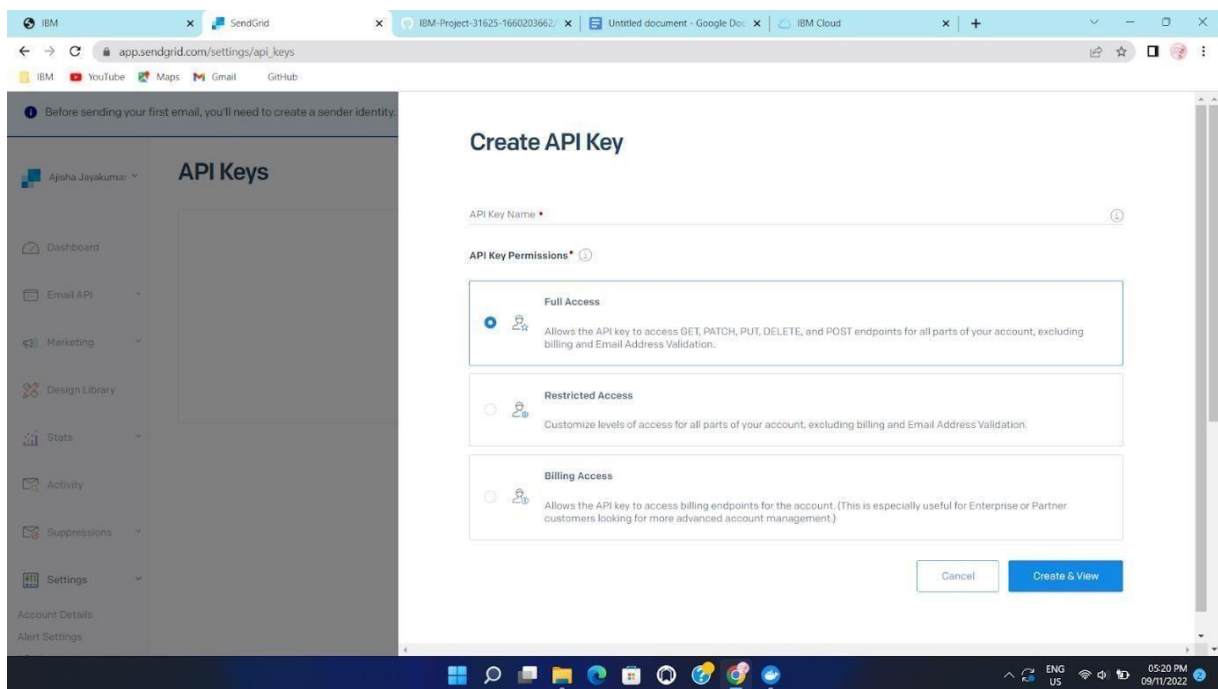


# PROJECT DEVELOPMENT PHASE

## SPRINT 4

Date	18 October 2022
Team ID	PNT2022TMID19420
Project Name	Smart Fashion Recommender
Maximum Marks	8 Marks

### 1. Sendgrid integration with python



IBM SendGrid IBM Project 31525-1660203642/ X Untitled document - Google Docs X IBM Cloud

apps.sendgrid.com/settings/api\_keys

Before sending your first email, you'll need to create a sender identity. [Create a sender identity](#)

## API Keys

Create API Key

NAME	API KEY	ACTION
trendy-fashion API Key ID: A7HtO1HRWPZ6LLNfrvg	*****	

Account Details  
Alert Settings

```
Administration: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> pip install sendgrid
Collecting sendgrid
  Downloading sendgrid-6.9.7-py3-none-any.whl (101 kB)
    |#####| 101.1/101.1 kB 447.4 kB/s eta 0:00:00
Collecting python-http-client>=3.2.1
  Downloading python_http_client-3.3.1-py3-none-any.whl (11.4 kB)
Collecting starkbank-ecdsa>=2.0.1
  Downloading starkbank-ecdsa-2.2.0.tar.gz (14 kB)
  Preparing metadata (setup.py) ... done
Installing collected packages: starkbank-ecdsa, python-http-client, sendgrid
DEPRECATION: starkbank-ecdsa is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
Running setup.py install for starkbank-ecdsa ... done
Successfully installed python-http-client-3.3.1 sendgrid-6.9.7 starkbank-ecdsa-2.2.0

[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Windows\system32>
```

