Assignment 5(Docker Containers)

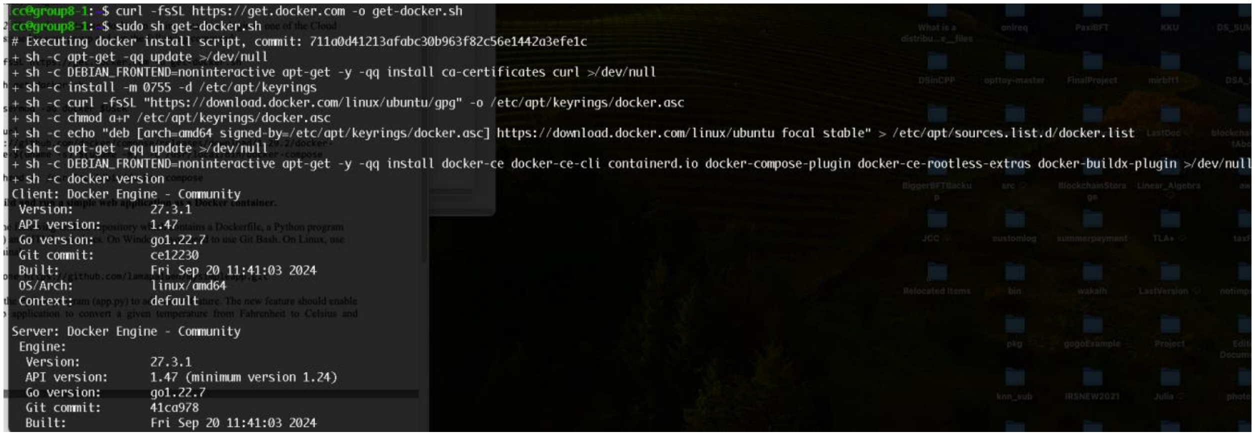
CS 5573 Cloud Computing
Submission by: Group 8:
Amjad Alqahtani, Nicholas Winkelmann, and Caleb Alva

Docker Installation

We selected option 2. We logged into the cloud VMs assigned to our group and executed the command instructions as follows.

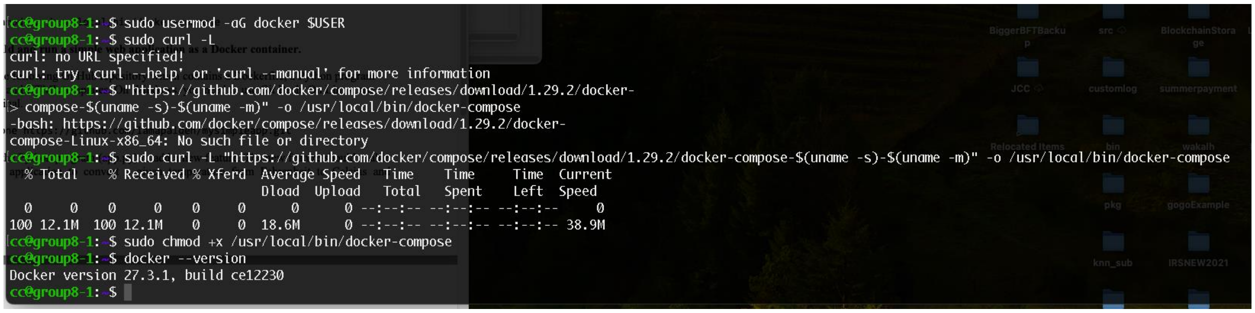
```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo usermod -aG docker $USER
sudo curl -L
"https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

The installation is done, and I confirm it by version command inquiry



