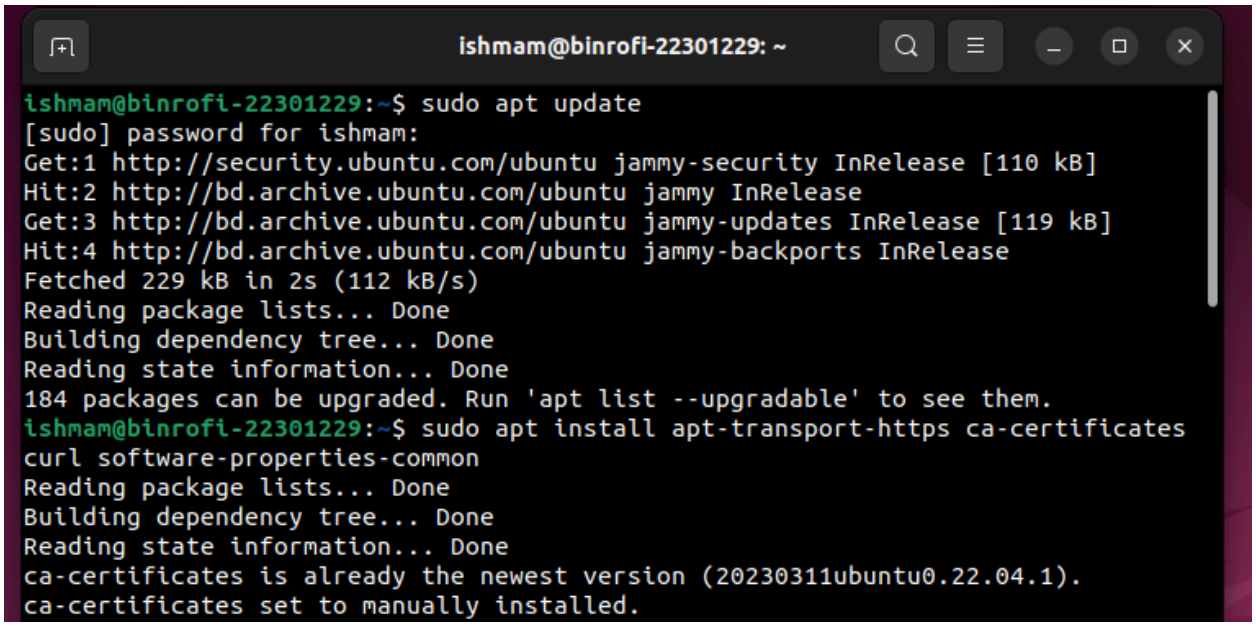
Name:

# 1. First Installing Docker on My Ubuntu

```
sudo apt update
```

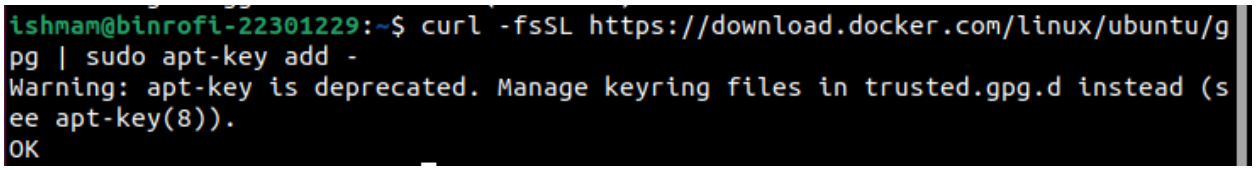
# Install a few prerequisite packages that let apt use packages over HTTPS

```
sudo apt install apt-transport-https ca-certificates curl
software-properties-common
```




# Adding the GPG key for the official Docker repository to my system

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add
-
```




# Adding the Docker repository to APT sources

```
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

```
ishmam@binrofi-22301229:~$ sudo add-apt-repository "deb [arch=amd64] https://dow
nload.docker.com/linux/ubuntu $(lsb_release -cs) stable"
[sudo] password for ishmam:
Sorry, try again.
[sudo] password for ishmam:
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu jammy sta
ble'
Description:
Archive for codename: jammy components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_co
```

# installing from the Docker repo instead of the default Ubuntu repo

**`apt-cache policy docker-ce`**

```
ishmam@binrofi-22301229:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:25.0.3-1~ubuntu.22.04~jammy
  Version table:
     5:25.0.3-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:25.0.2-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:25.0.1-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:25.0.0-1~ubuntu.22.04~jammy 500
```

# Install Docker

**`sudo apt install docker-ce`**

```
ishmam@binrofi-22301229:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit
  git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli
  docker-ce-rootless-extras docker-compose-plugin git git-man liberror-perl
  libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 180 not upgraded.
```

Now I am logging in to my dockerhub using my dockerhub pass and userid

```
ishmam@binrofi-22301229:~$ sudo docker login
Log in with your Docker ID or email address 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
.
You can log in with your password or a Personal Access Token (PAT). Using a limited
-scope PAT grants better security and is required for organizations using SSO. Lear
n more at https://docs.docker.com/go/access-tokens/

Username: ishmambr10
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
```

## 2. Show output of some basic docker commands

Pull: It pulls a docker image. First finds in local then goes for online repo

$ docker pull hello-world

```
ishmam@binrofi-22301229:~$ sudo docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:d000bc569937abbe195e20322a0bde6b2922d805332fd6d8a68b19f524b7d21d
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
```

Sudo docker images

It shows all the images there in my docker

```
ishmam@binrofi-22301229:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    d2c94e258dcb   9 months ago   13.3kB
```

Run: It runs a particular docker image on my pc

Docker run hello-world

```
ishmam@binrofi-22301229:~$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/
```

$ docker ps -a

docker ps -a: This command will show all containers (both running and stopped)

```
ishmam@binrofi-22301229:~$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED              STATUS
          PORTS     NAMES
d62b37af5026   hello-world   "/hello"   About a minute ago   Exited (0) About a min
ute ago             elastic_swanson
```