SendGrid

Sender Authentication /

Single Sender Verification

SENDERS

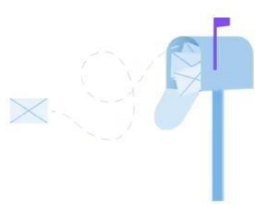
ADDRESS

Ponvarshini

FROM ponvarshiniarumugam23@gmail.com

REPLY mohamedasim71@gmail.com

Sri Krishna College of E  
Colmbatore, 641008 IN



Sender has been created

To verify sender identity, check your inbox  
at ponvarshiniarumugam23@gmail.com.

Resend email

Close

mail.google.com/mail/u/0/#inbox/FMfczGqRZZXpFxpRnfHqSfrVRJpbMhQ

IBM

YouTube

Maps

Gmail

GitHub

Compose

Inbox 613

Starred

Snoozed

Sent

Drafts 16

More

Labels +

Please Verify Your Single Sender

SendGrid

no-reply@sendgrid.com

Unsubscribe

11:09 PM (0 minutes ago)

Let's verify your single sender so you can start  
sending email.

[ajishaj503@gmail.com](#)

Your link is active for 48 hours. After that, you will need to resend the verification email.

Verify Single Sender

Sending request...

ENG US

11:09 PM

10/11/2022

The screenshot shows a Visual Studio Code editor window with a file explorer on the left. The file explorer shows a project named 'SMART FASHION RECOM...' with various files and folders. The main editor window displays a Python script named 'sendgrid.py'. The script uses the SendGrid Python library to send an email. The email content is: 'Sending with Twilio SendGrid is fun', 'html\_content' is set to '<strong>and easy to do anywhere, even with Python</strong>'. The script also includes error handling for exceptions.

```
1 # using SendGrid's Python Library
2 # https://github.com/sendgrid/sendgrid-python
3 import os
4 from sendgrid import SendGridAPIClient
5 from sendgrid.helpers.mail import Mail
6
7 message = Mail(
8     from_email='ajisha@gmail.com',
9     to_emails='amiranjai@gmail.com',
10    subject='Sending with Twilio SendGrid is fun',
11    html_content='<strong>and easy to do anywhere, even with Python</strong>')
12
13 try:
14     sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
15     response = sg.send(message)
16     print(response.status_code)
17     print(response.body)
18     print(response.headers)
19 except Exception as e:
20     print(e.message)
```

