

**Bansilal Ramnath Agarwal Charitable Trust's
VISHWAKARMA INSTITUTE OF INFORMATION
TECHNOLOGY,**

PUNE-48 Department of Information Technology

ITUA32202: CLOUD COMPUTING

Assignment-7

Shreyas Shripad Kulkarni

C2 Batch

Roll No. 333030

PRN: 22010443

AIM: Deploy a static website using Docker.

THEORY:

1) What is Docker

Docker is a containerization platform that allows developers to package an application and its dependencies into a container that can run on any machine with Docker installed.

2) Docker Architecture

Docker architecture includes three main components: the Docker daemon, the Docker client, and the Docker registry. The Docker daemon is the background service that manages the containers, images, and networks. The Docker client is a command-line interface that allows users to interact with the Docker daemon. The Docker registry is a place where Docker images can be stored and shared.

3) Difference between Docker and Virtual machine

Docker is different from a virtual machine in that it shares the host operating system's kernel and doesn't require a separate operating system for each container. This means that Docker containers are much lighter and faster to start up than virtual machines.

4) Docker Commands

Docker commands include `docker run`, `docker build`, `docker push`, `docker pull`, `docker ps`, and `docker logs`. These commands are used to manage Docker containers, images, and networks.

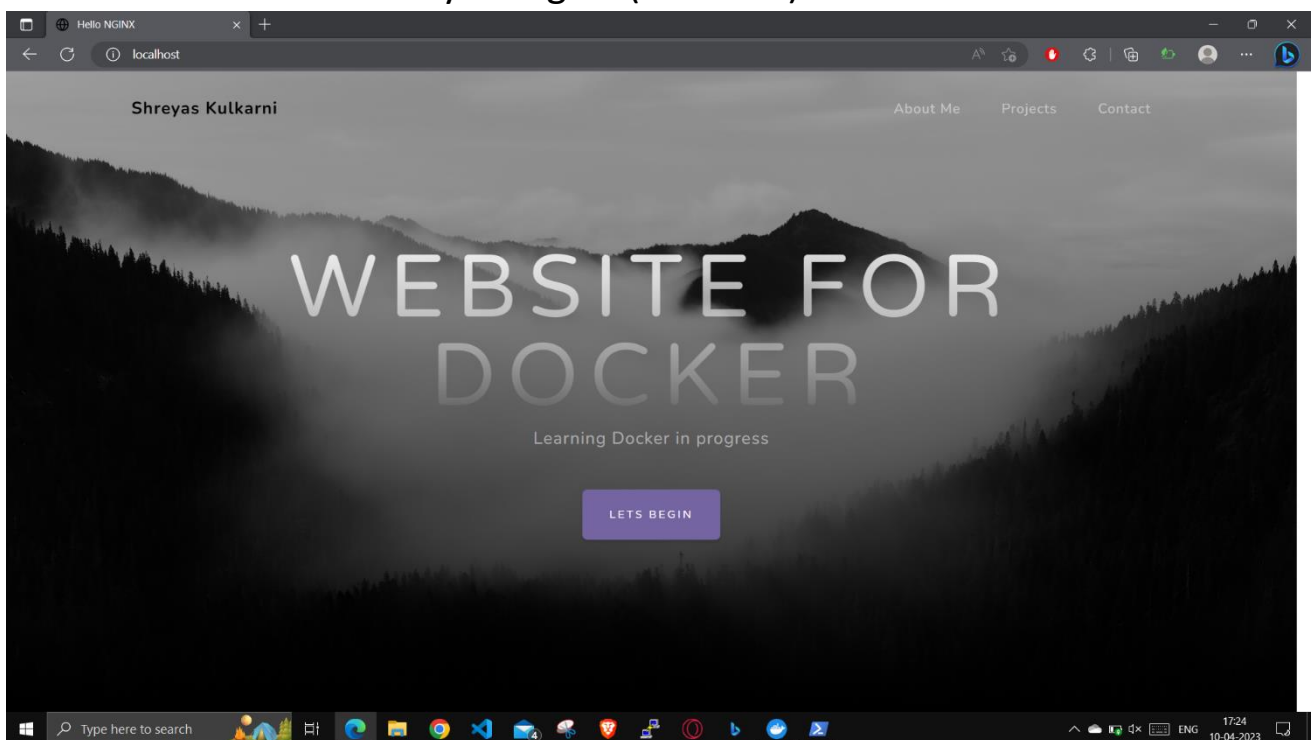
5) Dockerfile

A Dockerfile is a text file that contains instructions for building a Docker image. It specifies the base image, any additional software packages to install, and any configuration settings needed for the application to run.