```
cc@group8-1:~$ curl -fsSL https://get.docker.com -o get-docker.sh
cc@group8-1:~$ sudo sh get-docker.sh
# Executing docker install script, commit: 711a0d41213afabc30b963f82c56e1442a3efe1c
+ sh -c apt-get -qq update >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install ca-certificates curl >/dev/null
+ sh -c install -m 0755 -d /etc/apt/keyrings
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" -o /etc/apt/keyrings/docker.asc
+ sh -c chmod a+r /etc/apt/keyrings/docker.asc
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu focal stable" > /etc/apt/sources.list.d/docker.list
+ sh -c apt-get -qq update >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-ce-rootless-extras docker-buildx-plugin >/dev/null
+ sh -c docker version
Client: Docker Engine - Community
 Version:           27.3.1
 API version:       1.47
 Go version:        go1.22.7
 Git commit:        ce12230
 Built:             Fri Sep 20 11:41:03 2024
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          27.3.1
  API version:      1.47 (minimum version 1.24)
  Go version:       go1.22.7
  Git commit:       41ca978
  Built:            Fri Sep 20 11:41:03 2024
```

After the installation, to confirm everything is right, we run the command: docker --version

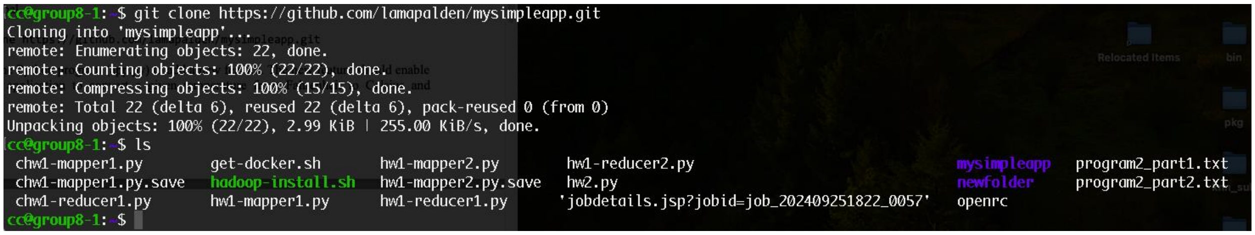


