Assignment - 1

Tasks To Be Performed:

1. Pull Ubuntu container

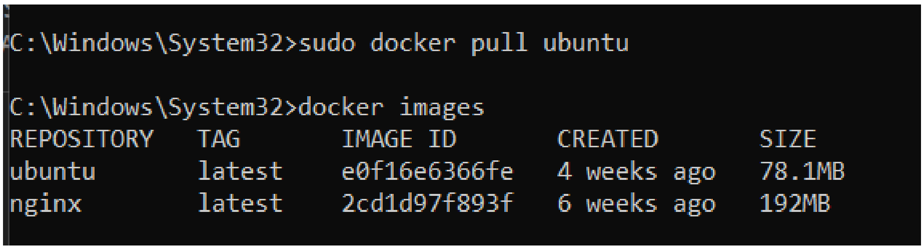
2. Run this container and map port 80 on the local

3. Install Apache2 on this container

4. Check if you are able to access the Apache page on your browser

==
==

1. Pull Ubuntu container

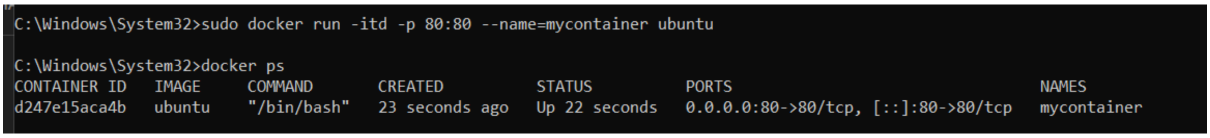


```
C:\Windows\System32>sudo docker pull ubuntu

C:\Windows\System32>docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
ubuntu       latest    e0f16e6366fe   4 weeks ago   78.1MB
nginx        latest    2cd1d97f893f   6 weeks ago   192MB
```

2. Run this container and map port 80 on the local

- We can create one container at one port
- Created container using the command below

  sudo docker run -itd -p 80:80 --name=mycontainer ubuntu

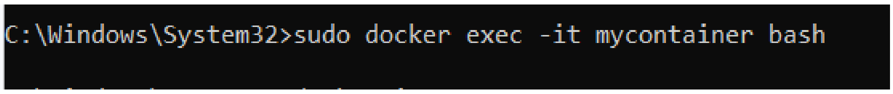


```
C:\Windows\System32>sudo docker run -itd -p 80:80 --name=mycontainer ubuntu

C:\Windows\System32>docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS          PORTS                                NAMES
d247e15aca4b   ubuntu    "/bin/bash"   23 seconds ago   Up 22 seconds   0.0.0.0:80->80/tcp, [::]:80->80/tcp   mycontainer
```

3. Install Apache2 on this container

- First we will go inside the container using the command below:

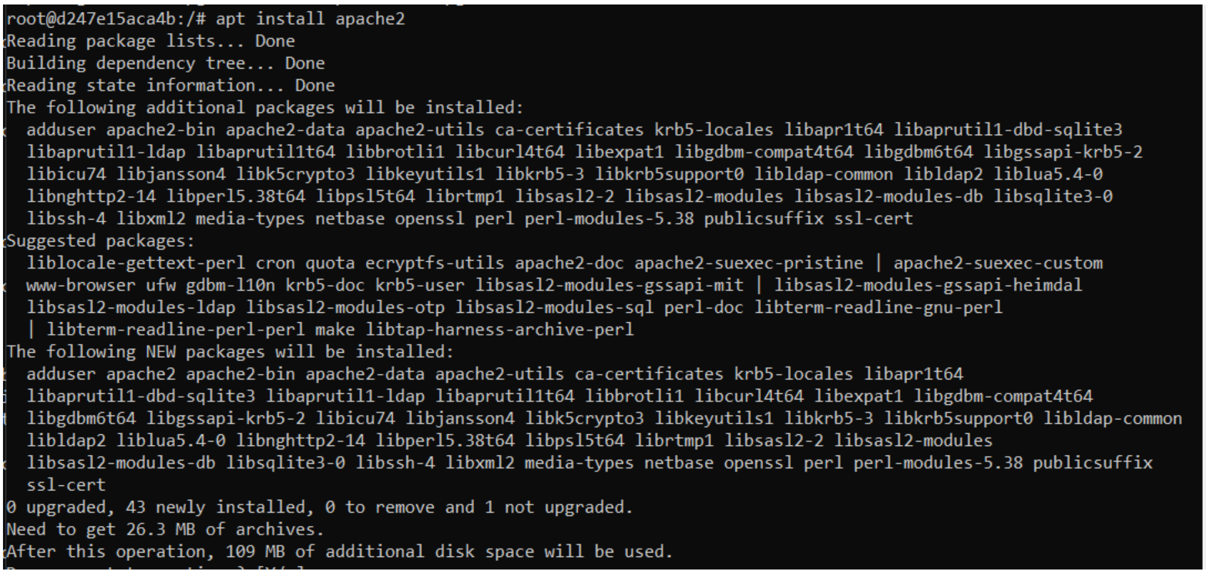
sudo docker exec -it a1container bash



```
C:\Windows\System32>sudo docker exec -it mycontainer bash
```