6) Docker-Compose and Docker-swarm

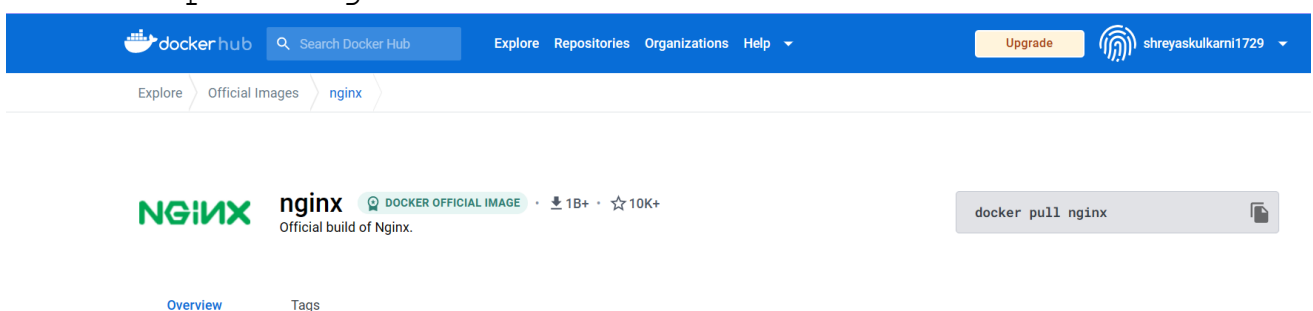
Docker-compose is a tool for defining and running multi-container Docker applications. Docker-swarm is a tool for managing a cluster of Docker nodes and deploying a Docker stack to that cluster.

-> Run the website locally on nginx (PORT 80)



-> Open docker desktop and open PowerShell

-> after checking the version, pull the nginx image from docker hub
`docker pull nginx`

