**Lab 07**

**Laboratory Exercise**

**Part 1: Manage docker container and image**

**LAB EXERCISE**

This LAB exercise demonstrates the management of images directly using docker command.

**Time to Complete**

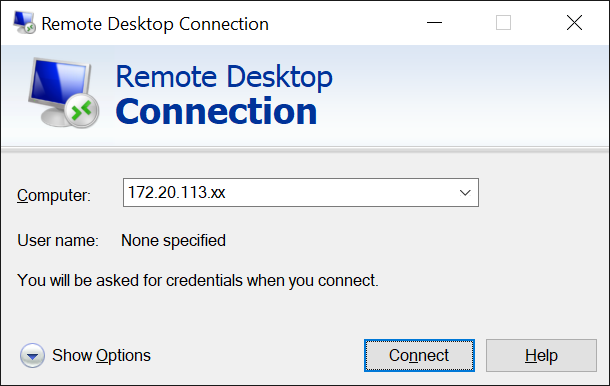
Approximately 30 Minutes

**What You Need**

* Lab 7 - Part 1 to be completed successfully.
* Docker packages are already installed in the ubuntu VM.

From your machine logged-in to RP VPN, run Remote Desktop Connection to connect to the ubuntu Linux Virtual Machine (VM). Please login based on your assigned VM as shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Name** | **VM** | **IP Address** | User Name | Password |
| 1 | ABDUL SALIM BIN ABDUL RASHITH | LABC03 - 172.20.115.50 | 172.20.115.50 | dockeradm | docker!2 |
| 2 | CASPER LEOW YU HAN (LIAO YU HANG) | LABC03 - 172.20.115.51 | 172.20.115.51 | dockeradm | docker!2 |
| 3 | CHAN JUN ZHI, GLENN | LABC03 - 172.20.115.52 | 172.20.115.52 | dockeradm | docker!2 |
| 4 | CHIA WAI TAT | LABC03 - 172.20.115.53 | 172.20.115.53 | dockeradm | docker!2 |
| 5 | HOI WAI TECK | LABC03 - 172.20.115.54 | 172.20.115.54 | dockeradm | docker!2 |
| 6 | KOH JIN CAI DAEMIAN | LABC03 - 172.20.115.55 | 172.20.115.55 | dockeradm | docker!2 |
| 7 | KYAW KYAW OO | LABC03 - 172.20.115.56 | 172.20.115.56 | dockeradm | docker!2 |
| 8 | LUM YOKE FAI | LABC03 - 172.20.115.57 | 172.20.115.57 | dockeradm | docker!2 |
| 9 | MUHAMMAD FADHLI BIN MOHAMED NOOR | LABC03 - 172.20.115.58 | 172.20.115.58 | dockeradm | docker!2 |
| 10 | MUHAMMAD HILMEE BIN MD ALI | LABC03 - 172.20.115.59 | 172.20.115.59 | dockeradm | docker!2 |
| 11 | NG SAY WEE | LABC03 - 172.20.115.60 | 172.20.115.60 | dockeradm | docker!2 |
| 12 | NGUI WEILY | LABC03 - 172.20.115.61 | 172.20.115.61 | dockeradm | docker!2 |
| 13 | NU'MAN HARITH BIN NORRAIMI | LABC03 - 172.20.115.62 | 172.20.115.62 | dockeradm | docker!2 |
| 14 | RULY JANUAR FACHMI | LABC03 - 172.20.115.63 | 172.20.115.63 | dockeradm | docker!2 |
| 15 | SEAH SHIH WEI GEROME | LABC03 - 172.20.115.64 | 172.20.115.64 | dockeradm | docker!2 |
| 16 | SEAN CHENG ZHI WEI | LABC03 - 172.20.115.65 | 172.20.115.65 | dockeradm | docker!2 |
| 17 | SEY KOK SIONG | LABC03 - 172.20.115.66 | 172.20.115.66 | dockeradm | docker!2 |
| 18 | TAN JOON YEE DOUGLAS | LABC03 - 172.20.115.67 | 172.20.115.67 | dockeradm | docker!2 |
| 19 | WU WAI TENG VANESSA | LABC03 - 172.20.115.68 | 172.20.115.68 | dockeradm | docker!2 |
| 20 | YAP KOON SING | LABC03 - 172.20.115.69 | 172.20.115.69 | dockeradm | docker!2 |
| 21 | YE CHENG LIM | LABC03 - 172.20.115.70 | 172.20.115.70 | dockeradm | docker!2 |
| 22 | SHAIFUL BIN ABDUL KARIM | LABC03 - 172.20.115.71 | 172.20.115.71 | dockeradm | docker!2 |
| 23 | CHAI RU YI | LABC03 - 172.20.115.72 | 172.20.115.72 | dockeradm | docker!2 |
| 24 | JWAY HWEE LING JULIE | LABC03 - 172.20.115.73 | 172.20.115.73 | dockeradm | docker!2 |
| 25 | SAMANTHA TEO XING YEE | LABC03 - 172.20.115.74 | 172.20.115.74 | dockeradm | docker!2 |
| 26 | ZIL AZZA HILMIAH BINTE RADUAN | LABC03 - 172.20.115.75 | 172.20.115.75 | dockeradm | docker!2 |



