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

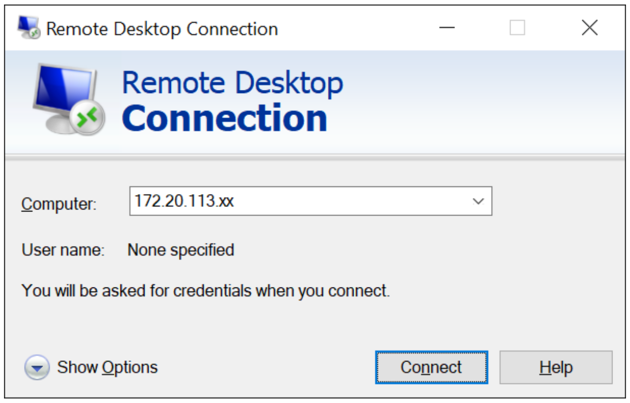
What You Need

- 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



Computer: 172.20.113.xx

User name: None specified

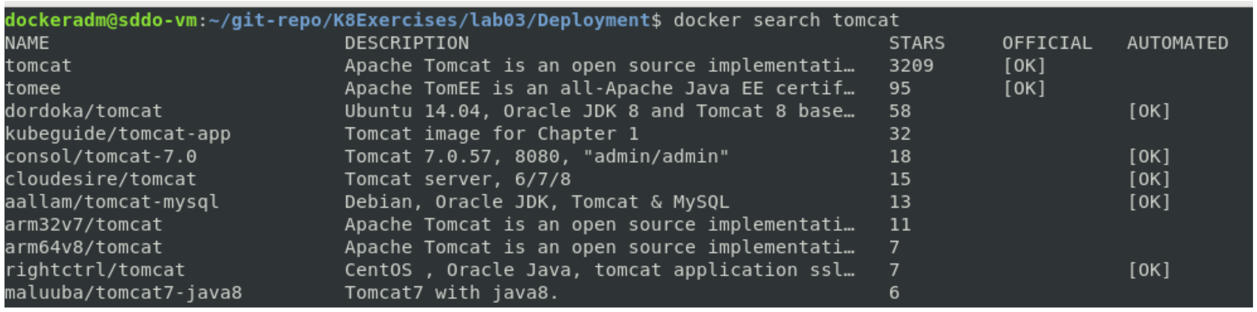
You will be asked for credentials when you connect.

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

Build a Docker container

1. Check docker versions
   `docker version`

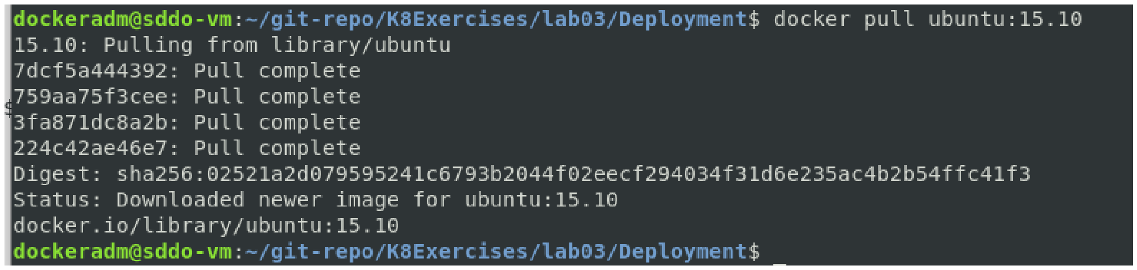
2. To search images on docker hub
   `docker search <word>`



```
dockeradm@sddo-vm:~/git-repo/K8Exercises/lab03/Deployment$ docker search tomcat
NAME                          DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
tomcat                        Apache Tomcat is an open source implementati…   3209      [OK]
tomee                         Apache TomEE is an all-Apache Java EE certif…   95        [OK]
dordoka/tomcat                Ubuntu 14.04, Oracle JDK 8 and Tomcat 8 base…   58                   [OK]
kubeguide/tomcat-app          Tomcat image for Chapter 1                      32
consol/tomcat-7.0             Tomcat 7.0.57, 8080, "admin/admin"              18                   [OK]
cloudesire/tomcat             Tomcat server, 6/7/8                            15                   [OK]
aallam/tomcat-mysql           Debian, Oracle JDK, Tomcat & MySQL              13                   [OK]
arm32v7/tomcat                Apache Tomcat is an open source implementati…   11
arm64v8/tomcat                Apache Tomcat is an open source implementati…   7
rightctrl/tomcat              CentOS , Oracle Java, tomcat application ssl…   7                    [OK]
maluuba/tomcat7-java8         Tomcat7 with java8.                             6
```