```
cc@group8-1:~$ sudo usermod -aG docker $USER
cc@group8-1:~$ sudo curl -L
curl: no URL specified!
curl: try 'curl --help' or 'curl --manual' for more information
cc@group8-1:~$ "https://github.com/docker/compose/releases/download/1.29.2/docker-
> compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
-bash: https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-Linux-x86_64: No such file or directory
cc@group8-1:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 12.1M  100 12.1M    0     0  18.6M      0 --:--:-- --:--:-- --:--:-- 38.9M
cc@group8-1:~$ sudo chmod +x /usr/local/bin/docker-compose
cc@group8-1:~$ docker --version
Docker version 27.3.1, build ce12230
cc@group8-1:~$
```

Build and run a simple web application as a Docker container.

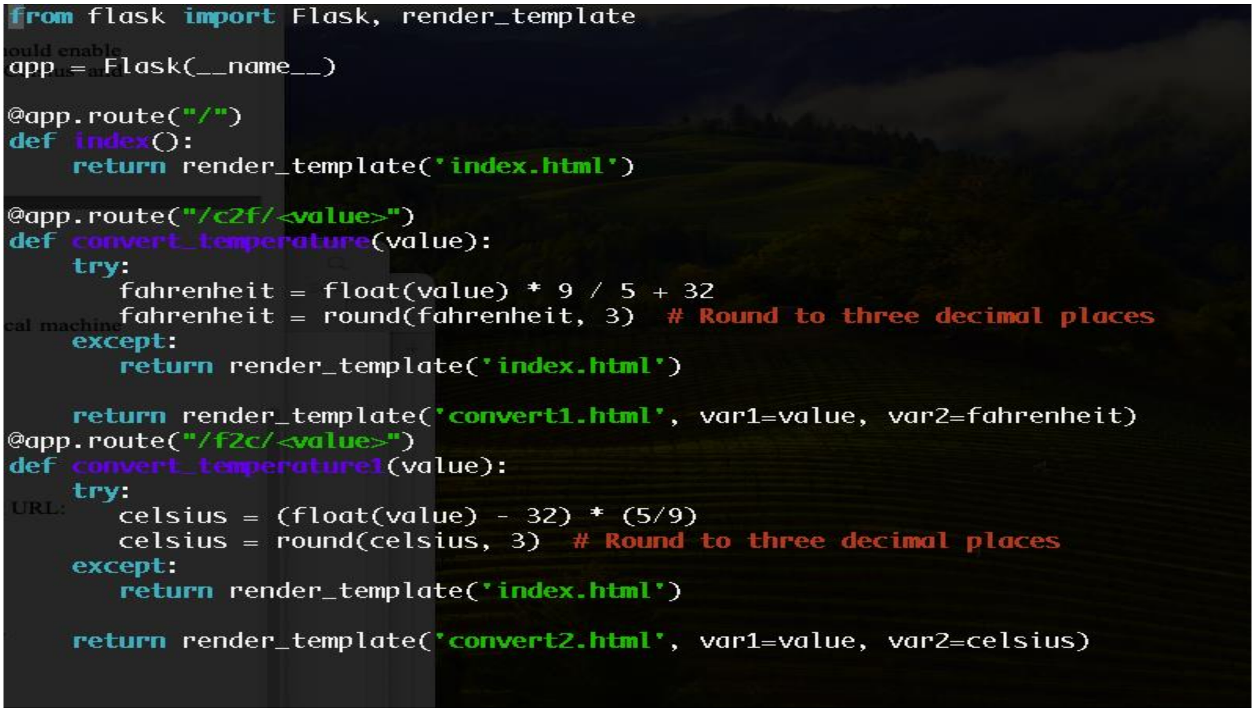
(a) Clone the following GitHub repository which contains a Dockerfile, a Python program (app.py) and HTML templates. On Windows, you need to use Git Bash. On Linux, use the terminal.
git clone https://github.com/lamapalden/mysimpleapp.git

We have taken this step.



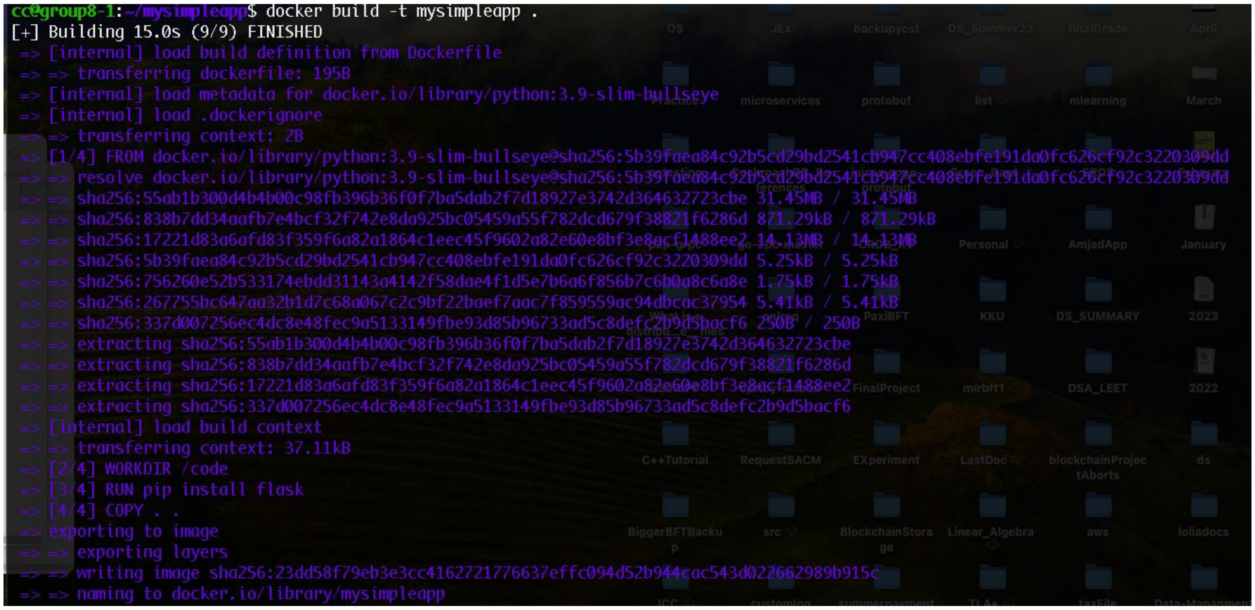
```
cc@group8-1:~$ git clone https://github.com/lamapalden/mysimpleapp.git
Cloning into 'mysimpleapp'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 22 (delta 6), reused 22 (delta 6), pack-reused 0 (from 0)
Unpacking objects: 100% (22/22), 2.99 KiB | 255.00 KiB/s, done.
cc@group8-1:~$ ls
 chw1-mapper1.py        get-docker.sh       hw1-mapper2.py        hw1-reducer2.py                              mysimpleapp   program2_part1.txt
 chw1-mapper1.py.save   hadoop-install.sh   hw1-mapper2.py.save   hw2.py                                       newfolder     program2_part2.txt
 chw1-reducer1.py       hw1-mapper1.py      hw1-reducer1.py      'jobdetails.jsp?jobid=job_202409251822_0057'  openrc
cc@group8-1:~$
```

(b) Update the Python program (app.py) to add a new feature. The new feature should enable the web application to convert a given temperature from Fahrenheit to Celsius and display the converted temperature when you open a web browser on your local machine
using the appropriate URL as described below.
If you are using Docker on local Windows machine, use the following URL:
localhost/f2c/<temperature>
For example,
localhost/f2c/32
If you are using Docker on the Cloud VM (Linux machine), use the following URL:
<VM’s public IP address>/f2c/<temperature>
For example,
129.114.27.107/f2c/32



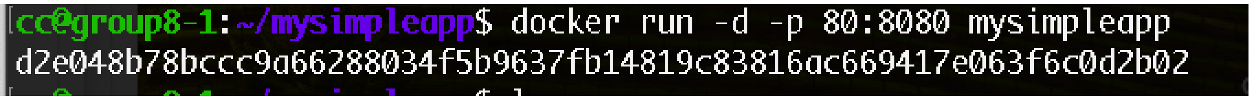
```python
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/")
def index():
    return render_template('index.html')

