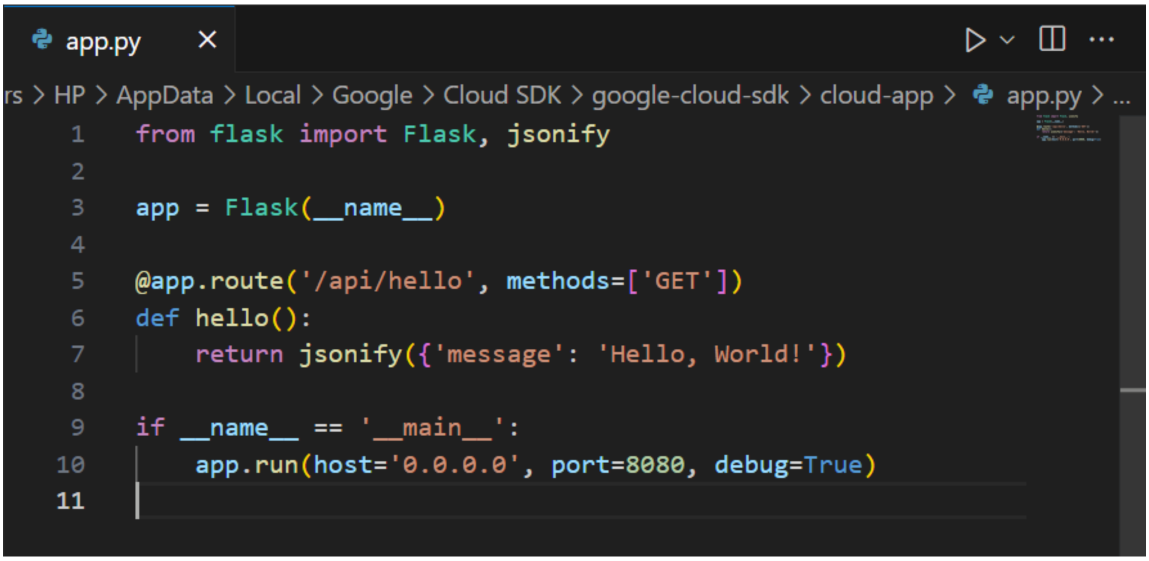
Cloud application development. Assignment 3 report

Managing APIs with google cloud endpoints

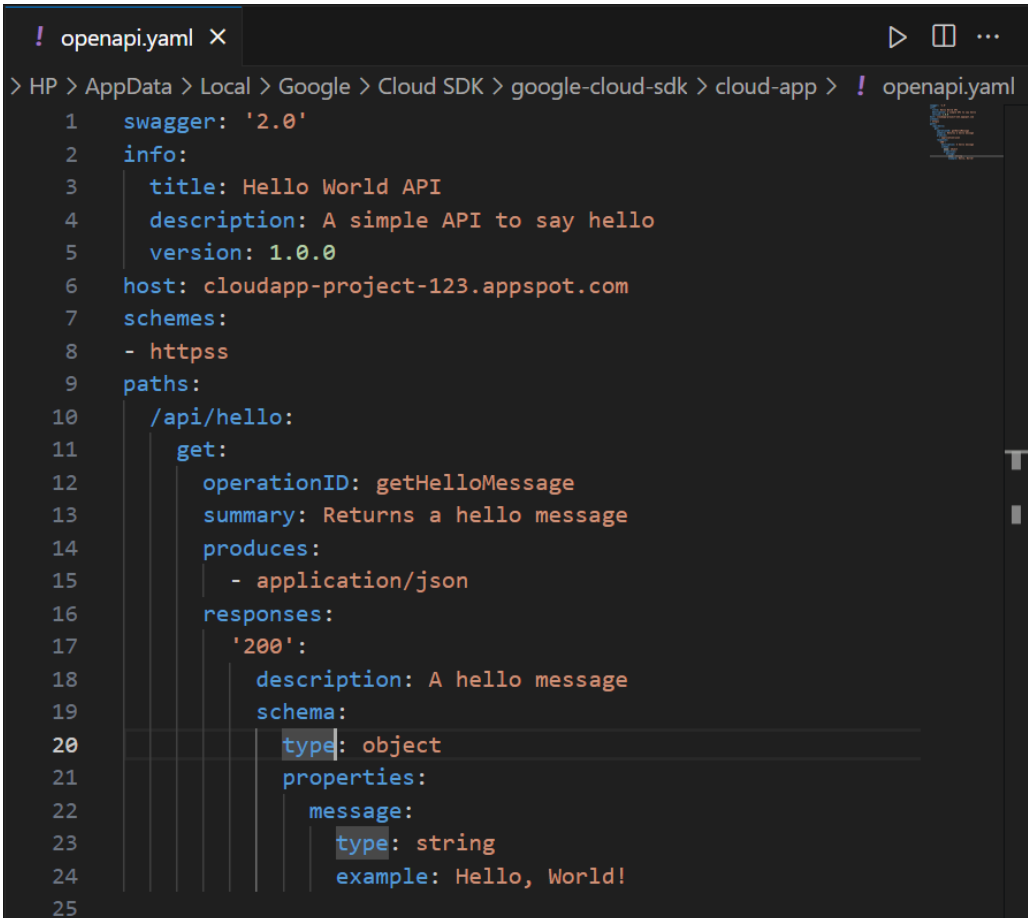
Deploy and manage an API using google cloud endpoints