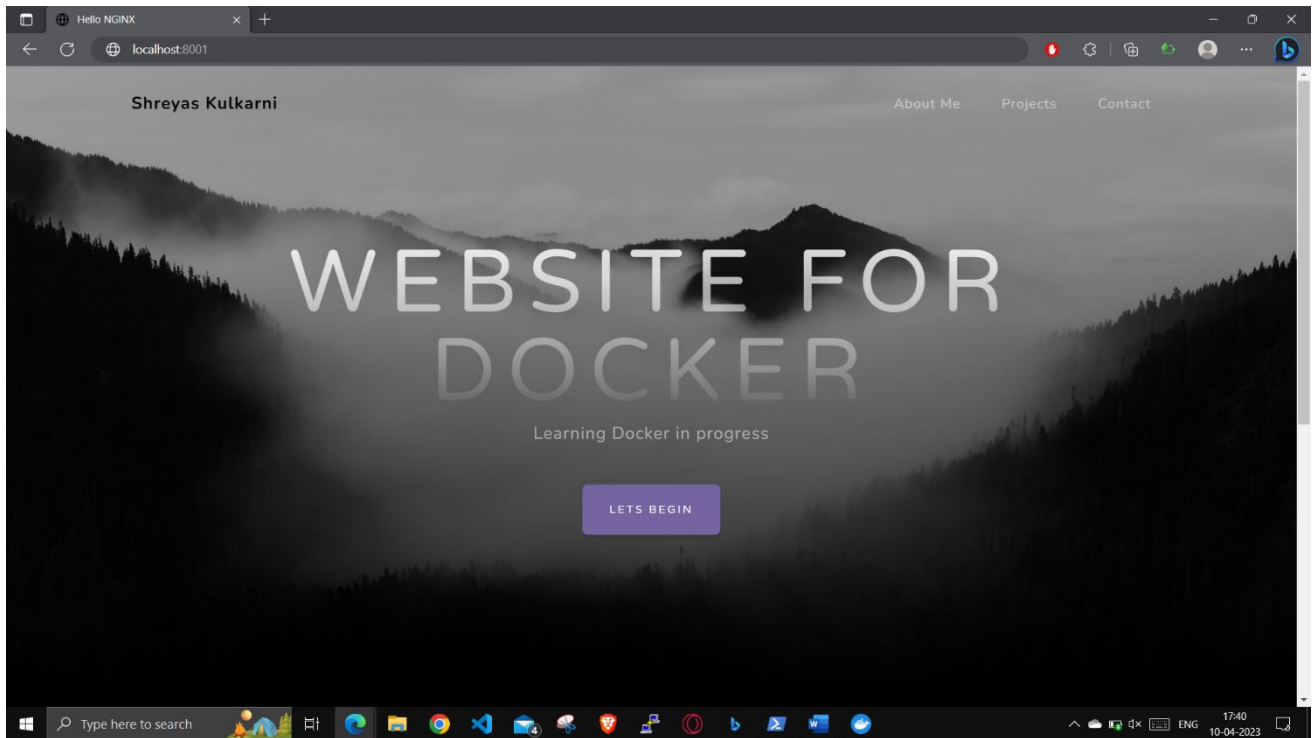


```
Windows PowerShell
PS C:\> docker --version
Docker version 20.10.23, build 7155243
PS C:\> docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
f1f26f570256: Pull complete
7f7f30930c6b: Pull complete
2836b727df80: Pull complete
e1eeb0f1c06b: Pull complete
86b2457cc2b0: Pull complete
9862f2ee2e8c: Pull complete
Digest: sha256:2ab30d6ac53580a6db8b657abf0f68d75360ff5cc1670a85acb5bd85ba1b19c0
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
PS C:\>
```

-> Perform Mount Bind which will create a container using the docker command and sync the Website folder with the folder inside the container

```
Administrator: Windows PowerShell
PS C:\users\hp\desktop\Website> docker run -d -p 8001:80 -v ${PWD}:/usr/share/nginx/html --name web-site nginx
084551cd0362c3a3d60d8771ce829eedb8a610848cca2c7da6969b0faed0918b
PS C:\users\hp\desktop\Website> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
084551cd0362   nginx    "/docker-entrypoint. ...."   10 seconds ago Up 9 seconds   0.0.0.0:8001->80/tcp   web-site
PS C:\users\hp\desktop\Website>
```

-> Verify the website by opening localhost:8001



We have successfully deployed your web application in the container.

#DOCKERFILE

Create a Directory structure like

APP

|--- Website

|--- Dockerfile

```
Dockerfile - Notepad
File Edit Format View Help
FROM nginx:latest
COPY ./Website/ /usr/share/nginx/html/
EXPOSE 80
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

-> Build image from docker file using the build command

```
docker build -t my-app:v1 .
```

```
Administrator: Windows PowerShell
PS C:\users\hp\desktop\App> docker build -t my-app:v1 .
[+] Building 0.2s (7/7) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 105B                                              0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/nginx:latest                 0.0s
=> [internal] load build context                                                 0.0s
=> => transferring context: 383B                                                0.0s
=> CACHED [1/2] FROM docker.io/library/nginx:latest                           0.0s
=> [2/2] COPY ./website/ /usr/share/nginx/html/                               0.0s
=> exporting to image                                                            0.0s
=> => exporting layers                                                            0.0s
=> => writing image sha256:9f9c078eb9a9baab18be133a2fb6c5295c0154c6da8b11f73f6ebd1d740f8bfc 0.0s
=> => naming to docker.io/library/my-app:v1                                    0.0s
PS C:\users\hp\desktop\App>
```

-> Using the docker images command check images

-> Login to DockerHub with credentials

