CSE- 322 **Cloud Computing**

NAME: ABHISHEK HARSH

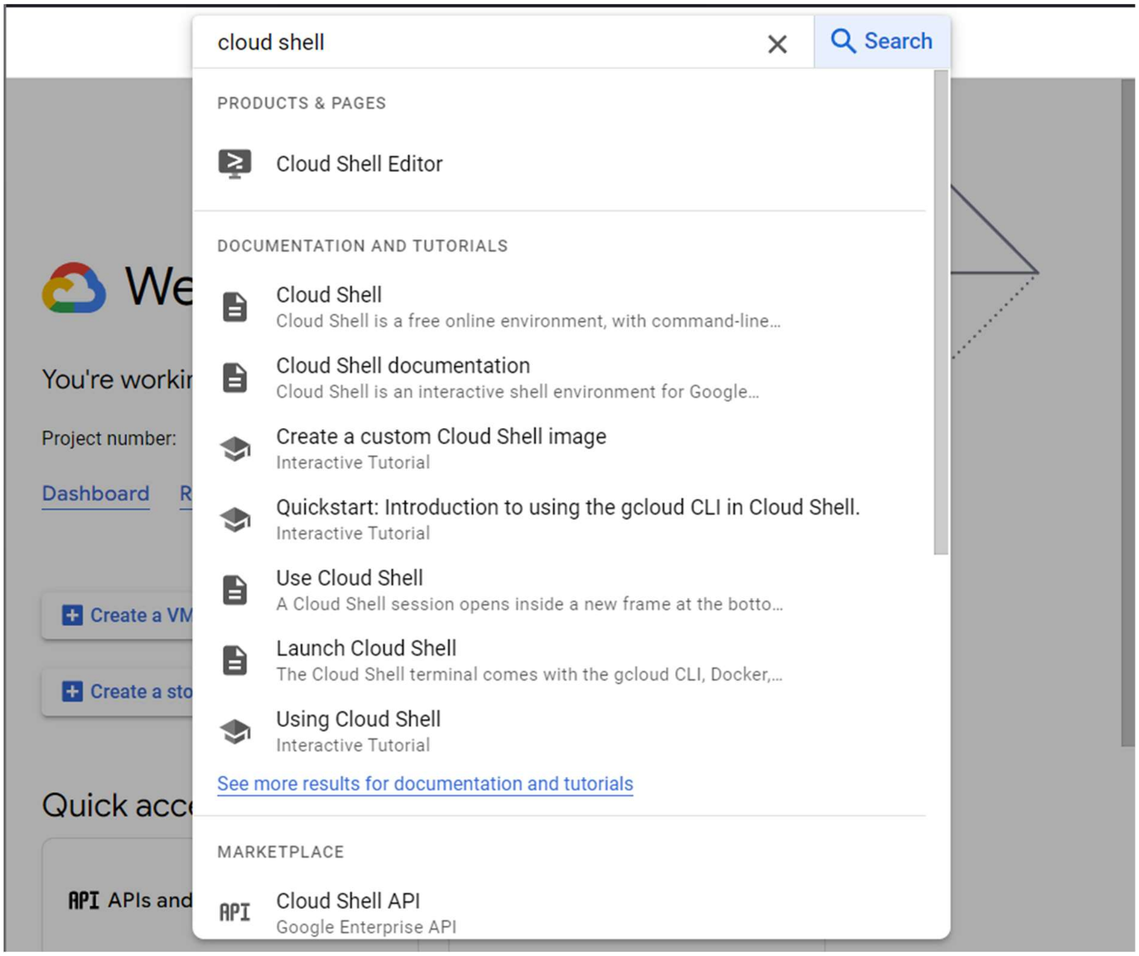
ROLL NO: 2021BCS0036

Topic: **Google Cloud Platform - Docker Container Implementation**

STEPS TO DO THIS:

STEP:1 1. Go to search bar and search for cloud shell

2. Then click Continue



PRODUCTS & PAGES

Cloud Shell Editor

DOCUMENTATION AND TUTORIALS

Cloud Shell
Cloud Shell is a free online environment, with command-line...

Cloud Shell documentation
Cloud Shell is an interactive shell environment for Google...