- Now install the Apache using

apt install apache2



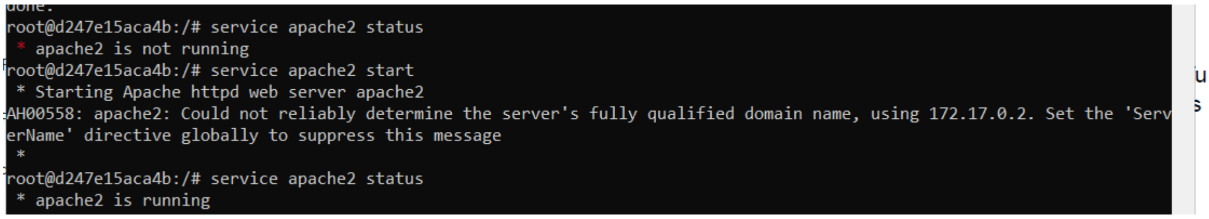
```
root@d247e15aca4b:/# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adduser apache2-bin apache2-data apache2-utils ca-certificates krb5-locales libapr1t64 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libaprutil1t64 libbrotli1 libcurl4t64 libexpat1 libgdbm-compat4t64 libgdbm6t64 libgssapi-krb5-2
  libicu74 libjansson4 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libldap-common libldap2 liblua5.4-0
  libnghttp2-14 libperl5.38t64 libpsl5t64 librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db libsqlite3-0
  libssh-4 libxml2 media-types netbase openssl perl perl-modules-5.38 publicsuffix ssl-cert
Suggested packages:
  liblocale-gettext-perl cron quota ecryptfs-utils apache2-doc apache2-suexec-pristine | apache2-suexec-custom
  www-browser ufw gdbm-l10n krb5-doc krb5-user libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal
  libsasl2-modules-ldap libsasl2-modules-otp libsasl2-modules-sql perl-doc libterm-readline-gnu-perl
  | libterm-readline-perl-perl make libtap-harness-archive-perl
The following NEW packages will be installed:
  adduser apache2 apache2-bin apache2-data apache2-utils ca-certificates krb5-locales libapr1t64
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 libbrotli1 libcurl4t64 libexpat1 libgdbm-compat4t64
  libgdbm6t64 libgssapi-krb5-2 libicu74 libjansson4 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libldap-common
  libldap2 liblua5.4-0 libnghttp2-14 libperl5.38t64 libpsl5t64 librtmp1 libsasl2-2 libsasl2-modules
  libsasl2-modules-db libsqlite3-0 libssh-4 libxml2 media-types netbase openssl perl perl-modules-5.38 publicsuffix
  ssl-cert
0 upgraded, 43 newly installed, 0 to remove and 1 not upgraded.
Need to get 26.3 MB of archives.
After this operation, 109 MB of additional disk space will be used.
```

- After installing Apache will check the status using the command below:

service apache2 status

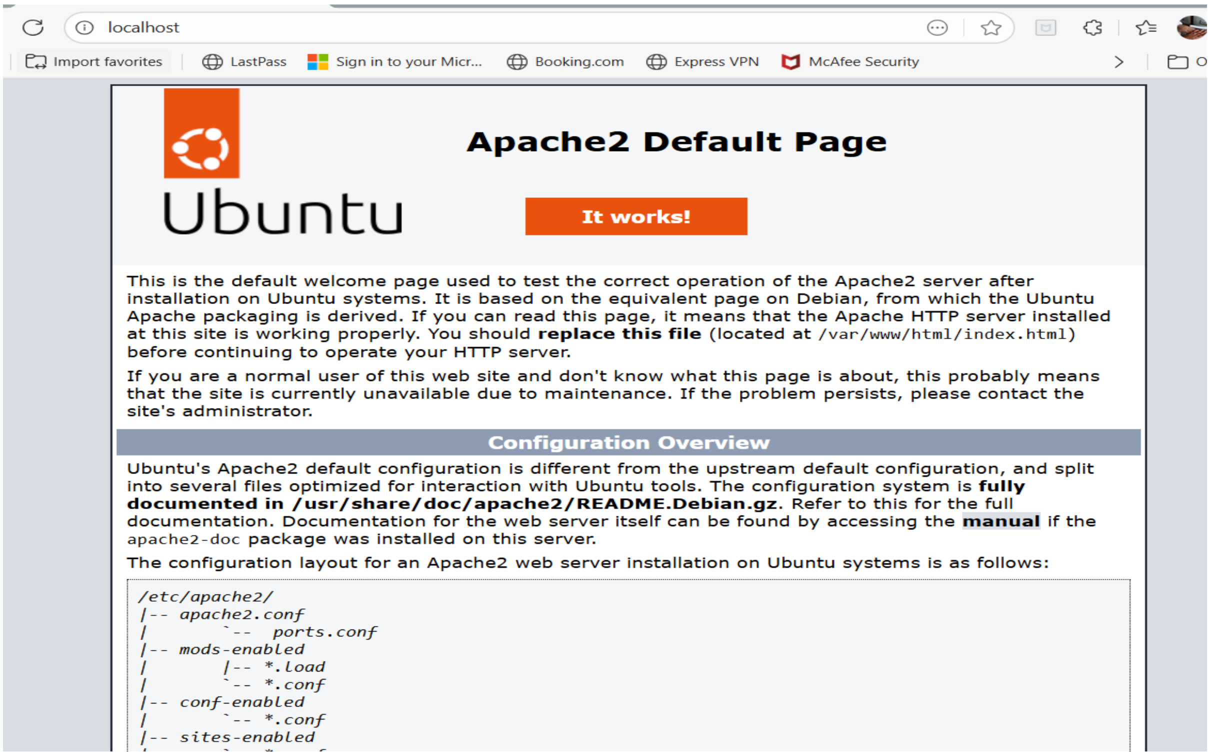
- As it is showing not running we will start Apache using the command below:

service apache2 start



```
root@d247e15aca4b:/# service apache2 status
 * apache2 is not running
root@d247e15aca4b:/# service apache2 start
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
 *
root@d247e15aca4b:/# service apache2 status
 * apache2 is running
```

