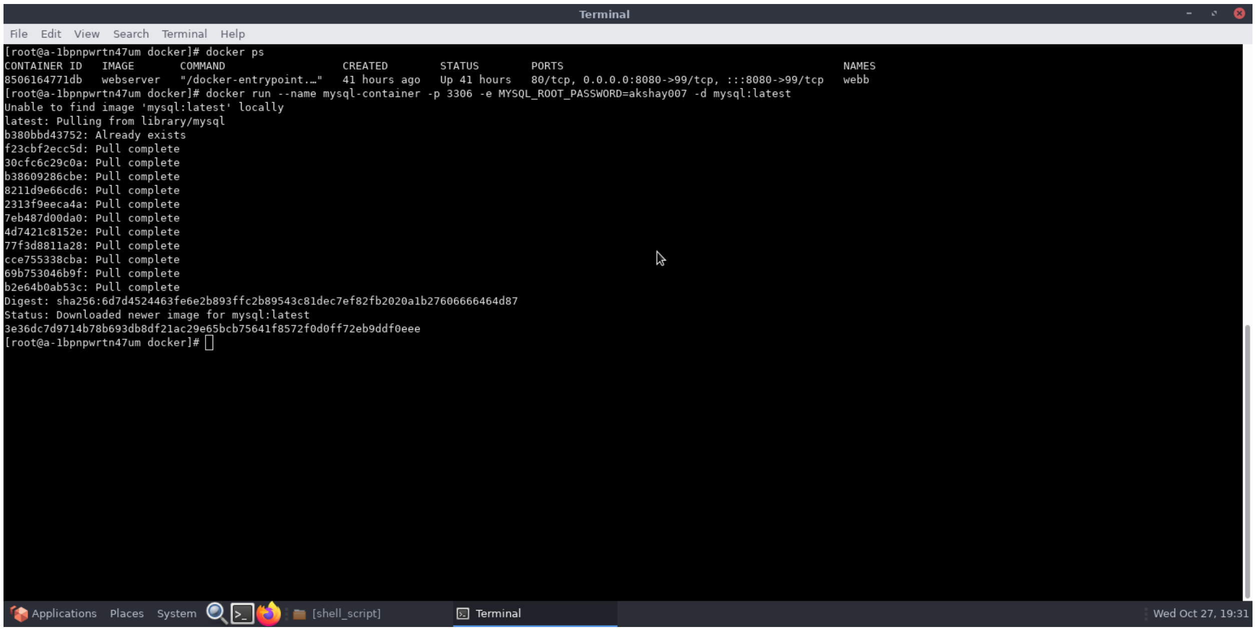
MYSQL

1. Create a MySql Database named **flask_example_app** inside the docker container named **mysql-container** running the image **mysql:latest**. Create a Table named **users** with attributes *username* and *password* of type *varchar*

Answer:

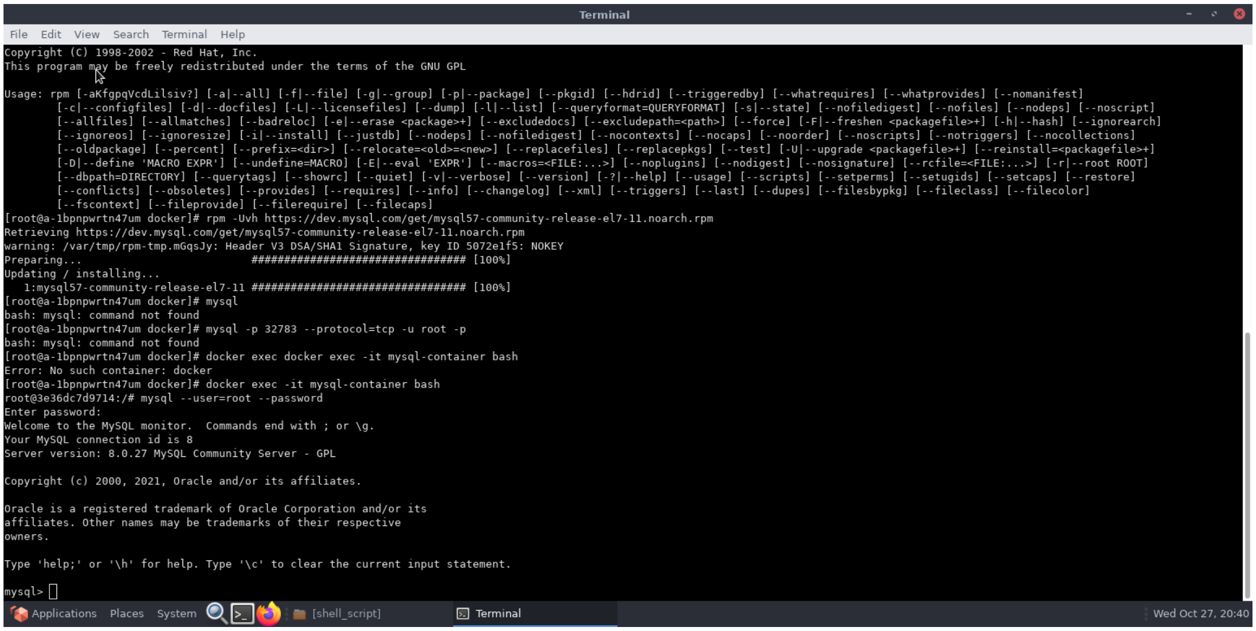
In support of this made a container named as mysql-container and after running it created the required database and table.

Here are the Screeenshots:

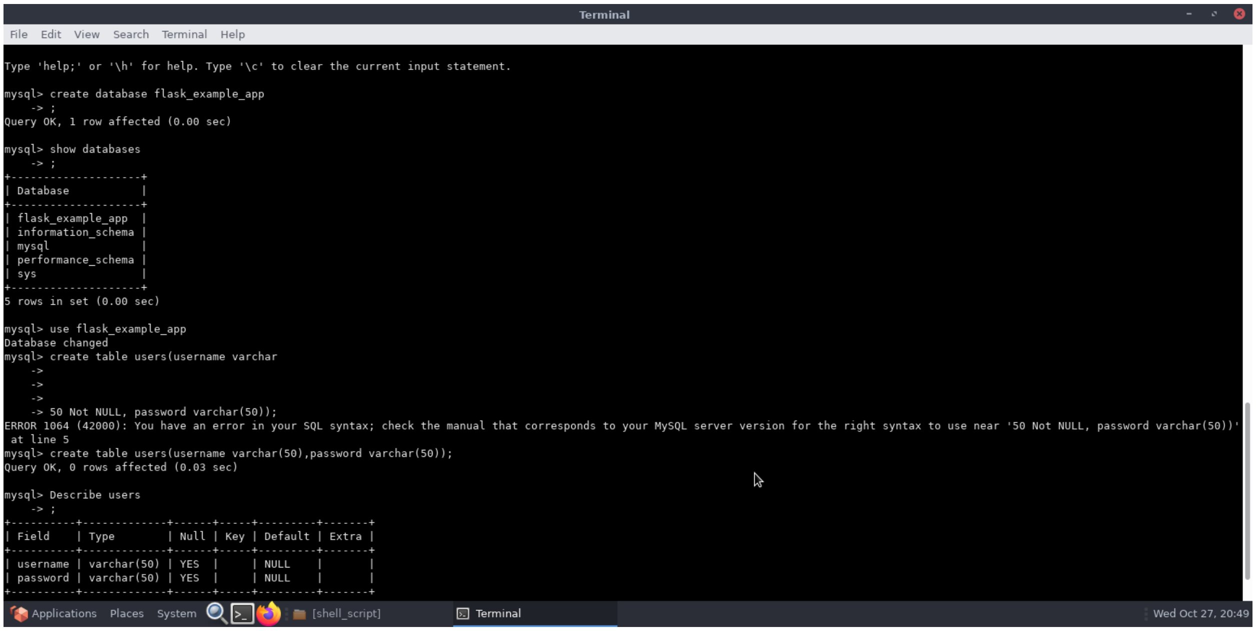


Terminal
File Edit View Search Terminal Help
Applications Places System
[shell_script]
Terminal
Wed Oct 27, 19:31

```
[root@a-1bpnpwrtn47um docker]# docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED        STATUS        PORTS                                                NAMES
8506164771db   webserver   "/docker-entrypoint.…"   41 hours ago   Up 41 hours   80/tcp, 0.0.0.0:8080->99/tcp, :::8080->99/tcp   webb
[root@a-1bpnpwrtn47um docker]# docker run --name mysql-container -p 3306 -e MYSQL_ROOT_PASSWORD=akshay007 -d mysql:latest
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
b380bbd43752: Already exists
f23cbf2ecc5d: Pull complete
30cfc6c29c0a: Pull complete
b38609286cbe: Pull complete
8211d9e66cd6: Pull complete
2313f9eeca4a: Pull complete
7eb487d00da0: Pull complete
4d7421c8152e: Pull complete
77f3d8811a28: Pull complete
cce755338cba: Pull complete
69b753046b9f: Pull complete
b2e64b0ab53c: Pull complete
Digest: sha256:6d7d4524463fe6e2b893ffc2b89543c81dec7ef82fb2020a1b27606666464d87
Status: Downloaded newer image for mysql:latest
3e36dc7d9714b78b693db8df21ac29e65bcb75641f8572f0d0ff72eb9ddf0eee
[root@a-1bpnpwrtn47um docker]# 
```