app.py – file defines how the server will handle http requests.


app.py
rs > HP > AppData > Local > Google > Cloud SDK > google-cloud-sdk > cloud-app > app.py > ...

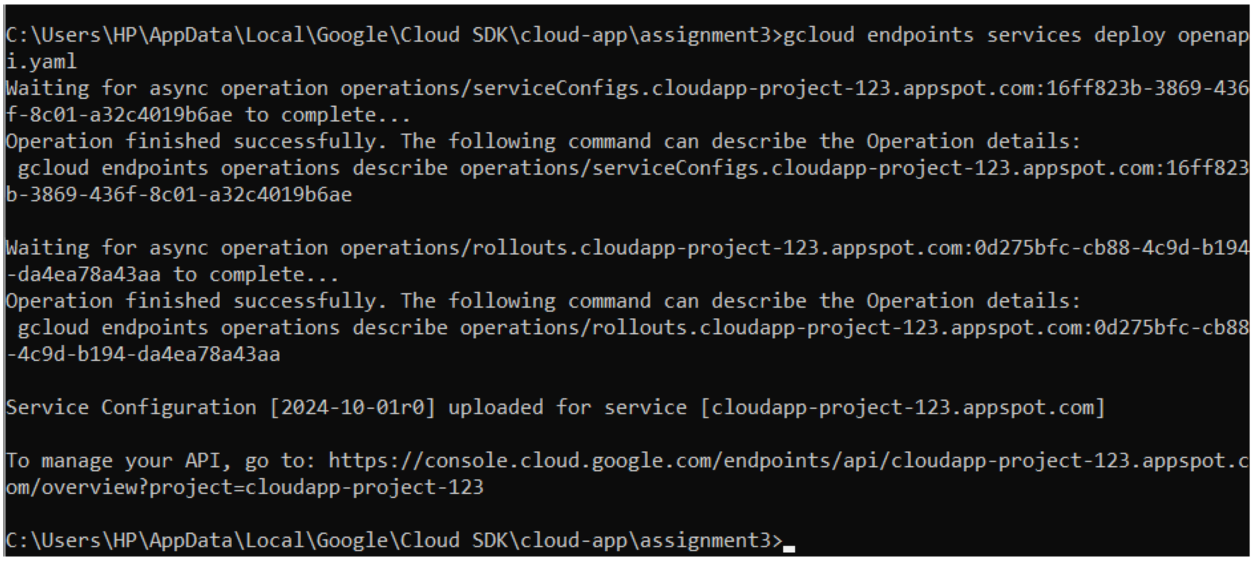
```python
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/api/hello', methods=['GET'])
def hello():
    return jsonify({'message': 'Hello, World!'})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, debug=True)

```

openapi.yaml – file describes the API for management through google cloud endpoints. Changes the API specification from openapi to swagger. Add host, delete content and description because they don't comply with the swagger specification. Added a unique identifier for the API (operationID: getHelloMessage).

