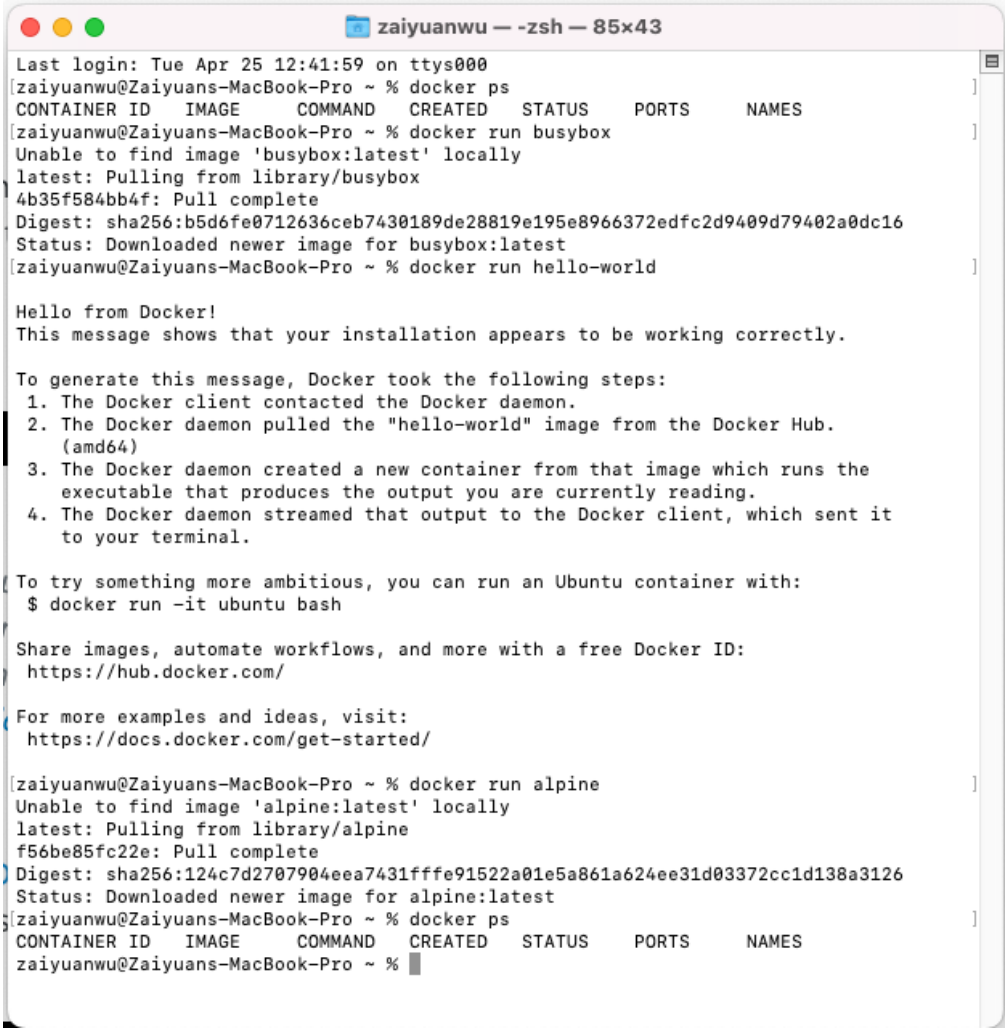
Task 01: Basic Docker Virtualization (6 Marks)

- Create a dockerized environment in your computer/ PC (Linux environment preferred. Can be hosted inside a VM) and run at least 3 well-known docker images from the docker hub (i.e., hello-world, BusyBox, Nginx, Redis, Alpine,...etc.).
- Show the running docker containers in your machine. Remove all the running containers and their images and show the list of images in the virtual domain.

Run three images: hello-world, BusyBox and alpine. Because these images are simple examples so it exits automatically after running. I use 'docker ps -a' to show all exited containers.




```
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker ps -a
CONTAINER ID   IMAGE         COMMAND     CREATED              STATUS                          PORTS     NAMES
4932d0a2707a   alpine        "sh"        About a minute ago   Exited (130) About a minute ago           practical_villani
5f06121d1392   alpine        "/bin/sh"   About a minute ago   Exited (130) About a minute ago           clever_volhard
9dc213db84e9   alpine        "/bin/sh"   2 minutes ago        Exited (130) 2 minutes ago                zen_chaum
1993e1c7519e   busybox       "sh"        2 minutes ago        Exited (0) 2 minutes ago                  sweet_beaver
50ef585c6cbb   hello-world   "/hello"    2 minutes ago        Exited (0) 2 minutes ago                  bold_mclean
91ee1650a50d   alpine        "/bin/sh"   8 minutes ago        Exited (0) 8 minutes ago                  friendly_euclid
060b3f048541   hello-world   "/hello"    8 minutes ago        Exited (0) 8 minutes ago                  jovial_nash
90f5296738f7   busybox       "sh"        8 minutes ago        Exited (0) 8 minutes ago                  goofy_goldstine
f27a3e639a06   hello-world   "/hello"    15 minutes ago       Exited (0) 15 minutes ago                 thirsty_roentgen
zaiyuanwu@Zaiyuans-MacBook-Pro ~ %
```