Replace xx with the IP address of the VM that you have been assigned.

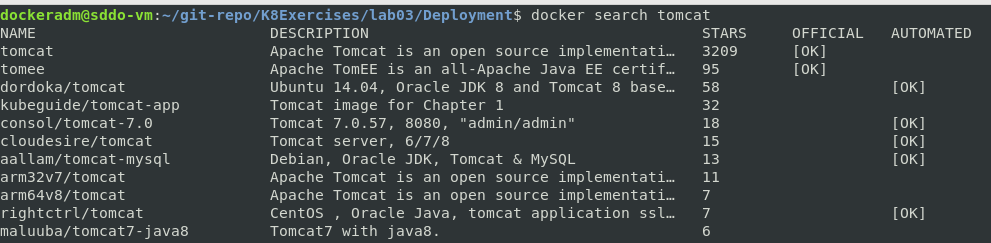
**Build a Docker container**

1. Check docker versions

docker version

1. To search images on docker hub

docker search <word>



1. To list images on local system

docker images

or

docker image ls

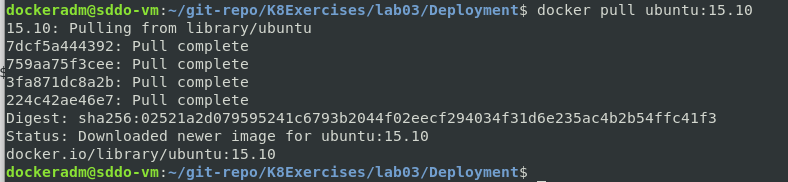
4. To pull a image from docker hub to local system.

docker pull <image name>

Noted: If not tag is specified, the tag ”latest” is used.

For example:

docker pull ubuntu:15.10

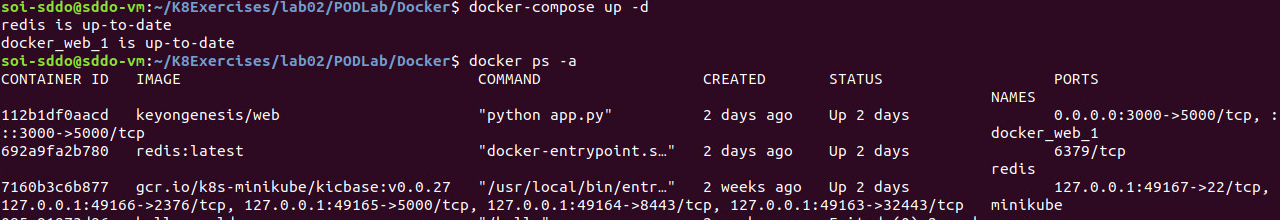


1. Check for the local images on the local system.

docker images

6. Check the container using the below command.

docker ps -a



6. Create a new container running on ubuntu OS 15.10

docker run --name testos10 -it ubuntu:15.10 /bin/sh

This example runs a container named testos10 using the ubuntu:15.10 image. The -it instructs Docker to allocate a pseudo-TTY connected to the container’s stdin; creating an interactive bash shell in the container. Once exit from the shell, the testos container stops.