3. 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.

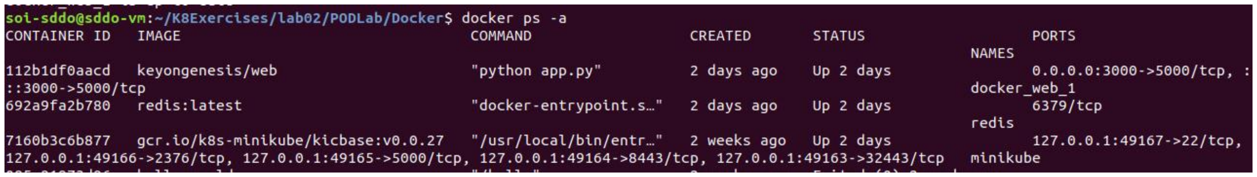
   For example:
   `docker pull ubuntu:15.10`



```
dockeradm@sddo-vm:~/git-repo/K8Exercises/lab03/Deployment$ docker pull ubuntu:15.10
15.10: Pulling from library/ubuntu
7dcf5a444392: Pull complete
759aa75f3cee: Pull complete
3fa871dc8a2b: Pull complete
224c42ae46e7: Pull complete
Digest: sha256:02521a2d079595241c6793b2044f02eecf294034f31d6e235ac4b2b54ffc41f3
Status: Downloaded newer image for ubuntu:15.10
docker.io/library/ubuntu:15.10
dockeradm@sddo-vm:~/git-repo/K8Exercises/lab03/Deployment$ _
```

5. Check for the local images on the local system.
   `docker images`

6. Check the container using the below command.
   `docker ps -a`



```
soi-sddo@sddo-vm:~/K8Exercises/lab02/PODLab/Docker$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS                 PORTS
                                                                                                        NAMES
112b1df0aacd   keyongenesis/web                     "python app.py"          2 days ago     Up 2 days              0.0.0.0:3000->5000/tcp, :
::3000->5000/tcp                                                                                        docker_web_1
692a9fa2b780   redis:latest                         "docker-entrypoint.s…"   2 days ago     Up 2 days              6379/tcp
                                                                                                        redis
7160b3c6b877   gcr.io/k8s-minikube/kicbase:v0.0.27   "/usr/local/bin/entr…"   2 weeks ago    Up 2 days              127.0.0.1:49167->22/tcp,
127.0.0.1:49166->2376/tcp, 127.0.0.1:49165->5000/tcp, 127.0.0.1:49164->8443/tcp, 127.0.0.1:49163->32443/tcp   minikube
```

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.

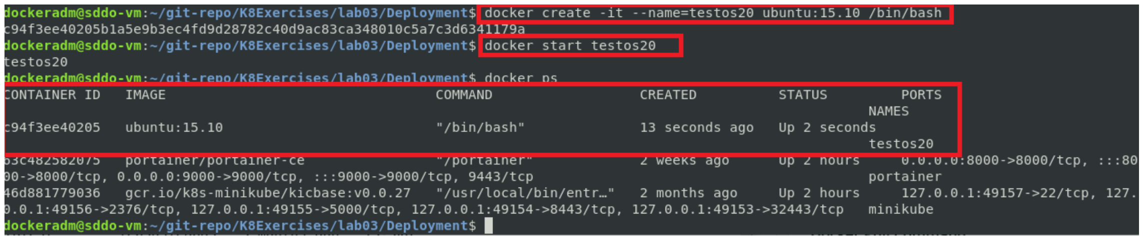
```
docker create -it --name=testos20 ubuntu:15.10 /bin/bash
```

Start the container

```
docker start testos20
```

Check the container is up and running.