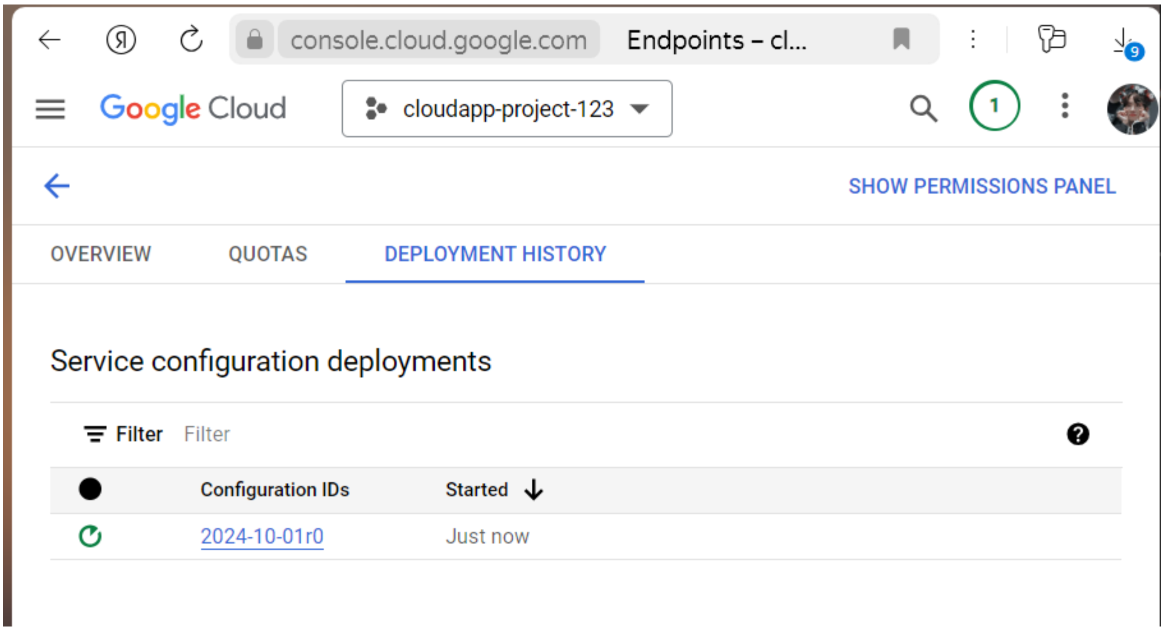

openapi.yaml
> HP > AppData > Local > Google > Cloud SDK > google-cloud-sdk > cloud-app > openapi.yaml

```yaml
swagger: '2.0'
info:
  title: Hello World API
  description: A simple API to say hello
  version: 1.0.0
host: cloudapp-project-123.appspot.com
schemes:
- httpss
paths:
  /api/hello:
    get:
      operationID: getHelloMessage
      summary: Returns a hello message
      produces:
        - application/json
      responses:
        '200':
          description: A hello message
          schema:
            type: object
            properties:
              message:
                type: string
                example: Hello, World!

```

gcloud endpoints services deploy openapi.yaml – deploy the API configuration to google cloud endpoints.



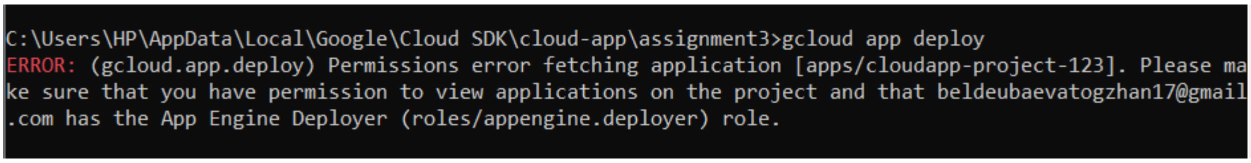
```
C:\Users\HP\AppData\Local\Google\Cloud SDK\cloud-app\assignment3>gcloud endpoints services deploy openap
i.yaml
Waiting for async operation operations/serviceConfigs.cloudapp-project-123.appspot.com:16ff823b-3869-436
f-8c01-a32c4019b6ae to complete...
Operation finished successfully. The following command can describe the Operation details:
 gcloud endpoints operations describe operations/serviceConfigs.cloudapp-project-123.appspot.com:16ff823
b-3869-436f-8c01-a32c4019b6ae

Waiting for async operation operations/rollouts.cloudapp-project-123.appspot.com:0d275bfc-cb88-4c9d-b194
-da4ea78a43aa to complete...
Operation finished successfully. The following command can describe the Operation details:
 gcloud endpoints operations describe operations/rollouts.cloudapp-project-123.appspot.com:0d275bfc-cb88
-4c9d-b194-da4ea78a43aa

Service Configuration [2024-10-01r0] uploaded for service [cloudapp-project-123.appspot.com]

To manage your API, go to: https://console.cloud.google.com/endpoints/api/cloudapp-project-123.appspot.c
om/overview?project=cloudapp-project-123

C:\Users\HP\AppData\Local\Google\Cloud SDK\cloud-app\assignment3>
```



