfile**:

```
FROM node:argon

# Create app directory
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app

# Install app dependencies
COPY package.json /usr/src/app/
RUN npm install

# Bundle app source
COPY . /usr/src/app

EXPOSE 8080
CMD [ "npm", "start" ]
```

3) Build your image
   a) In the docker terminal type "`docker build -t [your name]/node-web-app .`" where [your name] is your first initial followed by your last name. E.g. "Bill Wong" would be "BWong/node-web-app".
   The command "`docker build -t imageFile .`" takes the dockerfile in the current directory and builds an image called `imageFile` on your local machine.
   b) Verify the image has been created. **Show the image.**
4) Run the image in a container
   a) Type "`docker run -p 49160:8080 -d [your name]/node-web-app`"
   b) **Show the container**
      i) This should show that Docker is mapped to the 8080 port inside the container as well as to the port 49160 on your machine
   c) **Show the logs**
      i) https://docs.docker.com/engine/reference/commandline/
5) Test the application
   a) Find your localhost ip
   b) Run curl command to see server information
      i) Type "`curl -i [ip]:49160`"
   c) In your favorite browser test **server.js**
      i) In the address bar of a browser type: "`[ip]:49160`"
   d) **Take screenshots showing the output for 5b) and 5c)**
6) Stop and remove the container
   a) Type "`docker stop [container name]`"
   b) Type "`docker rm [container name]`"
   c) Check to make sure the container is no longer there
      i) Type "`docker ps -a`"

7) Save the image
   a) Type "`docker save [image id] | gzip > [your name]_webapp.tar.gz`"
   b) This new file should be in your directory

Install a Virtual Machine (If you don't already have it installed)

Install Ubuntu 16.04 in Virtualbox

1) Set the name as first-last-docker, the Type to Linux, and the Version to Ubuntu (64 bit)
   a) Set Memory to at least 1024MB
   b) Set size to at least 12GB

   VirtualBox has now created a blank machine to install an OS

Install Docker on Ubuntu

1) Start the new VM and update your APT sources from terminal
   a) In terminal type "`sudo apt-get update`"
   b) Type "`sudo apt-get install apt-transport-https ca-certificates`"
   c) Type "`sudo apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D`"
   d) Go to `/etc/apt/sources.list.d`
   e) Look for a **`docker.list`** file
      i) If the file doesn't exist create it
      ii) Open the file and remove existing entries
      iii) Add an entry for your Ubuntu OS "deb https://apt.dockerproject.org/repo ubuntu-xenial main"
   f) Purge old repos
      i) Type "`sudo apt-get purge lxc-docker sudo apt-get purge lxc-docker`"
   g) Verify APT is pulling from the correct repository
      i) Type "`sudo apt-cache policy docker-engine`"
   h) Restart your VM
      i) Type "`sudo reboot`"
   i) Once Ubuntu restarts, update your APT package index from terminal
      i) Type "`sudo apt-get update`"
2) Install Docker
   a) Type "`sudo apt-get install docker-engine`"

Transfer container from native environment to Ubuntu and start container

1) Transfer image to Ubuntu
2) Start docker
   a) In terminal verify docker is installed type "`sudo docker version`"
   b) Type "`sudo service docker start`"
3) Load image from ***.tar.gz**
   a) In terminal type "`sudo docker load < [your name]_webapp.tar.gz`"
4) **Show newly loaded docker image**
5) Run the image in a container and show it in a browser
   a) Run the webapp image
   b) **Inspect the container and find the container ip**
   c) **Use the ip to run the webapp in a browser and take a screenshot of the browser**

Additional Training (Optional)
https://training.docker.com/self-paced-training

- Create a PDF document containing the answers to the test section.
- Name the file <Last Name>_<First Name>_Lab07.pdf
- Send the file to me via Slack

References:
https://www.youtube.com
https://Docs.docker.com
https://Docker.com
https://www.virtualbox.org/wiki/Downloads
http://www.ubuntu.com/
https://nodejs.org/en/docs/guides/nodejs-docker-webapp/