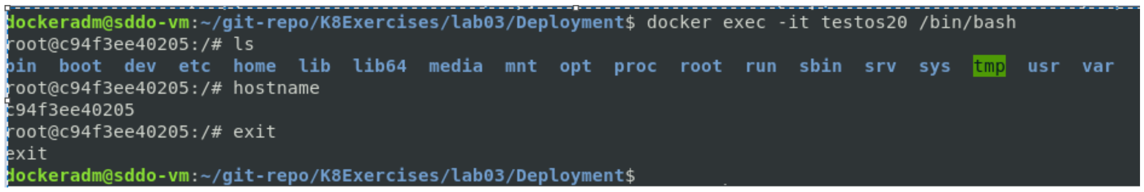
```
docker ps
```



```
dockeradm@sddo-vm:~/git-repo/K8Exercises/lab03/Deployment$ docker create -it --name=testos20 ubuntu:15.10 /bin/bash
c94f3ee40205b1a5e9b3ec4fd9d28782c40d9ac83ca348010c5a7c3d6341179a
dockeradm@sddo-vm:~/git-repo/K8Exercises/lab03/Deployment$ docker start testos20
testos20
dockeradm@sddo-vm:~/git-repo/K8Exercises/lab03/Deployment$ docker ps
CONTAINER ID   IMAGE                                 COMMAND                  CREATED          STATUS          PORTS
                                                                                                          NAMES
c94f3ee40205   ubuntu:15.10                          "/bin/bash"              13 seconds ago   Up 2 seconds
                                                                                                          testos20
63c482582075   portainer/portainer-ce                "/portainer"             2 weeks ago      Up 2 hours      0.0.0.0:8000->8000/tcp, :::80
00->8000/tcp, 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp, 9443/tcp                                         portainer
46d881779036   gcr.io/k8s-minikube/kicbase:v0.0.27   "/usr/local/bin/entr…"   2 months ago     Up 2 hours      127.0.0.1:49157->22/tcp, 127.
0.0.1:49156->2376/tcp, 127.0.0.1:49155->5000/tcp, 127.0.0.1:49154->8443/tcp, 127.0.0.1:49153->32443/tcp   minikube
dockeradm@sddo-vm:~/git-repo/K8Exercises/lab03/Deployment$
```

8. To access the shell of the container

```
docker exec -it testos20 /bin/bash
```



```
dockeradm@sddo-vm:~/git-repo/K8Exercises/lab03/Deployment$ docker exec -it testos20 /bin/bash
root@c94f3ee40205:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@c94f3ee40205:/# hostname
c94f3ee40205
root@c94f3ee40205:/# exit
exit
dockeradm@sddo-vm:~/git-repo/K8Exercises/lab03/Deployment$
```

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?

2. Use a browser and access localhost via port 3500

3. On Host OS, create an empty file
   ```
   touch /testfile
   ```

4. Copy /testfile from local host OS to the running container web-app
   ```
   docker cp /testfile web-app:/
   ```

5. Create a new image for this container
   ```
   docker commit web-app new-web-app
   ```

6. Check for the new image on local system
   ```
   docker images
   ```

7. Now, remove the existing container web-app
```
docker rm -f web-app
```

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

8. 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"
```

9. 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.

10. Now, remove the existing container web-app
```
docker rm -f new-web-app
```

11. 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"
```

12. 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?

