Configuration Manual

MSc Research Project
Programme Name

Bharat Goyal
Student ID: x19215860

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Bharat Goyal
Student ID:	x19215860
Programme:	Programme Name
Year:	2018
Module:	MSc Research Project
Supervisor:	Rashid Mijumbi
Submission Due Date:	20/12/2018
Project Title:	Configuration Manual
Word Count:	533
Page Count:	7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	31st January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

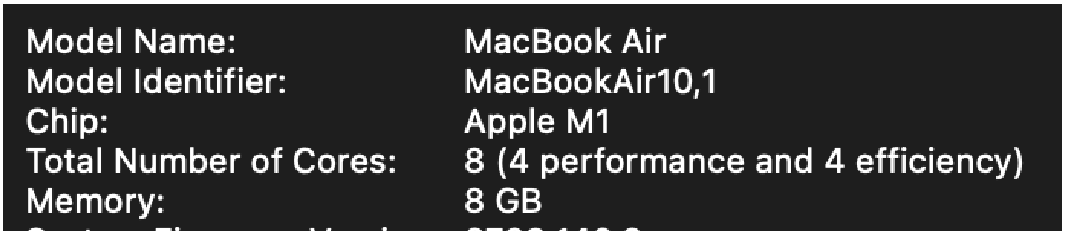
Configuration Manual

Bharat Goyal
x19215860

1 Local System Setup

1.1 System Configuration

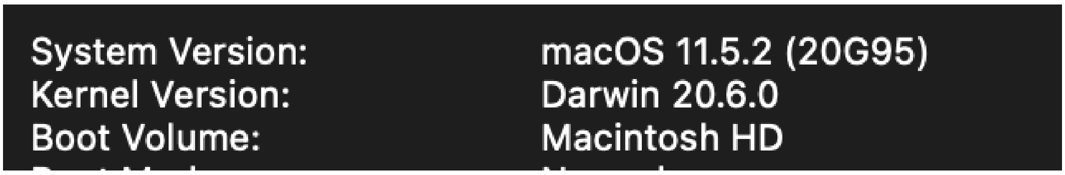
1.1.1 Hardware overview



```
Model Name:                 MacBook Air
Model Identifier:           MacBookAir10,1
Chip:                       Apple M1
Total Number of Cores:      8 (4 performance and 4 efficiency)
Memory:                     8 GB
```

Figure 1: Hardware Overview

1.1.2 Software overview



```
System Version:               macOS 11.5.2 (20G95)
Kernel Version:               Darwin 20.6.0
Boot Volume:                  Macintosh HD
```