Create a custom Cloud Shell image
Interactive Tutorial

Quickstart: Introduction to using the gcloud CLI in Cloud Shell.
Interactive Tutorial

Use Cloud Shell
A Cloud Shell session opens inside a new frame at the botto...

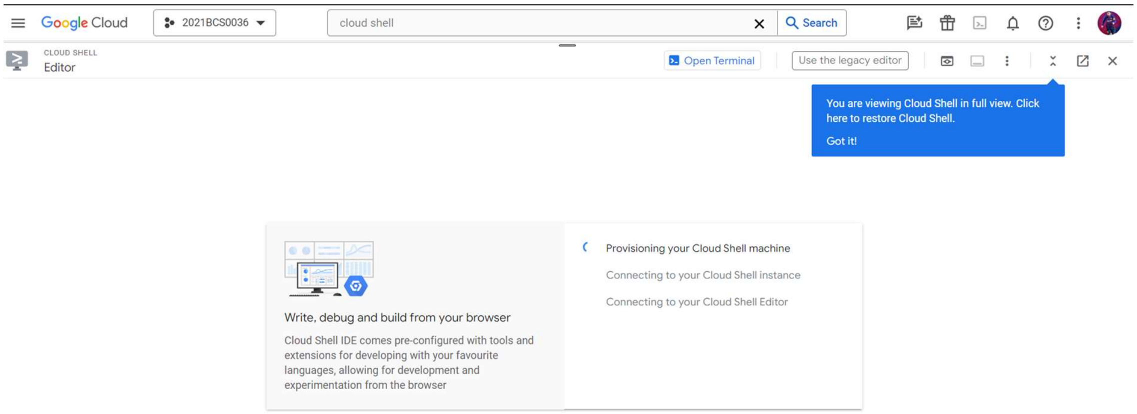
Launch Cloud Shell
The Cloud Shell terminal comes with the gcloud CLI, Docker,...

Using Cloud Shell
Interactive Tutorial

See more results for documentation and tutorials

MARKETPLACE

Cloud Shell API
Google Enterprise API



You are viewing Cloud Shell in full view. Click here to restore Cloud Shell.

Got it!

Write, debug and build from your browser

Cloud Shell IDE comes pre-configured with tools and extensions for developing with your favourite languages, allowing for development and experimentation from the browser