4. Check if you are able to access the Apache page on your browser



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
```

==

==

Assignment - 2

Tasks To Be Performed:

1. Save the image created in assignment 1 as a Docker image

2. Launch container from this new image and map the port to 81

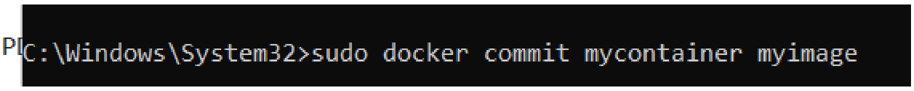
3. Go inside the container and start the Apache2 service

4. Check if you are able to access it on the browser

1. Save the image created in assignment 1 as a Docker image

- Once we create the image from the container which we created in assignment1 it will already have apache installed in it
- To create image from the container we will use command

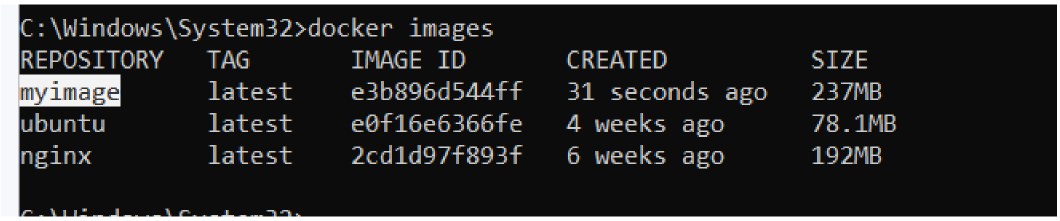
sudo docker commit <container name> <giveimage name>



```
PI C:\Windows\System32>sudo docker commit mycontainer myimage
```

Screen clipping taken: 27-08-2025 02:13 PM

- Now we will check the images using docker images command



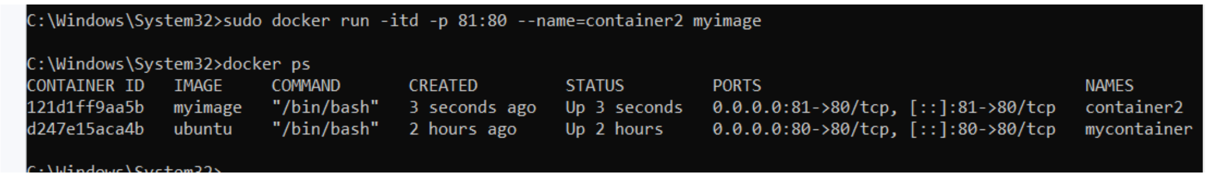
```
C:\Windows\System32>docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
myimage      latest    e3b896d544ff   31 seconds ago   237MB
ubuntu       latest    e0f16e6366fe   4 weeks ago      78.1MB
nginx        latest    2cd1d97f893f   6 weeks ago      192MB

C:\Windows\System32>
```

Screen clipping taken: 27-08-2025 02:14 PM

2. Launch container from this new image and map the port to 81

- Now we will create container using the same image using the command below

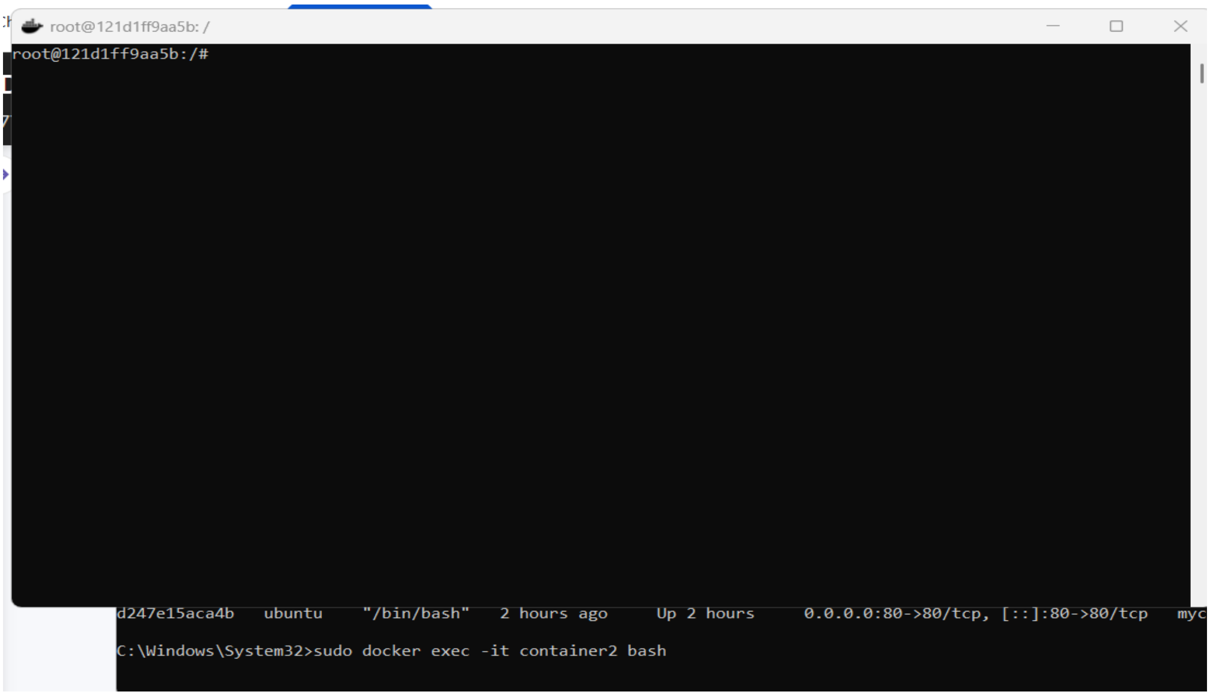
Sudo docker run -itd -p 81:80 --name=mycontainer2 myimage