Figure 2: Software Overview

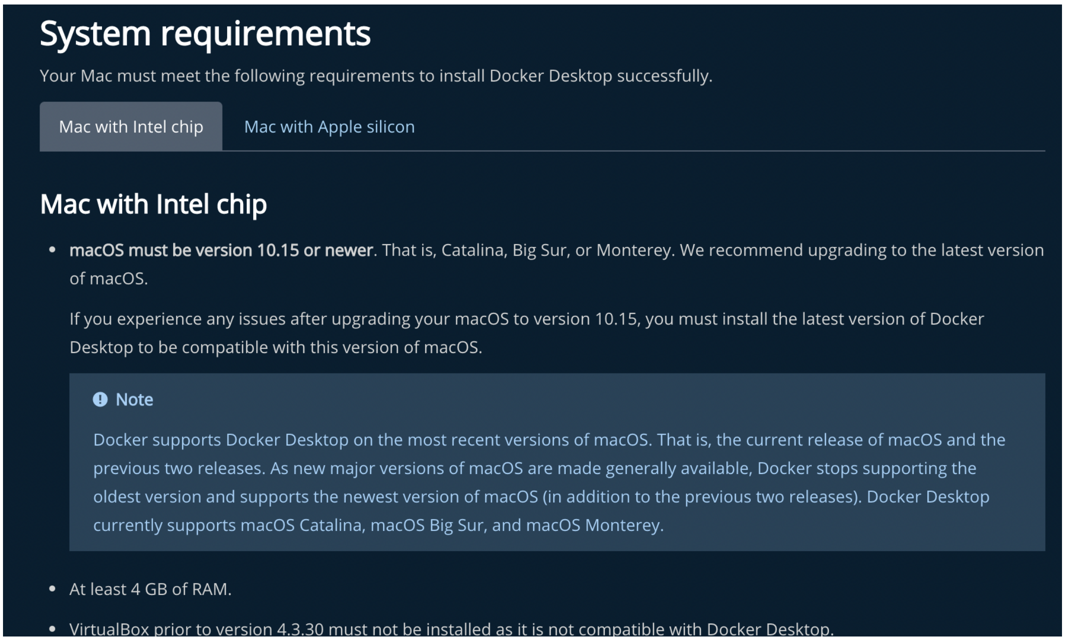
1.2 Docker Setup

1.2.1 Docker Desktop for Mac

1. Docker for Mac is straightforward to set up and can be downloaded from the below link https://docs.docker.com/desktop/mac/install/

2. Depending on the chipset (Apple Chipset or Intel Chipset) the installation file needs to be downloaded. Further installation steps on the same page can be followed to complete the installation

1.2.2 Minimum system requirements for intel chipset



System requirements

Your Mac must meet the following requirements to install Docker Desktop successfully.

Mac with Intel chip

- **macOS must be version 10.15 or newer**. That is, Catalina, Big Sur, or Monterey. We recommend upgrading to the latest version of macOS.

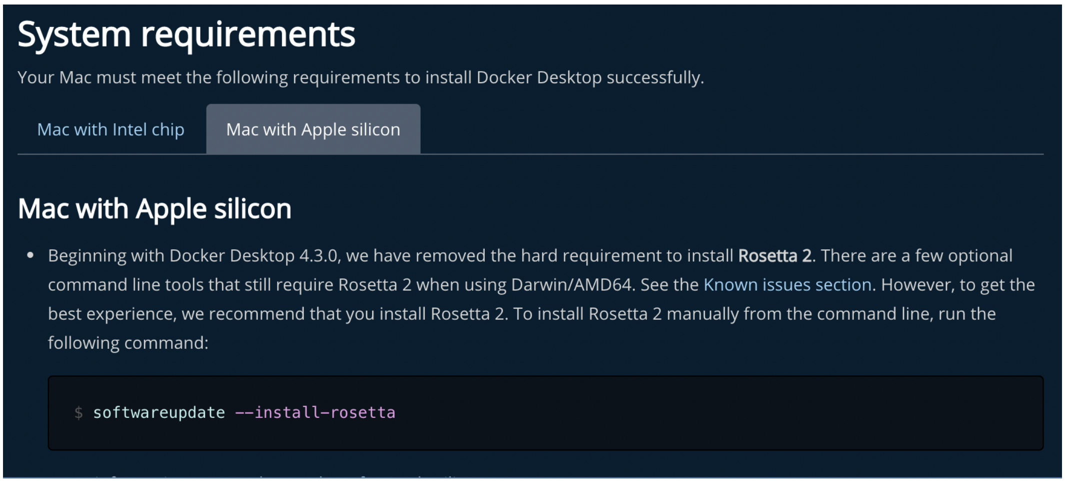
  If you experience any issues after upgrading your macOS to version 10.15, you must install the latest version of Docker Desktop to be compatible with this version of macOS.

  > Note
  >
  > Docker supports Docker Desktop on the most recent versions of macOS. That is, the current release of macOS and the previous two releases. As new major versions of macOS are made generally available, Docker stops supporting the oldest version and supports the newest version of macOS (in addition to the previous two releases). Docker Desktop currently supports macOS Catalina, macOS Big Sur, and macOS Monterey.

- At least 4 GB of RAM.
- VirtualBox prior to version 4.3.30 must not be installed as it is not compatible with Docker Desktop.

Figure 3: Minimum system requirements for Intel chipset

1.2.3 Minimum system requirements for Apple chipset



System requirements

Your Mac must meet the following requirements to install Docker Desktop successfully.