Check the container is down

docker ps

**7.** Create another container using docker create. It does not start the container.

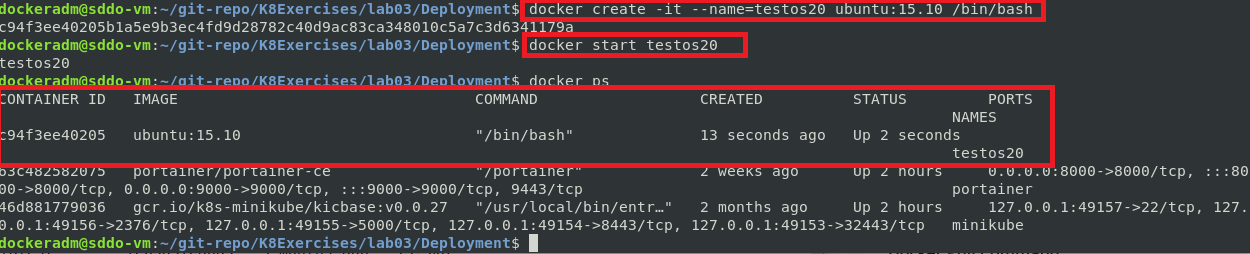
docker create -it –-name=testos20 ubuntu:15.10 /bin/bash

Start the container

docker start testos20

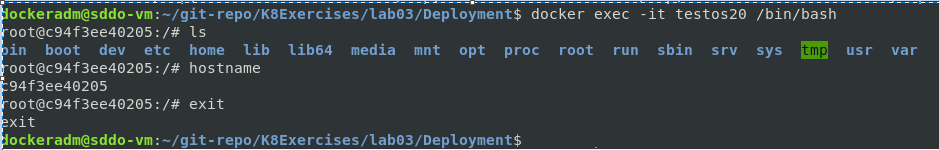
Check the container is up and running.

docker ps



8. To access the shell of the container

docker exec -it testos20 /bin/bash



9. What is the difference between docker create and docker run?

The docker create command creates a writeable container layer over the specified image and prepares it for running the specified command. This is similar to docker run -d except the container is never started. Use the docker start <container\_id> command to start the container at any point.

**Part 2: Deploy a Web Application**

**LAB EXERCISE**

This LAB exercise demonstrates the concept of deploying web application containers and be accessible by host OS.

**Time to Complete**

Approximately 30 Minutes

1. Create and run a tomcat container.

docker run -d -it -p 3500:8080 –-name=web-app tomcat:8.0 /bin/sh -c “catalina.sh run”

Do you know what does each parameter appear in the above command do?

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Use a browser and access localhost via port 3500
2. On Host OS, create an empty file

touch /testfile

1. Copy /testfile from local host OS to the running container web-app

docker cp /testfile web-app:/

1. Create a new image for this container

docker commit web-app new-web-app

1. Check for the new image on local system

docker images

1. Now, remove the existing container web-app

docker rm -f web-app

Do you know why -f is needed in the above command?

1. Create a new container using the new image “new-web-app”

docker run -d -it -–name=web-app10 new-web-app /bin/sh -c “catalina.sh run”

1. Now, check for the existing of the file /testfile

docker exec -it web-app10 /bin/bash

In the container, check for the testfile

ls /testfile

The file /testfile exists.

1. Now, remove the existing container web-app

docker rm -f new-web-app

1. Create a new container using the old image “tomcat:8.0”

docker run -d -it -–name=web-app tomcat:8.0 /bin/sh -c “catalina.sh run”

1. Now, check for the existing of the file /testfile

docker exec -it web-app /bin/bash

In the container, check for the testfile

ls /testfile

Does the /testfile exists?

Do you know why is it so?

1. Check the local images

docker images

1. Remove the newly created image

docker rmi new-web-app

**Part 3: Check details of container**

**LAB EXERCISE**

This LAB exercise demonstrates the concept of how to look for details of a container.