@app.route("/c2f/<value>")
def convert_temperature(value):
    try:
        fahrenheit = float(value) * 9 / 5 + 32
        fahrenheit = round(fahrenheit, 3)  # Round to three decimal places
    except:
        return render_template('index.html')

    return render_template('convert1.html', var1=value, var2=fahrenheit)
@app.route("/f2c/<value>")
def convert_temperature1(value):
    try:
        celsius = (float(value) - 32) * (5/9)
        celsius = round(celsius, 3)  # Round to three decimal places
    except:
        return render_template('index.html')

    return render_template('convert2.html', var1=value, var2=celsius)
```

(c) Use appropriate command to build the Docker image of your web application.
 (**docker build –t mysimpleapp .**)



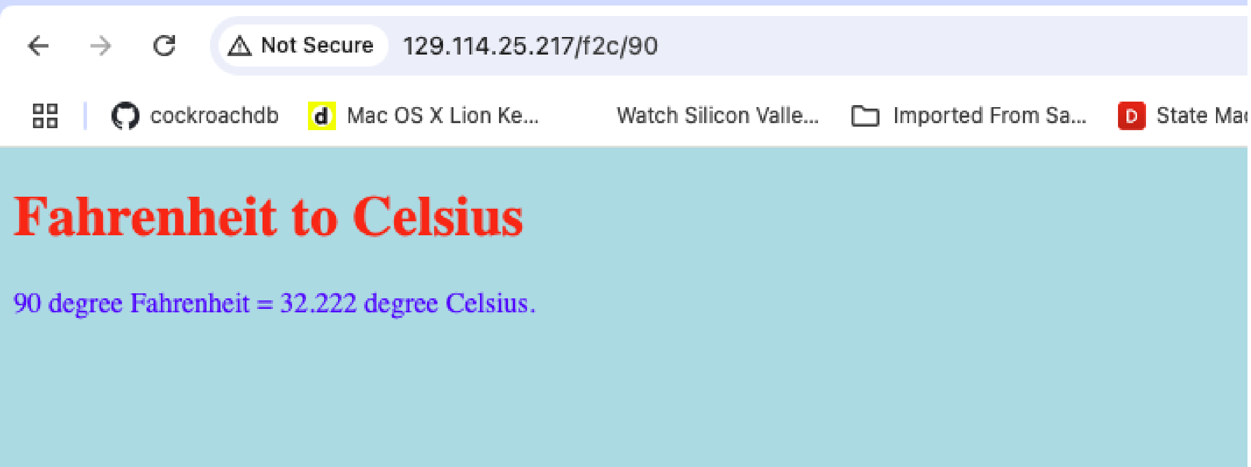
```
cc@group8-1:~/mysimpleapp$ docker build -t mysimpleapp .
[+] Building 15.0s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 195B
 => [internal] load metadata for docker.io/library/python:3.9-slim-bullseye
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/4] FROM docker.io/library/python:3.9-slim-bullseye@sha256:5b39faea84c92b5cd29bd2541cb947cc408ebfe191da0fc626cf92c3220309dd
 => => resolve docker.io/library/python:3.9-slim-bullseye@sha256:5b39faea84c92b5cd29bd2541cb947cc408ebfe191da0fc626cf92c3220309dd
 => => sha256:55ab1b300d4b4b00c98fb396b36f0f7ba5dab2f7d18927e3742d364632723cbe 31.45MB / 31.45MB
 => => sha256:838b7dd34aafb7e4bcf32f742e8da925bc05459a55f782dcd679f38821f6286d 871.29kB / 871.29kB
 => => sha256:17221d83a6afd83f359f6a82a1864c1eec45f9602a82e60e8bf3e8acf1488ee2 14.13MB / 14.13MB
 => => sha256:5b39faea84c92b5cd29bd2541cb947cc408ebfe191da0fc626cf92c3220309dd 5.25kB / 5.25kB
 => => sha256:756260e52b533174ebdd31143a4142f58dae4f1d5e7b6a6f856b7c6b0a8c6a8e 1.75kB / 1.75kB
 => => sha256:267755bc647aa32b1d7c68a067c2c9bf22baef7aac7f859559ac94dbcac37954 5.41kB / 5.41kB
 => => sha256:337d007256ec4dc8e48fec9a5133149fbe93d85b96733ad5c8defc2b9d5bacf6 250B / 250B
 => => extracting sha256:55ab1b300d4b4b00c98fb396b36f0f7ba5dab2f7d18927e3742d364632723cbe
 => => extracting sha256:838b7dd34aafb7e4bcf32f742e8da925bc05459a55f782dcd679f38821f6286d
 => => extracting sha256:17221d83a6afd83f359f6a82a1864c1eec45f9602a82e60e8bf3e8acf1488ee2
 => => extracting sha256:337d007256ec4dc8e48fec9a5133149fbe93d85b96733ad5c8defc2b9d5bacf6
 => [internal] load build context
 => => transferring context: 37.11kB
 => [2/4] WORKDIR /code
 => [3/4] RUN pip install flask
 => [4/4] COPY . .
 => exporting to image
 => => exporting layers
 => => writing image sha256:23dd58f79eb3e3cc4162721776637effc094d52b944cac543d022662989b915c
 => => naming to docker.io/library/mysimpleapp
```

(d) Use appropriate command to run the web application as a Docker container.
(**docker run –d –p 80:8080 mysimpleapp**)