Mac with Apple silicon

- Beginning with Docker Desktop 4.3.0, we have removed the hard requirement to install **Rosetta 2**. There are a few optional command line tools that still require Rosetta 2 when using Darwin/AMD64. See the Known issues section. However, to get the best experience, we recommend that you install Rosetta 2. To install Rosetta 2 manually from the command line, run the following command:

```
$ softwareupdate --install-rosetta
```

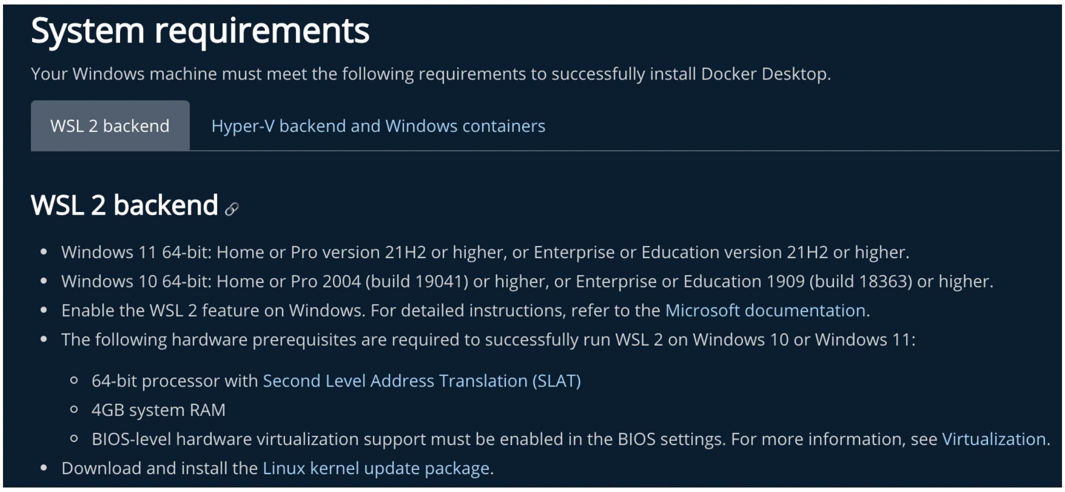
Figure 4: Minimum system requirements for Apple chipset

1.2.4 Docker Desktop for Windows

1. Docker for windows is straightforward to be set up and can be downloaded from https://docs.docker.com/desktop/windows/install/

2. Further installation steps in the same page can be followed to complete the installation

1.2.5 Minimum System Requirements for Windows



System requirements

Your Windows machine must meet the following requirements to successfully install Docker Desktop.

WSL 2 backend

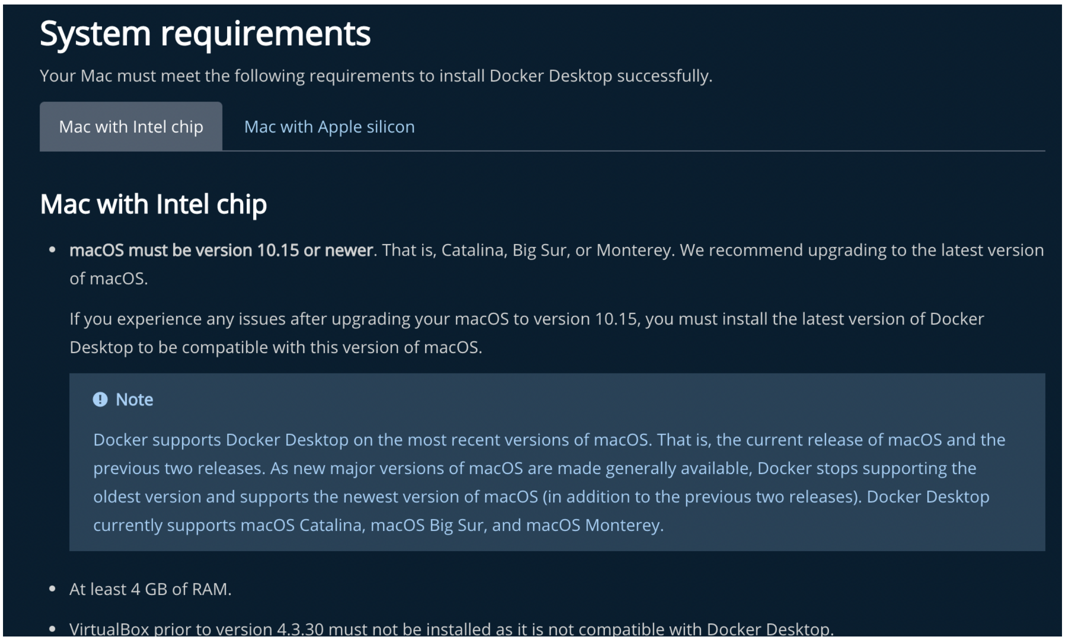
- Windows 11 64-bit: Home or Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.
- Windows 10 64-bit: Home or Pro 2004 (build 19041) or higher, or Enterprise or Education 1909 (build 18363) or higher.
- Enable the WSL 2 feature on Windows. For detailed instructions, refer to the Microsoft documentation.
- The following hardware prerequisites are required to successfully run WSL 2 on Windows 10 or Windows 11:
  - 64-bit processor with Second Level Address Translation (SLAT)
  - 4GB system RAM
  - BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see Virtualization.
- Download and install the Linux kernel update package.