Sudo docker rm docker id

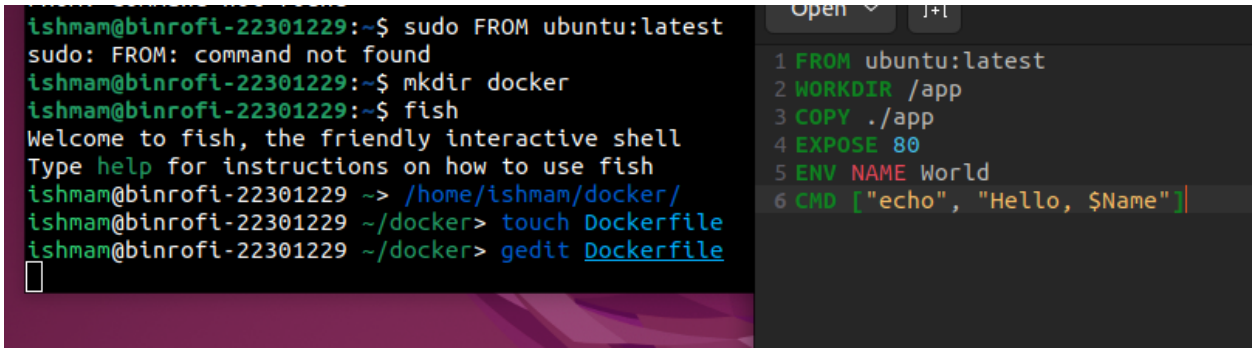
Removes the docker

```
ishmam@binrofi-22301229:~$ sudo docker rm d62b37af5026
d62b37af5026
ishmam@binrofi-22301229:~$
```

## 3. Create a Docker image using Dockerfile.

First I need to create a dockerfile. So, I am going to create a folder called docker first. Then I will create a file using touch named Dockerfile. Then I will edit the Dockerfile using the command gedit. Here is what I wrote in my Dockerfile

```
FROM ubuntu:latest
WORKDIR /app
COPY . /app
# Make port 80 available to the world outside this container
EXPOSE 80
# Define environment variable
ENV NAME World
# Run app.py when the container launches
CMD ["echo", "Hello, $NAME"]
```




Now lets build my docker image using the Dockerfile.

```
docker build -t hello-world-ishmam .
```

```
ishmam@binrofi-22301229 ~/docker [1]> gedit Dockerfile
^C⏎
ishmam@binrofi-22301229 ~/docker [SIGINT]> sudo docker build -t hello
-world-ishmam .
[+] Building 5.9s (8/8) FINISHED                          docker:default
 => [internal] load build definition from Dockerfile                0.0s
 => => transferring dockerfile: 108B                                0.0s
 => [internal] load metadata for docker.io/library/ubuntu:late      0.9s
 => [internal] load .dockerignore                                   0.0s
 => => transferring context: 2B                                     0.0s
 => [1/3] FROM docker.io/library/ubuntu:latest@sha256:f9d633ff      4.6s
 => => resolve docker.io/library/ubuntu:latest@sha256:f9d633ff      0.0s
 => => sha256:01007420e9b005dc14a8c8b0f996a2 29.54MB / 29.54MB      3.7s
 => => sha256:f9d633ff6640178c2d0525017174a688 1.13kB / 1.13kB      0.0s
 => => sha256:81bba8d1dde7fc1883b6e95cd46d6c9f4874 424B / 424B      0.0s
 => => sha256:3db8720ecbf5f5927d409cc61f9b4f7f 2.30kB / 2.30kB      0.0s
 => => extracting sha256:01007420e9b005dc14a8c8b0f996a2ad8e0d4      0.8s
 => [internal] load build context                                   0.0s
 => => transferring context: 108B                                   0.0s
 => [2/3] WORKDIR /app                                              0.2s
 => [3/3] COPY . /app                                               0.1s
 => exporting to image                                              0.0s
 => => exporting layers                                             0.0s
 => => writing image sha256:d6837ad87af71c642a7108efa269a0bf7a      0.0s
 => => naming to docker.io/library/hello-world-ishmam               0.0s
ishmam@binrofi-22301229 ~/docker>
```

## 4. Run a container as a single task, show outputs, and show the status of all containers (using docker ps -a)

I am running the docker image “hello-world” as a single container. Then I am showing the status of the all containers using ps -a

```
$ docker run hello-world
$ docker ps -a
```

```
See 'docker run --help'.
ishmam@binrofi-22301229 ~ [126]> sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working corre
ctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker
Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which r
uns the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, whic
h sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with
:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

ishmam@binrofi-22301229 ~> sudo docker ps -a
CONTAINER ID   IMAGE                COMMAND       CREATED
  STATUS                          PORTS     NAMES
cbdf2d3e7b35   hello-world          "/hello"      About a minute ago
  Exited (0) About a minute ago             confident_snyder
ee076b19dadd   hello-world-ishmam   "/bin/bash"   2 minutes ago
  Exited (0) 2 minutes ago                  elated_hodgkin
ishmam@binrofi-22301229 ~>
```

## 5. Run a container in iterative mode and install different packages in the container. Show each step.

I ran Ubuntu light image as an iterator on my terminal. Thus my docker pulled ubuntu from docker hub and ran it as an iterative process. Then I installed fish on my iterative terminal of Ubuntu
Commands:

**Sudo docker run -it ubuntu bash**

**Apt-get install -y fish curl**

```
ishmam@binrofi-22301229 ~ [1]> sudo docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
01007420e9b0: Already exists
Digest: sha256:f9d633ff6640178c2d0525017174a688e2c1aef28f0a0130b26bd5
554491f0da
Status: Downloaded newer image for ubuntu:latest
root@84f78bbd6e6a:/#
```

In the iterative container I installed fish and curl

```
root@84f78bbd6e6a:/# apt-get install -y fish curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bsdextrautils bzip2 ca-certificates file fish-common groff-base
  libbrotli1 libbsd0 libcurl4 libexpat1 libgdbm-compat4 libgdbm6
  libldap-2.5-0 libldap-common libmagic-mgc libmagic1 libmd0
  libmpdec3 libnghttp2-14 libpcre2-32-0 libperl5.34 libpipeline1
  libpsl5 libpython3-stdlib libpython3.10-minimal
  libpython3.10-stdlib libreadline8 librtmp1 libsasl2-2
  libsasl2-modules libsasl2-modules-db libsqlite3-0 libssh-4
```

## 6. Run a database container in the background. Then show logs of the running container. After, access the container in interactive mode. show some SQL queries inside the container.