Terminal
File Edit View Search Terminal Help
Applications Places System
[shell_script]
Terminal
Wed Oct 27, 20:40

```
Copyright (C) 1998-2002 - Red Hat, Inc.
This program may be freely redistributed under the terms of the GNU GPL

Usage: rpm [-aKfgpqVcdLilsiv?] [-a|--all] [-f|--file] [-g|--group] [-p|--package] [--pkgid] [--hdrid] [--triggeredby] [--whatrequires] [--whatprovides] [--nomanifest]
        [-c|--configfiles] [-d|--docfiles] [-L|--licensefiles] [--dump] [-l|--list] [--queryformat=QUERYFORMAT] [-s|--state] [--nofiledigest] [--nofiles] [--nodeps] [--noscript]
        [--allfiles] [--allmatches] [--badreloc] [-e|--erase <package>+] [--excludedocs] [--excludepath=<path>] [--force] [-F|--freshen <packagefile>+] [-h|--hash] [--ignorearch]
        [--ignoreos] [--ignoresize] [-i|--install] [--justdb] [--nodeps] [--nofiledigest] [--nocontexts] [--nocaps] [--noorder] [--noscripts] [--notriggers] [--nocollections]
        [--oldpackage] [--percent] [--prefix=<dir>] [--relocate=<old>=<new>] [--replacefiles] [--replacepkgs] [--test] [-U|--upgrade <packagefile>+] [--reinstall=<packagefile>+]
        [-D|--define 'MACRO EXPR'] [--undefine=MACRO] [-E|--eval 'EXPR'] [--macros=<FILE:...>] [--noplugins] [--nodigest] [--nosignature] [--rcfile=<FILE:...>] [-r|--root ROOT]
        [--dbpath=DIRECTORY] [--querytags] [--showrc] [--quiet] [-v|--verbose] [--version] [-?|--help] [--usage] [--scripts] [--setperms] [--setugids] [--setcaps] [--restore]
        [--conflicts] [--obsoletes] [--provides] [--requires] [--info] [--changelog] [--xml] [--triggers] [--last] [--dupes] [--filesbypkg] [--fileclass] [--filecolor]
        [--fscontext] [--fileprovide] [--filerequire] [--filecaps]
[root@a-1bpnpwrtn47um docker]# rpm -Uvh https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
Retrieving https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
warning: /var/tmp/rpm-tmp.mGqsJy: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:mysql57-community-release-el7-11 ################################# [100%]
[root@a-1bpnpwrtn47um docker]# mysql
bash: mysql: command not found
[root@a-1bpnpwrtn47um docker]# mysql -p 32783 --protocol=tcp -u root -p
bash: mysql: command not found
[root@a-1bpnpwrtn47um docker]# docker exec docker exec -it mysql-container bash
Error: No such container: docker
[root@a-1bpnpwrtn47um docker]# docker exec -it mysql-container bash
root@3e36dc7d9714:/# mysql --user=root --password
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, 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.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```


Terminal
File Edit View Search Terminal Help
Applications Places System
[shell_script]
Terminal
Wed Oct 27, 20:49

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database flask_example_app
    -> ;
Query OK, 1 row affected (0.00 sec)

mysql> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| flask_example_app  |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use flask_example_app
Database changed
mysql> create table users(username varchar
    ->
    ->
    ->
    -> 50 Not NULL, password varchar(50));
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '50 Not NULL, password varchar(50))'
 at line 5
mysql> create table users(username varchar(50),password varchar(50));
Query OK, 0 rows affected (0.03 sec)

mysql> Describe users
    -> ;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(50) | YES  |     | NULL    |       |
| password | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

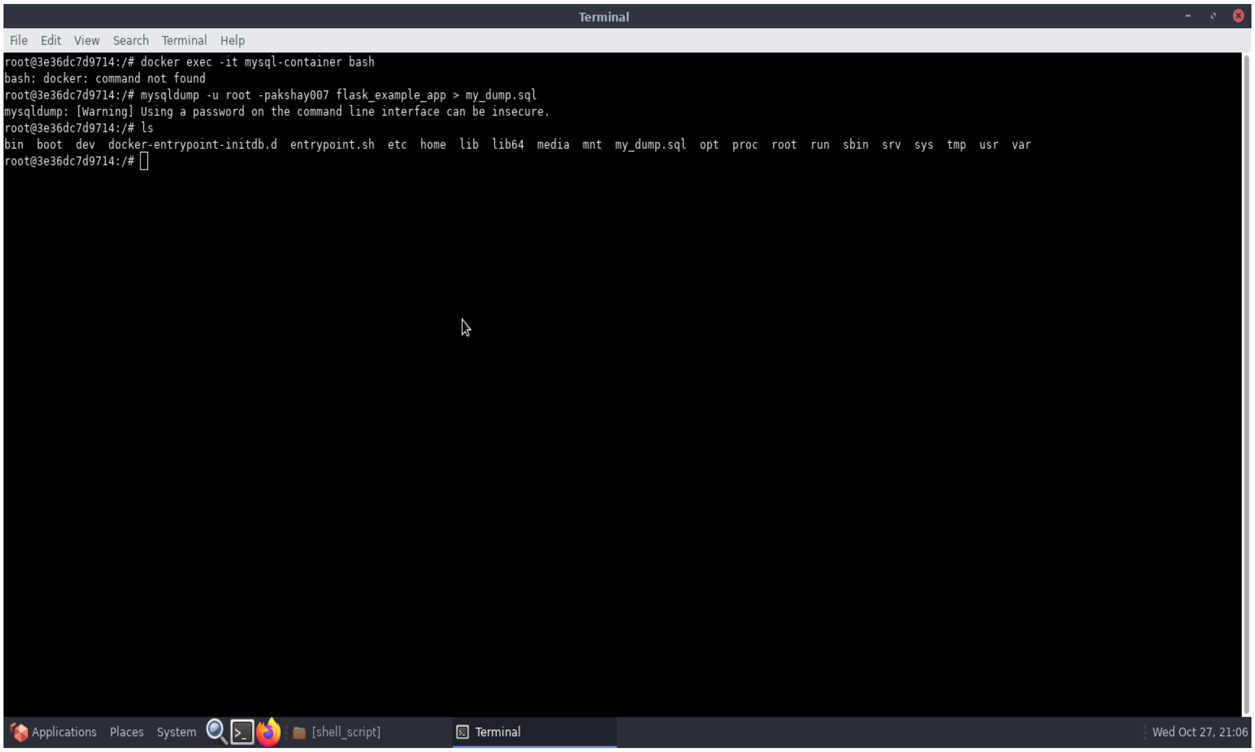
We can see here that the table is created successfully.

3. Create a **dump** of the MySql Database **flask_example_app** (The one you created in the first step)

   Answer:

   For this used the mysqldump command to process the above result. And made a .sql file as well.

   Here attaching the screenshot for the above.



Terminal
File Edit View Search Terminal Help
Applications Places System
[shell_script]
Terminal
Wed Oct 27, 21:06