```
C:\Windows\System32>sudo docker run -itd -p 81:80 --name=container2 myimage

C:\Windows\System32>docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS         PORTS                                 NAMES
121d1ff9aa5b   myimage   "/bin/bash"   3 seconds ago    Up 3 seconds   0.0.0.0:81->80/tcp, [::]:81->80/tcp   container2
d247e15aca4b   ubuntu    "/bin/bash"   2 hours ago      Up 2 hours     0.0.0.0:80->80/tcp, [::]:80->80/tcp   mycontainer

C:\Windows\System32>
```

3. Go inside the container and start the Apache2 service

- We will got inside container2

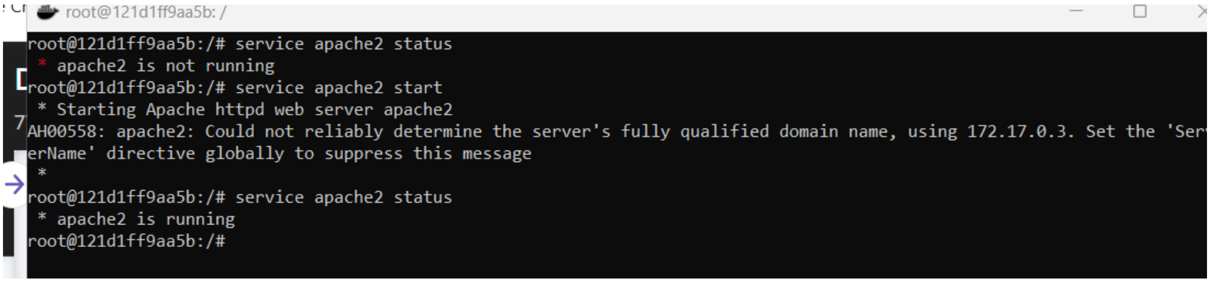


```
root@121d1ff9aa5b:/#
```

```
d247e15aca4b   ubuntu    "/bin/bash"   2 hours ago    Up 2 hours    0.0.0.0:80->80/tcp, [::]:80->80/tcp   myc

C:\Windows\System32>sudo docker exec -it container2 bash
```

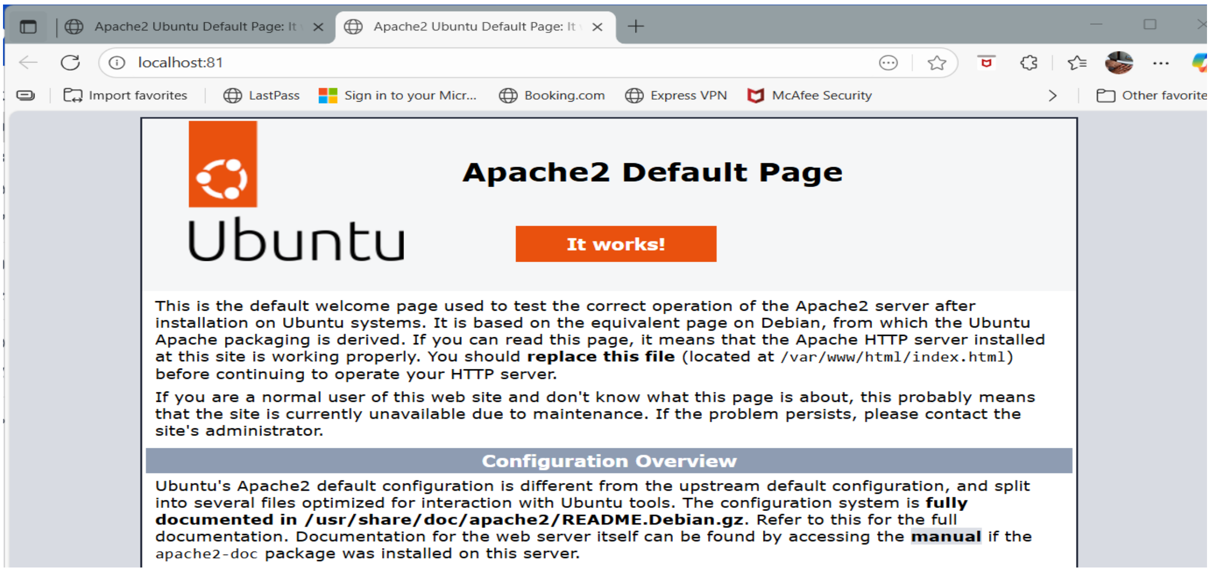
Screen clipping taken: 27-08-2025 02:25 PM

- Now we will start the apache2 service inside the container as it will not automatically start the services