Delete containers using command 'docker rm':

```
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker rm 4932d0a2707a 5f06121d1392 9dc213db84e9 1993e1c7519e 50ef585c6cbb 91ee1650a50d 060b3f048541 90f5296738f7 f27a3e639a06
4932d0a2707a
5f06121d1392
9dc213db84e9
1993e1c7519e
50ef585c6cbb
91ee1650a50d
060b3f048541
90f5296738f7
f27a3e639a06
```

Show docker images:

```
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
alpine        latest    9ed4aefc74f6   3 weeks ago     7.05MB
busybox       latest    7cfbbec8963d   5 weeks ago     4.86MB
hello-world   latest    feb5d9fea6a5   19 months ago   13.3kB
zaiyuanwu@Zaiyuans-MacBook-Pro ~ %
```

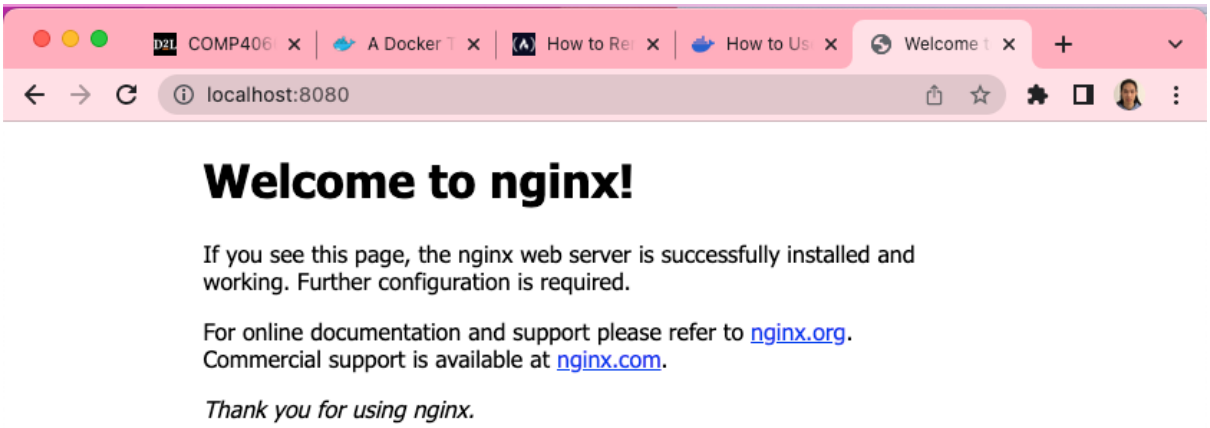
Delete images:

```
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
alpine        latest    9ed4aefc74f6   3 weeks ago     7.05MB
busybox       latest    7cfbbec8963d   5 weeks ago     4.86MB
hello-world   latest    feb5d9fea6a5   19 months ago   13.3kB
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker rmi alpine busybox hello-world
Untagged: alpine:latest
Untagged: alpine@sha256:124c7d2707904eea7431fffe91522a01e5a861a624ee31d03372cc1d138a3126
Deleted: sha256:9ed4aefc74f6792b5a804d1d146fe4b4a2299147b0f50eaf2b08435d7b38c27e
Deleted: sha256:f1417ff83b319fbdae6dd9cd6d8c9c88002dcd75ecf6ec201c8c6894681cf2b5
Untagged: busybox:latest
Untagged: busybox@sha256:b5d6fe0712636ceb7430189de28819e195e8966372edfc2d9409d79402a0dc16
Deleted: sha256:7cfbbec8963d8f13e6c70416d6592e1cc10f47a348131290a55d43c3acab3fb9
Deleted: sha256:baacf561cfff825708763ce7ee4a18293716c533e6ece3bd39009a5fb3c804d2
Untagged: hello-world:latest
Untagged: hello-world@sha256:4e83453afed1b4fa1a3500525091dbfca6ce1e66903fd4c01ff015dbcb1ba33e
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
Deleted: sha256:e07ee1baac5fae6a26f30cabfe54a36d3402f96afda318fe0a96cec4ca393359
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
zaiyuanwu@Zaiyuans-MacBook-Pro ~ %
```

- Follow the instructions at https://www.docker.com/blog/how-to-use-the-official-nginx-docker- image/ to run a basic Nginx web server in your PC/VM. Add a custom HTML message with your Group members' names and run the server. Show the outcomes.