How I am running my mysql database container. Here I run the database and set the name to some-mysql and used root user and set password my-secret-pw. And pulled mysql image.

```
$ docker run -d --name some-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw
mysql:latest
```

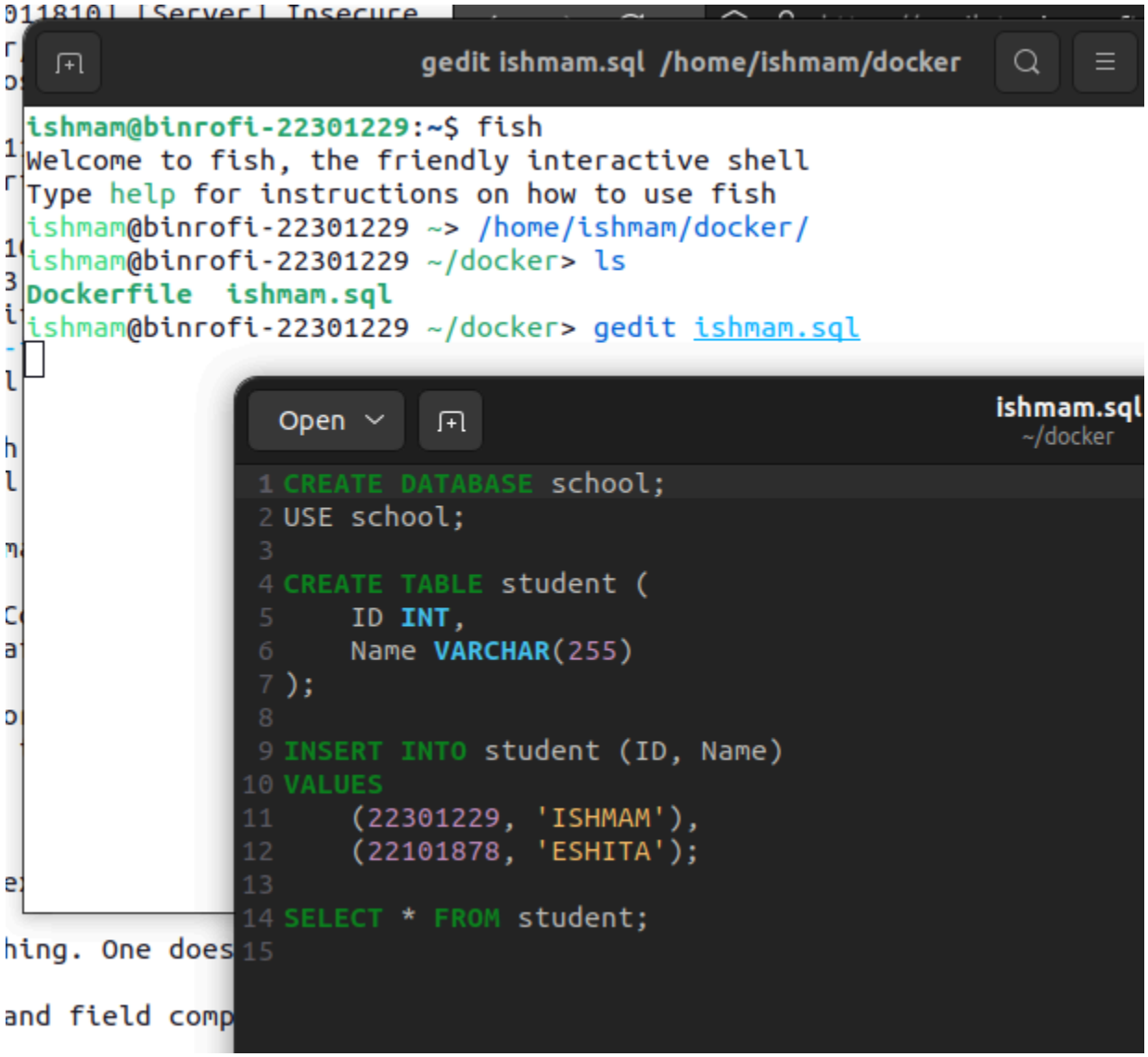
```
ishmam@binrofi-22301229 ~> sudo docker run -d --name some-mysql -e MY
SQL_ROOT_PASSWORD=my-secret-pw mysql:latest
[sudo] password for ishmam:
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
81badc5f380f: Pull complete
c490e5dd1a9d: Pull complete
87aeb61f1478: Pull complete
1cacbea6ceda: Pull complete
1e72891ace67: Waiting
42b720363d36: Download complete
6b3b50f9990a: Downloading  24.74MB/63.08MB
3811d52cfa61: Waiting
05bc7a0277d8: Waiting
cc0abd25a274: Waiting
```

Then I loaded my mysql logs using 'logs'