```
root@3e36dc7d9714:/# docker exec -it mysql-container bash
bash: docker: command not found
root@3e36dc7d9714:/# mysqldump -u root -pakshay007 flask_example_app > my_dump.sql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
root@3e36dc7d9714:/# ls
bin  boot  dev  docker-entrypoint-initdb.d  entrypoint.sh  etc  home  lib  lib64  media  mnt  my_dump.sql  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@3e36dc7d9714:/# 
```

4. Run two docker containers using *mysql:latest* image and configure them as Master-Slave setup.

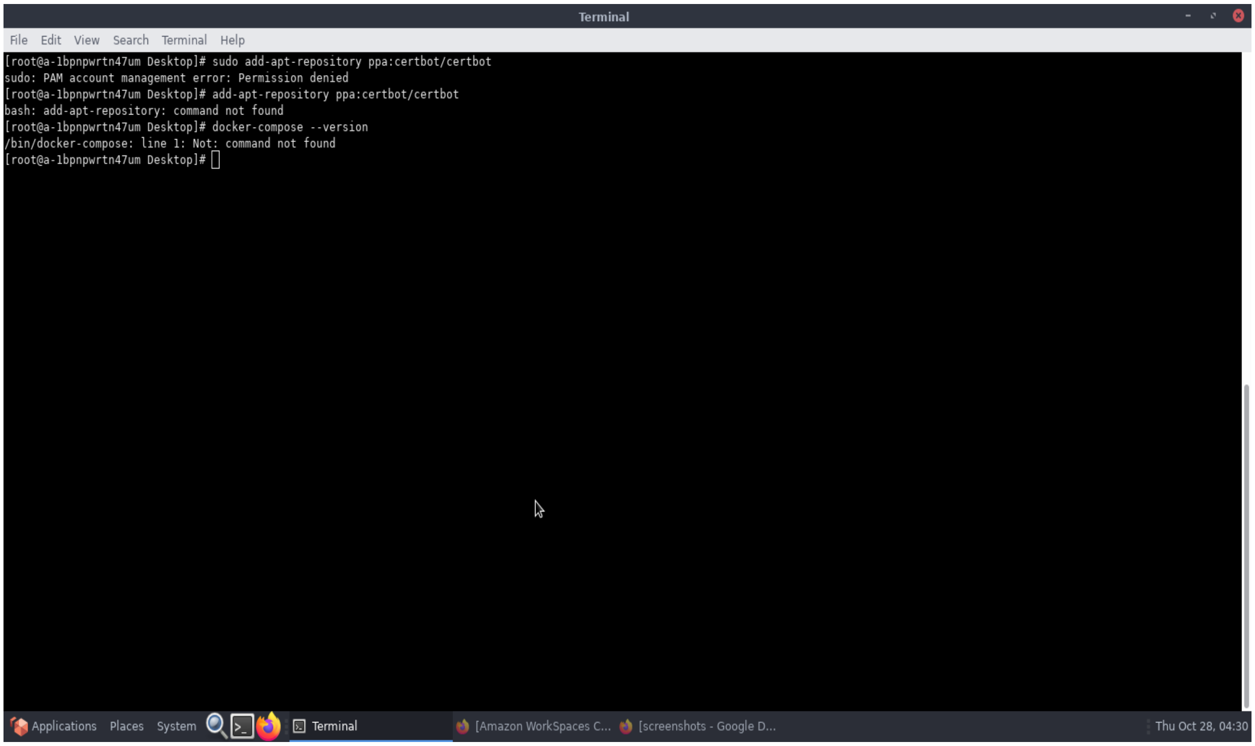
Answer:

For this configure two setup for master-slave system.

For this we first create docker-compose.yaml file with two mysql instances.

Now we need to configure master and slave file.

Due to unavailability of docker-compose and as it is not running so unable to complete it.


Terminal
File Edit View Search Terminal Help
Applications Places System
Terminal
[Amazon WorkSpaces C...
[screenshots - Google D...
Thu Oct 28, 04:30

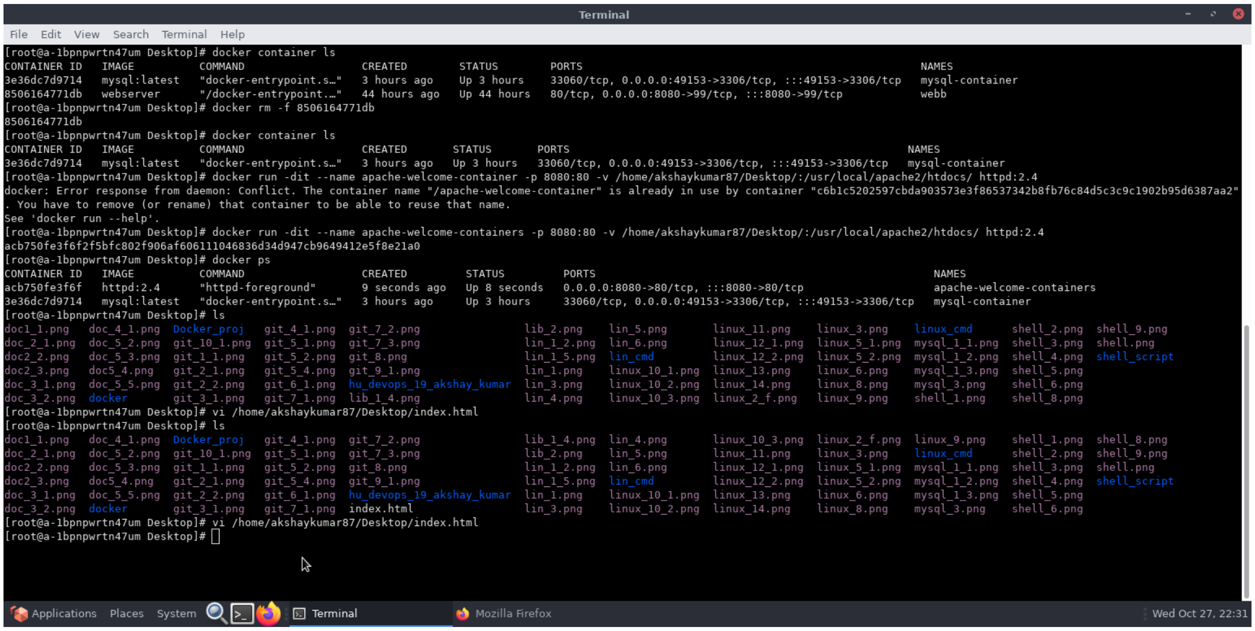
```
[root@a-1bpnpwrtn47um Desktop]# sudo add-apt-repository ppa:certbot/certbot
sudo: PAM account management error: Permission denied
[root@a-1bpnpwrtn47um Desktop]# add-apt-repository ppa:certbot/certbot
bash: add-apt-repository: command not found
[root@a-1bpnpwrtn47um Desktop]# docker-compose --version
/bin/docker-compose: line 1: Not: command not found
[root@a-1bpnpwrtn47um Desktop]# 
```

WEBSERVER

1. Run the *Apache server* on the docker container with the name **apache-welcome-container** and change the home page to "**Welcome to HashedIn**".

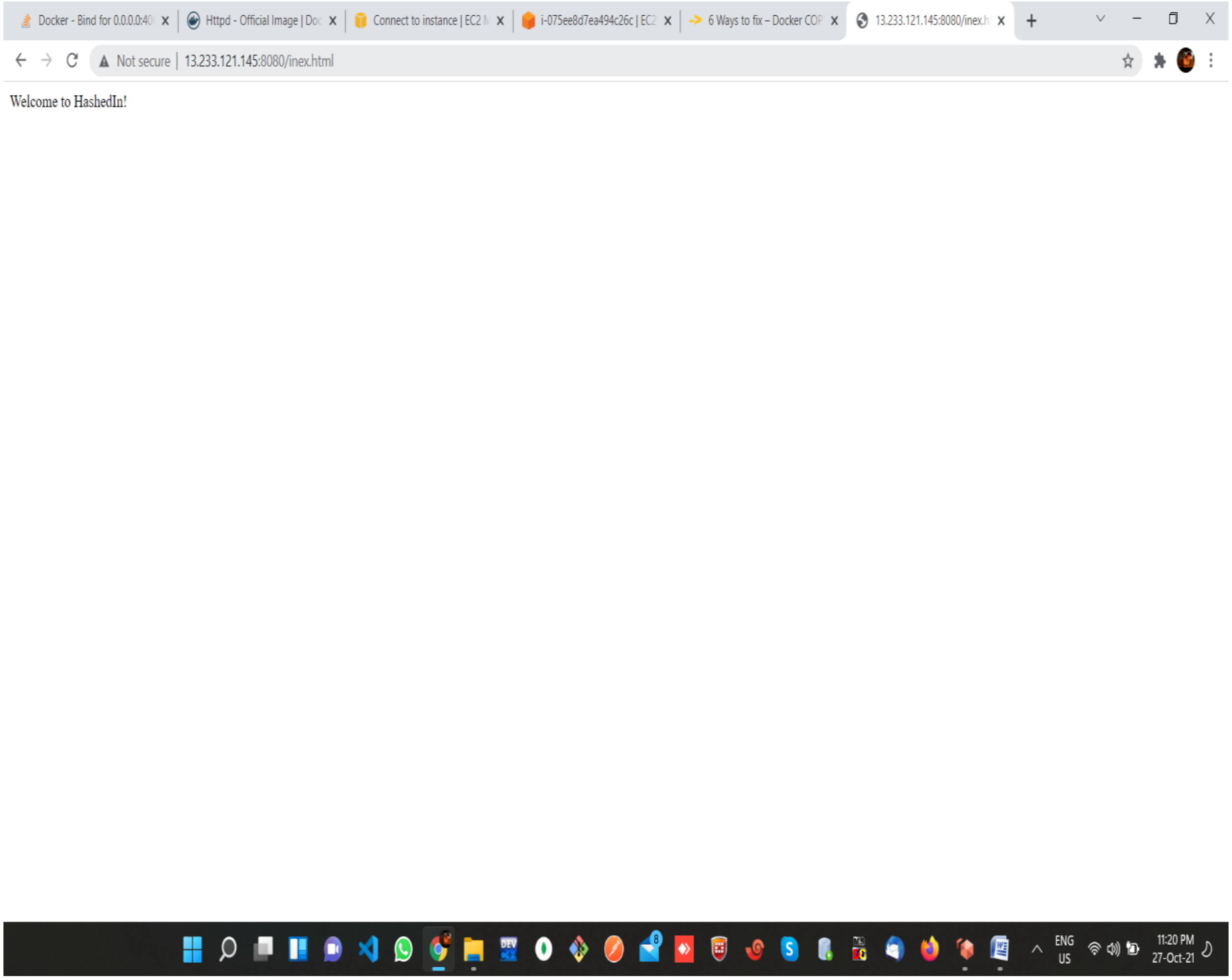
Answer:

For this made a docker container with the name specified and here is the screenshots of the above:


Terminal
File Edit View Search Terminal Help
Applications Places System
Terminal
Mozilla Firefox
Wed Oct 27, 22:31