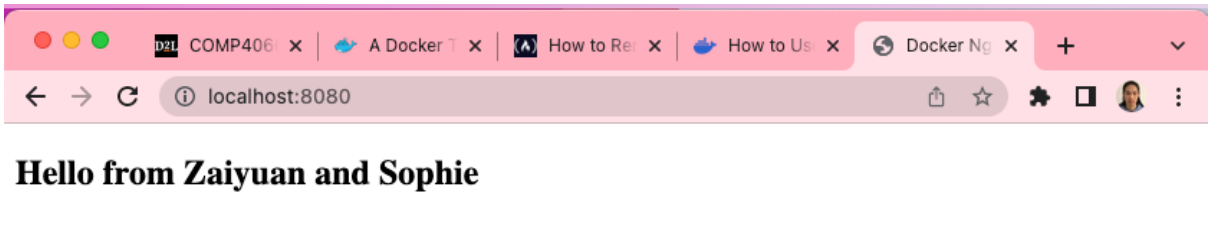
  Run the basic Nginx web server:

```
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker run -it --rm -d -p 8080:80 --name web nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
26c5c85e47da: Pull complete
4f3256bdf66b: Pull complete
2019c71d5655: Pull complete
8c767bdbc9ae: Pull complete
78e14bb05fd3: Pull complete
75576236abf5: Pull complete
Digest: sha256:63b44e8ddb83d5dd8020327c1f40436e37a6fffd3ef2498a6204df23be6e7e94
Status: Downloaded newer image for nginx:latest
ee49bfa44585fdfc4c7d2b7cacc0f7d0d6425be666ac693da6e023a3a5ab305e
zaiyuanwu@Zaiyuans-MacBook-Pro ~ %
```



Create a custom html page and run with -v flag to create a bind mount volume:

```
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker stop web
web
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % cd Desktop
zaiyuanwu@Zaiyuans-MacBook-Pro Desktop % mkdir site-content
zaiyuanwu@Zaiyuans-MacBook-Pro Desktop % cd site-content
zaiyuanwu@Zaiyuans-MacBook-Pro site-content % touch index.html
zaiyuanwu@Zaiyuans-MacBook-Pro site-content % docker run -it --rm -d -p 8080:80 --name web -v ~/Desktop/site-content:/usr/share/
nginx/html nginx
1e8af908842e2a1c56b35d4f527f829758f0d19f85cdbb4a028a2134182ae7fe
zaiyuanwu@Zaiyuans-MacBook-Pro site-content %
```



- Boot up a basic ubuntu container. Install any Linux package inside the container, such as nano or IP-Utils.

Start container:

```
zaiyuanwu@Zaiyuans-MacBook-Pro site-content % docker run -it ubuntu
root@d9ecb15ca619:/# apt-get update
```

Install nano:

```
root@d9ecb15ca619:/# apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 5 not upgraded.
Need to get 280 kB of archives.
After this operation, 881 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 nano amd64 6.2-1 [280 kB]
Fetched 280 kB in 0s (2326 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package nano.
(Reading database ... 4395 files and directories currently installed.)
Preparing to unpack .../archives/nano_6.2-1_amd64.deb ...
Unpacking nano (6.2-1) ...
Setting up nano (6.2-1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /usr/share/man/man1/nano.
1.gz (of link group editor) doesn't exist
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/pico.1.gz because associated file /usr/share/man/man1/nano.1.
gz (of link group pico) doesn't exist
root@d9ecb15ca619:/#
```

```
root@d9ecb15ca619:/# nano --version
 GNU nano, version 6.2
 (C) 1999-2011, 2013-2022 Free Software Foundation, Inc.
 (C) 2014-2022 the contributors to nano
 Compiled options: --disable-libmagic --enable-utf8
root@d9ecb15ca619:/#
```