**$ docker logs some-mysql**

```
ishmam@binrofi-22301229 ~> sudo docker logs some-mysql
2024-02-22 05:12:03+00:00 [Note] [Entrypoint]: Entrypoint script for
MySQL Server 8.3.0-1.el8 started.
2024-02-22 05:12:03+00:00 [Note] [Entrypoint]: Switching to dedicated
 user 'mysql'
2024-02-22 05:12:03+00:00 [Note] [Entrypoint]: Entrypoint script for
MySQL Server 8.3.0-1.el8 started.
2024-02-22 05:12:03+00:00 [Note] [Entrypoint]: Initializing database
files
2024-02-22T05:12:03.482384Z 0 [System] [MY-015017] [Server] MySQL Ser
ver Initialization - start.
2024-02-22T05:12:03.483294Z 0 [System] [MY-013169] [Server] /usr/sbin
/mysqld (mysqld 8.3.0) initializing of server in progress as process
80
2024-02-22T05:12:03.487789Z 1 [System] [MY-013576] [InnoDB] InnoDB in
itialization has started.
2024-02-22T05:12:04.087405Z 1 [System] [MY-013577] [InnoDB] InnoDB in
itialization has ended.
2024-02-22T05:12:05.108669Z 6 [Warning] [MY-010453] [Server] root@loc
alhost is created with an empty password ! Please consider switching
off the --initialize-insecure option.
2024-02-22T05:12:07.724468Z 0 [System] [MY-015018] [Server] MySQL Ser
ver Initialization - end.
2024-02-22 05:12:07+00:00 [Note] [Entrypoint]: Database files initial
ized
2024-02-22 05:12:07+00:00 [Note] [Entrypoint]: Starting temporary ser
ver
2024-02-22T05:12:07.774526Z 0 [System] [MY-015015] [Server] MySQL Ser
```

**Now I created a sql file using touch ishmam.sql then I edited the file and added some sql queries on that file.**



**Then I ran the sql on my container**
**$ mysql -u root -pmy-secret-pw mydb < /home/ishmam/docker/ishmam.sql**

```
ishmam@binrofi-22301229 ~> sudo docker exec -it some-mysql bash
bash-4.4# mysql -u root -pmy-secret-pw school < /home/ishmam/docker/i
shmam.sql
bash: /home/ishmam/docker/ishmam.sql: No such file or directory
bash-4.4# mysql -u root -pmy-secret-pw school /home/ishmam/docker/ish
mam.sql
mysql: [Warning] Using a password on the command line interface can b
e insecure.
mysql  Ver 8.3.0 for Linux on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Usage: mysql [OPTIONS] [database]
  -?, --help          Display this help and exit.
  -I, --help          Synonym for -?
  --auto-rehash       Enable automatic rehashing. One doesn't need to
 use
                      'rehash' to get table and field completion, but
 startup
```

## 7. Push your own image into the Docker public registry/Hub.

First I am tagging my docker image with a tag.

**docker tag hello-world-ishmam ishmambr10/hello-world-ishmam:latest**

```
ishmam@binrofi-22301229 ~ [1]> sudo docker tag hello-world-ishmam ish
mambr10/hello-world-ishmam:latest
ishmam@binrofi-22301229 ~>
```

Then I will push the image to my dockerhub using the command push and my tag

```
docker push ishmambr10/hello-world-ishmam:latest
```

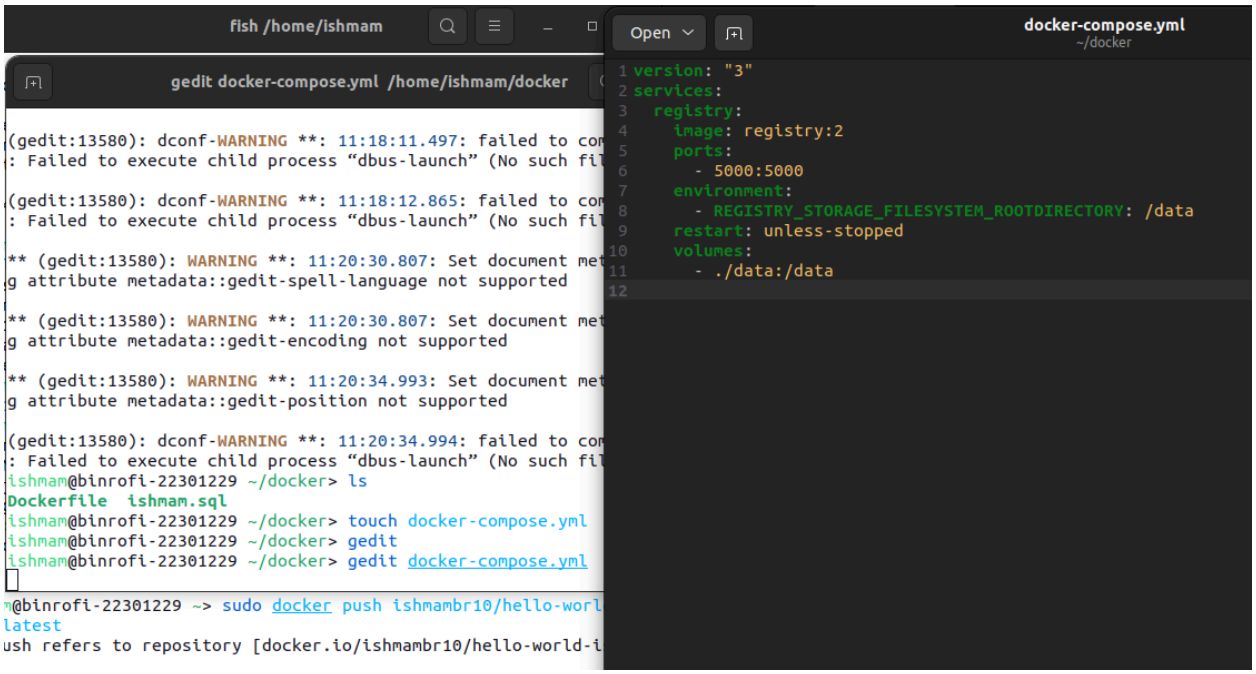
```
ishmam@binrofi-22301229 ~> sudo docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/co
nfig.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentia
ls-store

Login Succeeded
ishmam@binrofi-22301229 ~> sudo docker push ishmambr10/hello-world-is
hmam:latest
The push refers to repository [docker.io/ishmambr10/hello-world-ishma
m]
6af45fe21938: Pushed
541ca455b448: Pushed
d101c9453715: Mounted from library/ubuntu
latest: digest: sha256:e6a5650fdeeadc889e6aace4592b58adaddb04794d9fa3
b3ef9d4c95cfa8b2ea size: 942
ishmam@binrofi-22301229 ~>
```

