greatlearning
Learning for Life

Lab Brief

Course: Container and Microservices

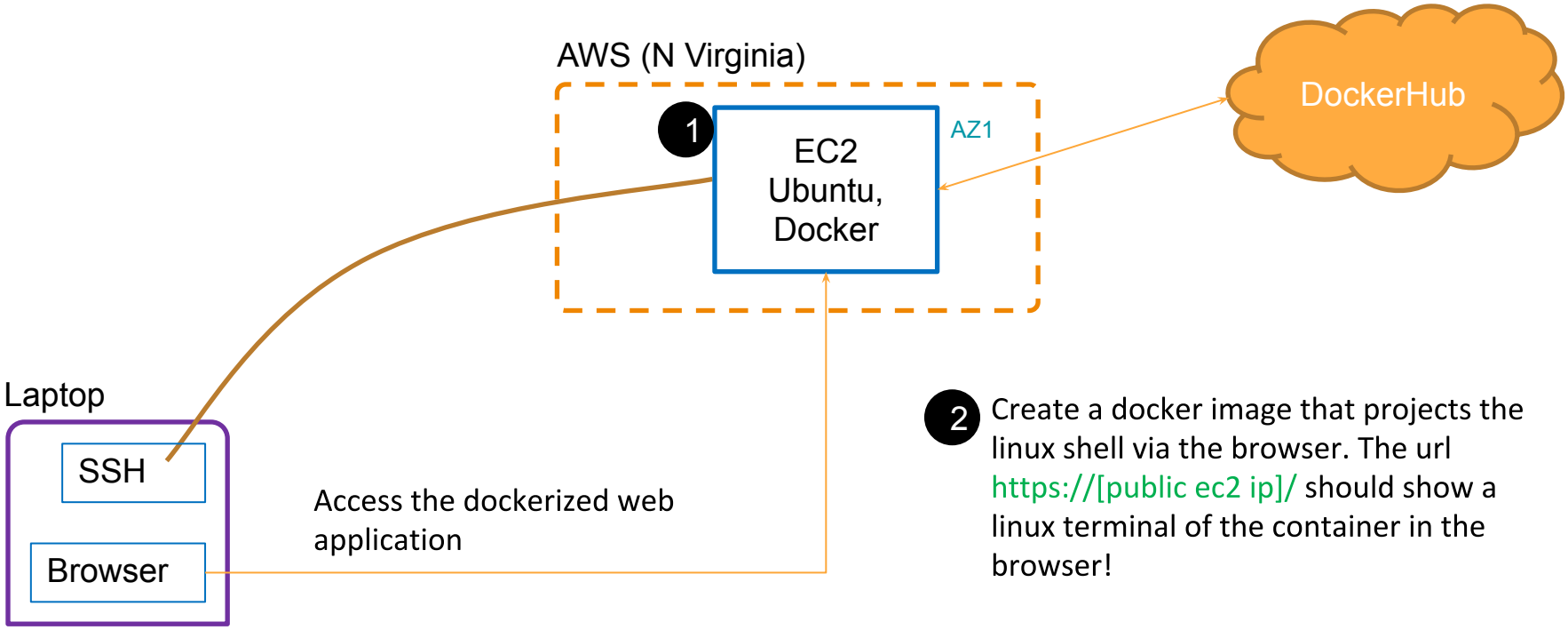
Docker | Images, containers, scripts

(Setup an instance with Docker, create multiple containers from existing images, create a custom image using Dockerfile)

Learning Outcomes

1. Working knowledge of EC2 instances with Ubuntu
2. Installing Docker from scratch
3. Working with images & containers
4. Understanding the Docker ecosystem

Final Goal



2 Create a docker image that projects the linux shell via the browser. The url https://[public ec2 ip]/ should show a linux terminal of the container in the browser!

What is needed?

1. AWS Account Credentials

2. EC2 Instances (Linux)

3. Terminal window for SSH

How to do it? - 1

Part A

1. Ensure your region is set to "N Virginia"
2. Create 1 EC2 instance using the 7 step workflow
   a) "T2 Small" instance with "Ubuntu 16.04 LTS"
   b) Download a new PEM file
   c) SSH to the instance
3. Create a new SG to open port 80 and assign to the EC2 instance
4. Install Docker from scratch (reference 1)

Part B

1. Run a tomcat instance with the following settings (reference 2)
   b) Should be JRE8
   c) Default tomcat port 8080 should be mapped to port 80 of host
   d) Be able to access the default tomcat page from the browser

How to do it? - 2

Part C

1. Stop the previous container running only Tomcat
2. To create a custom image of a java web app using Tomcat8 (reference 3)
   a) Download the HelloWorld.war in the EC2 instance
   b) Create a file by the name "Dockerfile", this file does not have any extension
   c) Create the custom image using the appropriate command
3. Launch a container from your custom image and access the application from the browser using the URL http://[ec2 public IP]/HelloWorld

Part D

1. Stop the previous container that is running the Java web application in Tomcat
2. Create a custom image with a static website running in httpd (reference 4)