13. Check the local images

```
docker images
```

14. Remove the newly created image

```
docker rmi new-web-app
```

Part 3: Check details of container

LAB EXERCISE

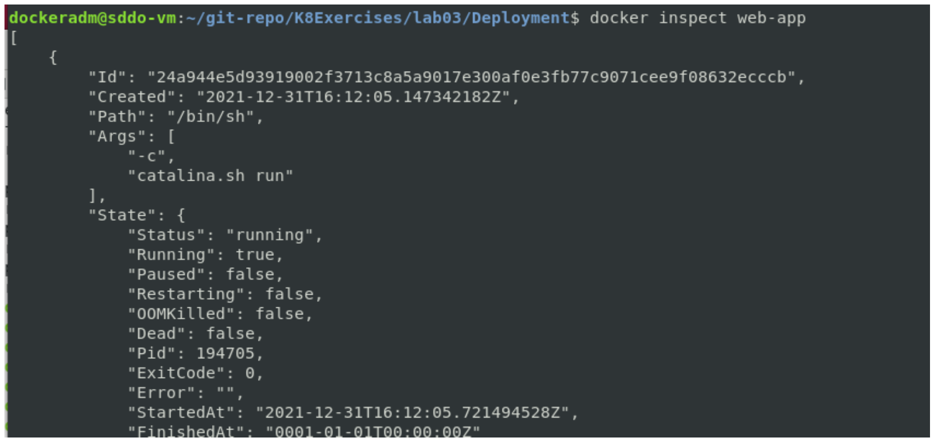
This LAB exercise demonstrates the concept of how to look for details of a container.

Time to Complete
Approximately 5 Minutes

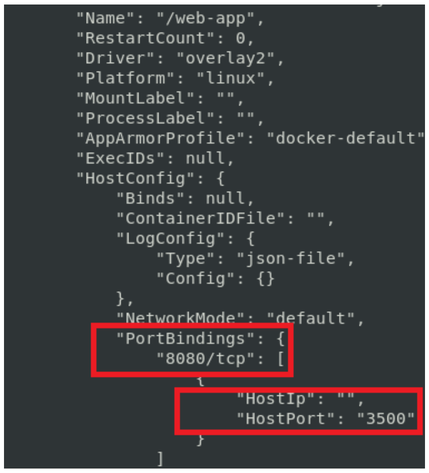
What You Need
- Part 2 to be completed successfully.

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`



```
dockeradm@sddo-vm:~/git-repo/K8Exercises/lab03/Deployment$ docker inspect web-app
[
    {
        "Id": "24a944e5d93919002f3713c8a5a9017e300af0e3fb77c9071cee9f08632ecccb",
        "Created": "2021-12-31T16:12:05.147342182Z",
        "Path": "/bin/sh",
        "Args": [
            "-c",
            "catalina.sh run"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 194705,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2021-12-31T16:12:05.721494528Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
```

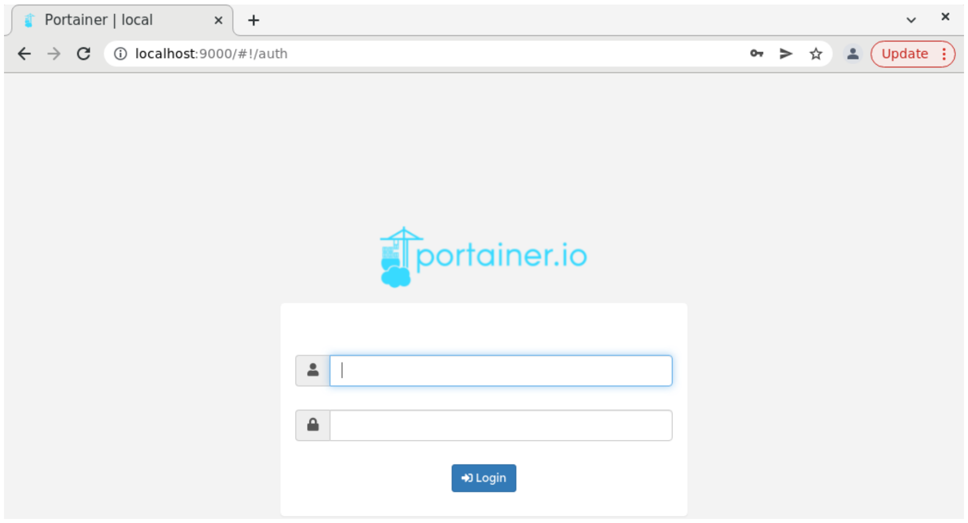


```
        "Name": "/web-app",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "docker-default"
        "ExecIDs": null,
        "HostConfig": {
            "Binds": null,
            "ContainerIDFile": "",
            "LogConfig": {
                "Type": "json-file",
                "Config": {}
            },
            "NetworkMode": "default",
            "PortBindings": {
                "8080/tcp": [
                    {
                        "HostIp": "",
                        "HostPort": "3500"
                    }
                ]
```

