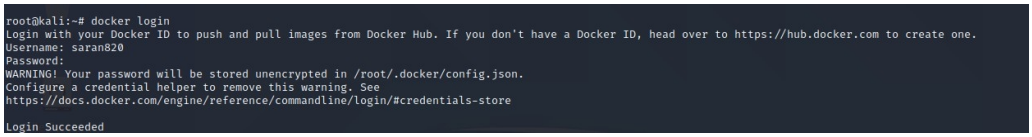
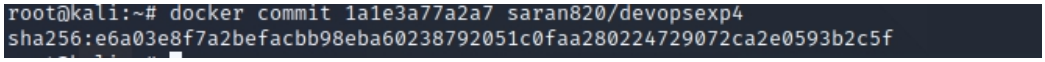
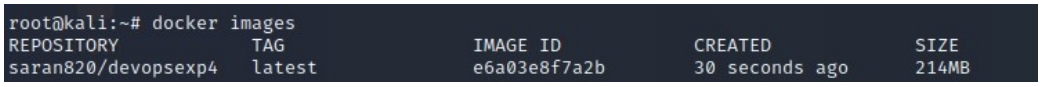
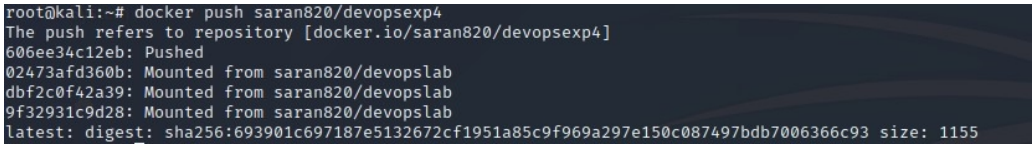
Semester	Semester VIII	
Subject	DevOps Lab	
Subject Professor In-charge	Prof. Yash Shah	
Laboratory	L11B	
Student Name	Ashwini Jadhav	
Roll Number	17101B0038	
Grade and Subject Teacher's Signature		

Experiment Number	4	
Experiment Title	To use Docker Hub for saving container repositories	
Resources / Apparatus Required	Hardware: Compatible Computer System	Kali Linux, Docker, Docker Hub
Objectives	Explore and implement DockerHub repositories	
Theory	**What is containerization?** It involves encapsulating or packaging up software code and all its dependencies so that it can run uniformly and consistently on any infrastructure. A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging. **Need of containerization:** ➢ Containerization reduces wasted resources because each container only holds the application and related binaries or libraries. ➢ By allowing more containers in the environment without the need for more servers, containerization increases scalability anywhere from 10 to 100 times that of traditional VM environments.	

	➢ The ability to rapidly spin up new containers also increases the capacity to handle website traffic load seamlessly. ➢ Using containerization helps your cloud environment efficiency; by deploying multiple containerized applications on to a single cloud instance, you get much closer to achieving 100% utilization. ➢ Improved security by isolating applications from the host system and from each other. ➢ Faster app start-up and easier scaling. ➢ Flexibility to work on virtualized infrastructures or on bare metal servers ➢ Easier management since install, upgrade, and rollback processes are built into the Kubernetes platform. **How to send repositories using Docker hub:** To login into Docker Hub: **docker login** Push the repository: **docker push *DockerHub_ID/repository_name*** Pull the repository: **docker pull *DockerHub_ID/repository_name***
Output	**Login to Docker Hub via terminal:**  **Committing the container:**  **List of images:**  **Push repository to Docker Hub:** 

Pulling a repository from DockerHub:

```
root@kali:~# docker pull yash1234shah/devopslab
Using default tag: latest
latest: Pulling from yash1234shah/devopslab
83ee3a23efb7: Already exists
db98fc6f11f0: Already exists
f611acd52c6c: Already exists
b6718bc6ac6b: Pull complete
Digest: sha256:08671f2a0c2f8b6d52228ef1caf3bae388386db0b35055b358d7c7b561743c5b
Status: Downloaded newer image for yash1234shah/devopslab:latest
docker.io/yash1234shah/devopslab:latest
```

Creating container from pulled image:

```
root@kali:~# docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
saran820/devopsexp4     latest              e6a03e8f7a2b        26 minutes ago      214MB
devopsexp4              latest              0031621bd405        44 minutes ago      214MB
devopsexp3              latest              522977051d9e        3 days ago          214MB
saran820/devopslab      latest              f7b2b30841ba        3 days ago          214MB
yash1234shah/devopslab  latest              5cdf4caa961e        3 days ago          214MB
webproject              latest              86c26621f3ce        10 days ago         214MB
ubuntu                  latest              f63181f19b2f        5 weeks ago         72.9MB
root@kali:~# docker run -it -d yash1234shah/devopslab
84a9580330e04e8aa64ca3cc17b4f9fbae567f9b636c3871a6f902f0d196cff9
root@kali:~# docker ps
CONTAINER ID        IMAGE                       COMMAND             CREATED             STATUS              PORTS               NAMES
84a9580330e0        yash1234shah/devopslab      "/bin/bash"         5 seconds ago       Up 3 seconds                            sleepy_johnson
1a1e3a77a2a7        ubuntu                      "/bin/bash"         56 minutes ago      Up 55 minutes                           zen_margulis
root@kali:~# docker commit 84a9580330e0 devopexp4pulled
sha256:e360dcf1f78f099e978b5d7314f63bc405025bf7d5823fb59c46fdb74d87f564
```

Port forwarding:

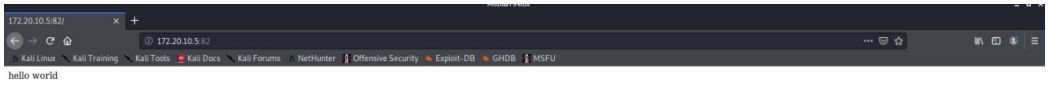
```
root@kali:~# docker run -it -d -p 82:80 yash1234shah/devopslab
af739dc2a8c20838e7cc0ff45447502374876e1fa3c35d5eb41af399ddba872f
```

Starting apache server within container:

```
root@kali:~# docker exec -it af739dc2a8c2 bash
root@af739dc2a8c2:/# service apache2 start
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.4. Set the 'ServerName' directive globally to suppress this message
 *
```

Result of the above:

```
root@kali:~# docker ps
CONTAINER ID        IMAGE                       COMMAND             CREATED             STATUS              PORTS                  NAMES
af739dc2a8c2        yash1234shah/devopslab      "/bin/bash"         54 seconds ago      Up 52 seconds       0.0.0.0:82->80/tcp     ecstatic_solomon
```



	List of images and containers and Stopping and removing containers:  root@kali:~# docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES af739dc2a8c2 yash1234shah/devopslab "/bin/bash" 6 minutes ago Up 6 minutes 0.0.0.0:82→80/tcp ecstatic_solomon 84a9580330e0 yash1234shah/devopslab "/bin/bash" 9 minutes ago Up 9 minutes sleepy_johnson 1a1e3a77a2a7 ubuntu "/bin/bash" About an hour ago Up About an hour zen_margulis root@kali:~# docker stop 84a9580330e0 84a9580330e0 root@kali:~# docker rm 84a9580330e0 84a9580330e0 root@kali:~# docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES af739dc2a8c2 yash1234shah/devopslab "/bin/bash" 7 minutes ago Up 7 minutes 0.0.0.0:82→80/tcp ecstatic_solomon 1a1e3a77a2a7 ubuntu "/bin/bash" About an hour ago Up About an hour zen_margulis root@kali:~#
Conclusion	Thus, we have implemented containerization using Docker and saved repository to Docker Hub.