```
PS C:\Users\hp\Desktop\APP> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
my-app        v1        aeee2b85e233   11 seconds ago 144MB
nginx         latest    6efc10a0510f   2 weeks ago   142MB
PS C:\Users\hp\Desktop\APP> docker login -u shreyaskulkarni1729
Password:
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/a
ccess-tokens/
PS C:\Users\hp\Desktop\APP> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
my-app        v1        aeee2b85e233   53 seconds ago 144MB
nginx         latest    6efc10a0510f   2 weeks ago   142MB
PS C:\Users\hp\Desktop\APP> docker tag my-web:v1 shreyaskulkarni1729/newapp
```

PUSH Images to DockerHub

-> docker images

```
PS C:\Users\hp\Desktop\APP> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
my-app        v1        aeee2b85e233   53 seconds ago 144MB
nginx         latest    6efc10a0510f   2 weeks ago   142MB
```

-> docker tag (old image name)

```
Windows PowerShell
PS C:\Users\hp\Desktop\APP> docker tag my-app:v1 shreyaskulkarni1729/my-app:v1
```

-> docker images

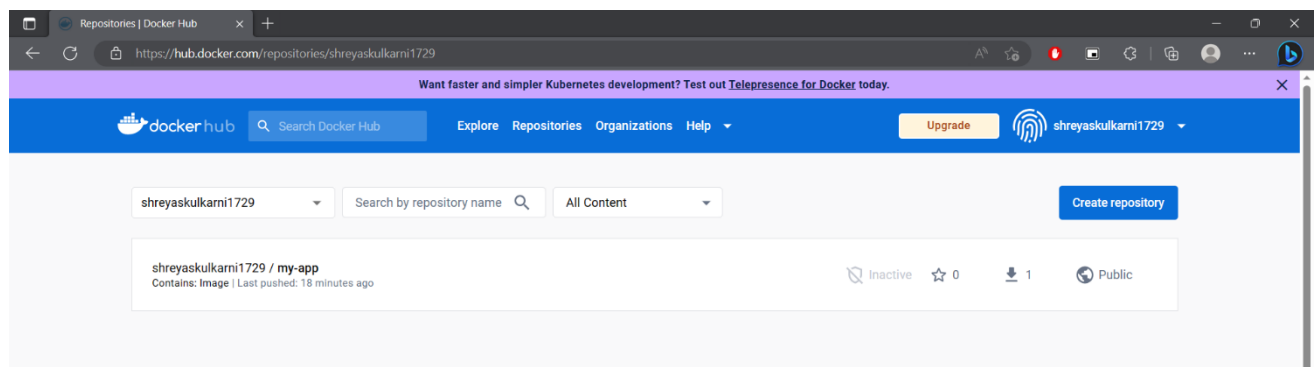
```
PS C:\Users\hp\Desktop\APP> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
my-app	v1	aeee2b85e233	About an hour ago	144MB
shreyaskulkarni1729/my-app	v1	aeee2b85e233	About an hour ago	144MB
nginx	latest	6efc10a0510f	2 weeks ago	142MB

-> docker push shreyaskulkarni1729/my-app

```
PS C:\Users\hp\Desktop\APP> docker push shreyaskulkarni1729/my-app:v1
The push refers to repository [docker.io/shreyaskulkarni1729/my-app]
b044e0594e2a: Pushed
9d907f11dc74: Mounted from library/nginx
79974a1a12aa: Mounted from library/nginx
f12d4345b7f3: Mounted from library/nginx
935b5bd454e1: Mounted from library/nginx
fb6d57d46ad5: Mounted from library/nginx
ed7b0ef3bf5b: Mounted from library/nginx
v1: digest: sha256:bfbba6b90a50a053e403bfbce85c641830c10e5ed7a6625c83b17b37d9e5520 size: 1781
PS C:\Users\hp\Desktop\APP>
```

-> The docker image is now pushed on the DockerHub !



Now we can share this image with anyone with running nginx and our web application.

Conclusion: We have successfully deployed a static website using Docker.