Provisioning your Cloud Shell machine

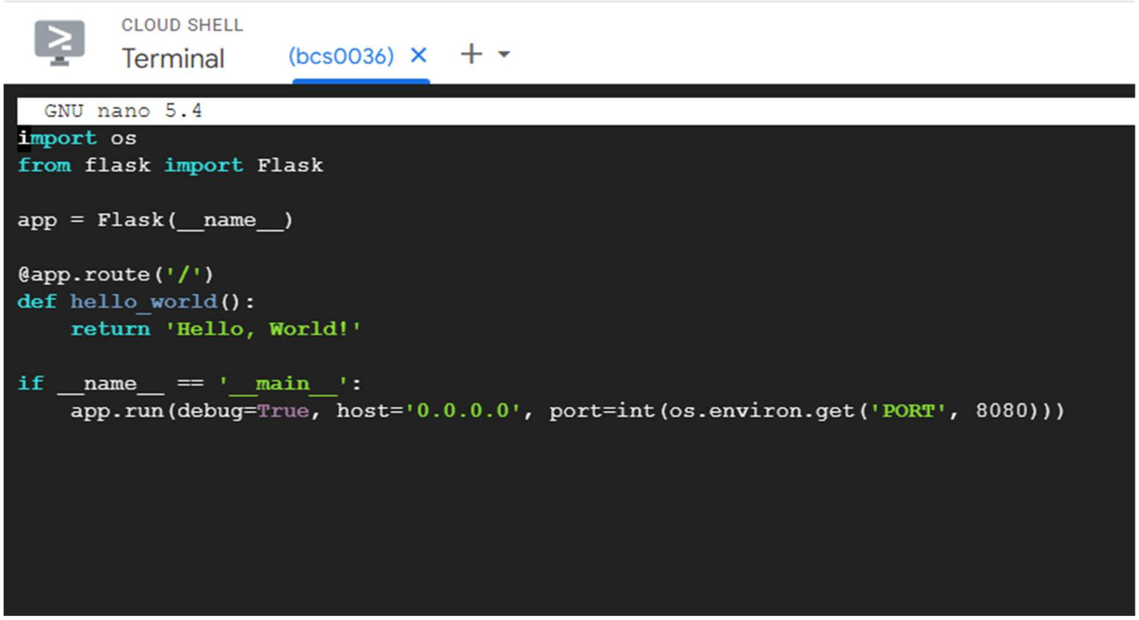
Connecting to your Cloud Shell instance

Connecting to your Cloud Shell Editor

STEP 2:

1. Make a new directory ,Go to that directory then code app

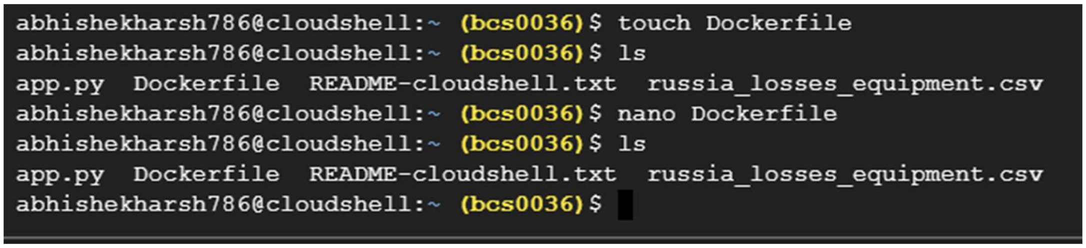
2. We will create a new file “app.py”.



```
  GNU nano 5.4
import os
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=int(os.environ.get('PORT', 8080)))
```

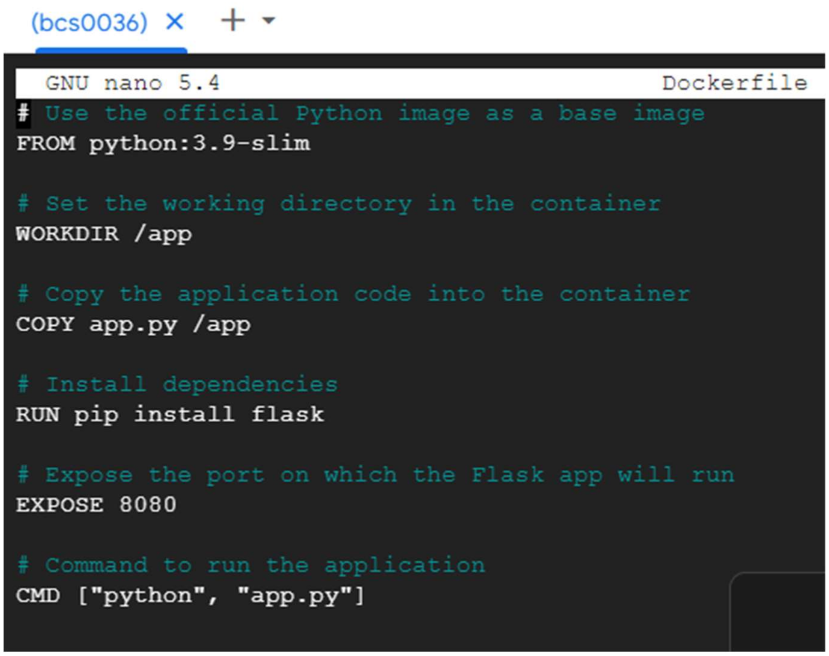
3 . Making Dockerfile

Create a Dockerfile in the same directory as your **app.py** with the following content:



```
abhishekharsh786@cloudshell:~ (bcs0036)$ touch Dockerfile
abhishekharsh786@cloudshell:~ (bcs0036)$ ls
app.py  Dockerfile  README-cloudshell.txt  russia_losses_equipment.csv
abhishekharsh786@cloudshell:~ (bcs0036)$ nano Dockerfile
abhishekharsh786@cloudshell:~ (bcs0036)$ ls
app.py  Dockerfile  README-cloudshell.txt  russia_losses_equipment.csv
abhishekharsh786@cloudshell:~ (bcs0036)$
```