Figure 5: Minimum System Requirements for Windows

2 Docker Setup for ubuntu VM(GCP)

1. Docker for Mac is straightforward to set up and can be downloaded from the below link https://docs.docker.com/desktop/mac/install/
2. Depending on the chipset (Apple Chipset or Intel Chipset) the installation file needs to be downloaded. Further installation steps on the same page can be followed to complete the installation

2.0.1 Minimum system requirements for intel chipset



System requirements

Your Mac must meet the following requirements to install Docker Desktop successfully.

Mac with Intel chip

- **macOS must be version 10.15 or newer**. That is, Catalina, Big Sur, or Monterey. We recommend upgrading to the latest version of macOS.

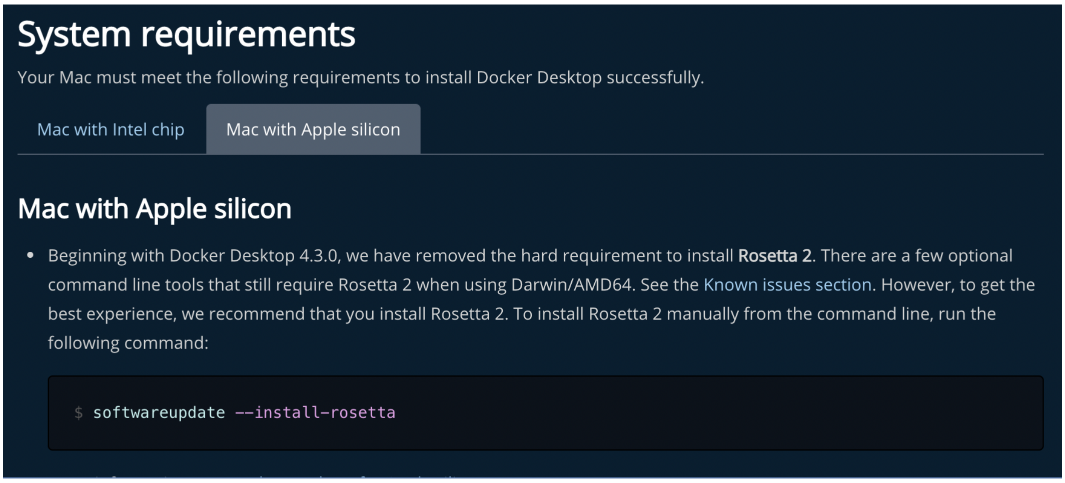
  If you experience any issues after upgrading your macOS to version 10.15, you must install the latest version of Docker Desktop to be compatible with this version of macOS.

  > Note
  >
  > Docker supports Docker Desktop on the most recent versions of macOS. That is, the current release of macOS and the previous two releases. As new major versions of macOS are made generally available, Docker stops supporting the oldest version and supports the newest version of macOS (in addition to the previous two releases). Docker Desktop currently supports macOS Catalina, macOS Big Sur, and macOS Monterey.

- At least 4 GB of RAM.
- VirtualBox prior to version 4.3.30 must not be installed as it is not compatible with Docker Desktop.

Figure 6: Minimum system requirements for Intel chipset

2.0.2 Minimum system requirements for Apple chipset



System requirements

Your Mac must meet the following requirements to install Docker Desktop successfully.