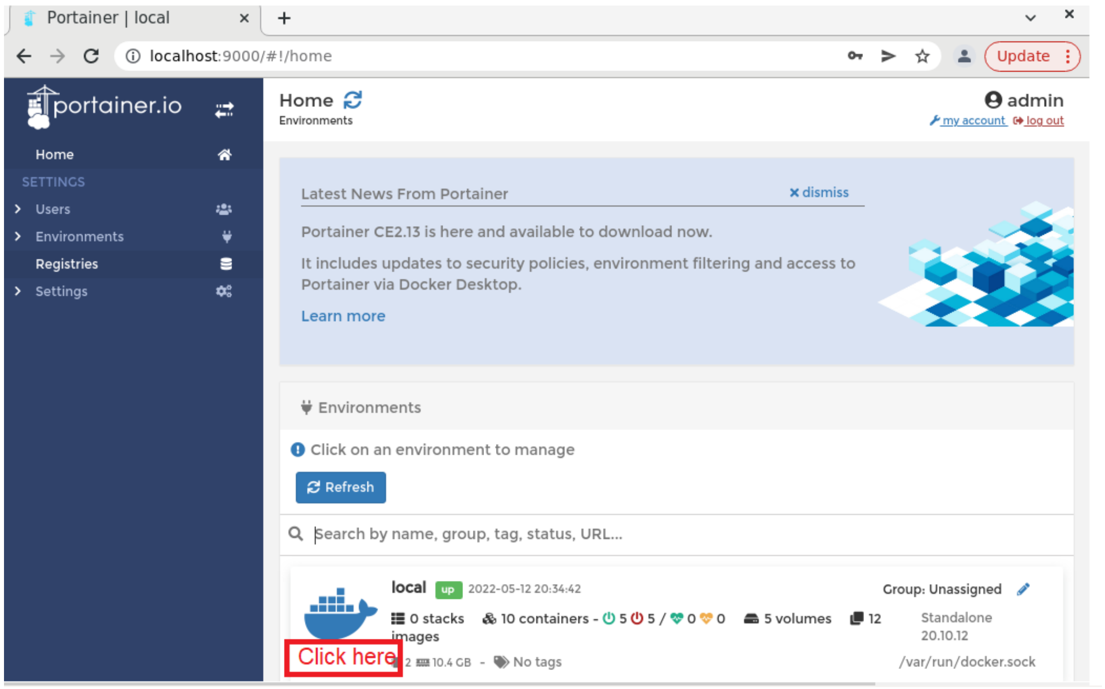
Part 4: Check details of container via Portainer

Access `http://localhost:9000`

Username: `admin`
Password: `admin!234`



After login:



Latest News From Portainer

Portainer CE2.13 is here and available to download now.

It includes updates to security policies, environment filtering and access to Portainer via Docker Desktop.