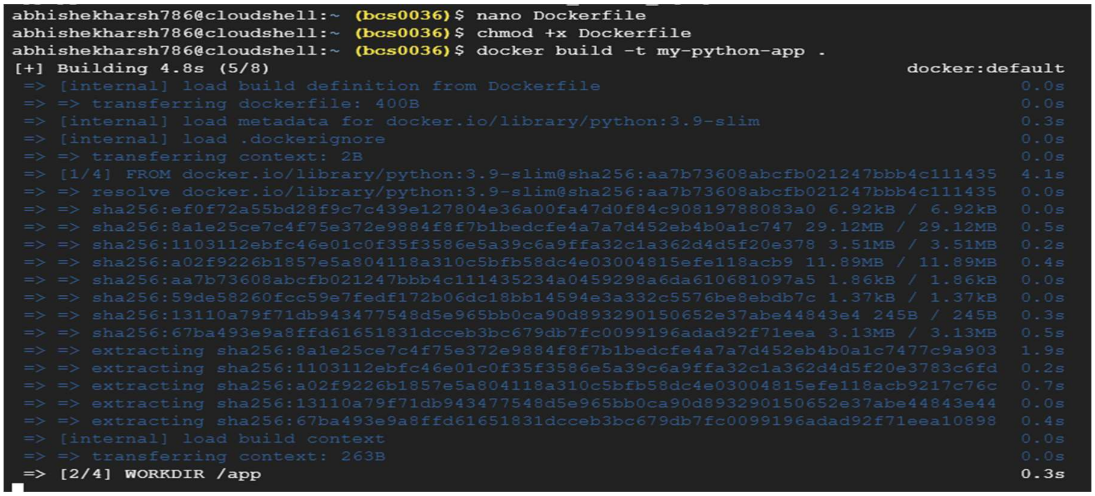
Contents of Dockerfile



```
  GNU nano 5.4                                  Dockerfile
# Use the official Python image as a base image
FROM python:3.9-slim

# Set the working directory in the container
WORKDIR /app

# Copy the application code into the container
COPY app.py /app

# Install dependencies
RUN pip install flask

# Expose the port on which the Flask app will run
EXPOSE 8080

# Command to run the application
CMD ["python", "app.py"]
```