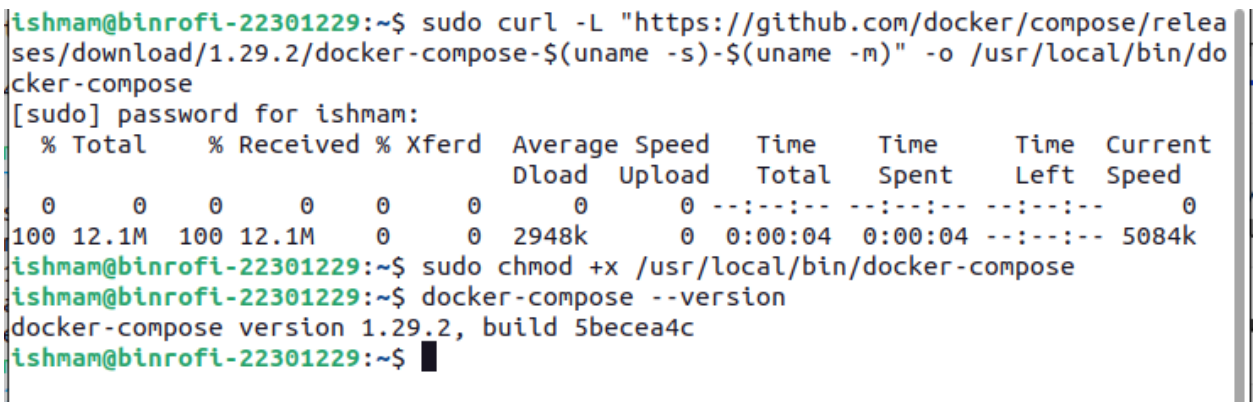
## 8. How to make your own private registry? Show steps.

First I have to create a docker compose file. Here I will create a docker-compose.yml file. Then I will add these into my yml file:

```
version: "3"
services:
  registry:
    image: registry:2
    ports:
      - 5000:5000
    environment:
      - REGISTRY_STORAGE_FILESYSTEM_ROOTDIRECTORY:
/data
    restart: unless-stopped
    volumes:
```

```
    - ./data:/data
```


fish /home/ishmam
gedit docker-compose.yml /home/ishmam/docker
(gedit:13580): dconf-WARNING **: 11:18:11.497: failed to con
: Failed to execute child process “dbus-launch” (No such fil
(gedit:13580): dconf-WARNING **: 11:18:12.865: failed to con
: Failed to execute child process “dbus-launch” (No such fil
** (gedit:13580): WARNING **: 11:20:30.807: Set document met
g attribute metadata::gedit-spell-language not supported
** (gedit:13580): WARNING **: 11:20:30.807: Set document met
g attribute metadata::gedit-encoding not supported
** (gedit:13580): WARNING **: 11:20:34.993: Set document met
g attribute metadata::gedit-position not supported
(gedit:13580): dconf-WARNING **: 11:20:34.994: failed to con
: Failed to execute child process “dbus-launch” (No such fil
ishmam@binrofi-22301229 ~/docker> ls
Dockerfile ishmam.sql
ishmam@binrofi-22301229 ~/docker> touch docker-compose.yml
ishmam@binrofi-22301229 ~/docker> gedit
ishmam@binrofi-22301229 ~/docker> gedit docker-compose.yml
m@binrofi-22301229 ~> sudo docker push ishmambr10/hello-worl
latest
ush refers to repository [docker.io/ishmambr10/hello-world-i
Open
docker-compose.yml
~/docker
1 version: "3"
2 services:
3 registry:
4 image: registry:2
5 ports:
6 - 5000:5000
7 environment:
8 - REGISTRY_STORAGE_FILESYSTEM_ROOTDIRECTORY: /data
9 restart: unless-stopped
10 volumes:
11 - ./data:/data
12


ishmam@binrofi-22301229:~$ sudo curl -L "https://github.com/docker/compose/relea
ses/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/do
cker-compose
[sudo] password for ishmam:
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0
100 12.1M 100 12.1M 0 0 2948k 0 0:00:04 0:00:04 --:--:-- 5084k
ishmam@binrofi-22301229:~$ sudo chmod +x /usr/local/bin/docker-compose
ishmam@binrofi-22301229:~$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c
ishmam@binrofi-22301229:~$

**Then I will save the file and run docker-compose up -d to launch my registry.**

```
ishmam@binrofi-22301229 ~/docker [1]> sudo docker-compose up -d
Creating network "docker_default" with the default driver
Pulling registry (registry:2)...
2: Pulling from library/registry
619be1103602: Pull complete
2ba4b87859f5: Pull complete
0da701e3b4d6: Pull complete
14a4d5d702c7: Pull complete
d1a4f6454cb2: Pull complete
Digest: sha256:f4e1b878d4bc40a1f65532d68c94dcfbab56aa8cba1f00e355a206
e7f6cc9111
Status: Downloaded newer image for registry:2
Creating docker_registry_1 ... done
ishmam@binrofi-22301229 ~/docker>
```

Now I will tag an image using a path which resolves to my registry.