Mac with Apple silicon

- Beginning with Docker Desktop 4.3.0, we have removed the hard requirement to install **Rosetta 2**. There are a few optional command line tools that still require Rosetta 2 when using Darwin/AMD64. See the Known issues section. However, to get the best experience, we recommend that you install Rosetta 2. To install Rosetta 2 manually from the command line, run the following command:

```
$ softwareupdate --install-rosetta
```

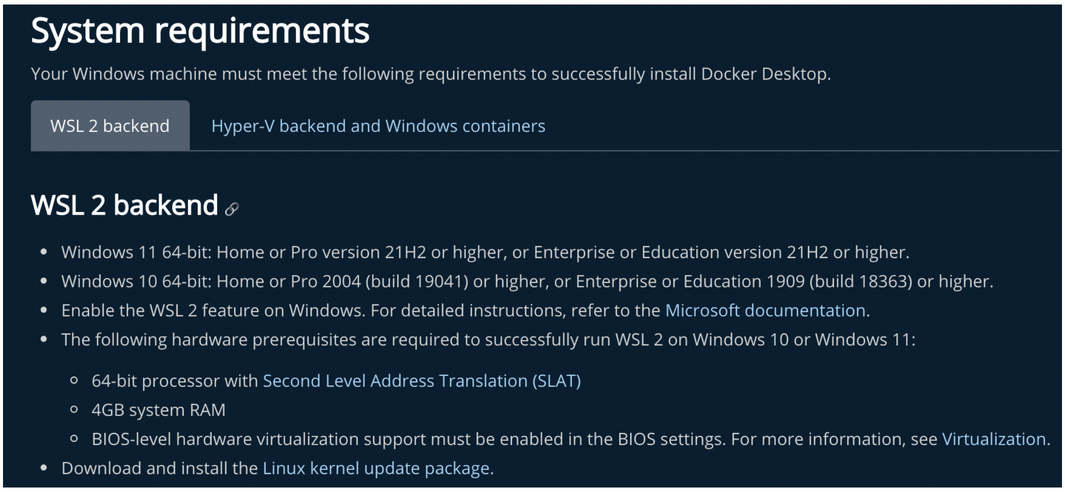
Figure 7: Minimum system requirements for Apple chipset

2.0.3 Docker Desktop for Windows

1. Docker for windows is straightforward to be set up and can be downloaded from https://docs.docker.com/desktop/windows/install/

2. Further installation steps in the same page can be followed to complete the installation

2.0.4 Minimum System Requirements for Windows



System requirements

Your Windows machine must meet the following requirements to successfully install Docker Desktop.

WSL 2 backend

- Windows 11 64-bit: Home or Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.
- Windows 10 64-bit: Home or Pro 2004 (build 19041) or higher, or Enterprise or Education 1909 (build 18363) or higher.
- Enable the WSL 2 feature on Windows. For detailed instructions, refer to the Microsoft documentation.
- The following hardware prerequisites are required to successfully run WSL 2 on Windows 10 or Windows 11:
  - 64-bit processor with Second Level Address Translation (SLAT)
  - 4GB system RAM
  - BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see Virtualization.
- Download and install the Linux kernel update package.

Figure 8: Minimum System Requirements for Windows

2.1 Code Setup

2.1.1 Start the containers using without cluster mode

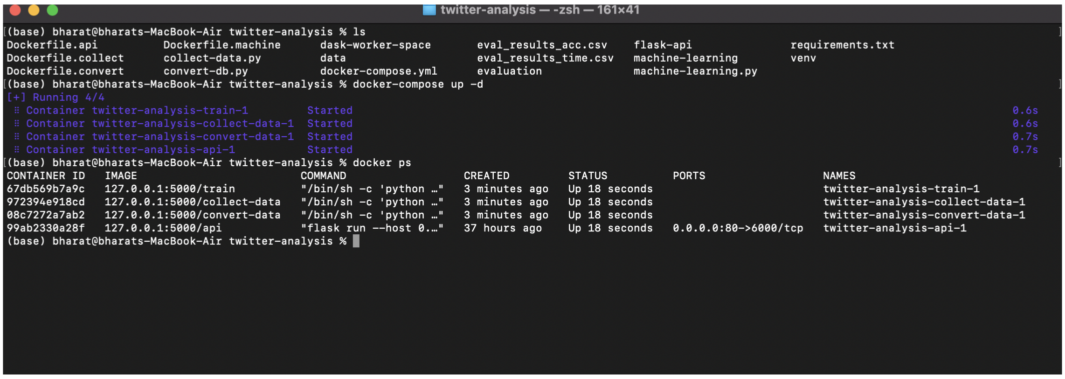
1. Unzip the code folder.