```
root@121d1ff9aa5b:/# service apache2 status
 * apache2 is not running
root@121d1ff9aa5b:/# service apache2 start
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'Serv
erName' directive globally to suppress this message
 *
root@121d1ff9aa5b:/# service apache2 status
 * apache2 is running
root@121d1ff9aa5b:/#
```

4. Check if you are able to access it on the browser



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

Screen clipping taken: 27-08-2025 02:30 PM

==

==

Assignment - 3

Tasks To Be Performed:

1. Use the saved image in the previous assignment

2. Upload this image on Docker Hub

3. On a separate machine pull this Docker Hub image and launch it on port

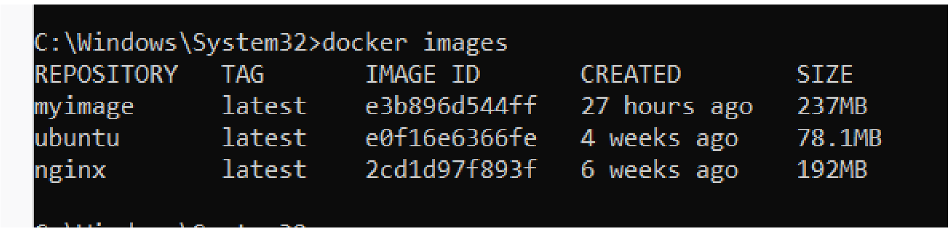
80

4. Start the Apache2 service

5. Verify if you are able to see the Apache2 service

1. Use the saved image in the previous assignment

- We will check the image which we needs to upload on docker hub

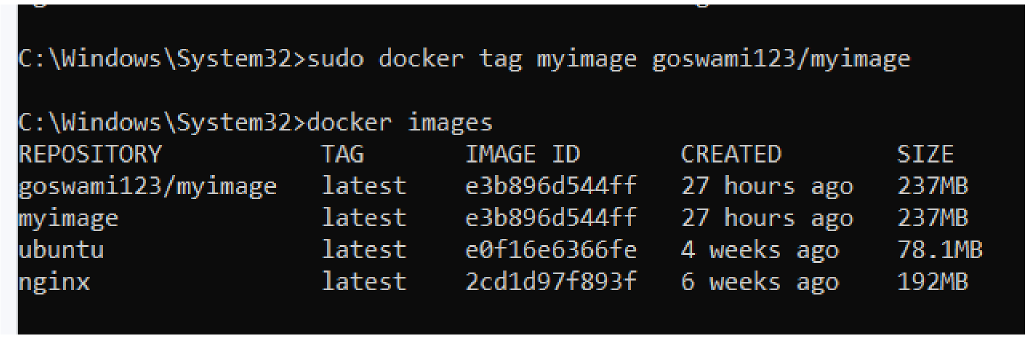


```
C:\Windows\System32>docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
myimage       latest    e3b896d544ff   27 hours ago    237MB
ubuntu        latest    e0f16e6366fe   4 weeks ago     78.1MB
nginx         latest    2cd1d97f893f   6 weeks ago     192MB
```

- This image should be in a format in order to upload on docker hub
- Logged in to docker hub

2. Upload this image on Docker Hub

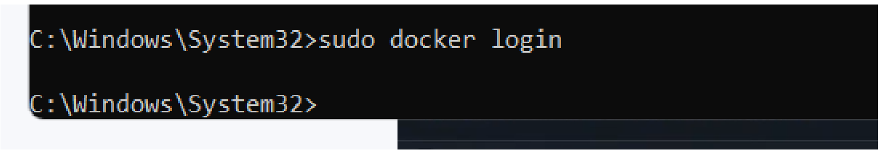
- In docker hub my username is goswami123/myimage (dockerusername/imagename)
- We will rename this image using **sudo docker tag <imagename> <newimagename>**



```
C:\Windows\System32>sudo docker tag myimage goswami123/myimage

C:\Windows\System32>docker images
REPOSITORY           TAG       IMAGE ID       CREATED         SIZE
goswami123/myimage   latest    e3b896d544ff   27 hours ago    237MB
myimage              latest    e3b896d544ff   27 hours ago    237MB
ubuntu               latest    e0f16e6366fe   4 weeks ago     78.1MB
nginx                latest    2cd1d97f893f   6 weeks ago     192MB
```

Screen clipping taken: 28-08-2025 05:25 PM

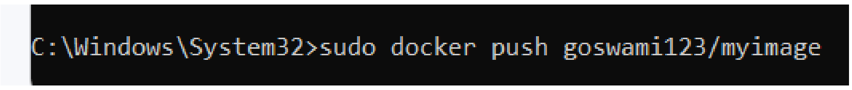
- Now we will login using sudo docker login, it will ask for credentials and authenticate



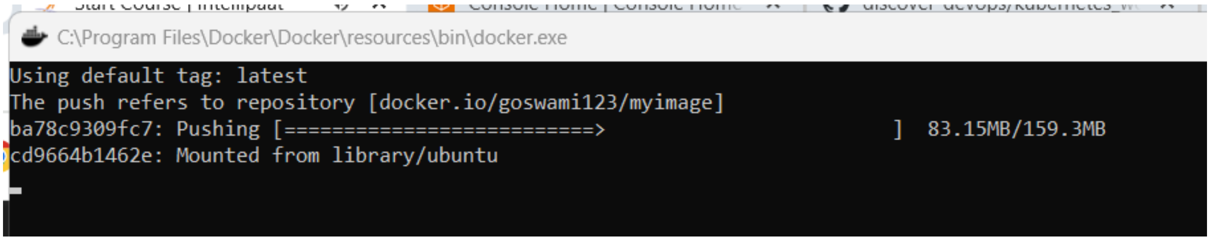
```
C:\Windows\System32>sudo docker login