**Time to Complete**

Approximately 5 Minutes

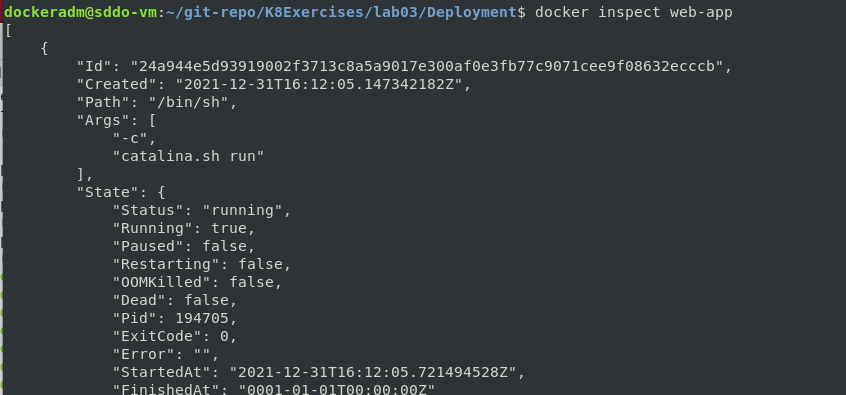
**What You Need**

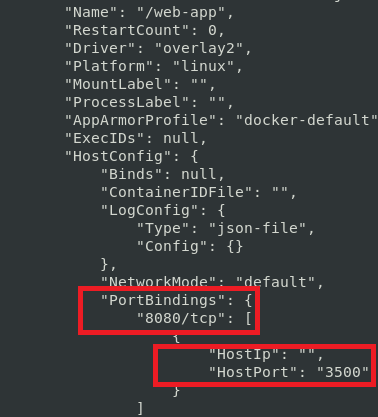
* Part 2 to be completed successfully.

1. If the web app is remove, re-deploy the Application

docker run -d -it -–name=web-app tomcat:8.0 /bin/sh -c “catalina.sh run”