```
[root@a-1bpnpwrtn47um Desktop]# docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                                   NAMES
3e36dc7d9714   mysql:latest   "docker-entrypoint.s…"   3 hours ago    Up 3 hours    33060/tcp, 0.0.0.0:49153->3306/tcp, :::49153->3306/tcp   mysql-container
8506164771db   webserver      "/docker-entrypoint.…"   44 hours ago   Up 44 hours   80/tcp, 0.0.0.0:8080->99/tcp, :::8080->99/tcp             webb
[root@a-1bpnpwrtn47um Desktop]# docker rm -f 8506164771db
8506164771db
[root@a-1bpnpwrtn47um Desktop]# docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED       STATUS       PORTS                                                   NAMES
3e36dc7d9714   mysql:latest   "docker-entrypoint.s…"   3 hours ago   Up 3 hours   33060/tcp, 0.0.0.0:49153->3306/tcp, :::49153->3306/tcp   mysql-container
[root@a-1bpnpwrtn47um Desktop]# docker run -dit --name apache-welcome-container -p 8080:80 -v /home/akshaykumar87/Desktop/:/usr/local/apache2/htdocs/ httpd:2.4
docker: Error response from daemon: Conflict. The container name "/apache-welcome-container" is already in use by container "c6b1c5202597cbda903573e3f86537342b8fb76c84d5c3c9c1902b95d6387aa2"
. You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[root@a-1bpnpwrtn47um Desktop]# docker run -dit --name apache-welcome-containers -p 8080:80 -v /home/akshaykumar87/Desktop/:/usr/local/apache2/htdocs/ httpd:2.4
acb750fe3f6f2f5bfc802f906af606111046836d34d947cb9649412e5f8e21a0
[root@a-1bpnpwrtn47um Desktop]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS                                                   NAMES
acb750fe3f6f   httpd:2.4      "httpd-foreground"       9 seconds ago   Up 8 seconds   0.0.0.0:8080->80/tcp, :::8080->80/tcp                   apache-welcome-containers
3e36dc7d9714   mysql:latest   "docker-entrypoint.s…"   3 hours ago     Up 3 hours     33060/tcp, 0.0.0.0:49153->3306/tcp, :::49153->3306/tcp   mysql-container
[root@a-1bpnpwrtn47um Desktop]# ls
doc1_1.png   doc_4_1.png  Docker_proj   git_4_1.png  git_7_2.png                 lib_2.png    lin_5.png       linux_11.png    linux_3.png    linux_cmd      shell_2.png  shell_9.png
doc_2_1.png  doc_5_2.png  git_10_1.png  git_5_1.png  git_7_3.png                 lin_1_2.png  lin_6.png       linux_12_1.png  linux_5_1.png  mysql_1_1.png  shell_3.png  shell.png
doc2_2.png   doc_5_3.png  git_1_1.png   git_5_2.png  git_8.png                   lin_1_5.png  lin_cmd         linux_12_2.png  linux_5_2.png  mysql_1_2.png  shell_4.png  shell_script
doc2_3.png   doc5_4.png   git_2_1.png   git_5_4.png  git_9_1.png                 lin_1.png    linux_10_1.png  linux_13.png    linux_6.png    mysql_1_3.png  shell_5.png
doc_3_1.png  doc_5_5.png  git_2_2.png   git_6_1.png  hu_devops_19_akshay_kumar  lin_3.png    linux_10_2.png  linux_14.png    linux_8.png    mysql_3.png    shell_6.png
doc_3_2.png  docker       git_3_1.png   git_7_1.png  lib_1_4.png                 lin_4.png    linux_10_3.png  linux_2_f.png   linux_9.png    shell_1.png    shell_8.png
[root@a-1bpnpwrtn47um Desktop]# vi /home/akshaykumar87/Desktop/index.html
[root@a-1bpnpwrtn47um Desktop]# ls
doc1_1.png   doc_4_1.png  Docker_proj   git_4_1.png  git_7_2.png                 lib_1_4.png  lin_4.png       linux_10_3.png  linux_2_f.png  linux_9.png    shell_1.png  shell_8.png
doc_2_1.png  doc_5_2.png  git_10_1.png  git_5_1.png  git_7_3.png                 lib_2.png    lin_5.png       linux_11.png    linux_3.png    linux_cmd      shell_2.png  shell_9.png
doc2_2.png   doc_5_3.png  git_1_1.png   git_5_2.png  git_8.png                   lin_1_2.png  lin_6.png       linux_12_1.png  linux_5_1.png  mysql_1_1.png  shell_3.png  shell.png
doc2_3.png   doc5_4.png   git_2_1.png   git_5_4.png  git_9_1.png                 lin_1_5.png  lin_cmd         linux_12_2.png  linux_5_2.png  mysql_1_2.png  shell_4.png  shell_script
doc_3_1.png  doc_5_5.png  git_2_2.png   git_6_1.png  hu_devops_19_akshay_kumar  lin_1.png    linux_10_1.png  linux_13.png    linux_6.png    mysql_1_3.png  shell_5.png
doc_3_2.png  docker       git_3_1.png   git_7_1.png  index.html                  lin_3.png    linux_10_2.png  linux_14.png    linux_8.png    mysql_3.png    shell_6.png
[root@a-1bpnpwrtn47um Desktop]# vi /home/akshaykumar87/Desktop/index.html
[root@a-1bpnpwrtn47um Desktop]# 
```

Here is the output after running in the browser:

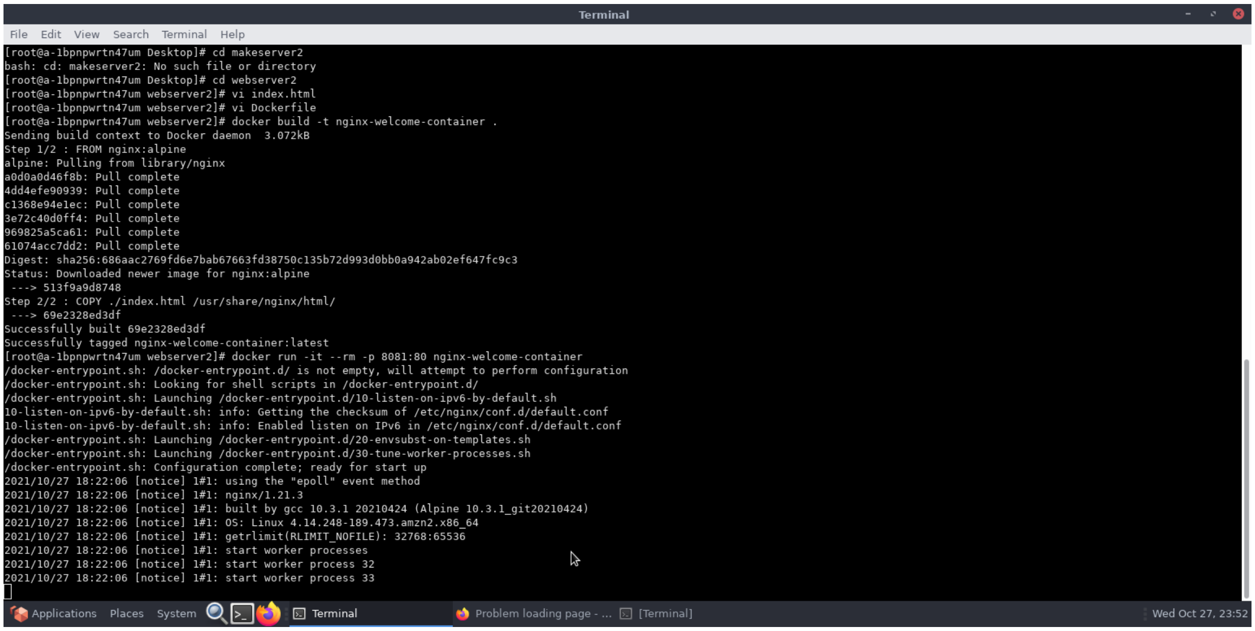

Docker - Bind for 0.0.0.0:40
Httpd - Official Image | Doc
Connect to instance | EC2 M
i-075ee8d7ea494c26c | EC2
6 Ways to fix – Docker COP
13.233.121.145:8080/inex.h
Not secure
13.233.121.145:8080/inex.html
ENG
US
11:20 PM
27-Oct-21

Welcome to HashedIn!

2. Run the *Nginx server* on the docker container with the name **nginx-welcome-container** and change the home page to “**Welcome to HashedIn**

Answer:

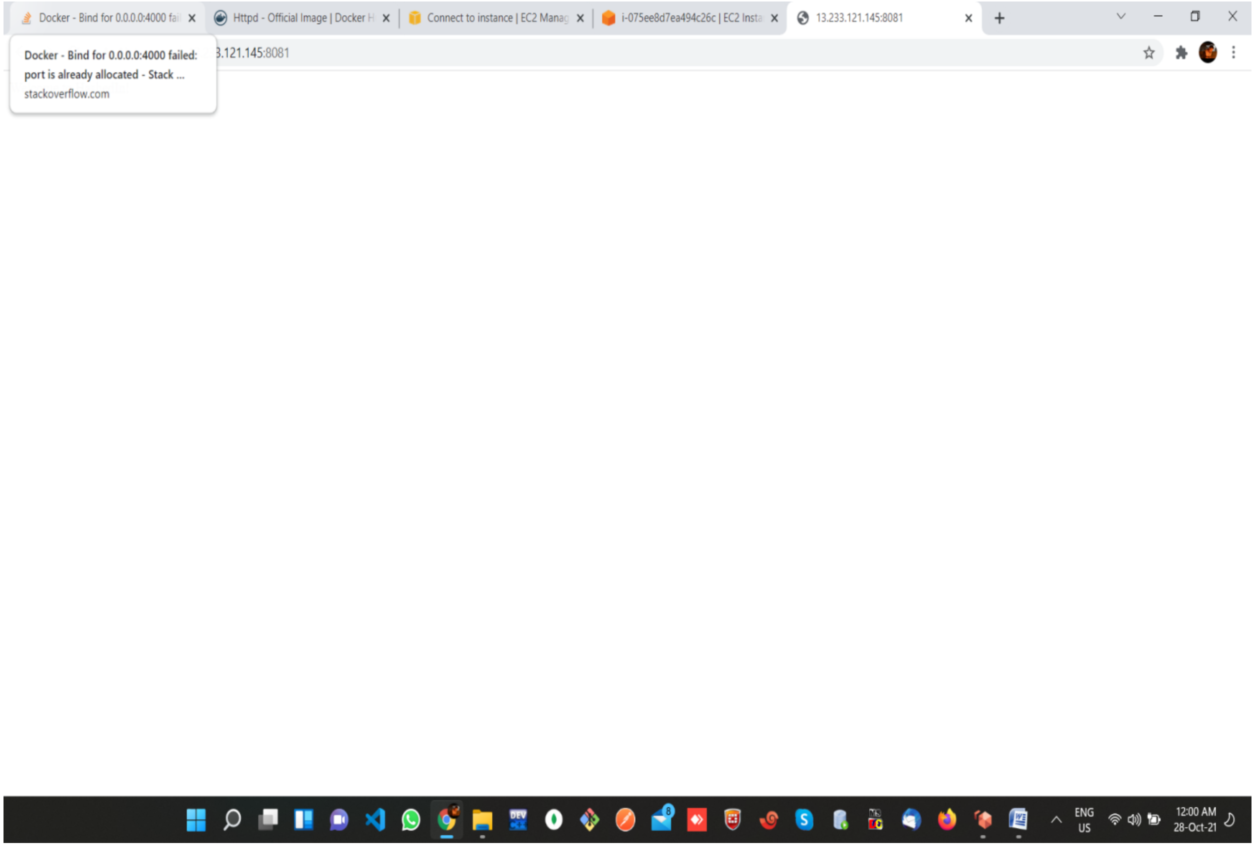
For this made a docker container with the name specified and here is the screenshots of the above:



Terminal
File Edit View Search Terminal Help
Applications Places System
Terminal
Problem loading page - ...
[Terminal]
Wed Oct 27, 23:52