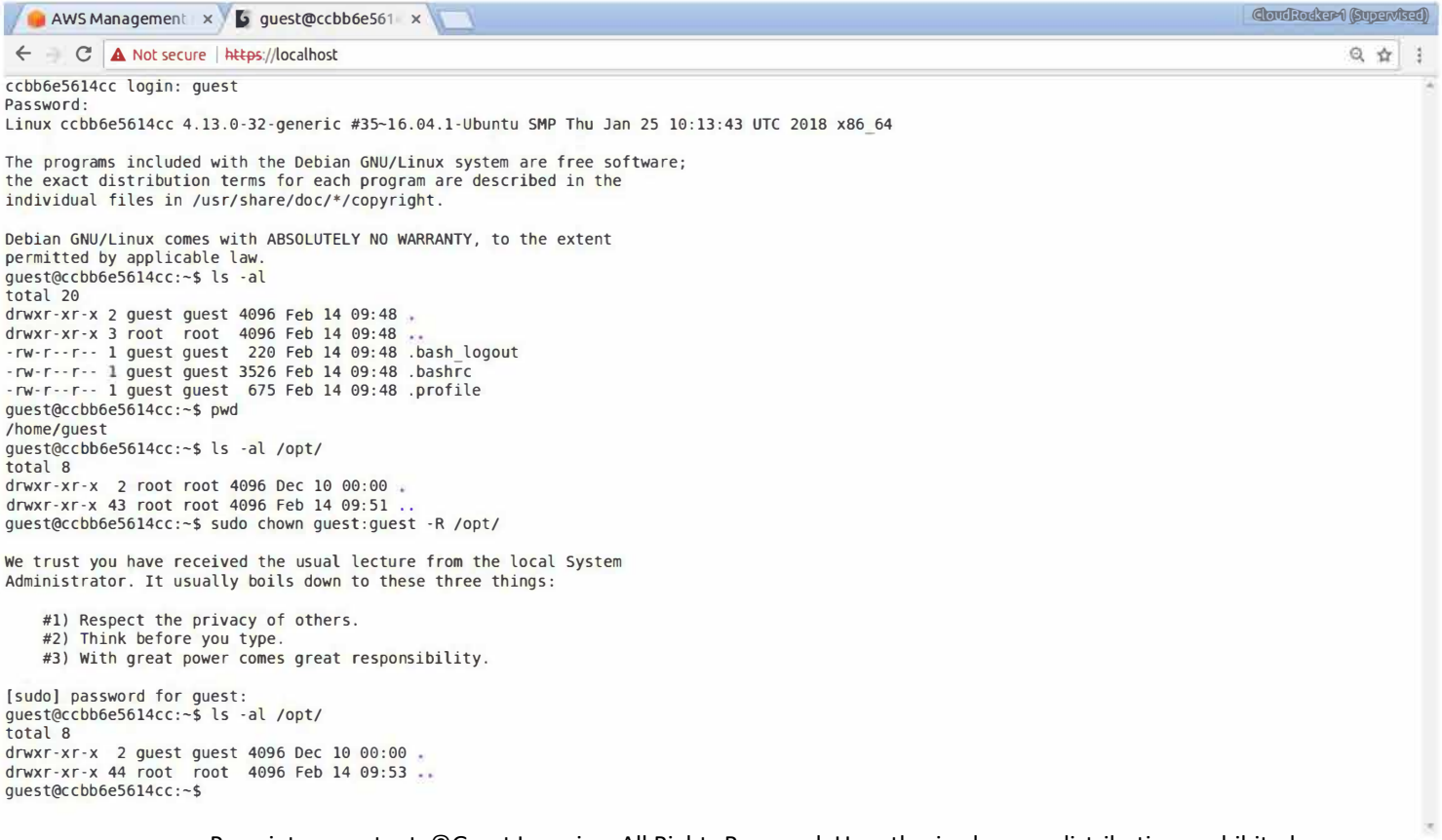
How to do it? – 3 (advanced)

Part E

mkdir /opt/siab
cd /opt/siab
wget
https://storage.googleapis.com/skl-training/aws-codelabs/aws-intro/SIABDockerfile
https://storage.googleapis.com/skl-training/aws-codelabs/aws-intro/SIABDockerfile

Follow the instructions given in the dockerfile. Your final output should look like below
-

How to do it? – 3 (advanced)



Command reference - 1

```
sudo apt update
sudo apt install docker.io
sudo docker version
sudo usermod -a -G docker ubuntu

#Ensure you restart the shell (close the terminal window and SSH again)
#After SSH type the below command (notice there is no sudo)

docker version

#The above command should show the client and server versions and other details
#The installation is now successful

#For further exercises do the following
sudo chown ubuntu:ubuntu -R /opt
cd /opt
```

Command reference - 2

docker images

docker ps -a

docker run --rm busybox:latest /bin/echo "Hello world"

docker run -d -p 80:8080 tomcat:jre8

docker rm [container id]

docker rmi [image id]

docker build -t [your image name without spaces] .

docker stop [first 3 characters of your container ID]

docker start [first 3 characters of your container ID]

Command reference - 3

mkdir /opt/HelloWorld
cd /opt/HelloWorld

wget https://storage.googleapis.com/skl-training/aws-codelabs/aws-intro/HelloWorld.war

The content of the Dockerfile is given below

```
FROM tomcat:jre8
MAINTAINER [Put in your name here]
COPY HelloWorld.war /usr/local/tomcat/webapps/
```

Command reference - 4

mkdir /opt/smartfin
cd /opt/smartfin

wget https://storage.googleapis.com/skl-training/docker/smart_finance.tar.gz
tar -zxf smart_finance.tar.gz

The partial content of the Dockerfile is given below

docker build -t smartfin .
docker run --rm -d -p 80:80 smartfin

FROM httpd
MAINTAINER [Put in your name here]
run mv /usr/local/apache2/htdocs/index.html /usr/local/apache2/htdocs/index.html.old
[TBD - what command needs to be used to push the site files]

greatlearning
Learning for Life

Resource Clean-Up

1. Cloud is always **pay per use model** and all resources/services that we consume are chargeable. Cleaning up when you've completed your lab or project is always necessary. This is true whether you're doing a lab or implementing a project at your workplace.

2. **After completing with the lab, make sure to delete each resource created in the reverse chronological order.**

3. Check resources in each cloud region that you have worked on before logging off.

4. Since the dashboard doesn't show cross-region resources, it is up to you to find and delete them.