

 Vidyalankar Institute of Technology ACCREDITED A+ BY NAAC	Department of Information Technology
--	--------------------------------------

Semester	B.E. Semester VIII – INFT
Subject	DevOps Lab
Subject Professor In –charge	Prof. Rohit Barve

Student Name	Nupur Sawarkar
Roll Number	18101B0030
Grade and Subject Teacher's Signature	

Experiment Number	4
Experiment Title	To use Docker Hub for saving container repositories.
Resources / Apparatus Required	<div>Hardware:</div> <ul style="list-style-type: none"> ● Intel Core i3/i5/i7 Processor with Intel VT-X support. ● 4 GB RAM ● 500 GB Hard disk <div>Software:</div> <p>Operating systems: Windows</p> <p>Virtual Box, Docker Hub</p>
Theory	<p>What is Docker Hub?</p> <p>Docker Hub is Docker's official cloud-based registry for Docker images. It hosts over 100,000 images including official images for MongoDB, nginx, Apache, Ubuntu, and MySQL that have all been downloaded over a billion times each.</p> <p>In addition to the public repositories that anyone can pull from, Docker Hub offers private repositories where individuals or teams can host images they</p>

wish to restrict access to. Docker Hub also offers features such as GitHub and Bitbucket integrations that help automate development build processes and support for webhooks which can act as triggers for everything from automated tests to notifications.

Why Use Docker Hub?

Trusted sources

Docker Hub enables users to access various types of trusted images, which are reviewed for security and quality standards. These types include:

- Verified Publisher images—the software vendor is verified by Docker.
- Docker Certified images—Docker verifies and checks the images against best practices and known vulnerabilities.
- Official images—curated by Docker, helping developers to establish a strong foundation within their project, based upon a collection of essential base images and best practices. Learn more in our guide to Docker Official Images ›

Each type of image offers a different level of trust. You can reference these trust levels when choosing images to download or when publishing an image on Docker Hub.

Free tier

Docker Hub’s free tier makes useful capabilities of container repositories accessible, providing:

- Unlimited public repositories
- One private repository with a maximum of three collaborators.

You can leverage basic testing capabilities within the free tier, and get familiar with the functions and user experience of Docker Hub. However, the terms of service applicable to free tier accounts limit the number of pushes and pulls allowed, per six hours. This limitation makes the free tier

unsuitable for most modern development and DevOps workflows.

Security features

Docker Hub allows all accounts to benefit from local image vulnerability scans. However, only team accounts can access audit-logs and use multifactor authentication (MFA) to secure their repositories.

Steps

After creating container and starting services in experiment 3, Now come out of the container and type “**sudo docker ps**” to see which containers are running.

```
root@f0494df7c42d:/var/www/html# exit
exit
manali@manali-VirtualBox:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
f0494df7c42d   ubuntu   "bash"    12 minutes ago   Up 12 minutes   container2
35d46bfb9d6f   ubuntu   "bash"    14 minutes ago   Up 14 minutes   charming_nobel
manali@manali-VirtualBox:~$ sudo docker ps -a
```

Now stop one container and see the status of that container.

```
manali@manali-VirtualBox:~$ sudo docker stop 7507d2c0fff1
7507d2c0fff1
manali@manali-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
f0494df7c42d   ubuntu   "bash"    14 minutes ago   Up 14 minutes   container2
35d46bfb9d6f   ubuntu   "bash"    16 minutes ago   Up 16 minutes   charming_nobel
6dd39230ea19   ubuntu   "bash"    7 days ago       Exited (0) 7 days ago   sweet_feynman
9ff8dcea8852   ubuntu   "bash"    7 days ago       Exited (255) 7 days ago   container1
7507d2c0fff1   ubuntu   "bash"    7 days ago       Exited (255) 7 days ago   nifty_yalow
```

Now remove the container using command “**sudo docker rm [containerID]**” and then again type “**sudo docker ps -a**” , we can see that the container is removed.

```
manali@manali-VirtualBox: ~  
7507d2c0fff1 ubuntu bash 7 days ago Exited (255) 7 days ago  
nifty_yalow  
manali@manali-VirtualBox:~$ sudo docker rm 7507d2c0fff1  
7507d2c0fff1  
manali@manali-VirtualBox:~$ sudo docker ps -a  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
f0494df7c42d ubuntu "bash" 15 minutes ago Up 15 minutes container2  
35d46bfb9d6f ubuntu "bash" 16 minutes ago Up 16 minutes charming_nobel  
6dd39230ea19 ubuntu "bash" 7 days ago Exited (0) 7 days ago sweet_feynman  
9ff8dcea8852 ubuntu "bash" 7 days ago Exited (255) 7 days ago container1
```