**`docker tag ubuntu localhost:5000/ubuntu`**

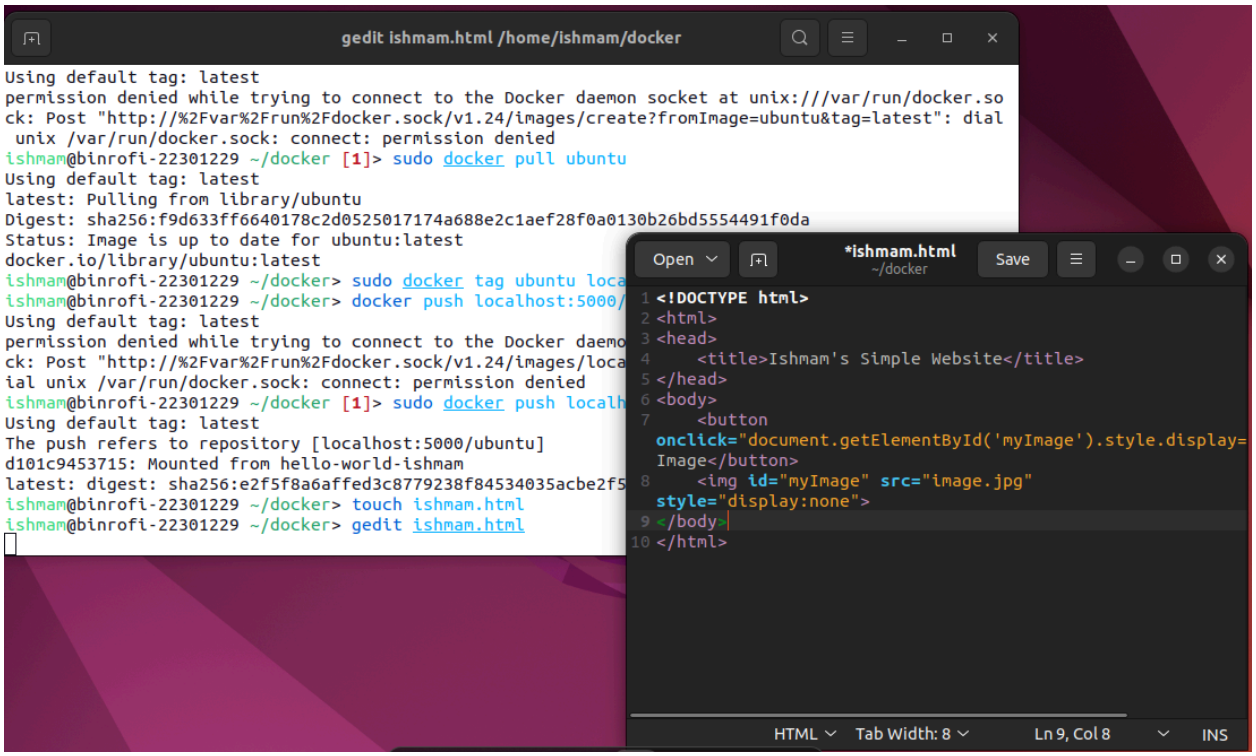
**`docker push localhost:5000/ubuntu`**

```
ishmam@binrofi-22301229 ~/docker [1]> sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:f9d633ff6640178c2d0525017174a688e2c1aef28f0a0130b26bd5554491f0da
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest
ishmam@binrofi-22301229 ~/docker> sudo docker tag ubuntu localhost:5000/ubuntu
ishmam@binrofi-22301229 ~/docker> docker push localhost:5000/ubuntu
Using default tag: latest
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.so
ck: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/images/localhost:5000/ubuntu/push?tag=latest": d
ial unix /var/run/docker.sock: connect: permission denied
ishmam@binrofi-22301229 ~/docker [1]> sudo docker push localhost:5000/ubuntu
Using default tag: latest
The push refers to repository [localhost:5000/ubuntu]
d101c9453715: Mounted from hello-world-ishmam
latest: digest: sha256:e2f5f8a6affed3c8779238f84534035acbe2f53dcc404d7cbbe0ecdf4ac23999 size: 529
ishmam@binrofi-22301229 ~/docker>
```

## 9. Create a simple website using docker

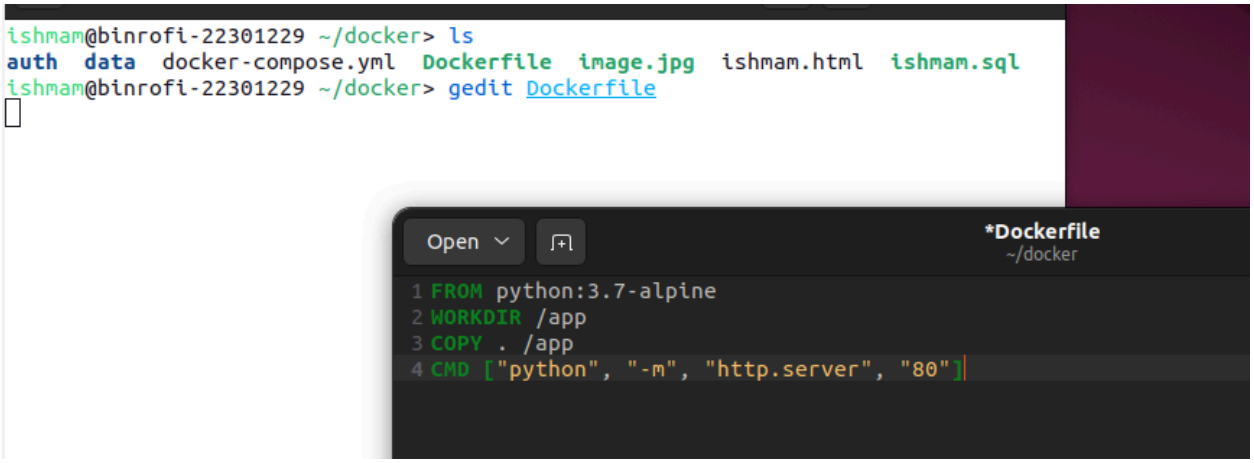
First I will create a html file named ishmam.html. Here the docker file will just show a picture that I have in my folder.

```
<!DOCTYPE html>
<html>
<head>
    <title>My Simple Website</title>
</head>
<body>
    <button
onclick="document.getElementById('myImage').style.display='block
'">Show Image</button>
    <img id="myImage" src="image.jpg" style="display:none">
</body>
</html>
```



Then I have to edit my Dockerfile and setup a python server. Here I will write this on my Dockerfile