Learn more

Environments

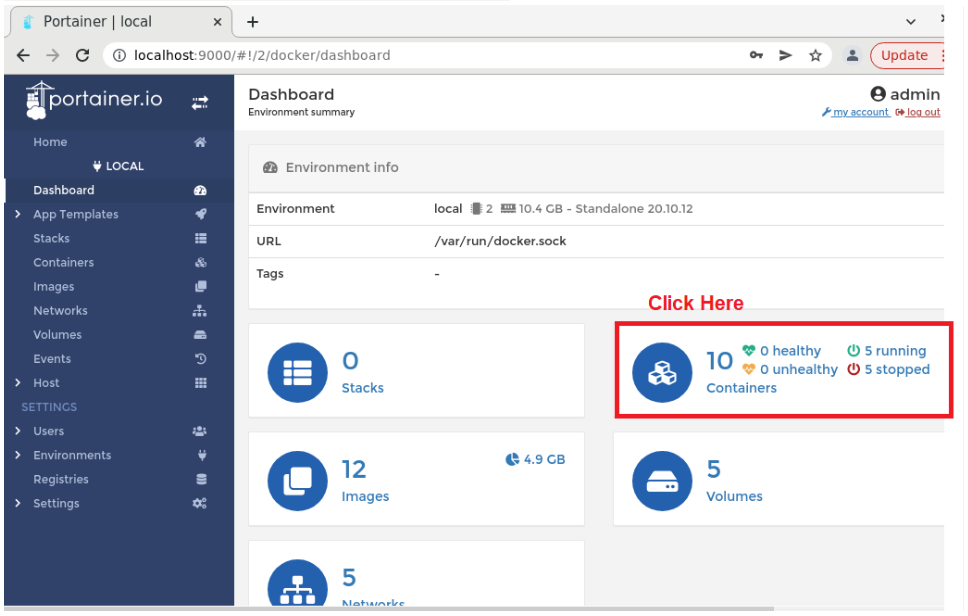
Click on an environment to manage

Refresh

Search by name, group, tag, status, URL...

local up 2022-05-12 20:34:42 Group: Unassigned
0 stacks 10 containers - 5 5 / 0 0 5 volumes 12 images Standalone 20.10.12
2 10.4 GB - No tags /var/run/docker.sock

Click on Container icon as shown below:



Dashboard
Environment summary

admin
my account log out

Environment info

Environment	local 2 10.4 GB - Standalone 20.10.12
URL	/var/run/docker.sock
Tags	-