- Create a directory with your group name inside the container. Inside the container, create a text file and include the group members' names in it. Show the created directory and the file content.

```
root@d9ecb15ca619:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@d9ecb15ca619:/# mkdir Zaiyuan_Sophie
root@d9ecb15ca619:/# cd Zaiyuan_Sophie/
root@d9ecb15ca619:/Zaiyuan_Sophie# touch name.txt
root@d9ecb15ca619:/Zaiyuan_Sophie# nano name.txt
root@d9ecb15ca619:/Zaiyuan_Sophie# cat name.txt
Zaiyuan
Sophie
root@d9ecb15ca619:/Zaiyuan_Sophie#
```

- Commit the Ubuntu container and push its image to the docker hub. Mention the link to the exported docker hub image.

```
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS                     PORTS     NAMES
c2c0614293dc   ubuntu    "/bin/bash"   3 minutes ago   Exited (0) 10 seconds ago            dazzling_engelbart
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker commit c2c0614293dc task1
sha256:074c8d2ed951d42e5c4db8b6b6e371bb4df436a40f5cff9b9974c2be653a5f55
```

```
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker tag task1 willinie/task1
zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker push willinie/task1
Using default tag: latest
The push refers to repository [docker.io/willinie/task1]
d126fb792465: Pushed
b93c1bd012ab: Pushed
latest: digest: sha256:60307d4f4f512df7f0d0e765048b0765e7962719a0a8e20d7b6fcd3c11ad84c3 size: 741
zaiyuanwu@Zaiyuans-MacBook-Pro ~ %
```

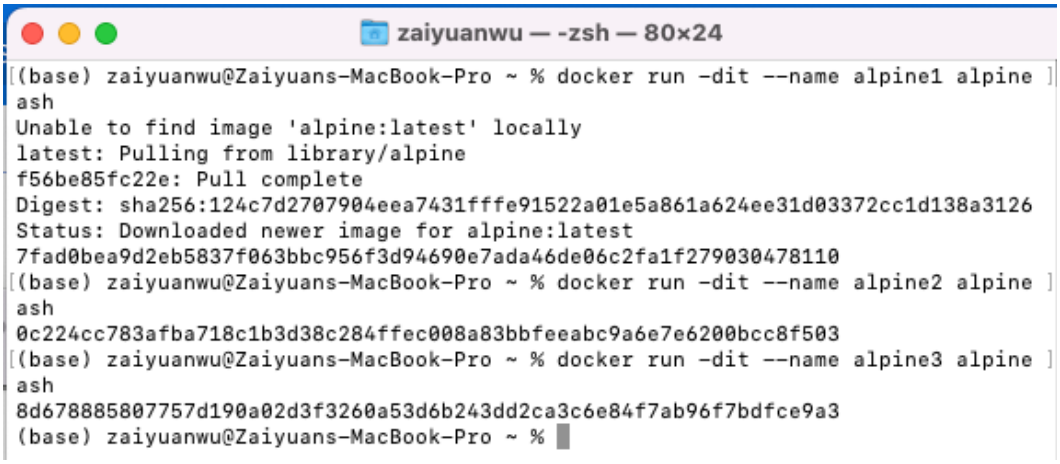
Link: https://hub.docker.com/r/willinie/task1/tags

Task 02: Docker Networking (6 Marks)

https://docs.docker.com/network/

- Follow the instructions at https://docs.docker.com/network/network-tutorial-standalone/ use a default bridge to connect three alpine containers and run ping commands to test their connectivity. Show the ping results.

Run 3 alpine containers:



Test ping command:

```
(base) zaiyuanwu@Zaiyuans-MacBook-Pro ~ % docker attach alpine1
/ # ping alpine2
ping: bad address 'alpine2'
/ # ping -c 2 google.com
PING google.com (172.253.116.138): 56 data bytes
64 bytes from 172.253.116.138: seq=0 ttl=36 time=46.491 ms
64 bytes from 172.253.116.138: seq=1 ttl=36 time=65.231 ms

--- google.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 46.491/55.861/65.231 ms
/ # ping -c 2 172.17.0.3
PING 172.17.0.3 (172.17.0.3): 56 data bytes
64 bytes from 172.17.0.3: seq=0 ttl=64 time=0.173 ms
64 bytes from 172.17.0.3: seq=1 ttl=64 time=0.144 ms

--- 172.17.0.3 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.144/0.158/0.173 ms
/ #
```

- Establish an Inter-Process Communication (IPC) channel between two Ubuntu containers following the instructions at https://medium.com/techanic/docker-containers-ipc-using-sockets-part-1-2ee90885602c
- Leveraging the created IPC channel, send parametric values (at least 50) from container 1 to container 2, emulating an offloading scenario; compute the mean, median, and standard deviation of these parameters at container 2. You might have to extend the python script to establish the IPC communication. Send the computed stats to container 1 and display all the results at each container.