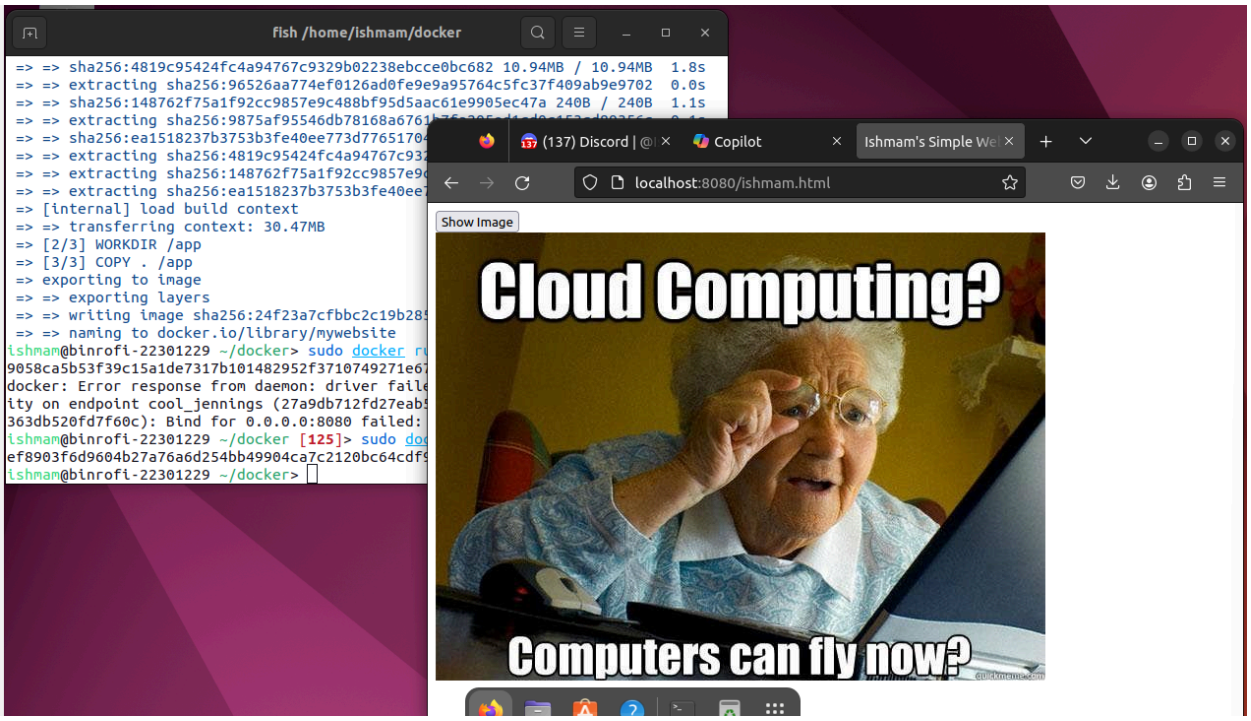
```
FROM python:3.7-alpine
WORKDIR /app
COPY . /app
CMD [“python”, “-m”, “http.server”, “80”]
```




Then I will build my docker image usind docker build -t mywebsite .

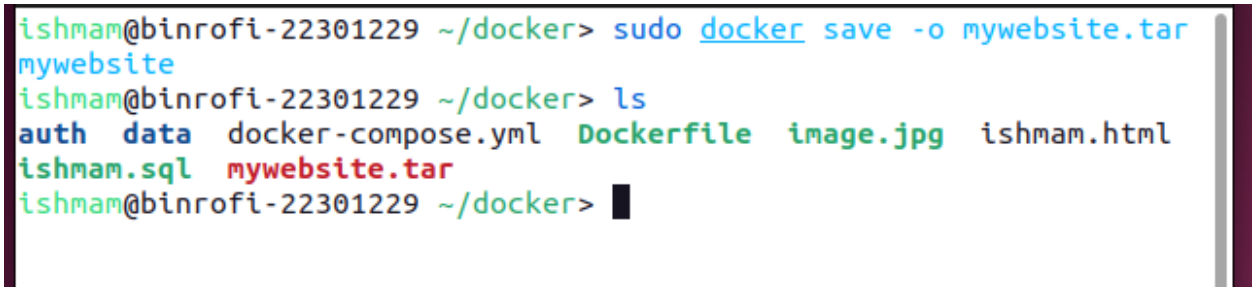
```
ishmam@binrofi-22301229 ~/docker [SIGINT]> sudo docker build -t mywebsite .
[sudo] password for ishmam:
[+] Building 6.0s (9/9) FINISHED                                docker:default
 => [internal] load build definition from Dockerfile                        0.0s
 => => transferring dockerfile: 127B                                        0.0s
 => [internal] load metadata for docker.io/library/python:3.7-alpine        3.4s
 => [auth] library/python:pull token for registry-1.docker.io               0.0s
 => [internal] load .dockerignore                                           0.0s
 => => transferring context: 2B                                             0.0s
 => [1/3] FROM docker.io/library/python:3.7-alpine@sha256:f3d31c8677d03f0   2.3s
 => => resolve docker.io/library/python:3.7-alpine@sha256:f3d31c8677d03f0   0.0s
```

Then I will run and up my docker image . So it will connect to my local host, so now I will open my browser and go to “localhost:8080/ishmam.html”. Now I can see my static website.




## 10. Migrate the new container having the application into another machine. Again run the container and browse the URL. It should work.