Service configuration deployments

●	Configuration IDs	Started ↓
✓	2024-10-01r0	Just now

gcloud app deploy – deploy the application to google app engine. Need a billing account



```
C:\Users\HP\AppData\Local\Google\Cloud SDK\cloud-app\assignment3>gcloud app deploy
ERROR: (gcloud.app.deploy) Permissions error fetching application [apps/cloudapp-project-123]. Please ma
ke sure that you have permission to view applications on the project and that beldeubaevatogzhan17@gmail
.com has the App Engine Deployer (roles/appengine.deployer) role.
```

To test curl http://

JSON

{

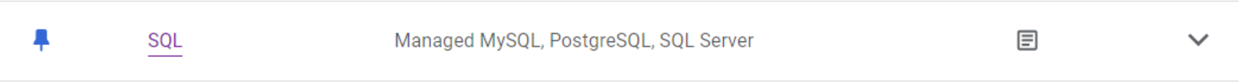
“message”: “Hello. World!”

}

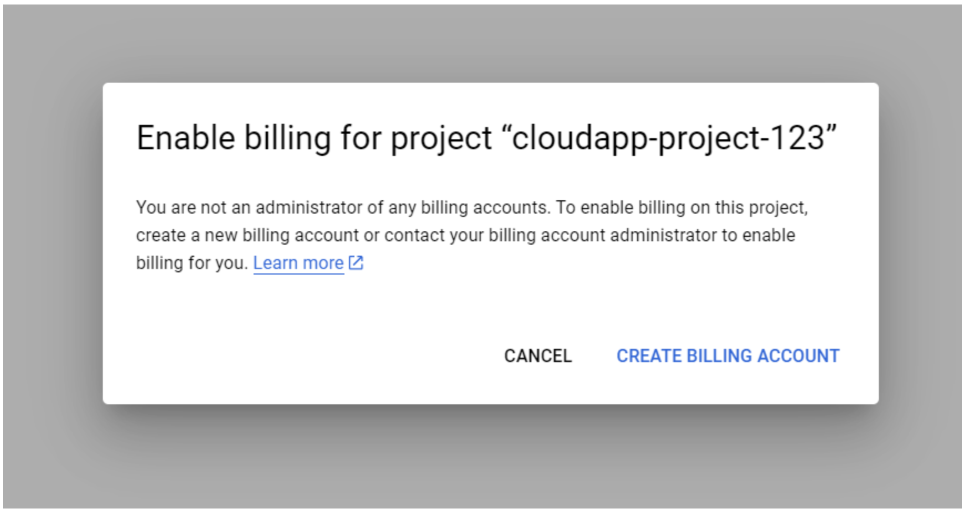
Google cloud databases

Set up and interact with a google cloud sql database

To set up google cloud sql, a billing account is required.



SQL   Managed MySQL, PostgreSQL, SQL Server

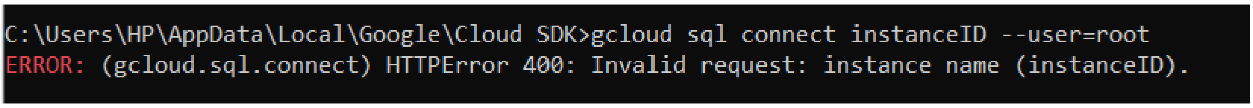


Enable billing for project "cloudapp-project-123"

You are not an administrator of any billing accounts. To enable billing on this project, create a new billing account or contact your billing account administrator to enable billing for you. Learn more