Now commit the current created container.

```
manali@manali-VirtualBox:~$ sudo docker commit f0494df7c42d ubuntu  
sha256:22975f69c22efb55b20a0bb10562874c4da9a9b102cb72f5cea731a832a03d1e
```

Now type “sudo docker images” to see which images exist.

```
manali@manali-VirtualBox:~$ sudo docker images  
REPOSITORY TAG IMAGE ID CREATED SIZE  
ubuntu latest 22975f69c22e 10 seconds ago 262MB  
ubuntu <none> ff0fea8310f3 11 days ago 72.8MB  
ubuntu <none> 2b4cba85892a 3 weeks ago 72.8MB
```

Now to do port forwarding type “**sudo docker run -it -d -p 82:80 ubuntu**”, this Docker Desktop makes whatever is running on port 82 in the container available on port 80 of localhost.

```
manali@manali-VirtualBox: ~  
ubuntu <none> 2b4cba85892a 3 weeks ago 72.8MB  
manali@manali-VirtualBox:~$ sudo docker run -it -d -p 82:80 ubuntu  
f09da217f5c9ecc221a19fefe002db1dc1231d05835d38a1c439b74003bcd8ee  
manali@manali-VirtualBox:~$ sudo docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS  
f09da217f5c9 ubuntu "bash" 2 minutes ago Up 2 minutes 0.0.0.0:82->80/tcp, :::82->80/tcp  
f0494df7c42d ff0fea8310f3 "bash" 23 minutes ago Up 23 minutes container2  
35d46bfb9d6f ff0fea8310f3 "bash" 24 minutes ago Up 24 minutes charming_nobel
```

After that start the Apache service and check whether it runs on that Localhost.

```
manali@manali-VirtualBox:~$ sudo docker exec -it f09da217f5c9 bash  
root@f09da217f5c9:/# 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  
*  
root@f09da217f5c9:/# exit  
exit
```

Now login to docker hub.

```
manali@manali-VirtualBox: ~  
root@f09da217f5c9:/# exit  
exit  
manali@manali-VirtualBox:~$ sudo docker login  
Login with your Docker ID to push and pull images from Docker Hub. If you don't  
have a Docker ID, head over to https://hub.docker.com to create one.  
Username: ms2310  
Password:  
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.  
Configure a credential helper to remove this warning. See  
https://docs.docker.com/engine/reference/commandline/login/#credentials-store  
Login Succeeded
```

Now commit the container and push the image to the docker Hub.

```
manali@manali-VirtualBox:~$ sudo docker commit f09da217f5c9 ms2310/1stcommit  
sha256:4a165428d41607f548e276f2b8b47d24b598b8773787226293f9394621b4f94b  
manali@manali-VirtualBox:~$ sudo docker images  
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE  
ms2310/1stcommit    latest       4a165428d416     7 seconds ago   262MB  
ubuntu              latest       22975f69c22e     24 minutes ago  262MB  
ubuntu              <none>       ff0fea8310f3     12 days ago     72.8MB  
ubuntu              <none>       2b4cba85892a     3 weeks ago     72.8MB  
manali@manali-VirtualBox:~$ sudo docker push ms2310/1stcommit  
Using default tag: latest  
The push refers to repository [docker.io/ms2310/1stcommit]  
679d918b99e7: Pushed  
24d4526310fb: Pushed  
867d0767a47c: Mounted from library/ubuntu  
latest: digest: sha256:d1fa4a2cf13ff172b132bc8fb1602d06ad640d4e0cab2d4674e7019d  
02923824 size: 948  
manali@manali-VirtualBox:~$
```

The screenshot shows a web browser window with the Docker Hub repository page for 'ms2310/1stcommit'. The browser tabs include 'manali-shinde/Gith', 'Apache2 Ubuntu Defa', and 'Docker Hub'. The address bar shows the URL 'https://hub.docker.com/repository/docker/ms2310/1stcommit'. The page content includes a message about cleaning up unused content, the repository name 'ms2310/1stcommit', a note that it has no description, the last push time '10 minutes ago', a 'Public View' button, and a section for 'Docker commands' with the command 'docker push ms2310/1stcommit:tagname' in a code block. At the bottom, there is a 'Tags and Scans' section and a security notice 'VULNERABILITY SCANNING - DISABLED'.

Conclusion:	Thus, we came to know how to do port forwarding and pushing the container on the Docker Hub. Docker Hub is a cloud-based repository that lets you create, test, store, and deploy Docker container images. It provides access to public open source image repositories and lets you create your own private repositories
-------------	--