```
[root@a-1bpnpwrtn47um Desktop]# cd makeserver2
bash: cd: makeserver2: No such file or directory
[root@a-1bpnpwrtn47um Desktop]# cd webserver2
[root@a-1bpnpwrtn47um webserver2]# vi index.html
[root@a-1bpnpwrtn47um webserver2]# vi Dockerfile
[root@a-1bpnpwrtn47um webserver2]# docker build -t nginx-welcome-container .
Sending build context to Docker daemon  3.072kB
Step 1/2 : FROM nginx:alpine
alpine: Pulling from library/nginx
a0d0a0d46f8b: Pull complete
4dd4efe90939: Pull complete
c1368e94e1ec: Pull complete
3e72c40d0ff4: Pull complete
969825a5ca61: Pull complete
61074acc7dd2: Pull complete
Digest: sha256:686aac2769fd6e7bab67663fd38750c135b72d993d0bb0a942ab02ef647fc9c3
Status: Downloaded newer image for nginx:alpine
 ---> 513f9a9d8748
Step 2/2 : COPY ./index.html /usr/share/nginx/html/
 ---> 69e2328ed3df
Successfully built 69e2328ed3df
Successfully tagged nginx-welcome-container:latest
[root@a-1bpnpwrtn47um webserver2]# docker run -it --rm -p 8081:80 nginx-welcome-container
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2021/10/27 18:22:06 [notice] 1#1: using the "epoll" event method
2021/10/27 18:22:06 [notice] 1#1: nginx/1.21.3
2021/10/27 18:22:06 [notice] 1#1: built by gcc 10.3.1 20210424 (Alpine 10.3.1_git20210424)
2021/10/27 18:22:06 [notice] 1#1: OS: Linux 4.14.248-189.473.amzn2.x86_64
2021/10/27 18:22:06 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 32768:65536
2021/10/27 18:22:06 [notice] 1#1: start worker processes
2021/10/27 18:22:06 [notice] 1#1: start worker process 32
2021/10/27 18:22:06 [notice] 1#1: start worker process 33
```

Also here we can see the output:


Docker - Bind for 0.0.0.0:4000 fai
Httpd - Official Image | Docker H
Connect to instance | EC2 Manag
i-075ee8d7ea494c26c | EC2 Insta
13.233.121.145:8081
Docker - Bind for 0.0.0.0:4000 failed:
port is already allocated - Stack ...
stackoverflow.com
3.121.145:8081
ENG
US
12:00 AM
28-Oct-21

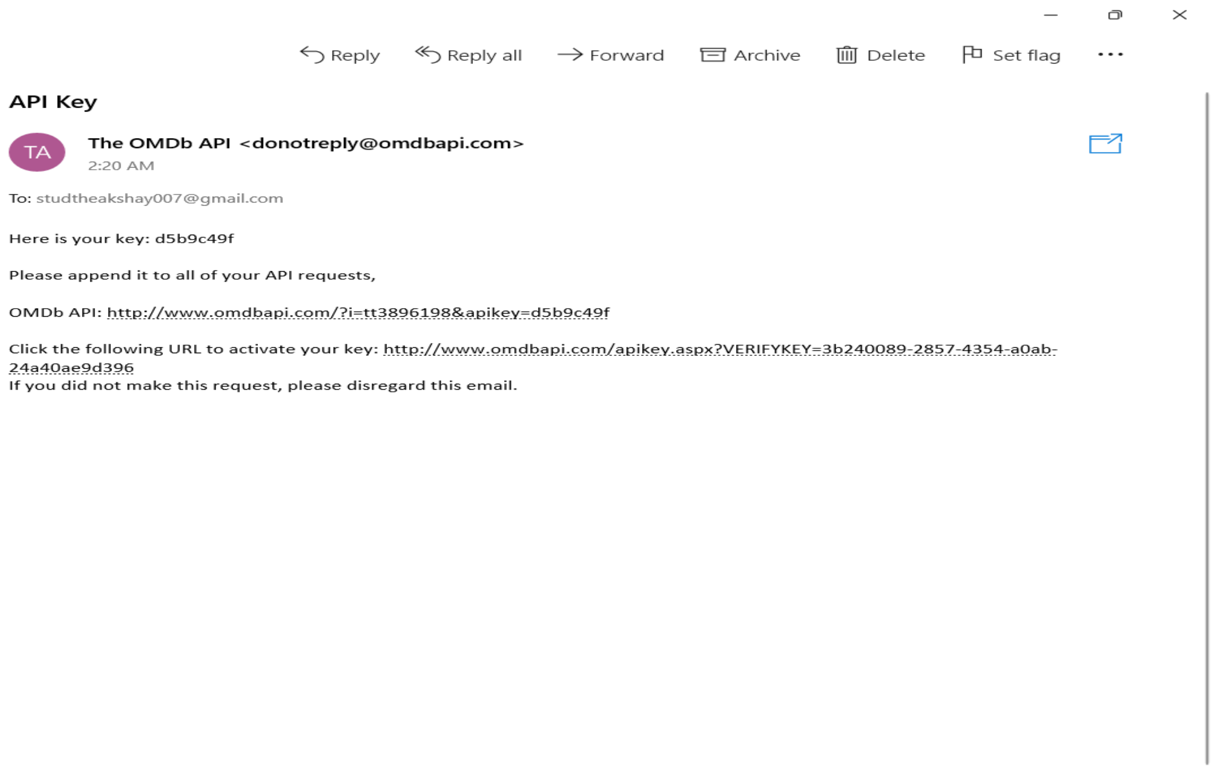
API

1. Write a python program and run it inside the docker container with the name **python-api-container** that fetches details of a movie provided as an input by the user. Use OmdbApi to fetch the movie. Display Year, Director, Actors, Plot, and ImdbRating as the output.
   API URL: http://www.omdbapi.com/

Answer;

For this need to make a api key. For this gone to the given link and made the free 1000 times access key account. Also made a docker file for the same.

Here is the output of the given explanations:



API Key

The OMDb API <donotreply@omdbapi.com>
2:20 AM

To: studtheakshay007@gmail.com

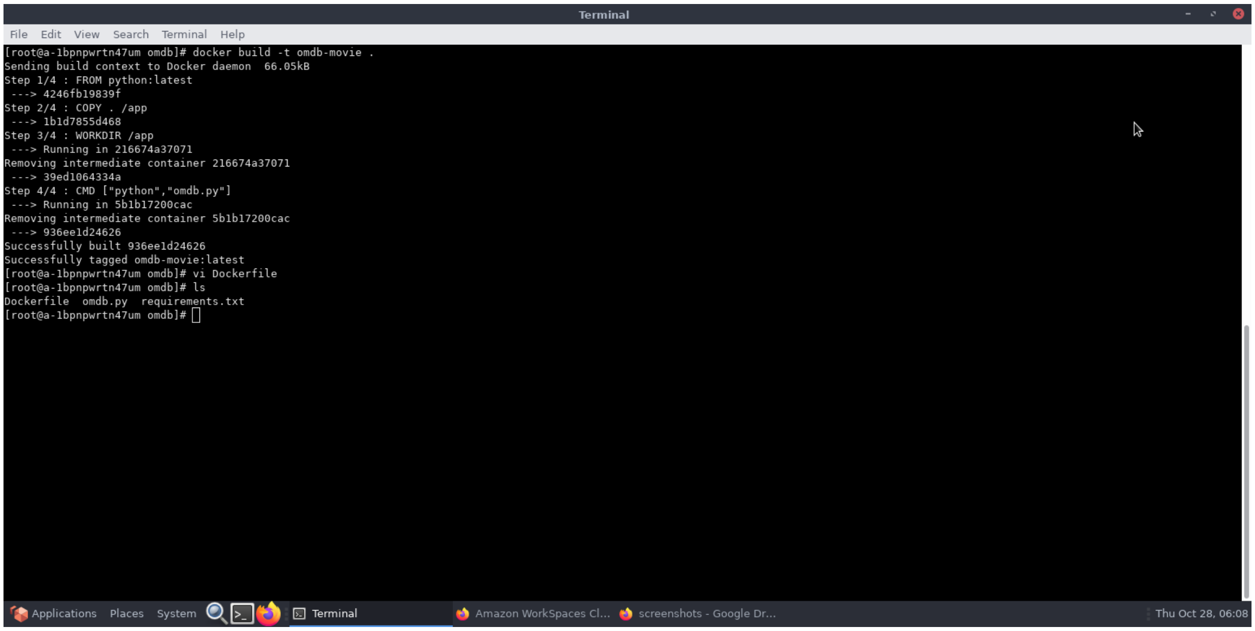
Here is your key: d5b9c49f

Please append it to all of your API requests,

OMDb API: http://www.omdbapi.com/?i=tt3896198&apikey=d5b9c49f

Click the following URL to activate your key: http://www.omdbapi.com/apikey.aspx?VERIFYKEY=3b240089-2857-4354-a0ab-24a40ae9d396
If you did not make this request, please disregard this email.

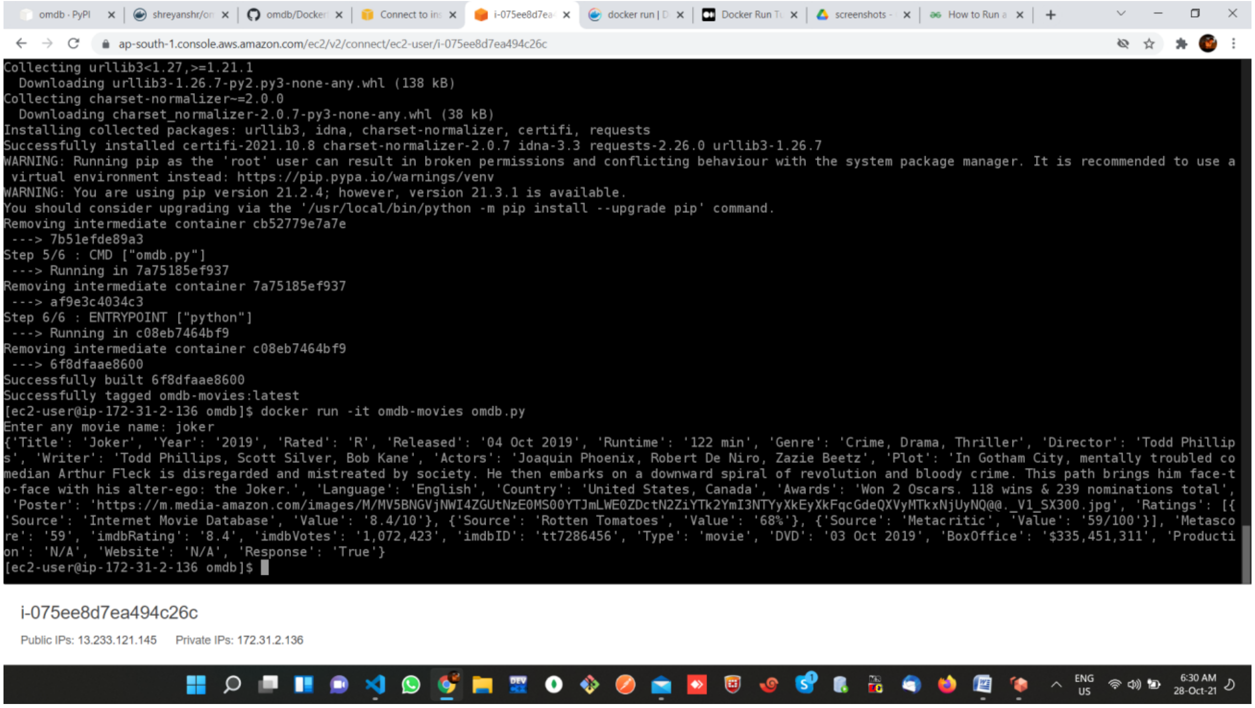
Showing through the screenshot:



```
[root@a-1bpnpwrtn47um omdb]# docker build -t omdb-movie .
Sending build context to Docker daemon  66.05kB
Step 1/4 : FROM python:latest
 ---> 4246fb19839f
Step 2/4 : COPY . /app
 ---> 1b1d7855d468
Step 3/4 : WORKDIR /app
 ---> Running in 216674a37071
Removing intermediate container 216674a37071
 ---> 39ed1064334a
Step 4/4 : CMD ["python","omdb.py"]
 ---> Running in 5b1b17200cac
Removing intermediate container 5b1b17200cac
 ---> 936ee1d24626
Successfully built 936ee1d24626
Successfully tagged omdb-movie:latest
[root@a-1bpnpwrtn47um omdb]# vi Dockerfile
[root@a-1bpnpwrtn47um omdb]# ls
Dockerfile  omdb.py  requirements.txt
[root@a-1bpnpwrtn47um omdb]# 
```

Now Showing the output:

Given the input “joker”:



```
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.7-py2.py3-none-any.whl (138 kB)
Collecting charset-normalizer~=2.0.0
  Downloading charset_normalizer-2.0.7-py3-none-any.whl (38 kB)
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2021.10.8 charset-normalizer-2.0.7 idna-3.3 requests-2.26.0 urllib3-1.26.7
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a
 virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.2.4; however, version 21.3.1 is available.
You should consider upgrading via the '/usr/local/bin/python -m pip install --upgrade pip' command.
Removing intermediate container cb52779e7a7e
 ---> 7b51efde89a3
Step 5/6 : CMD ["omdb.py"]
 ---> Running in 7a75185ef937
Removing intermediate container 7a75185ef937
 ---> af9e3c4034c3
Step 6/6 : ENTRYPOINT ["python"]
 ---> Running in c08eb7464bf9
Removing intermediate container c08eb7464bf9
 ---> 6f8dfaae8600
Successfully built 6f8dfaae8600
Successfully tagged omdb-movies:latest
[ec2-user@ip-172-31-2-136 omdb]$ docker run -it omdb-movies omdb.py
Enter any movie name: joker
{'Title': 'Joker', 'Year': '2019', 'Rated': 'R', 'Released': '04 Oct 2019', 'Runtime': '122 min', 'Genre': 'Crime, Drama, Thriller', 'Director': 'Todd Phillip
s', 'Writer': 'Todd Phillips, Scott Silver, Bob Kane', 'Actors': 'Joaquin Phoenix, Robert De Niro, Zazie Beetz', 'Plot': 'In Gotham City, mentally troubled co
median Arthur Fleck is disregarded and mistreated by society. He then embarks on a downward spiral of revolution and bloody crime. This path brings him face-t
o-face with his alter-ego: the Joker.', 'Language': 'English', 'Country': 'United States, Canada', 'Awards': 'Won 2 Oscars. 118 wins & 239 nominations total',
 'Poster': 'https://m.media-amazon.com/images/M/MV5BNGVjNWI4ZGUtNzE0MS00YTJmLWE0ZDctN2ZiYTk2YmI3NTYyXkEyXkFqcGdeQXVyMTkxNjUyNQ@@._V1_SX300.jpg', 'Ratings': [{
'Source': 'Internet Movie Database', 'Value': '8.4/10'}, {'Source': 'Rotten Tomatoes', 'Value': '68%'}, {'Source': 'Metacritic', 'Value': '59/100'}], 'Metasco
re': '59', 'imdbRating': '8.4', 'imdbVotes': '1,072,423', 'imdbID': 'tt7286456', 'Type': 'movie', 'DVD': '03 Oct 2019', 'BoxOffice': '$335,451,311', 'Producti
on': 'N/A', 'Website': 'N/A', 'Response': 'True'}
[ec2-user@ip-172-31-2-136 omdb]$ 
```

i-075ee8d7ea494c26c

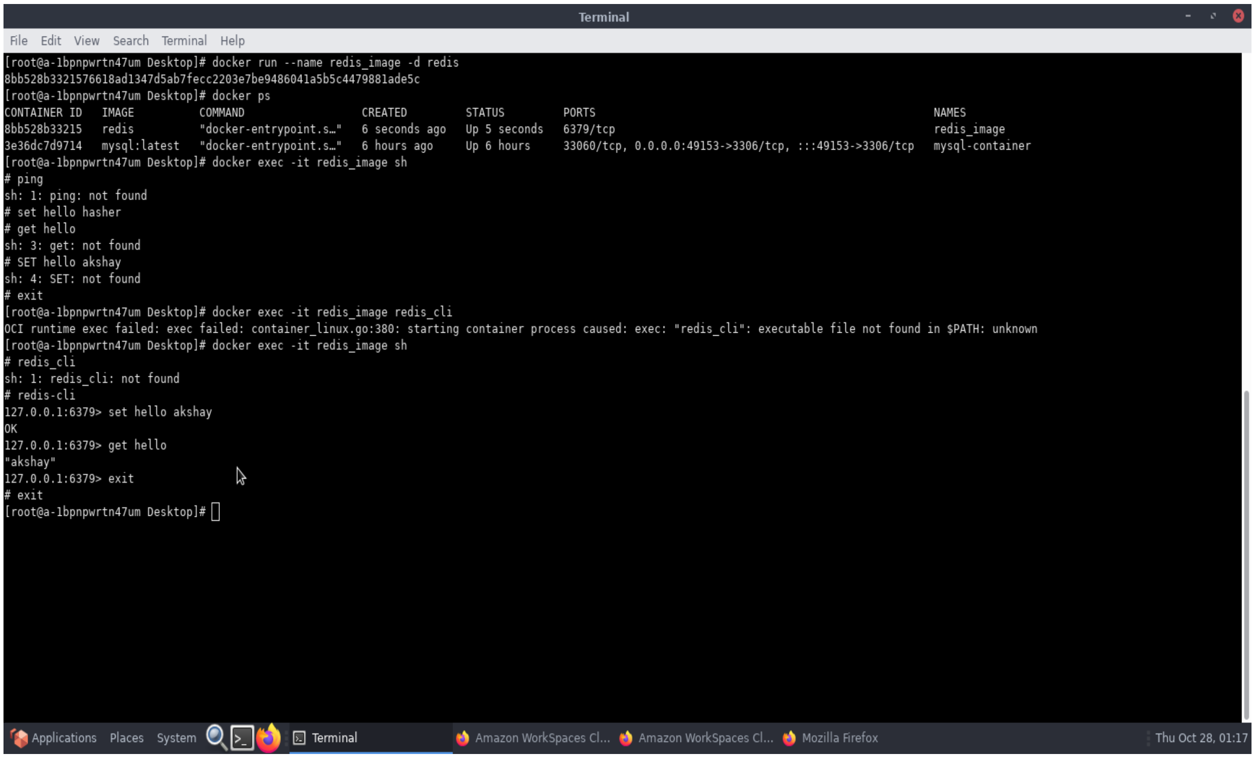
Public IPs: 13.233.121.145   Private IPs: 172.31.2.136

REDIS

1. Configure Redis by running it inside a docker container and test the same using **redis-cli**.

Answer:

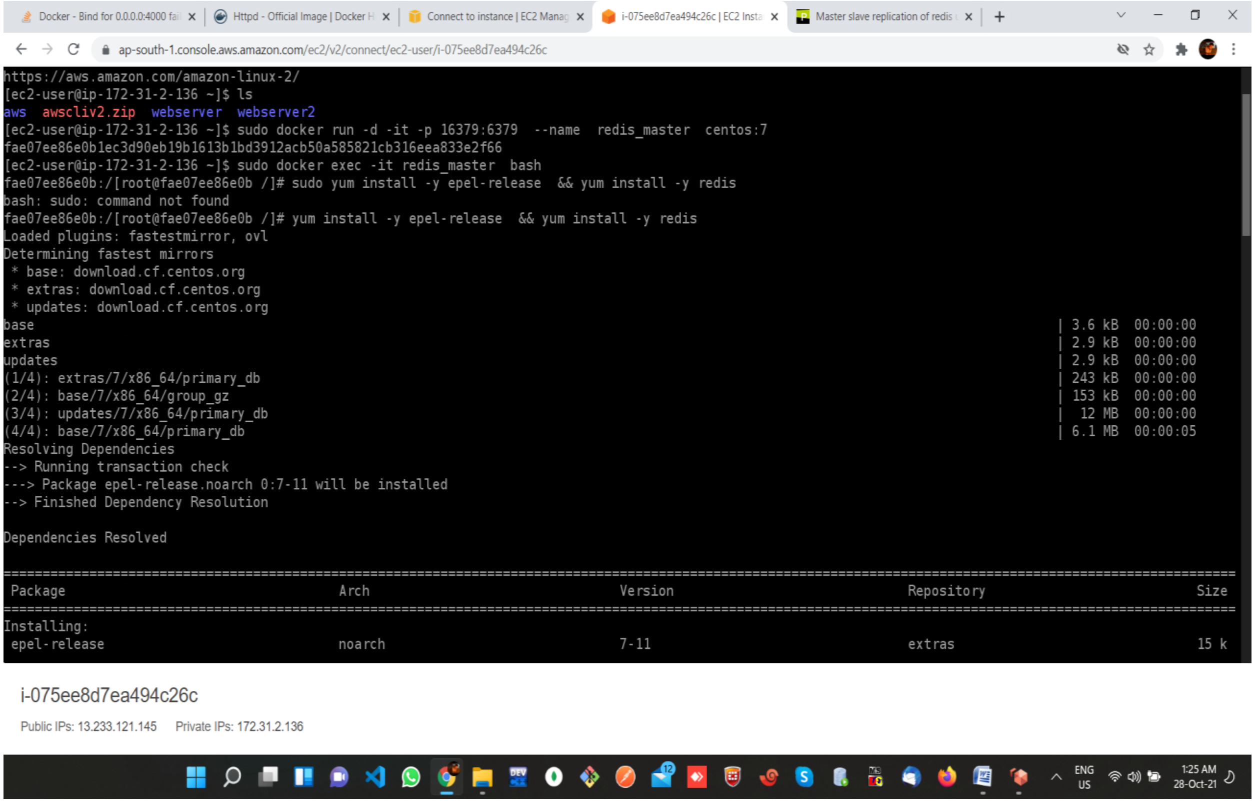
Here is the output of the above :