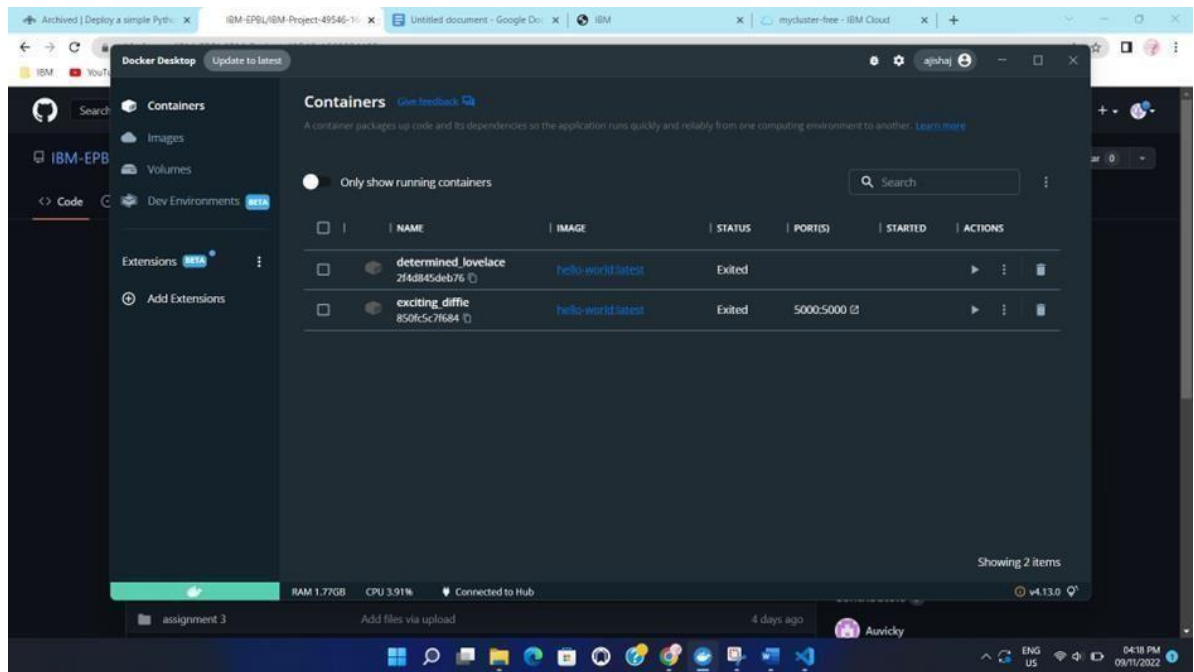
## 2. Containerize the application

The screenshot shows a Visual Studio Code editor window with a file explorer on the left. The file explorer shows a project named 'FLASK' with various files and folders. The main editor window displays a Dockerfile. The Dockerfile contains instructions to build a container image for a Flask application. The instructions include: FROM python:2.7, LABEL maintainer='Kunal Malhotra, kunal.malhotra@ibm.com', RUN apt-get update, RUN mkdir /app, WORKDIR /app, COPY . /app, RUN pip install -r requirements.txt, EXPOSE 5000, ENTRYPOINT [ "python" ], CMD [ "app.py" ].

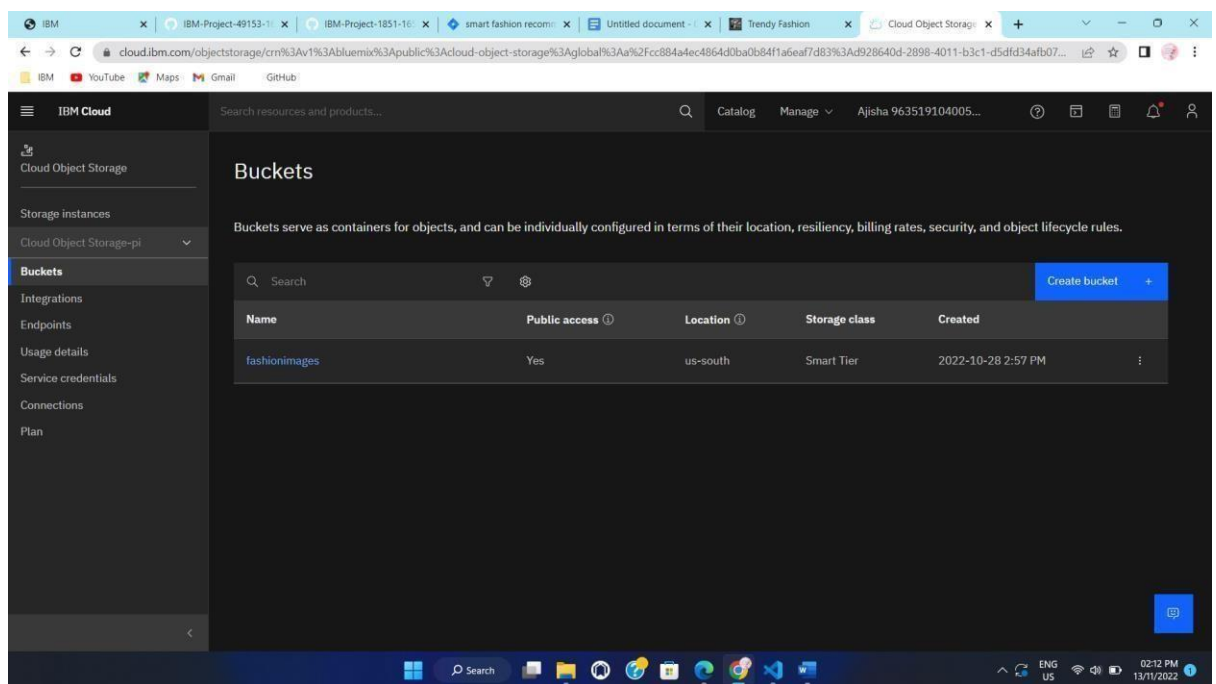
```
1 FROM python:2.7
2 LABEL maintainer="Kunal Malhotra, kunal.malhotra@ibm.com"
3 RUN apt-get update
4 RUN mkdir /app
5 WORKDIR /app
6 COPY . /app
7 RUN pip install -r requirements.txt
8 EXPOSE 5000
9 ENTRYPOINT [ "python" ]
10 CMD [ "app.py" ]
```