2. Go to terminal and locate folder using the cd command.

3. Run the command docker-compose up -d.

4. The above command Builds, (re)creates, starts, and attaches to containers for a service. The above command does these things in the detached mode so that processes are not shown. If anyone wants to view the processes then the docker-compose up command runs the containers in detached mode.

5. The above process might take some time depending on the network bandwidth as for the first time all the python images needs to be downloaded and container images need to be built.



```
(base) bharat@bharats-MacBook-Air twitter-analysis % ls
Dockerfile.api          Dockerfile.machine      dask-worker-space       eval_results_acc.csv    flask-api               requirements.txt
Dockerfile.collect      collect-data.py         data                    eval_results_time.csv   machine-learning        venv
Dockerfile.convert      convert-db.py           docker-compose.yml      evaluation              machine-learning.py
(base) bharat@bharats-MacBook-Air twitter-analysis % docker-compose up -d
[+] Running 4/4
 ⠿ Container twitter-analysis-train-1         Started                                                                                           0.6s
 ⠿ Container twitter-analysis-collect-data-1  Started                                                                                           0.6s
 ⠿ Container twitter-analysis-convert-data-1  Started                                                                                           0.7s
 ⠿ Container twitter-analysis-api-1           Started                                                                                           0.7s
(base) bharat@bharats-MacBook-Air twitter-analysis % docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED          STATUS          PORTS                    NAMES
67db569b7a9c   127.0.0.1:5000/train           "/bin/sh -c 'python …"   3 minutes ago    Up 18 seconds                            twitter-analysis-train-1
972394e918cd   127.0.0.1:5000/collect-data    "/bin/sh -c 'python …"   3 minutes ago    Up 18 seconds                            twitter-analysis-collect-data-1
08c7272a7ab2   127.0.0.1:5000/convert-data    "/bin/sh -c 'python …"   3 minutes ago    Up 18 seconds                            twitter-analysis-convert-data-1
99ab2330a28f   127.0.0.1:5000/api             "flask run --host 0.…"   37 hours ago     Up 18 seconds   0.0.0.0:80->6000/tcp     twitter-analysis-api-1
(base) bharat@bharats-MacBook-Air twitter-analysis % 
```

Figure 9: Start the containers using without cluster mode

2.1.2 Deploy the containers on a cluster using Docker Swarm

1. Unzip the code folder.

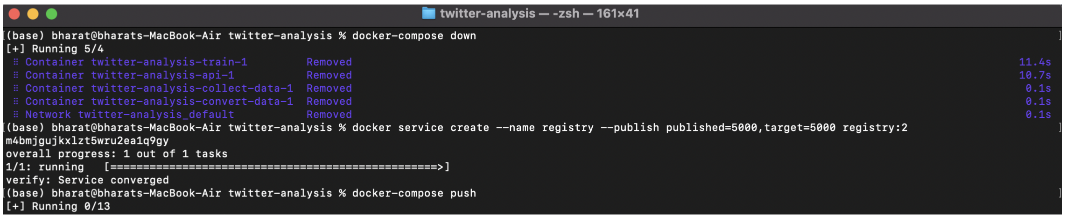
2. Go to terminal and locate folder using the cd command.

3. Run the command *docker-compose build*

4. Run the command *docker-compose up -d*

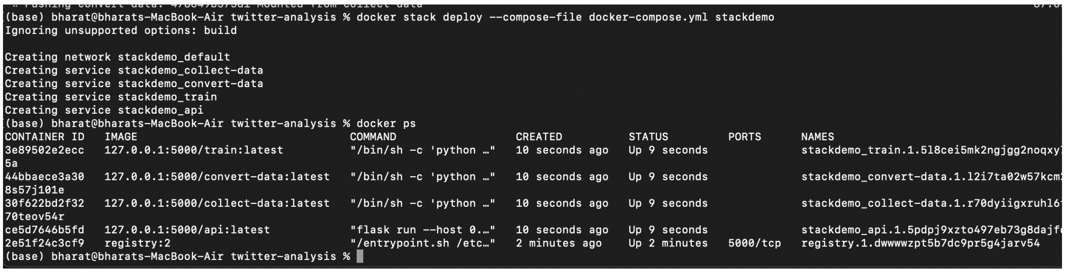
5. The above two command Builds, (re)creates, starts, and attaches to containers for a service. The above command does these things in the detached mode so that processes are not shown. If anyone wants to view the processes then the docker-compose up command runs the containers in detached mode.