```
[root@a-1bpnpwrtn47um Desktop]# docker run --name redis_image -d redis
8bb528b3321576618ad1347d5ab7fecc2203e7be9486041a5b5c4479881ade5c
[root@a-1bpnpwrtn47um Desktop]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS                                                     NAMES
8bb528b33215   redis          "docker-entrypoint.s…"   6 seconds ago   Up 5 seconds   6379/tcp                                                  redis_image
3e36dc7d9714   mysql:latest   "docker-entrypoint.s…"   6 hours ago     Up 6 hours     33060/tcp, 0.0.0.0:49153->3306/tcp, :::49153->3306/tcp   mysql-container
[root@a-1bpnpwrtn47um Desktop]# docker exec -it redis_image sh
# ping
sh: 1: ping: not found
# set hello hasher
# get hello
sh: 3: get: not found
# SET hello akshay
sh: 4: SET: not found
# exit
[root@a-1bpnpwrtn47um Desktop]# docker exec -it redis_image redis_cli
OCI runtime exec failed: exec failed: container_linux.go:380: starting container process caused: exec: "redis_cli": executable file not found in $PATH: unknown
[root@a-1bpnpwrtn47um Desktop]# docker exec -it redis_image sh
# redis_cli
sh: 1: redis_cli: not found
# redis-cli
127.0.0.1:6379> set hello akshay
OK
127.0.0.1:6379> get hello
"akshay"
127.0.0.1:6379> exit
# exit
[root@a-1bpnpwrtn47um Desktop]# 
```

2. Configure master-slave architecture of Redis using docker containers.

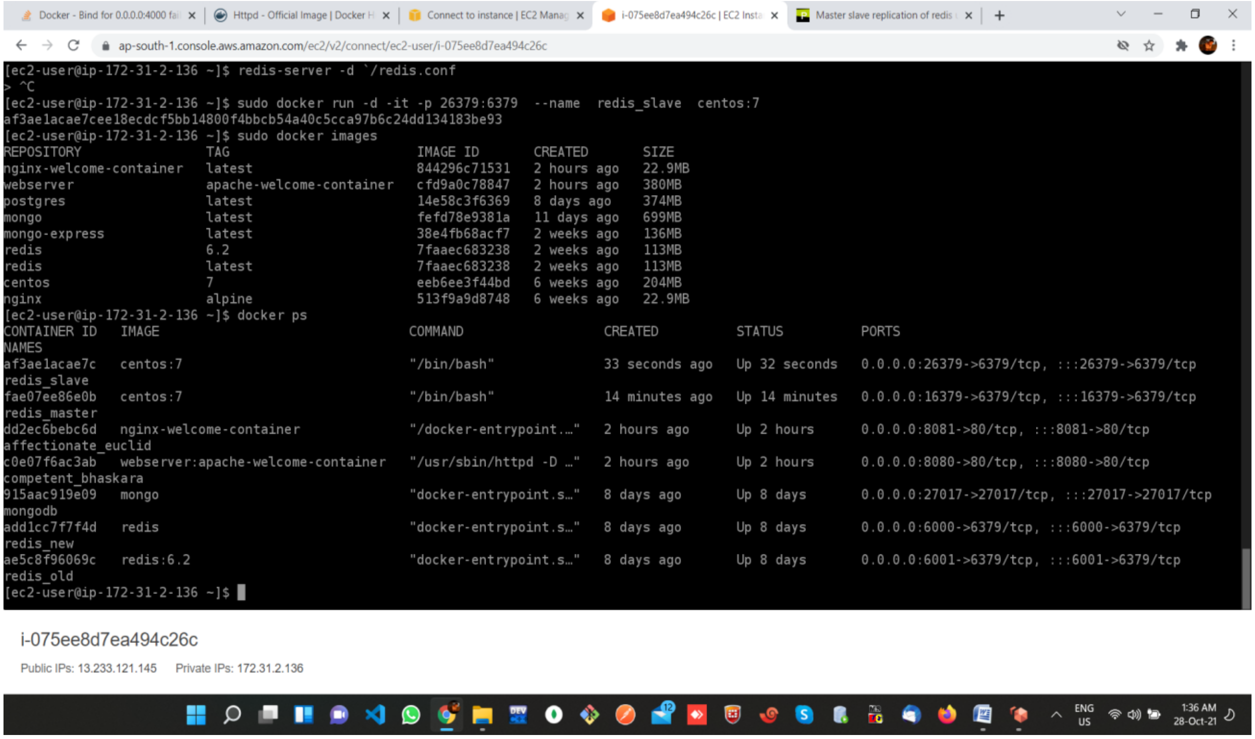
Answer: Idea behind this starts with making docker-compose.yml file. We need to configure both master and slave conf file. Here I'm sharing some screenshots for this.


Docker - Bind for 0.0.0.0:4000 fai
Httpd - Official Image | Docker H
Connect to instance | EC2 Manag
i-075ee8d7ea494c26c | EC2 Insta
Master slave replication of redis
ap-south-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-075ee8d7ea494c26c

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-2-136 ~]$ ls
aws  awscliv2.zip  webserver  webserver2
[ec2-user@ip-172-31-2-136 ~]$ sudo docker run -d -it -p 16379:6379  --name  redis_master  centos:7
fae07ee86e0b1ec3d90eb19b1613b1bd3912acb50a585821cb316eea833e2f66
[ec2-user@ip-172-31-2-136 ~]$ sudo docker exec -it redis_master  bash
fae07ee86e0b:/[root@fae07ee86e0b /]# sudo yum install -y epel-release  && yum install -y redis
bash: sudo: command not found
fae07ee86e0b:/[root@fae07ee86e0b /]# yum install -y epel-release  && yum install -y redis
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: download.cf.centos.org
 * extras: download.cf.centos.org
 * updates: download.cf.centos.org
base                                                        | 3.6 kB  00:00:00
extras                                                      | 2.9 kB  00:00:00
updates                                                     | 2.9 kB  00:00:00
(1/4): extras/7/x86_64/primary_db                           | 243 kB  00:00:00
(2/4): base/7/x86_64/group_gz                               | 153 kB  00:00:00
(3/4): updates/7/x86_64/primary_db                          |  12 MB  00:00:00
(4/4): base/7/x86_64/primary_db                             | 6.1 MB  00:00:05
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

==================================================================================================
 Package                      Arch                Version             Repository              Size
==================================================================================================
Installing:
 epel-release                 noarch              7-11                extras                  15 k
```