2. Check the details of the container   
 docker inspect web-app



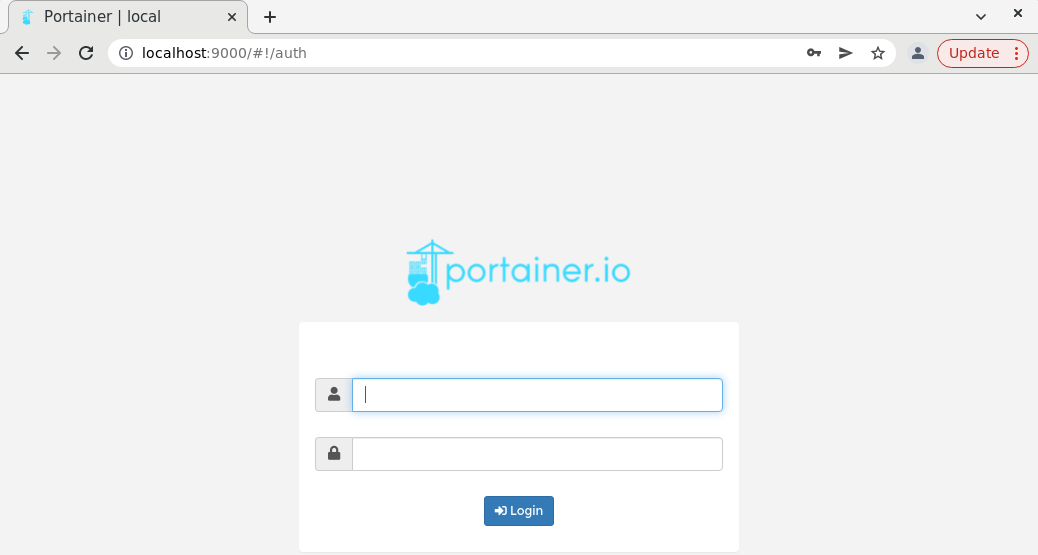


**Part 4: Check details of container via Portainer**

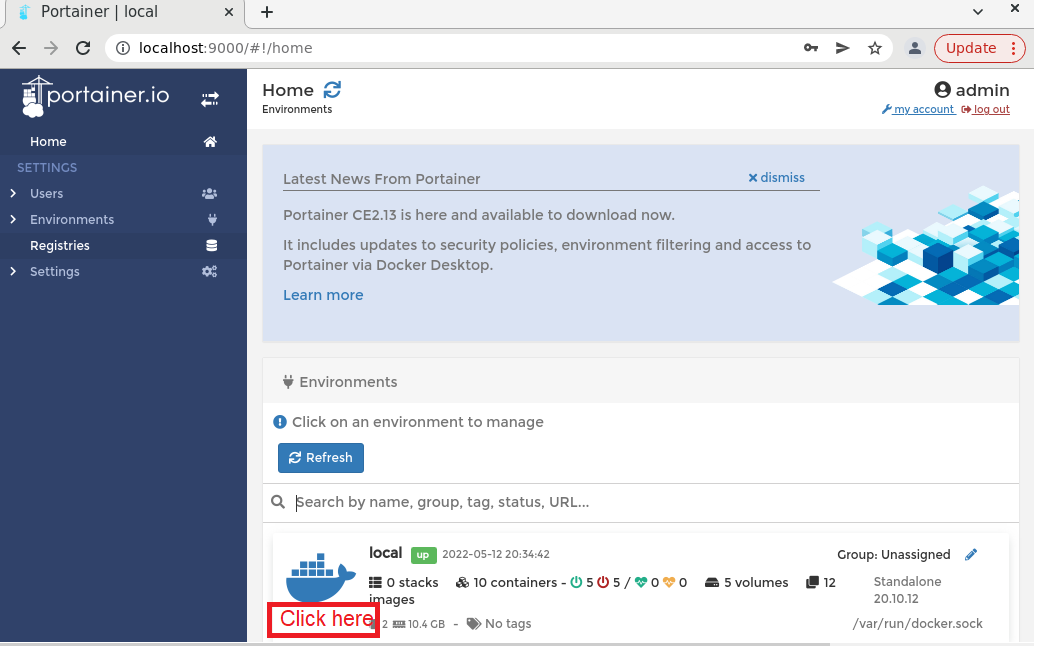
Access <http://localhost:9000>

Username: admin

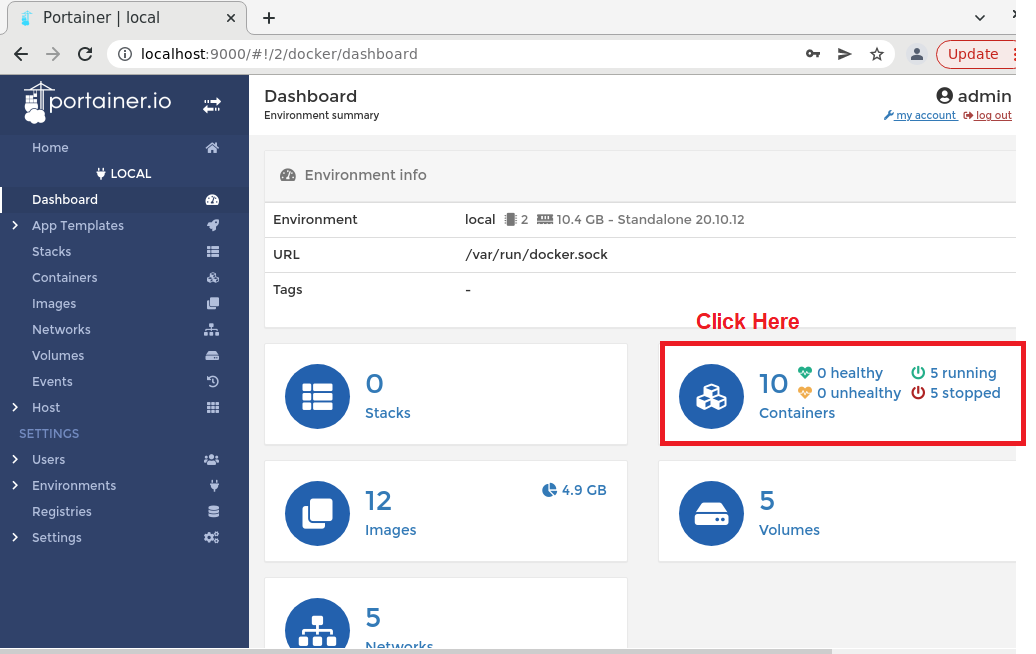
Password: admin!234



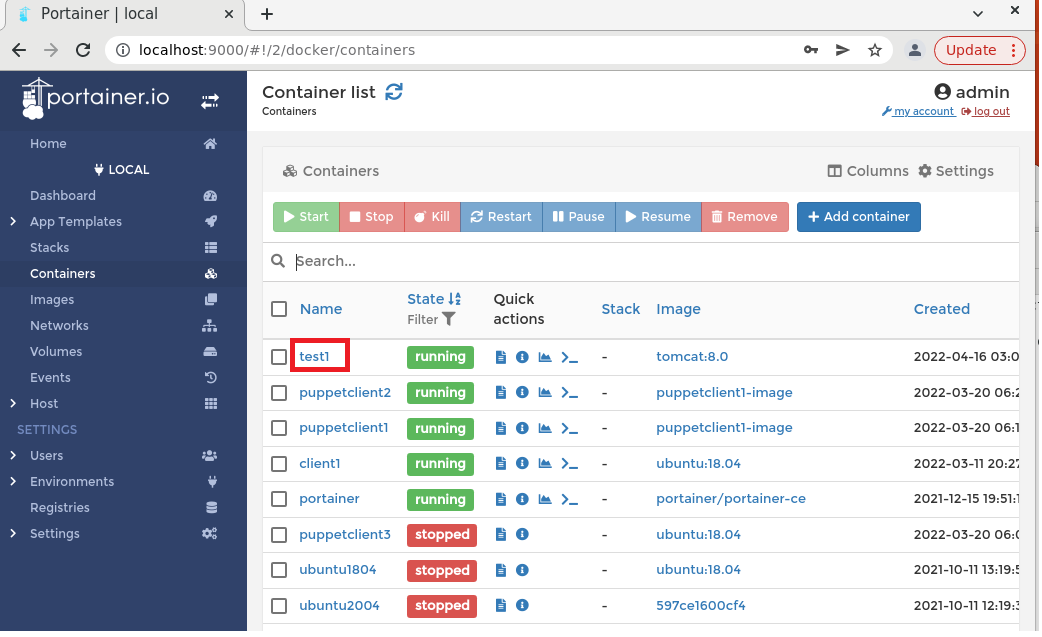