C:\Windows\System32>
```

Screen clipping taken: 28-08-2025 05:32 PM

- Now we will push the image using sudo docker push <image name>



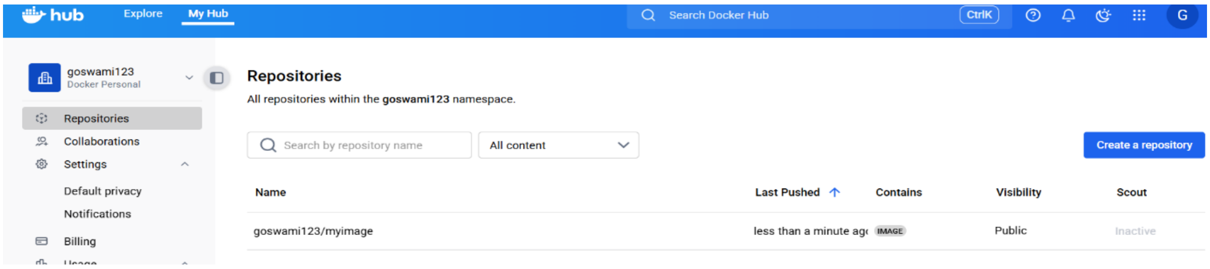
```
C:\Windows\System32>sudo docker push goswami123/myimage
```



```
Using default tag: latest
The push refers to repository [docker.io/goswami123/myimage]
ba78c9309fc7: Pushing [========================>                          ]  83.15MB/159.3MB
cd9664b1462e: Mounted from library/ubuntu
```

Screen clipping taken: 28-08-2025 05:35 PM

- Now we will see on docker hub the image will be there



Repositories

All repositories within the **goswami123** namespace.

Search by repository name | All content

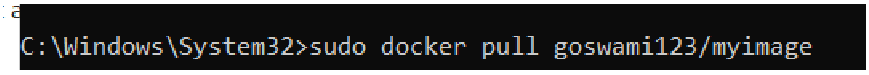
Name	Last Pushed ↑	Contains	Visibility	Scout
goswami123/myimage	less than a minute ago	IMAGE	Public	Inactive

3. On a separate machine pull this Docker Hub image and launch it on port

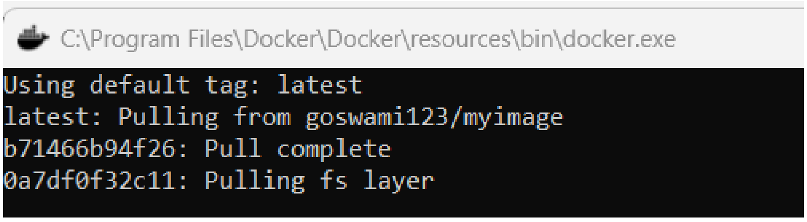
80

- Now we will pull the image from dockerhub using

sudo docker pull goswami123/myimage



```
C:\Windows\System32>sudo docker pull goswami123/myimage
```

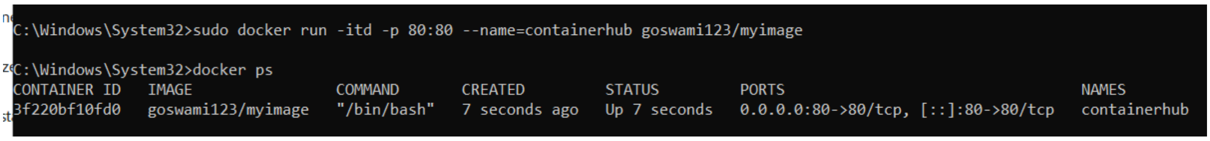


C:\Program Files\Docker\Docker\resources\bin\docker.exe

```
Using default tag: latest
latest: Pulling from goswami123/myimage
b71466b94f26: Pull complete
0a7df0f32c11: Pulling fs layer
```

- Now we will luanch the container at port number 80 using

sudo docker run -itd -p 80:80 --name=containerhub goswami123/myimage



```
C:\Windows\System32>sudo docker run -itd -p 80:80 --name=containerhub goswami123/myimage