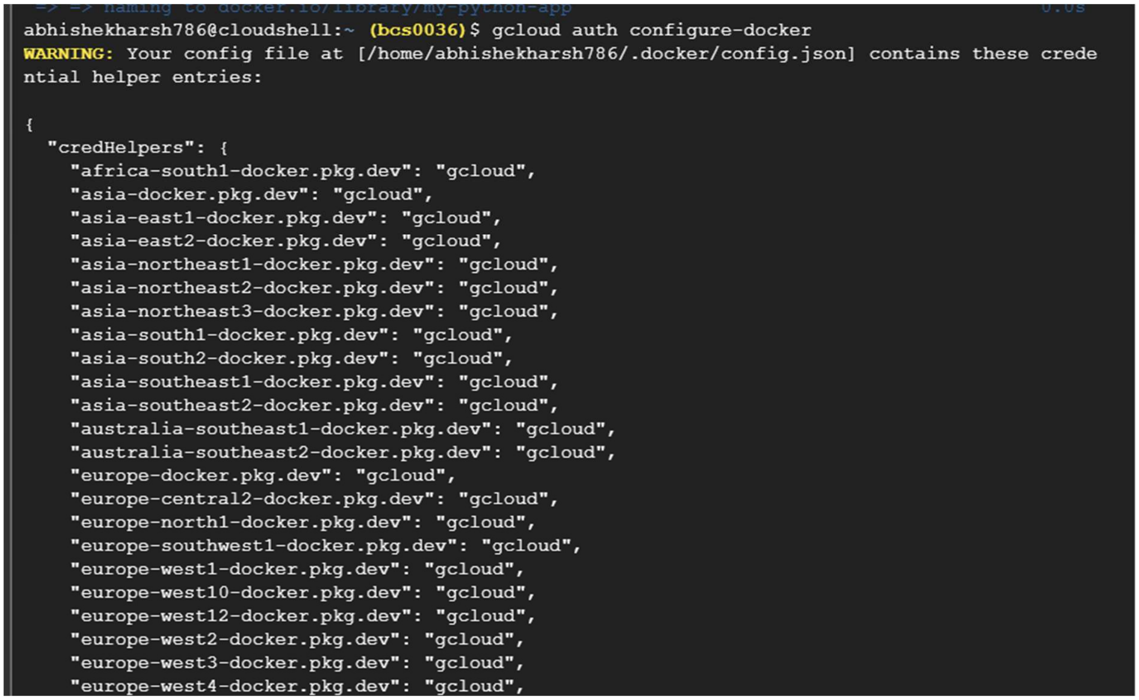
3. Building the container image

Open a terminal, navigate to the directory containing your Dockerfile and **app.py**, and run the following command to build your Docker image:

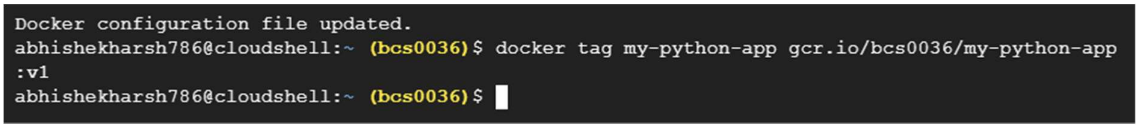


```
abhishekharsh786@cloudshell:~ (bcs0036)$ nano Dockerfile
abhishekharsh786@cloudshell:~ (bcs0036)$ chmod +x Dockerfile
abhishekharsh786@cloudshell:~ (bcs0036)$ docker build -t my-python-app .
[+] Building 4.8s (5/8)                                                    docker:default
 => [internal] load build definition from Dockerfile                                 0.0s
 => => transferring dockerfile: 400B                                                 0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim                   0.3s
 => [internal] load .dockerignore                                                    0.0s
 => => transferring context: 2B                                                      0.0s
 => [1/4] FROM docker.io/library/python:3.9-slim@sha256:aa7b73608abcfb021247bbb4c111435  4.1s
 => => resolve docker.io/library/python:3.9-slim@sha256:aa7b73608abcfb021247bbb4c111435  0.0s
 => => sha256:ef0f72a55bd28f9c7c439e127804e36a00fa47d0f84c90819788083a0 6.92kB / 6.92kB  0.0s
 => => sha256:8a1e25ce7c4f75e372e9884f8f7b1bedcfe4a7a7d452eb4b0a1c747 29.12MB / 29.12MB  0.5s
 => => sha256:1103112ebfc46e01c0f35f3586e5a39c6a9ffa32c1a362d4d5f20e378 3.51MB / 3.51MB  0.2s
 => => sha256:a02f9226b1857e5a804118a310c5bfb58dc4e03004815efe118acb9 11.89MB / 11.89MB  0.4s
 => => sha256:aa7b73608abcfb021247bbb4c111435234a0459298a6da610681097a5 1.86kB / 1.86kB  0.0s
 => => sha256:59de58260fcc59e7fedf172b06dc18bb14594e3a332c5576be8ebdb7c 1.37kB / 1.37kB  0.0s
 => => sha256:13110a79f71db943477548d5e965bb0ca90d893290150652e37abe44843e4 245B / 245B  0.3s
 => => sha256:67ba493e9a8ffd61651831dcceb3bc679db7fc0099196adad92f71eea 3.13MB / 3.13MB  0.5s
 => => extracting sha256:8a1e25ce7c4f75e372e9884f8f7b1bedcfe4a7a7d452eb4b0a1c7477c9a903  1.9s
 => => extracting sha256:1103112ebfc46e01c0f35f3586e5a39c6a9ffa32c1a362d4d5f20e3783c6fd  0.2s
 => => extracting sha256:a02f9226b1857e5a804118a310c5bfb58dc4e03004815efe118acb9217c76c  0.7s
 => => extracting sha256:13110a79f71db943477548d5e965bb0ca90d893290150652e37abe44843e44  0.0s
 => => extracting sha256:67ba493e9a8ffd61651831dcceb3bc679db7fc0099196adad92f71eea10898  0.4s
 => [internal] load build context                                                    0.0s
 => => transferring context: 263B                                                    0.0s
 => [2/4] WORKDIR /app                                                               0.3s
```