6. The above process might take some time depending on the network bandwidth as for the first time all the python images needs to be downloaded and container images need to be built.



```
(base) bharat@bharats-MacBook-Air twitter-analysis % docker-compose down
[+] Running 5/4
 ⠿ Container twitter-analysis-train-1         Removed                                                                                          11.4s
 ⠿ Container twitter-analysis-api-1           Removed                                                                                          10.7s
 ⠿ Container twitter-analysis-collect-data-1  Removed                                                                                           0.1s
 ⠿ Container twitter-analysis-convert-data-1  Removed                                                                                           0.1s
 ⠿ Network twitter-analysis_default           Removed                                                                                           0.1s
(base) bharat@bharats-MacBook-Air twitter-analysis % docker service create --name registry --publish published=5000,target=5000 registry:2
m4bmjgujkxlzt5wru2ea1q9gy
overall progress: 1 out of 1 tasks
1/1: running   [==================================================>]
verify: Service converged
(base) bharat@bharats-MacBook-Air twitter-analysis % docker-compose push
[+] Running 0/13
```

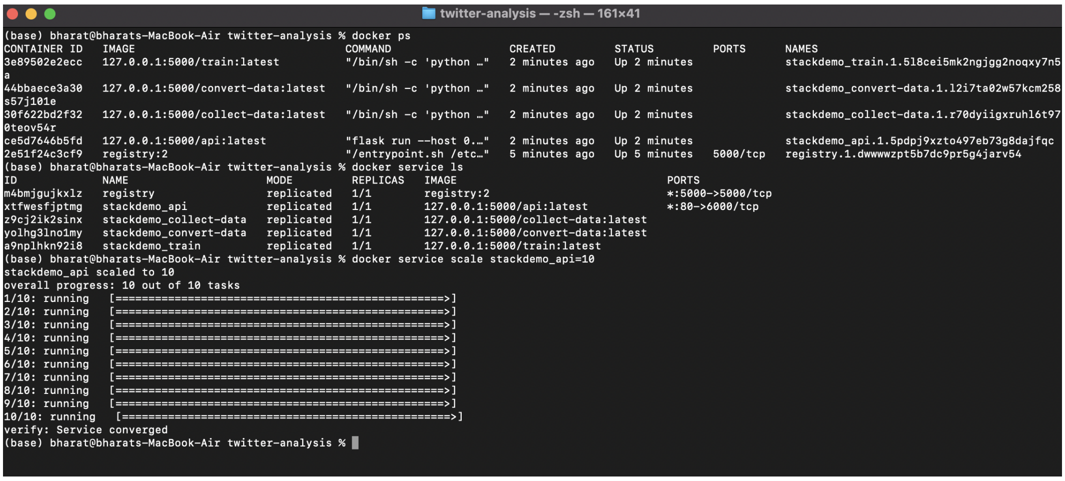
Figure 10: Deploy the containers on cluster using Docker Swarm



```
(base) bharat@bharats-MacBook-Air twitter-analysis % docker stack deploy --compose-file docker-compose.yml stackdemo
Ignoring unsupported options: build

Creating network stackdemo_default
Creating service stackdemo_collect-data
Creating service stackdemo_convert-data
Creating service stackdemo_train
Creating service stackdemo_api
(base) bharat@bharats-MacBook-Air twitter-analysis % docker ps
CONTAINER ID   IMAGE                               COMMAND                  CREATED          STATUS         PORTS      NAMES
3e89502e2ecc   127.0.0.1:5000/train:latest         "/bin/sh -c 'python …"   10 seconds ago   Up 9 seconds              stackdemo_train.1.518cei5mk2ngjgg2noqxy
5a
44bbaece3a30   127.0.0.1:5000/convert-data:latest   "/bin/sh -c 'python …"   10 seconds ago   Up 9 seconds              stackdemo_convert-data.1.l2i7ta02w57kcm
8s57j101e
30f622bd2f32   127.0.0.1:5000/collect-data:latest   "/bin/sh -c 'python …"   10 seconds ago   Up 9 seconds              stackdemo_collect-data.1.r70dyiigxruhl6
70teov54r
ce5d7646b5fd   127.0.0.1:5000/api:latest           "flask run --host 0.…"   10 seconds ago   Up 9 seconds              stackdemo_api.1.5pdpj9xzto497eb73g8dajf
2e51f24c3cf9   registry:2                          "/entrypoint.sh /etc…"   2 minutes ago    Up 2 minutes   5000/tcp   registry.1.dwwwwzpt5b7dc9pr5g4jarv54
(base) bharat@bharats-MacBook-Air twitter-analysis % 
```