C:\Windows\System32>docker ps
CONTAINER ID   IMAGE                COMMAND       CREATED         STATUS         PORTS                                 NAMES
3f220bf10fd0   goswami123/myimage   "/bin/bash"   7 seconds ago   Up 7 seconds   0.0.0.0:80->80/tcp, [::]:80->80/tcp   containerhub
```

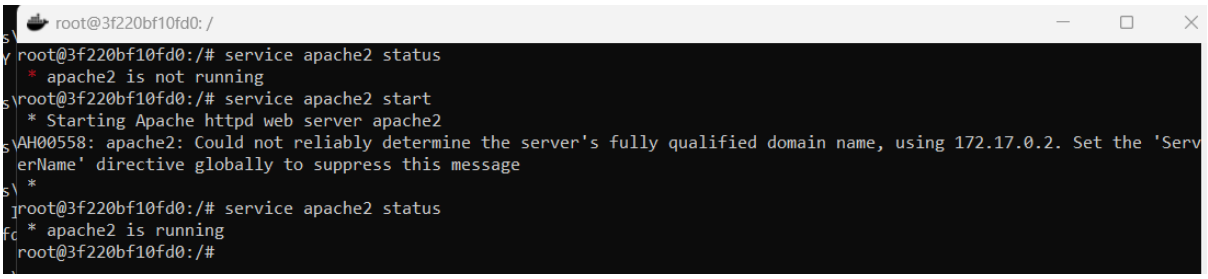
Screen clipping taken: 28-08-2025 05:44 PM

4. Start the Apache2 service

- We will go inside the container using

  sudo docker exec -it containerhub bash

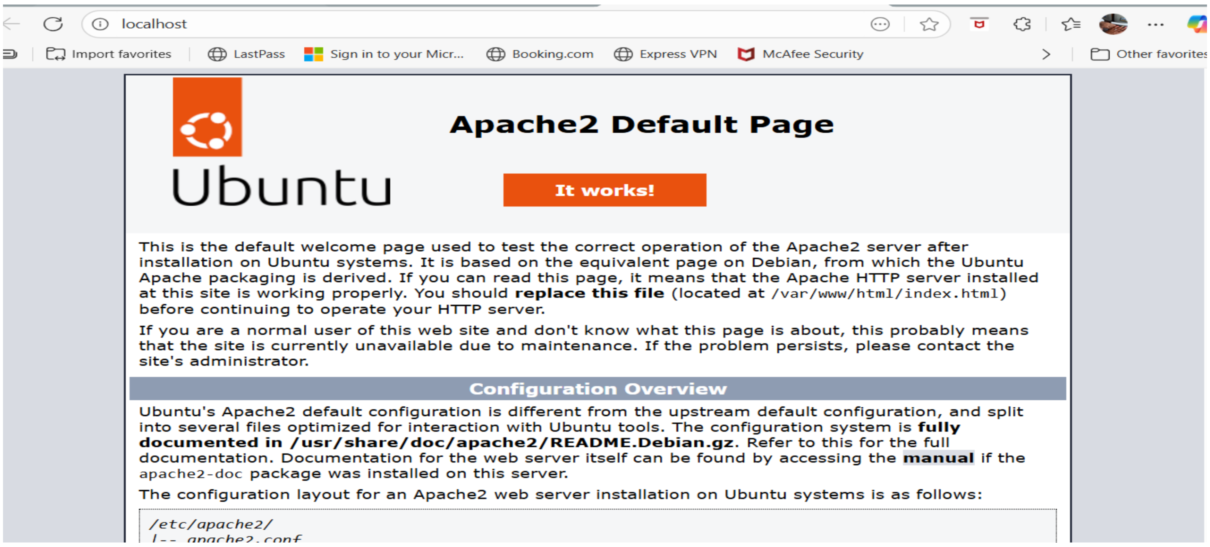
- We will start the apache service



root@3f220bf10fd0: /

```
root@3f220bf10fd0:/# service apache2 status
 * apache2 is not running
root@3f220bf10fd0:/# service apache2 start
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'Serv
erName' directive globally to suppress this message
 *
root@3f220bf10fd0:/# service apache2 status
 * apache2 is running
root@3f220bf10fd0:/#
```

5. Verify if you are able to see the Apache2 service



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
```

Screen clipping taken: 28-08-2025 05:47 PM

==

Assignment - 4

Tasks To Be Performed:

1. Create a Dockerfile with the following specs:

● Ubuntu container

● Apache2 installed

● Apache2 should automatically run once the container starts

2. Submit the Dockerfile for assignment completion

1. Create a sample HTML file

2. Use the Dockerfile from the previous task

3. Replace this sample HTML file inside the Docker container with the default

Page

- First we will create docker file

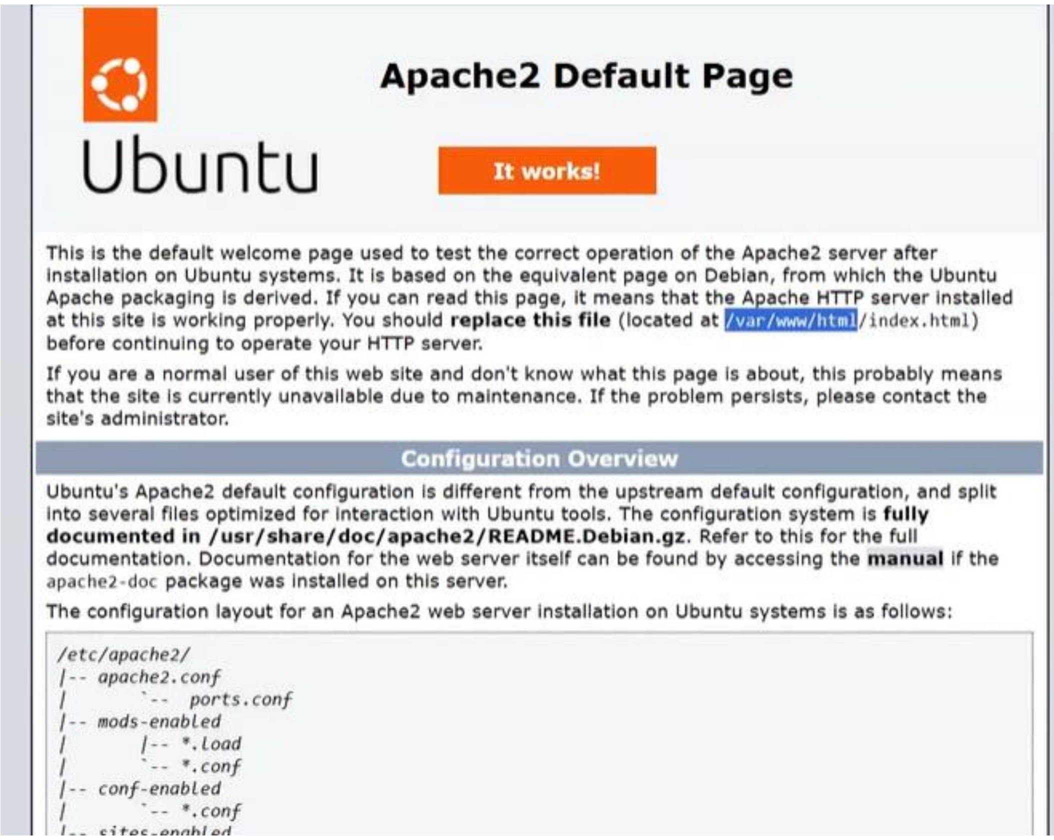
FROM ubuntu (as we need ubuntu image in first step)

