Try it out objective

Use this hands-on to learn how to install docker, creating multiple containers from existing images and creating a custom image using a docker file.

The goal

Following are the goals of this hands-on:

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

A. Hands-on: Launch an Instance

1. Open the EC2 management console at https://console.aws.amazon.com/ec2/ (you will be required to sign in)
2. Change the region to **N Virginia** (if it is not already selected).
3. From the EC2 management console, click on **Launch Instance**.
4. In **Name and Tags** field, type out any name of your instance
5. Then choose an Amazon Machine Image (AMI) page that displays a list of basic configurations. Click on **Ubuntu** and select **Ubuntu Server 20.04 LTS**.
6. On the **Choose an Instance Type** page, select the **t2.small** instance type.
7. Under **Key pair (login)**, click on **create a new key pair**
8. Enter the key pair name of your choice, and then click on **Create key pair.**
9. Under **Network settings**, click on **Edit.**
10. **Port 22** for SSH should already be present in the security group. Click on **Add security group rule** and create the rule with the following parameters
    a. Type : HTTP
    b. Port Range : 80
    c. Source Type : Anywhere

11. At last, click on **Launch instance.**

B. Hands-on: Installation of docker

1. Select your running instance by clicking on the checkbox present on the left side.

2. Click on **Connect** button.

3. Click on the **Connect** button again present at the bottom of the page.
   NOTE: You must be under 'EC2 Instance Connect'

4. This will land you to the EC2 Instance Connect browser terminal.

5. Execute the below commands one by one in the terminal window.
   NOTE: You are already in your EC2 server, you need not SSH again.

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

NOTE: Ensure you restart the shell (Refresh the page) .

3. Now , type the below command (notice there is no sudo)

```
docker version
```

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

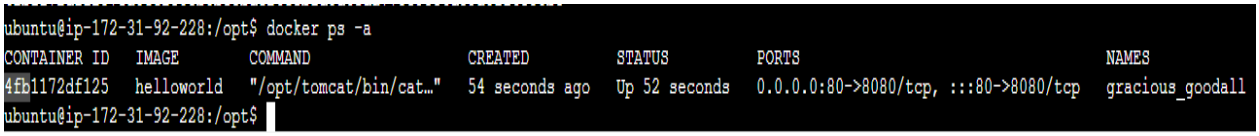
C. Hands-on: Working with images and containers

1. Type out the below commands in the terminal window-

```
sudo chown ubuntu:ubuntu -R /opt
cd /opt
docker images
docker run --rm busybox:latest /bin/echo "Hello world"
wget https://d6opu47qoi4ee.cloudfront.net/project-container/Dockerfile
docker build -t helloworld .
docker run -d -p 80:8080 helloworld
docker ps -a
```

NOTE: Please note down the first three characters of the container's id. Below commands can be used to remove , build , stop and start the containers and images . Fill the container and image ids as required in the respective Placeholders.
Please refer to the screenshot attached below.

```
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]
```



2. Hit the **public IP address** of the running instance on a new web browser tab to see the Apache Tomcat web page.

3. Stop the running container and remove it using the below commands.

```
docker stop [first 3 characters of your container ID]

docker rm [container id]
```

D. Hands-on: Shell In A Box (Advanced)

Objective

Use this hands-on part to learn how to deploy a web based terminal using a container.

The goal

Following are the goals of this hands-on:

1. Pulling the debian OS image.
2. Creating the user guest and update/install dependencies in the bash.
3. Accessing the browser from the local host.
4. Logging into the webpage as a guest user.

Steps to be followed

1. Select your running instance by clicking on the checkbox present on the left side.

2. Click on **Connect** button.

3. Click on the **Connect** button again present at the bottom of the page.
   NOTE: You must be under 'EC2 Instance Connect'

4. This will take you to the EC2 Instance Connect browser terminal.

5. Execute the below 4 commands one by one in the terminal .

```
sudo apt update
sudo mkdir /opt/siab
cd /opt/siab
sudo wget https://d6opu47qoi4ee.cloudfront.net/SIABDockerfile
```

6. Pull the debian OS image by entering the below 2 commands in the terminal one at a time

```
docker pull debian
docker images
```

7. Start and exec in the container (siab = shell in a box) by entering the following 3 commands

```
docker run --name siabdebian -d -it debian tail -f /dev/null
docker ps -a
docker exec -it siabdebian bash
```

8. Now, create the user guest & update/install dependencies in the bash.

```
apt update
apt upgrade -y
apt install -y apt-utils procps sudo wget curl
adduser guest
passwd guest  -> set to welcome [Optional step]
New password
Retype new password:
Changing the username information for guest
Enter the new value , or press ENTER for the default

        Enter full name []:  [Optional step]          // Press enter through to skip
        Room Number []:   [Optional step]
        Work Phone []:       [Optional step]
        Home Phone []:       [Optional step]
        Other []:            [Optional step]

Is the information correct? [Y/n]                  // To save the information , Press Y and hit enter.
```

```
adduser guest sudo
exit
```

9. The below 3 commands will create the Base image .Please enter your name in the Author Placeholder.

```
docker stop siabdebian
docker commit -m "Debian with guest user" -a "[Author]" siabdebian debian-siab-base:v1
docker images
```

10. Create a security group and open port 443 for the https protocol set to Anywhere . Attach the security group to the ec2 instance . To perform this, follow the below instructions .

11. Select your ec2 instance → Go to Actions → Security → Change Security Groups → Click on the Security group to be attached to your ec2 instance → Click on Add Security Group → Click on save .

12. Build and run the image as follows (ignore the red lines that may appear in the output)

```
docker build -f SIABDockerfile -t siab .
docker run -d -p 443:4200 --name siab -i siab
```

13. Access the shell from a new browser tab using the below URL
https://<public Ip of ec2 instance>

14. As your connection is not Private , click on advanced and proceed to “your_public_ip_of _your _EC2_instance(unsafe)”.

15. Now ,you are in the Shell In A Box !! . To login , enter the Username and Password created in the Bash.

```
Login : guest
Password :
guest@XXXXXXXXXXXX:~$ sudo su

[sudo] password for guest:

root@XXXXXXXXXXXX:/home/guest# exit
guest@XXXXXXXXXXXX:~$
```

16. Go back to the terminal , stop and remove the container.

```
docker stop [first 3 characters of your container ID]
docker rm [first 3 characters of your container ID]
```

17. Enter exit to go out of the ec2 instance.

D. Hands-on: Terminating / deleting an instance

1. Go back to the browser tab EC2 management console

2. Click on Instances in the left navigation

3. Click on the checkbox to the left of your running instance (no action is necessary if the checkbox is already selected)

4. Click on the Instance state dropdown towards the top right side of the screen

5. Select Terminate instance option

6. Click on the Terminate button on the confirmation popup window

7. The instance will show in Shutting down status and a few moments later will show as Terminated

8. Terminated instances do not attract any costs and will be auto removed from the instance listing in about 2 hours (or less)

9. Go back to the Lab environment and reset the Lab work area using the reset button in the top right of the screen to avoid any unintended credit charges .