Figure 11: Deploy the containers on cluster using Docker Swarm



```
(base) bharat@bharats-MacBook-Air twitter-analysis % docker ps
CONTAINER ID   IMAGE                               COMMAND                  CREATED         STATUS         PORTS      NAMES
3e89502e2ecc   127.0.0.1:5000/train:latest         "/bin/sh -c 'python …"   2 minutes ago   Up 2 minutes              stackdemo_train.1.518cei5mk2ngjgg2noqxy7n5
a
44bbaece3a30   127.0.0.1:5000/convert-data:latest   "/bin/sh -c 'python …"   2 minutes ago   Up 2 minutes              stackdemo_convert-data.1.l2i7ta02w57kcm258
s57j101e
30f622bd2f32   127.0.0.1:5000/collect-data:latest   "/bin/sh -c 'python …"   2 minutes ago   Up 2 minutes              stackdemo_collect-data.1.r70dyiigxruhl6t97
0teov54r
ce5d7646b5fd   127.0.0.1:5000/api:latest           "flask run --host 0.…"   2 minutes ago   Up 2 minutes              stackdemo_api.1.5pdpj9xzto497eb73g8dajfqc
2e51f24c3cf9   registry:2                          "/entrypoint.sh /etc…"   5 minutes ago   Up 5 minutes   5000/tcp   registry.1.dwwwwzpt5b7dc9pr5g4jarv54
(base) bharat@bharats-MacBook-Air twitter-analysis % docker service ls
ID             NAME                     MODE         REPLICAS   IMAGE                                PORTS
m4bmjgujkxlz   registry                 replicated   1/1        registry:2                           *:5000->5000/tcp
xtfwesfjptmg   stackdemo_api            replicated   1/1        127.0.0.1:5000/api:latest            *:80->6000/tcp
z9cj2ik2sinx   stackdemo_collect-data   replicated   1/1        127.0.0.1:5000/collect-data:latest
yolhg3lno1my   stackdemo_convert-data   replicated   1/1        127.0.0.1:5000/convert-data:latest
a9nplhkn92i8   stackdemo_train          replicated   1/1        127.0.0.1:5000/train:latest
(base) bharat@bharats-MacBook-Air twitter-analysis % docker service scale stackdemo_api=10
stackdemo_api scaled to 10
overall progress: 10 out of 10 tasks
1/10: running   [==================================================>]
2/10: running   [==================================================>]
3/10: running   [==================================================>]
4/10: running   [==================================================>]
5/10: running   [==================================================>]
6/10: running   [==================================================>]
7/10: running   [==================================================>]
8/10: running   [==================================================>]
9/10: running   [==================================================>]
10/10: running   [==================================================>]
verify: Service converged
(base) bharat@bharats-MacBook-Air twitter-analysis % 
```

Figure 12: Deploy the containers on cluster using Docker Swarm

3 Docker Installation on Linux virtual machines.

Run the following commands in the Linux terminal.

1. sudo apt-get install ca-certificates curl gnupg lsb-release
2. curl -fsSL https://download.docker.com/linux/ubuntu/gpg — sudo gpg –dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
3. sudo apt-get update
4. sudo apt-get install docker-ce docker-ce-cli containerd.io

References

https://docs.docker.com/engine/install/