RUN apt update (we need to update the ubuntu)

RUN apt install apache2 -y (we need to install apache2)

ENTRYPOINT apachectl -D FOREGROUND (this will make sure that apache2 is keeps running in the background when the container will be started)

- Default page of apache2 is shown below, where the html path is mentioned



Ubuntu

Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
```

Screen clipping taken: 16-09-2025 10:37 AM

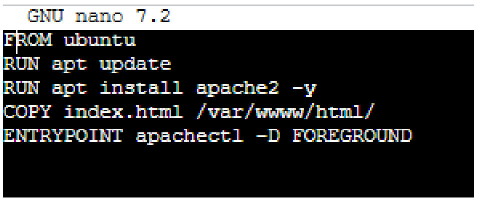
/var/www/html/index.html

- Now we will copy the index.html file to the above destination

So we will add this also in the docker file

COPY index.html (file which we want to copy) /var/www/html/ (destination

- Created docker file

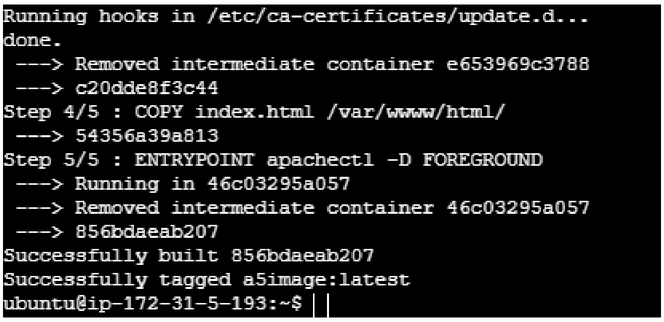


```
  GNU nano 7.2
FROM ubuntu
RUN apt update
RUN apt install apache2 -y
COPY index.html /var/wwww/html/
ENTRYPOINT apachectl -D FOREGROUND
```

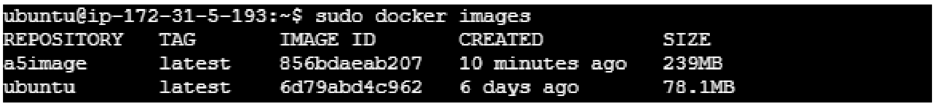
Screen clipping taken: 16-09-2025 11:58 AM

- Created indexfile
- Created image using

Docker build . -t a5image



```
Running hooks in /etc/ca-certificates/update.d...
done.
 ---> Removed intermediate container e653969c3788
 ---> c20dde8f3c44
Step 4/5 : COPY index.html /var/wwww/html/
 ---> 54356a39a813
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
 ---> Running in 46c03295a057
 ---> Removed intermediate container 46c03295a057
 ---> 856bdaeab207
Successfully built 856bdaeab207
Successfully tagged a5image:latest
ubuntu@ip-172-31-5-193:~$ 
```

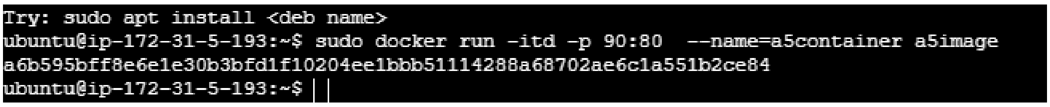


```
ubuntu@ip-172-31-5-193:~$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
a5image      latest    856bdaeab207   10 minutes ago   239MB
ubuntu       latest    6d79abd4c962   6 days ago       78.1MB
```

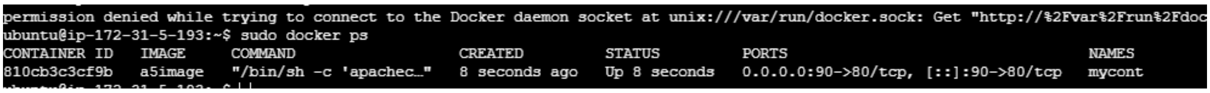
Screen clipping taken: 16-09-2025 12:10 PM

- Now we will create the container using this image at port number 90

Sudo docker run -itd -p 90:80 --name=a5container a5image

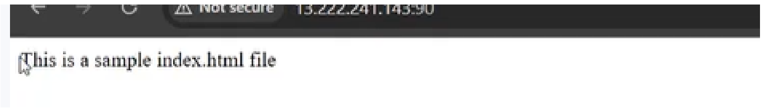


```
Try: sudo apt install <deb name>
ubuntu@ip-172-31-5-193:~$ sudo docker run -itd -p 90:80  --name=a5container a5image
a6b595bff8e6e1e30b3bfd1f10204ee1bbb51114288a68702ae6c1a551b2ce84
ubuntu@ip-172-31-5-193:~$ 
```



```
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdoc
ubuntu@ip-172-31-5-193:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                 NAMES
810cb3c3cf9b   a5image   "/bin/sh -c 'apachec…"   8 seconds ago   Up 8 seconds   0.0.0.0:90->80/tcp, [::]:90->80/tcp   mycont
```

- Now we will access the page and it will show this



This is a sample index.html file