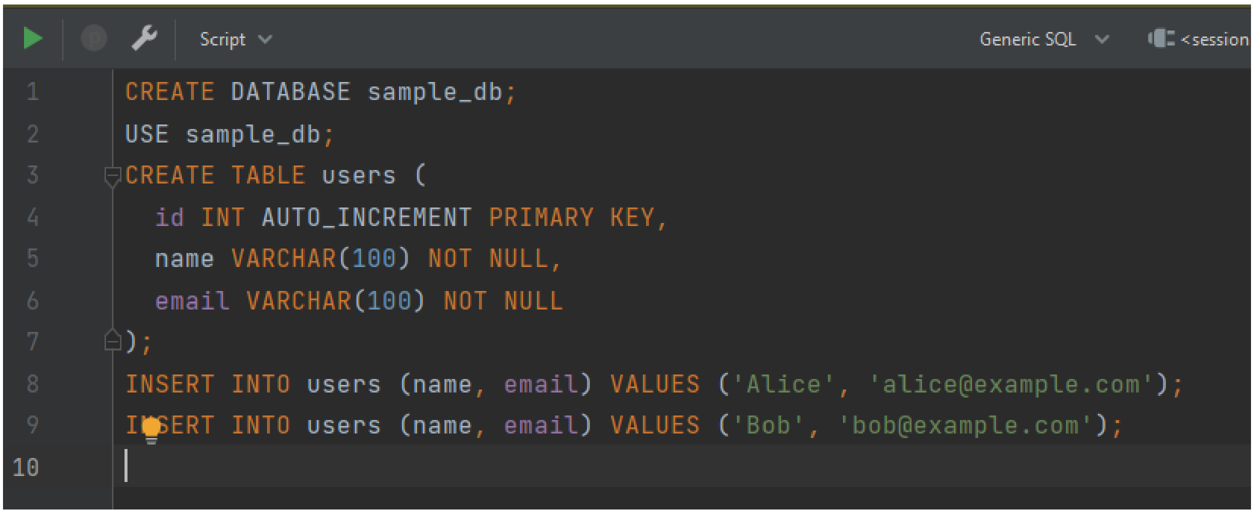
If work: create instance, choose a database type and write configuration settings (instanceID, password, …)

gcloud sql connect [instanceID] -user=root – connects to a google cloud sql instance using postgresql



```
C:\Users\HP\AppData\Local\Google\Cloud SDK>gcloud sql connect instanceID --user=root
ERROR: (gcloud.sql.connect) HTTPError 400: Invalid request: instance name (instanceID).
```

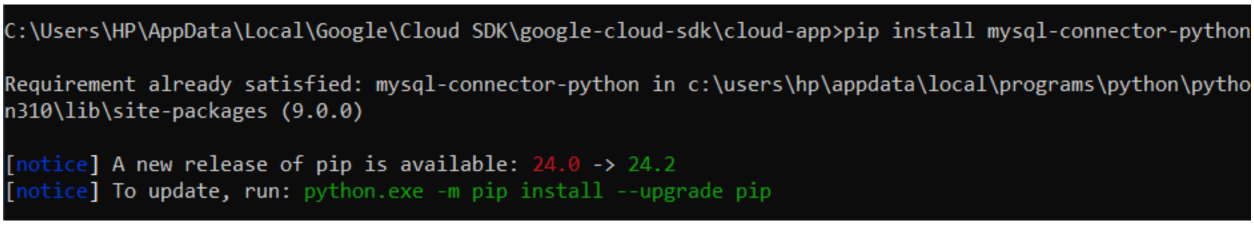
database.sql – file creates a database and a table



```
CREATE DATABASE sample_db;
USE sample_db;
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  email VARCHAR(100) NOT NULL
);
INSERT INTO users (name, email) VALUES ('Alice', 'alice@example.com');
INSERT INTO users (name, email) VALUES ('Bob', 'bob@example.com');
```