0 Stacks

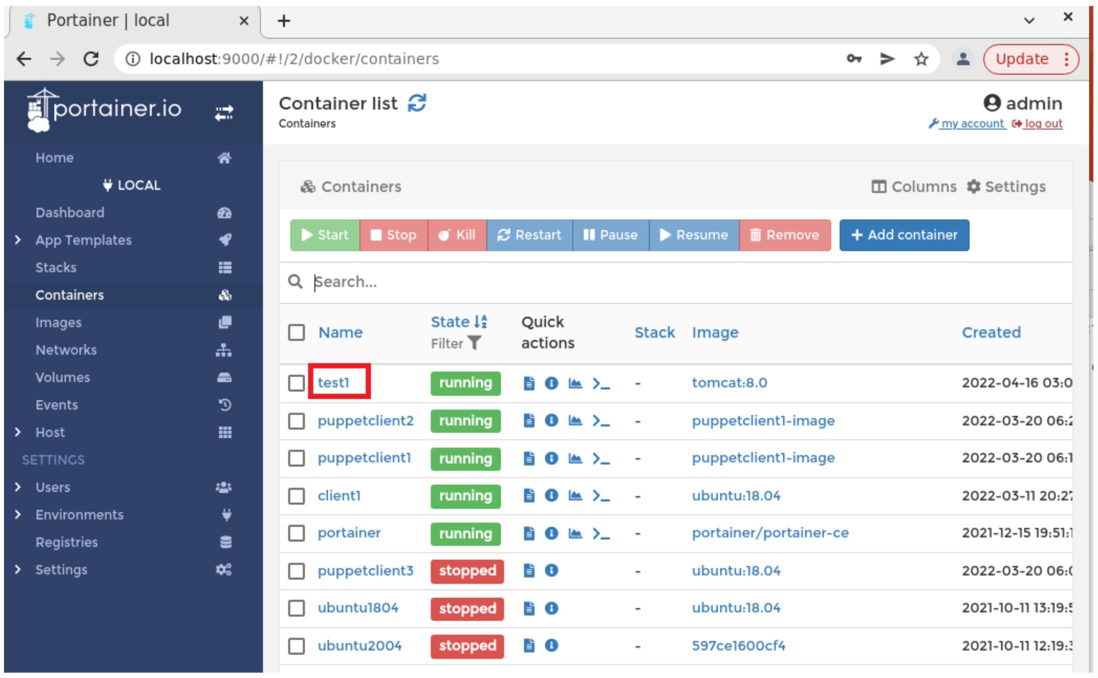
10 Containers

12 Images 4.9 GB

5 Volumes

5 Networks

Click on one of the container for more details:



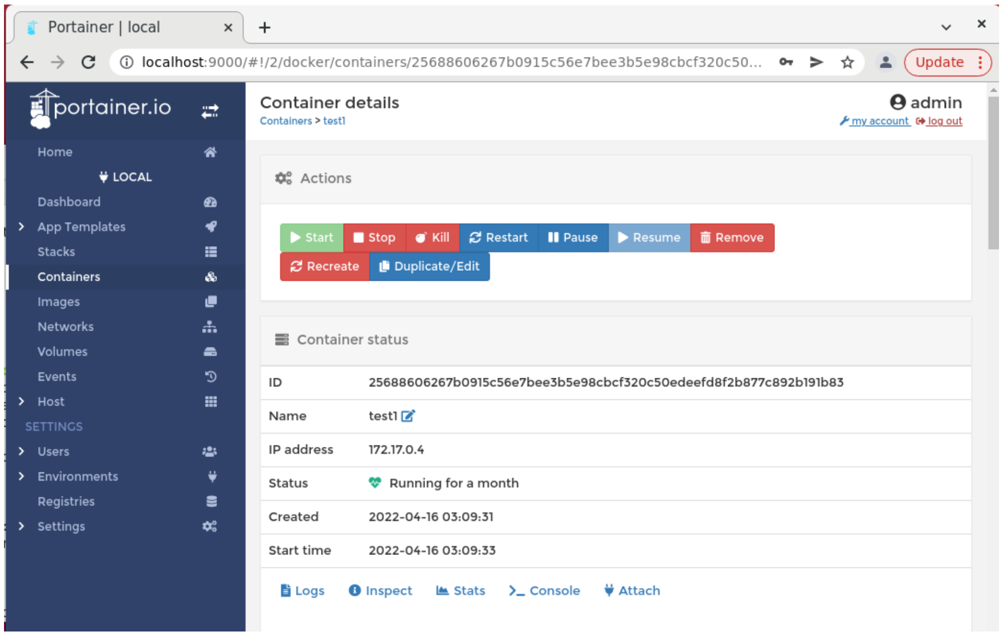
Container list
Containers

admin
my account log out

Containers

Name	State Filter	Quick actions	Stack	Image	Created
test1	running		-	tomcat:8.0	2022-04-16 03:0
puppetclient2	running		-	puppetclient1-image	2022-03-20 06:2
puppetclient1	running		-	puppetclient1-image	2022-03-20 06:1
client1	running		-	ubuntu:18.04	2022-03-11 20:2
portainer	running		-	portainer/portainer-ce	2021-12-15 19:51:
puppetclient3	stopped		-	ubuntu:18.04	2022-03-20 06:0
ubuntu1804	stopped		-	ubuntu:18.04	2021-10-11 13:19:
ubuntu2004	stopped		-	597ce1600cf4	2021-10-11 12:19:

`The detail of the selected container is shown.`



ID	25688606267b0915c56e7bee3b5e98cbcf320c50edeefd8f2b877c892b191b83
Name	test1
IP address	172.17.0.4
Status	Running for a month
Created	2022-04-16 03:09:31
Start time	2022-04-16 03:09:33

-- End of Lab ---