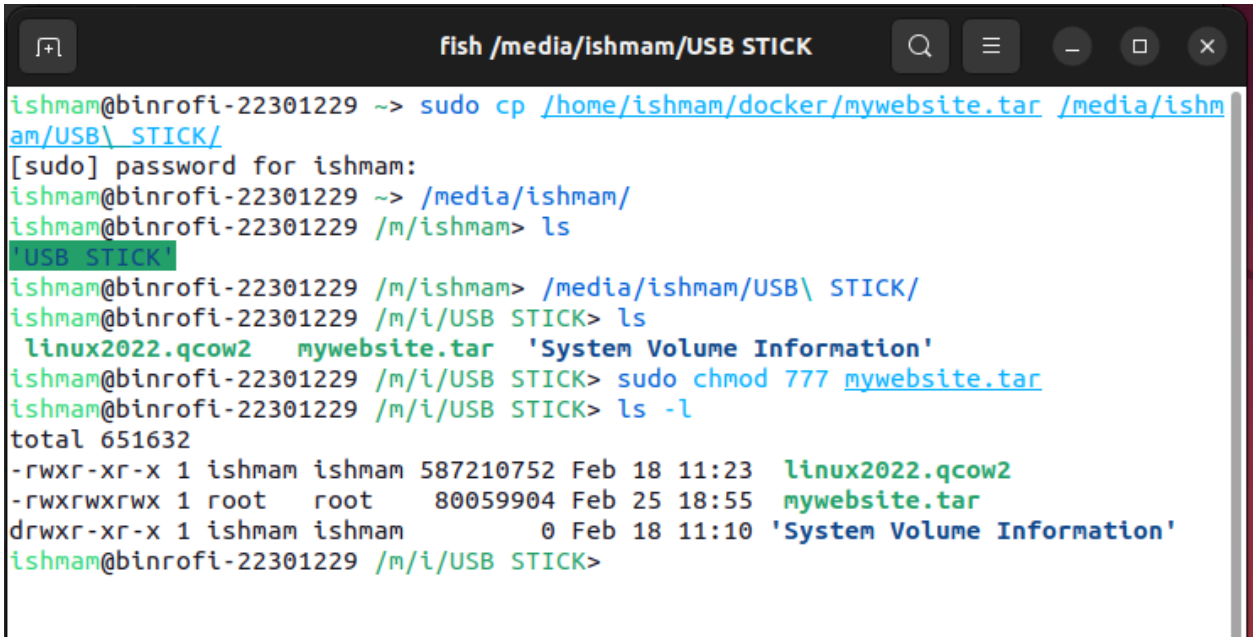
First on my device I saved my docker image using the save -o command and made my mywebsite as mywebsite.tar ….
Then I made my tar file 777 permission.

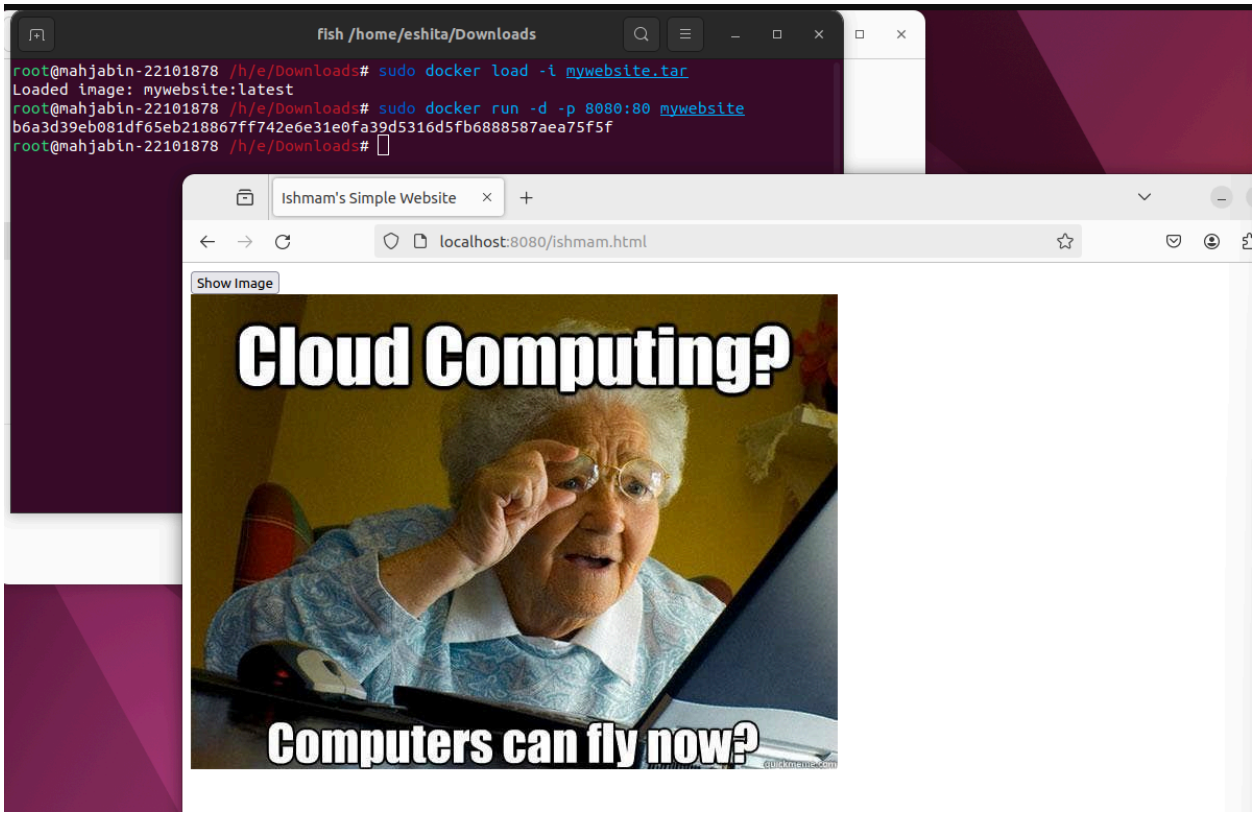


Now I copied my mywebsite.tar file to a usb stick. Now I entered the usb to my friend Eshita's pc and then loaded my website using

```
'''
sudo docker load -i mywebsite.tar
Then I ran the docker file using
sudo docker run -d -p 8080:80 mywebsite
'''
```

Now on Eshita's pc I can run my website using the previous url. Here we can see that my website worked on Eshita's pc.




fish /home/eshita/Downloads
root@mahjabin-22101878 /h/e/Downloads# sudo docker load -i mywebsite.tar
Loaded image: mywebsite:latest
root@mahjabin-22101878 /h/e/Downloads# sudo docker run -d -p 8080:80 mywebsite
b6a3d39eb081df65eb218867ff742e6e31e0fa39d5316d5fb6888587aea75f5f
root@mahjabin-22101878 /h/e/Downloads#
Ishmam's Simple Website
localhost:8080/ishmam.html
Show Image
Cloud Computing?
Computers can fly now?