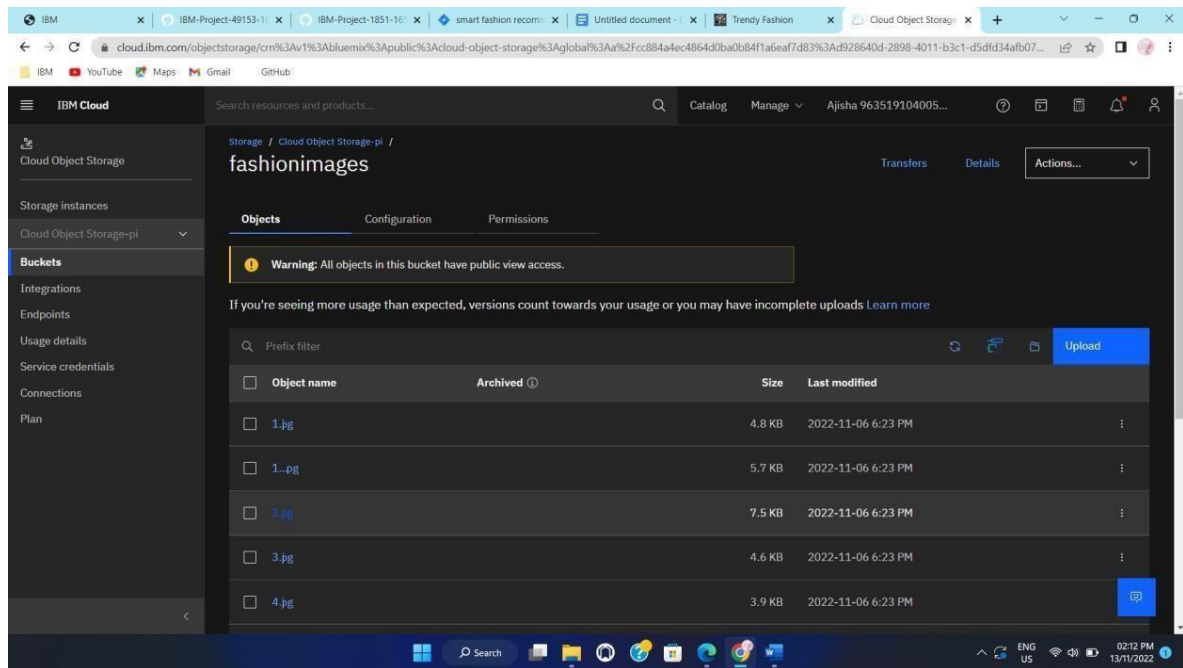
The bottom panel of the editor shows the 'TERMINAL' tab with a list of Docker command-line options and their descriptions:

- runtime string: Runtime to use for this container
- security-opt list: Security Options
- shm-size bytes: Size of /dev/shm
- sig-proxy: Proxy received signals to the process (default true)
- stop-signal string: Signal to stop a container (default "15")
- stop-timeout int: Timeout (in seconds) to stop a container
- storage-opt list: Storage driver options for the container
- sysctl map: Sysctl options (default map[])
- tmpfs list: Mount a tmpfs directory
- t, --tty: Allocate a pseudo-TTY
- ulimit ulimit: Ulimit options (default [])
- u, --user string: Username or UID (format: <name[:uid][:group[:gid]])
- userns string: User namespace to use
- uts string: UTS namespace to use
- v, --volume list: Bind mount a volume
- volume-driver string: Optional volume driver for the container
- volumes-from list: Mount volumes from the specified container(s)
- w, --workdir string: Working directory inside the container



### 3. Upload images to cloud





#### 4. Create responsive design for the application