i-075ee8d7ea494c26c

Public IPs: 13.233.121.145   Private IPs: 172.31.2.136

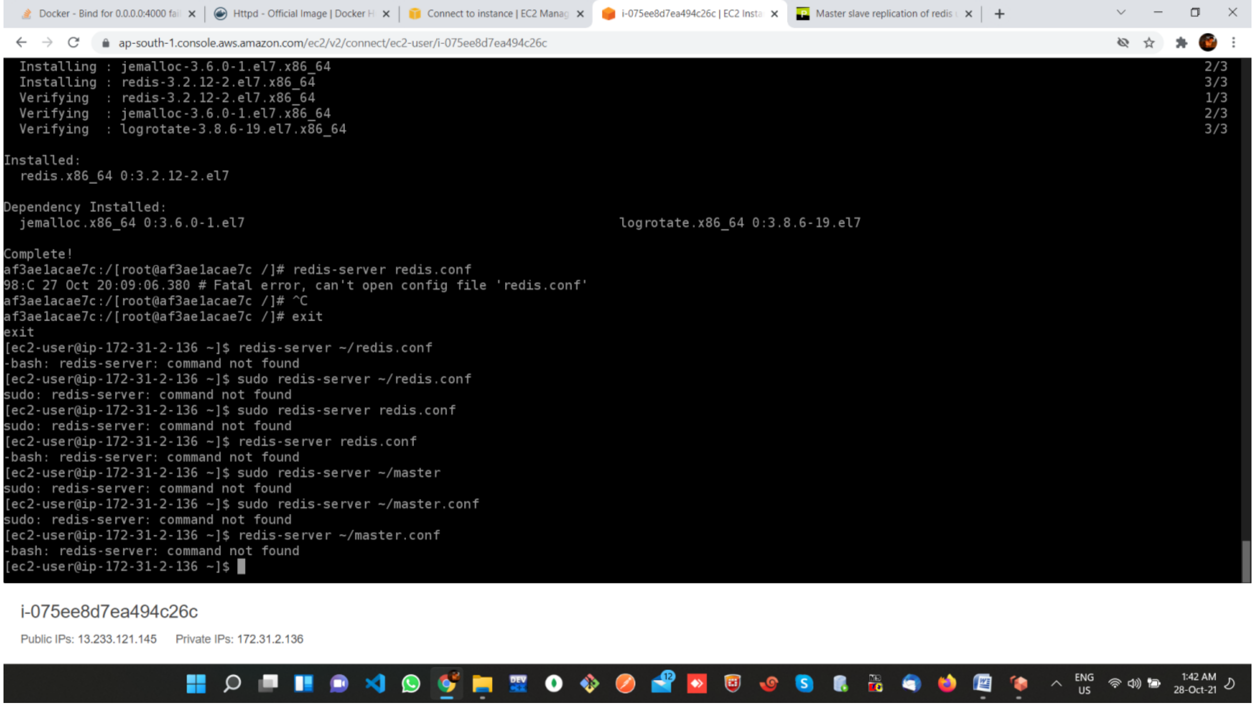


Docker - Bind for 0.0.0.0:4000 fai
Httpd - Official Image | Docker H
Connect to instance | EC2 Manag
i-075ee8d7ea494c26c | EC2 Insta
Master slave replication of redis
ap-south-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-075ee8d7ea494c26c
ENG
US
1:36 AM
28-Oct-21

```
[ec2-user@ip-172-31-2-136 ~]$ redis-server -d `/redis.conf
> ^C
[ec2-user@ip-172-31-2-136 ~]$ sudo docker run -d -it -p 26379:6379  --name  redis_slave  centos:7
af3ae1acae7cee18ecdcf5bb14800f4bbcb54a40c5cca97b6c24dd134183be93
[ec2-user@ip-172-31-2-136 ~]$ sudo docker images
REPOSITORY               TAG                        IMAGE ID       CREATED        SIZE
nginx-welcome-container  latest                     844296c71531   2 hours ago    22.9MB
webserver                apache-welcome-container   cfd9a0c78847   2 hours ago    380MB
postgres                 latest                     14e58c3f6369   8 days ago     374MB
mongo                    latest                     fefd78e9381a   11 days ago    699MB
mongo-express            latest                     38e4fb68acf7   2 weeks ago    136MB
redis                    6.2                        7faaec683238   2 weeks ago    113MB
redis                    latest                     7faaec683238   2 weeks ago    113MB
centos                   7                          eeb6ee3f44bd   6 weeks ago    204MB
nginx                    alpine                     513f9a9d8748   6 weeks ago    22.9MB
[ec2-user@ip-172-31-2-136 ~]$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS
NAMES
af3ae1acae7c   centos:7                             "/bin/bash"              33 seconds ago   Up 32 seconds   0.0.0.0:26379->6379/tcp, :::26379->6379/tcp
redis_slave
fae07ee86e0b   centos:7                             "/bin/bash"              14 minutes ago   Up 14 minutes   0.0.0.0:16379->6379/tcp, :::16379->6379/tcp
redis_master
dd2ec6bebc6d   nginx-welcome-container              "/docker-entrypoint.…"   2 hours ago      Up 2 hours      0.0.0.0:8081->80/tcp, :::8081->80/tcp
affectionate_euclid
c0e07f6ac3ab   webserver:apache-welcome-container   "/usr/sbin/httpd -D …"   2 hours ago      Up 2 hours      0.0.0.0:8080->80/tcp, :::8080->80/tcp
competent_bhaskara
915aac919e09   mongo                                "docker-entrypoint.s…"   8 days ago       Up 8 days       0.0.0.0:27017->27017/tcp, :::27017->27017/tcp
mongodb
add1cc7f7f4d   redis                                "docker-entrypoint.s…"   8 days ago       Up 8 days       0.0.0.0:6000->6379/tcp, :::6000->6379/tcp
redis_new
ae5c8f96069c   redis:6.2                            "docker-entrypoint.s…"   8 days ago       Up 8 days       0.0.0.0:6001->6379/tcp, :::6001->6379/tcp
redis_old
[ec2-user@ip-172-31-2-136 ~]$ █
```

i-075ee8d7ea494c26c

Public IPs: 13.233.121.145   Private IPs: 172.31.2.136



Docker - Bind for 0.0.0.0:4000 fai
Httpd - Official Image | Docker H
Connect to instance | EC2 Manag
i-075ee8d7ea494c26c | EC2 Insta
Master slave replication of redis
ap-south-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-075ee8d7ea494c26c
ENG
US
1:42 AM
28-Oct-21

```
  Installing : jemalloc-3.6.0-1.el7.x86_64                                                                        2/3
  Installing : redis-3.2.12-2.el7.x86_64                                                                          3/3
  Verifying  : redis-3.2.12-2.el7.x86_64                                                                          1/3
  Verifying  : jemalloc-3.6.0-1.el7.x86_64                                                                        2/3
  Verifying  : logrotate-3.8.6-19.el7.x86_64                                                                      3/3

Installed:
  redis.x86_64 0:3.2.12-2.el7

Dependency Installed:
  jemalloc.x86_64 0:3.6.0-1.el7                                logrotate.x86_64 0:3.8.6-19.el7

Complete!
af3ae1acae7c:/[root@af3ae1acae7c /]# redis-server redis.conf
98:C 27 Oct 20:09:06.380 # Fatal error, can't open config file 'redis.conf'
af3ae1acae7c:/[root@af3ae1acae7c /]# ^C
af3ae1acae7c:/[root@af3ae1acae7c /]# exit
exit
[ec2-user@ip-172-31-2-136 ~]$ redis-server ~/redis.conf
-bash: redis-server: command not found
[ec2-user@ip-172-31-2-136 ~]$ sudo redis-server ~/redis.conf
sudo: redis-server: command not found
[ec2-user@ip-172-31-2-136 ~]$ sudo redis-server redis.conf
sudo: redis-server: command not found
[ec2-user@ip-172-31-2-136 ~]$ redis-server redis.conf
-bash: redis-server: command not found
[ec2-user@ip-172-31-2-136 ~]$ sudo redis-server ~/master
sudo: redis-server: command not found
[ec2-user@ip-172-31-2-136 ~]$ sudo redis-server ~/master.conf
sudo: redis-server: command not found
[ec2-user@ip-172-31-2-136 ~]$ redis-server ~/master.conf
-bash: redis-server: command not found
[ec2-user@ip-172-31-2-136 ~]$ █
```

i-075ee8d7ea494c26c

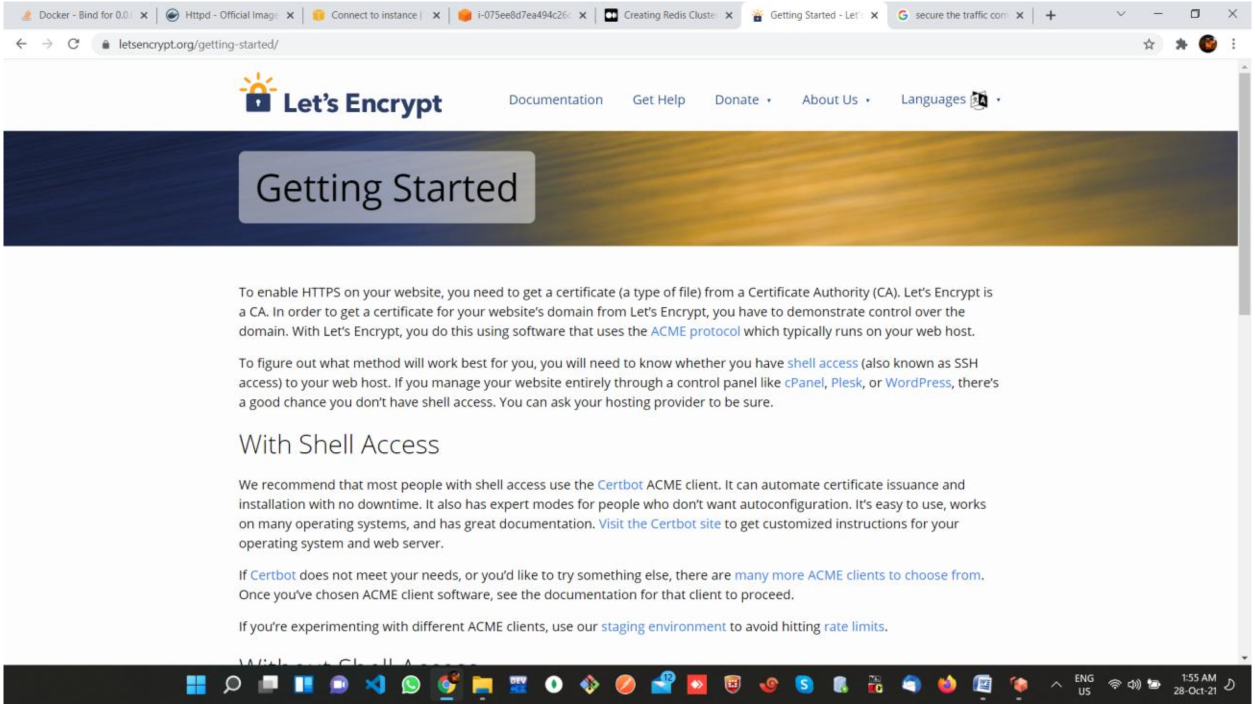
Public IPs: 13.233.121.145   Private IPs: 172.31.2.136

SECURITY

1. **1.** Generate an SSL certificate using **Let's Encrypt** and secure the traffic coming to the **Nginx web server** running as a container.

Answer:

For this I first need to go the given site, that is:



Getting Started

To enable HTTPS on your website, you need to get a certificate (a type of file) from a Certificate Authority (CA). Let's Encrypt is a CA. In order to get a certificate for your website's domain from Let's Encrypt, you have to demonstrate control over the domain. With Let's Encrypt, you do this using software that uses the ACME protocol which typically runs on your web host.

To figure out what method will work best for you, you will need to know whether you have shell access (also known as SSH access) to your web host. If you manage your website entirely through a control panel like cPanel, Plesk, or WordPress, there's a good chance you don't have shell access. You can ask your hosting provider to be sure.

With Shell Access

We recommend that most people with shell access use the Certbot ACME client. It can automate certificate issuance and installation with no downtime. It also has expert modes for people who don't want autoconfiguration. It's easy to use, works on many operating systems, and has great documentation. Visit the Certbot site to get customized instructions for your operating system and web server.

If Certbot does not meet your needs, or you'd like to try something else, there are many more ACME clients to choose from. Once you've chosen ACME client software, see the documentation for that client to proceed.

If you're experimenting with different ACME clients, use our staging environment to avoid hitting rate limits.

Without Shell Access

THANK YOU!