4. Pushing the image to container registry

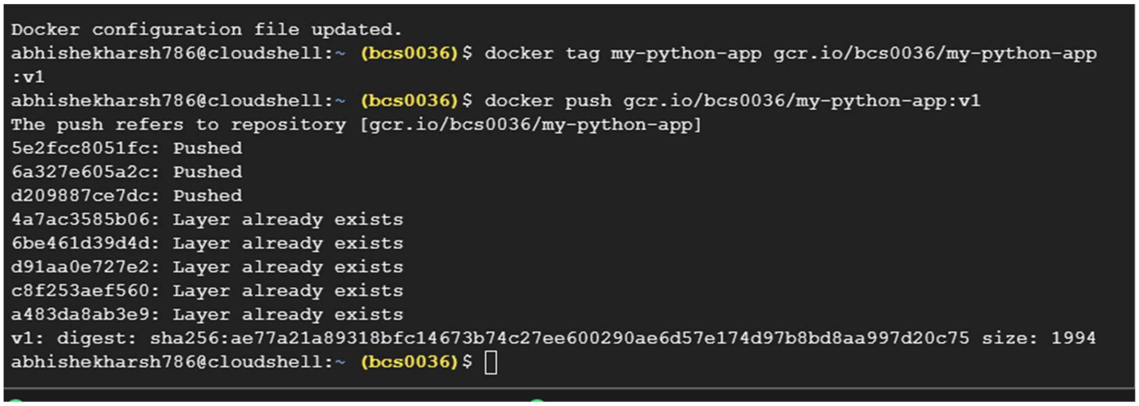


```
 => => naming to docker.io/library/my-python-app                              0.0s
abhishekharsh786@cloudshell:~ (bcs0036)$ gcloud auth configure-docker
WARNING: Your config file at [/home/abhishekharsh786/.docker/config.json] contains these crede
ntial helper entries:

{
  "credHelpers": {
    "africa-south1-docker.pkg.dev": "gcloud",
    "asia-docker.pkg.dev": "gcloud",
    "asia-east1-docker.pkg.dev": "gcloud",
    "asia-east2-docker.pkg.dev": "gcloud",
    "asia-northeast1-docker.pkg.dev": "gcloud",
    "asia-northeast2-docker.pkg.dev": "gcloud",
    "asia-northeast3-docker.pkg.dev": "gcloud",
    "asia-south1-docker.pkg.dev": "gcloud",
    "asia-south2-docker.pkg.dev": "gcloud",
    "asia-southeast1-docker.pkg.dev": "gcloud",
    "asia-southeast2-docker.pkg.dev": "gcloud",
    "australia-southeast1-docker.pkg.dev": "gcloud",
    "australia-southeast2-docker.pkg.dev": "gcloud",
    "europe-docker.pkg.dev": "gcloud",
    "europe-central2-docker.pkg.dev": "gcloud",
    "europe-north1-docker.pkg.dev": "gcloud",
    "europe-southwest1-docker.pkg.dev": "gcloud",
    "europe-west1-docker.pkg.dev": "gcloud",
    "europe-west10-docker.pkg.dev": "gcloud",
    "europe-west12-docker.pkg.dev": "gcloud",
    "europe-west2-docker.pkg.dev": "gcloud",
    "europe-west3-docker.pkg.dev": "gcloud",
    "europe-west4-docker.pkg.dev": "gcloud",
```