  Images Links:

  https://hub.docker.com/r/willinie/task2part1_server

  https://hub.docker.com/r/willinie/task2part1_client

  Build images:

```
(base) zaiyuanwu@Zaiyuans-MacBook-Pro server % docker build -t my_ipc_server .
[+] Building 0.8s (8/8) FINISHED
```

```
(base) zaiyuanwu@Zaiyuans-MacBook-Pro client % docker build -t my_ipc_client .
[+] Building 1.2s (9/9) FINISHED
```

  Define bridge network:

```
(base) zaiyuanwu@Zaiyuans-MacBook-Pro server % docker network create my_socket_ipc_networ
k
463b5a28568c13adeeb9d465a7bb9ce17a9ce4f7726cabaa9b0f149b63cfa203
(base) zaiyuanwu@Zaiyuans-MacBook-Pro server % docker network ls
NETWORK ID     NAME                   DRIVER    SCOPE
1ecd11d54ebe   bridge                 bridge    local
0f8e5189eb24   host                   host      local
463b5a28568c   my_socket_ipc_network  bridge    local
8ef9ffea9575   none                   null      local
(base) zaiyuanwu@Zaiyuans-MacBook-Pro server %
```

  Run containers separately:

```
(base) zaiyuanwu@Zaiyuans-MacBook-Pro server % docker run --rm --network=my_socket_ipc_ne
twork --name ipc_server_dns_name my_ipc_server
Connected by ('172.19.0.3', 40698)
Sent Average: 24.5, Median: 24.5, Stdev: 14.577379737113251
(base) zaiyuanwu@Zaiyuans-MacBook-Pro server %
```

```
(base) zaiyuanwu@Zaiyuans-MacBook-Pro client % docker run --rm --network=my_socket_ipc_network my_ipc_client
Received '0'
Received '1'
Received '2'
Received '3'
```

```
Received '48'
Received '49'
Received Average: 24.5, Median: 24.5, Stdev: 14.577379737113251
(base) zaiyuanwu@Zaiyuans-MacBook-Pro client %
```

- Emulate another offloading scenario that is relevant for future applications.

  Another scenario we implemented in this task in asymmetrical encryption with rsa-key pair. In this Part2 of this task, we emulated a client-server transmission, including the following steps:

  1. server generate an rsa-key pair, sending public key to client
  2. client receive public key, then encrypt a secret message with it
  3. client send encrypted message to server
  4. server decrypt message with private key

  Images Links:

  https://hub.docker.com/r/willinie/task2part2_server

  https://hub.docker.com/r/willinie/task2part2_client

  Presentation:

  1. Similar to last part, build two images:

```
(base) zaiyuanwu@Zaiyuans-MacBook-Pro server % docker build -t my_ipc_server .
[+] Building 3.5s (10/10) FINISHED
```

```
(base) zaiyuanwu@Zaiyuans-MacBook-Pro client % docker build -t my_ipc_client .
[+] Building 0.5s (9/9) FINISHED
```

  2. Run containers separately:

```
(base) zaiyuanwu@Zaiyuans-MacBook-Pro server % docker run --rm --network=my_socket_ipc_network --name ipc_server_dns_name my_ipc
_server
[rsa-key]generated
[rsa-key]saved to files
Connected by ('172.19.0.3', 45056)
[socket]HEADER sent
[socket]public key sent
[socket]encrypted message received
[message]decrypted -- This is a secret message! Don't share it with anyone.
(base) zaiyuanwu@Zaiyuans-MacBook-Pro server %
```

```
(base) zaiyuanwu@Zaiyuans-MacBook-Pro client % docker run --rm --network=my_socket_ipc_network my_ipc_client
Connected by 172.19.0.2
[socket]HEADER received
public key info: public.pem -- 251bytes
[socket]public key received
-----BEGIN RSA PUBLIC KEY-----
MIGJAoGBALiP2iUVYJSNiM/5A3F3NrEq6ADhTU1RTuyw3mPd35Ifpb742brEkV/H
QhlFZ7wDOrT8Pj7dzhhZgS88JQdFakoZFpkJnW7IwIaZvx7j6C2gJ0aOW4KJ7xcw
6h7vh/YuEl7B4nBjBD7rC3YAzNF6aLSECyIqMdvs04GR7o16o3pDAgMBAAE=
-----END RSA PUBLIC KEY-----

[message] -- This is a secret message! Don't share it with anyone.
[message]encrypted with public key
[socket]encrypted message sent to server
(base) zaiyuanwu@Zaiyuans-MacBook-Pro client %
```