After login:



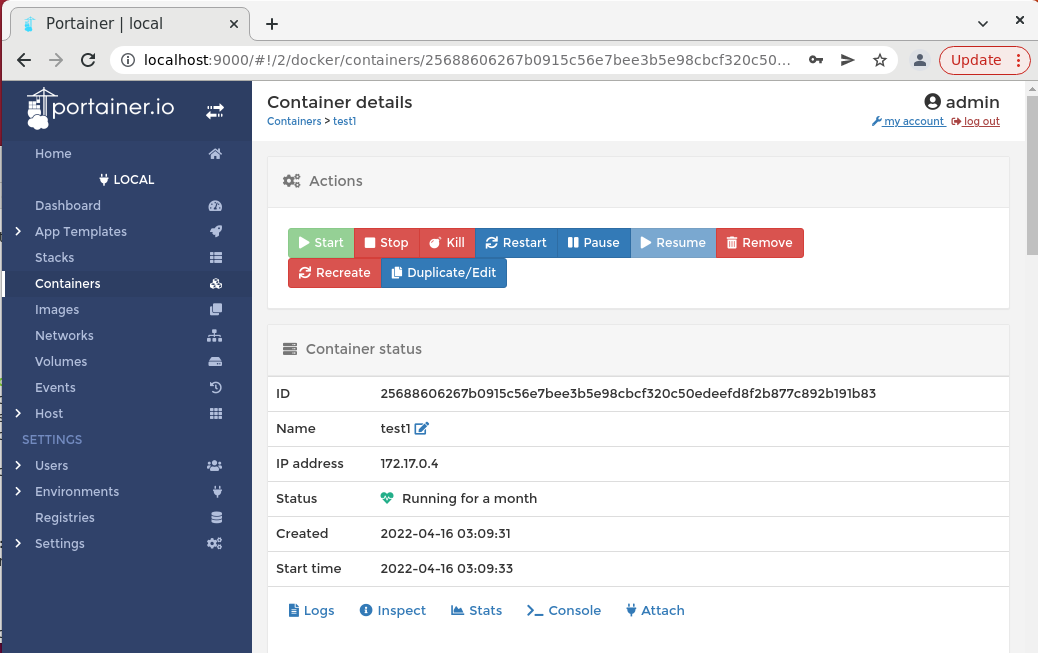
Click on Container icon as shown below:



Click on one of the container for more details:



The detail of the selected container is shown.



**------------------------------------------------ End of Lab --------------------------------------------------------**