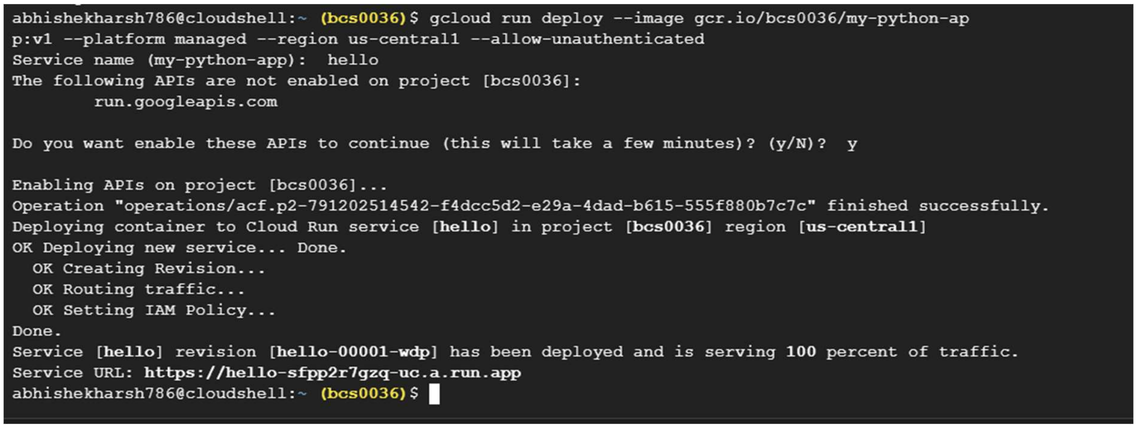
```
Docker configuration file updated.
abhishekharsh786@cloudshell:~ (bcs0036)$ docker tag my-python-app gcr.io/bcs0036/my-python-app
:v1
abhishekharsh786@cloudshell:~ (bcs0036)$ █
```

Tagging our Docker image with the registry path:



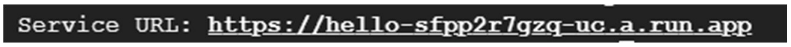
```
Docker configuration file updated.
abhishekharsh786@cloudshell:~ (bcs0036)$ docker tag my-python-app gcr.io/bcs0036/my-python-app
:v1
abhishekharsh786@cloudshell:~ (bcs0036)$ docker push gcr.io/bcs0036/my-python-app:v1
The push refers to repository [gcr.io/bcs0036/my-python-app]
5e2fcc8051fc: Pushed
6a327e605a2c: Pushed
d209887ce7dc: Pushed
4a7ac3585b06: Layer already exists
6be461d39d4d: Layer already exists
d91aa0e727e2: Layer already exists
c8f253aef560: Layer already exists
a483da8ab3e9: Layer already exists
v1: digest: sha256:ae77a21a89318bfc14673b74c27ee600290ae6d57e174d97b8bd8aa997d20c75 size: 1994
abhishekharsh786@cloudshell:~ (bcs0036)$ 
```

Last step is running and deploying the application



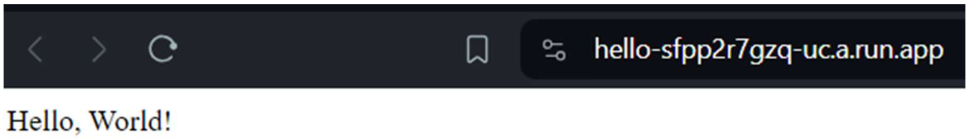
```
abhishekharsh786@cloudshell:~ (bcs0036)$ gcloud run deploy --image gcr.io/bcs0036/my-python-ap
p:v1 --platform managed --region us-central1 --allow-unauthenticated
Service name (my-python-app):  hello
The following APIs are not enabled on project [bcs0036]:
        run.googleapis.com

Do you want enable these APIs to continue (this will take a few minutes)? (y/N)?  y

Enabling APIs on project [bcs0036]...
Operation "operations/acf.p2-791202514542-f4dcc5d2-e29a-4dad-b615-555f880b7c7c" finished successfully.
Deploying container to Cloud Run service [hello] in project [bcs0036] region [us-central1]
OK Deploying new service... Done.
  OK Creating Revision...
  OK Routing traffic...
  OK Setting IAM Policy...
Done.
Service [hello] revision [hello-00001-wdp] has been deployed and is serving 100 percent of traffic.
Service URL: https://hello-sfpp2r7gzq-uc.a.run.app
abhishekharsh786@cloudshell:~ (bcs0036)$ 
```

6 . Deployment done Our app is working fine



```
Service URL: https://hello-sfpp2r7gzq-uc.a.run.app
```

We can use the above url to access our app



Hello, World!