```
cc@group8-1:~/mysimpleapp$ docker run -d -p 80:8080 mysimpleapp
d2e048b78bccc9a66288034f5b9637fb14819c83816ac669417e063f6c0d2b02
```

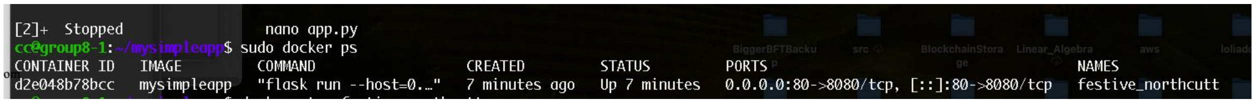
(e) Open a web browser and use appropriate URL to convert any temperature from Fahrenheit to Celsius.



Fahrenheit to Celsius

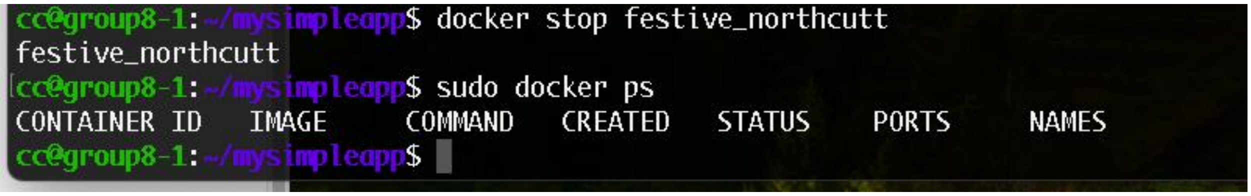
90 degree Fahrenheit = 32.222 degree Celsius.

(f) Use appropriate command to find the name of the running container.
(**sudo docker ps**)



```
[2]+  Stopped                 nano app.py
cc@group8-1:~/mysimpleapp$ sudo docker ps
CONTAINER ID   IMAGE         COMMAND                  CREATED         STATUS         PORTS                                   NAMES
d2e048b78bcc   mysimpleapp   "flask run --host=0.…"   7 minutes ago   Up 7 minutes   0.0.0.0:80->8080/tcp, [::]:80->8080/tcp   festive_northcutt
```

(g) Use appropriate command to stop the running container.
(**docker stop festive_northcutt**)



```
cc@group8-1:~/mysimpleapp$ docker stop festive_northcutt
festive_northcutt
cc@group8-1:~/mysimpleapp$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
cc@group8-1:~/mysimpleapp$ 
```

Hints: Formula to convert Fahrenheit to Celsius is as follows:
Celsius = (Fahrenheit - 32) × 5/9

Group Tasks

Amjad Alqahtani – wkh221

1. I contributed to setting up the environment, ensuring the system is up and running
2. I contributed to debugging issues that arose during testing
3. I documented our processes, including setup instructions
4. I actively participated in group meetings and sharing insights
5. I contributed to the final report

Nicholas Winkelmann - mhj052

1. Helped test application in Docket container
2. Helped write the documentation for the assignment
3. Participated in our group meetings
4. Tested code.

Caleb Alva – uxy606

1. Helped add new feature to the app
2. Helped write documentation
3. Participated in group meetings
4. Tested the code on the container.