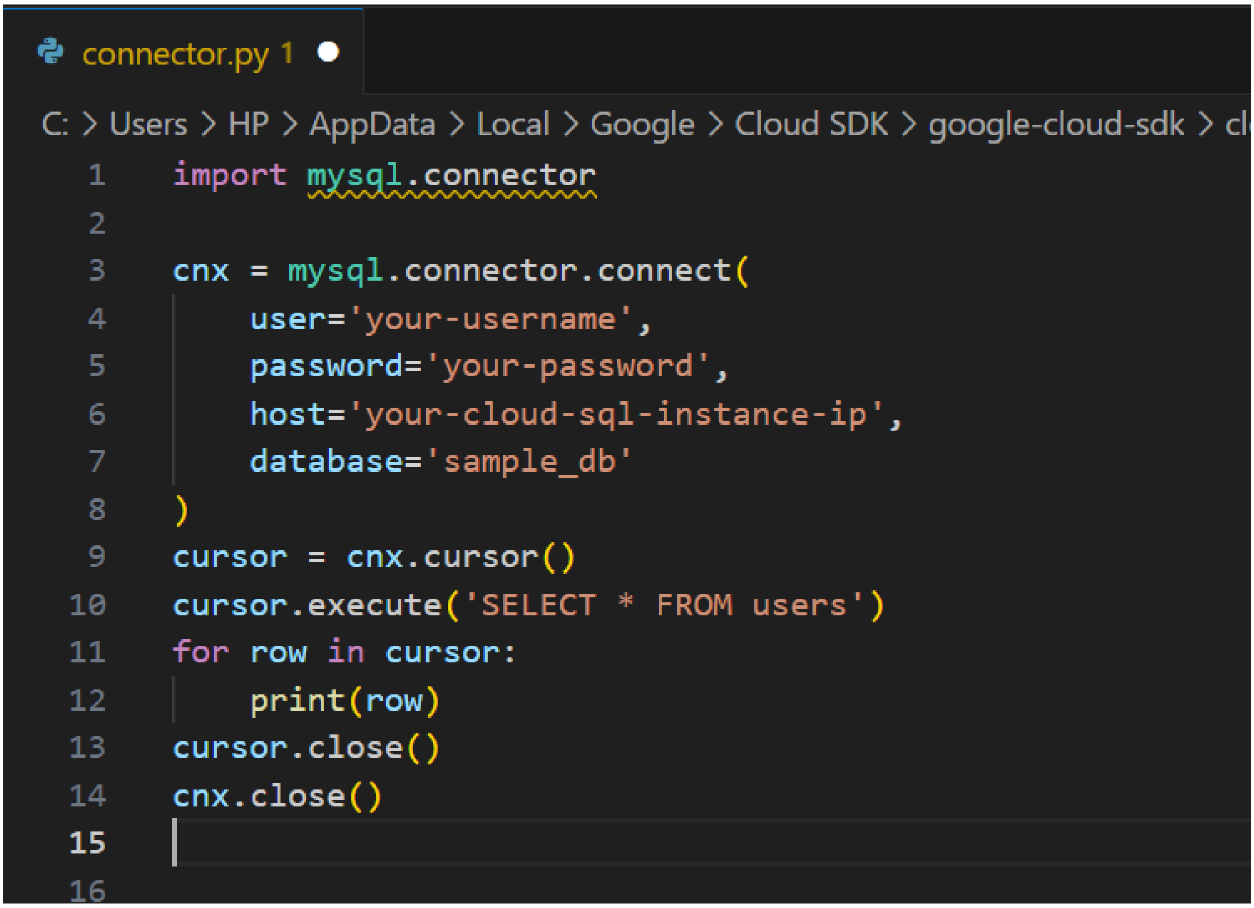
source database.sql – executes sql queries

Install mysql connector for python



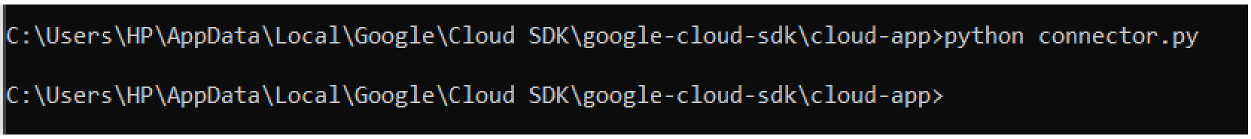
```
C:\Users\HP\AppData\Local\Google\Cloud SDK\google-cloud-sdk\cloud-app>pip install mysql-connector-python

Requirement already satisfied: mysql-connector-python in c:\users\hp\appdata\local\programs\python\pytho
n310\lib\site-packages (9.0.0)

[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

connect.py – establishes a database connection in google cloud sql and data interaction.


connector.py 1 ●
C: > Users > HP > AppData > Local > Google > Cloud SDK > google-cloud-sdk > cl

```python
import mysql.connector

cnx = mysql.connector.connect(
    user='your-username',
    password='your-password',
    host='your-cloud-sql-instance-ip',
    database='sample_db'
)
cursor = cnx.cursor()
cursor.execute('SELECT * FROM users')
for row in cursor:
    print(row)
cursor.close()
cnx.close()
```

Launch via

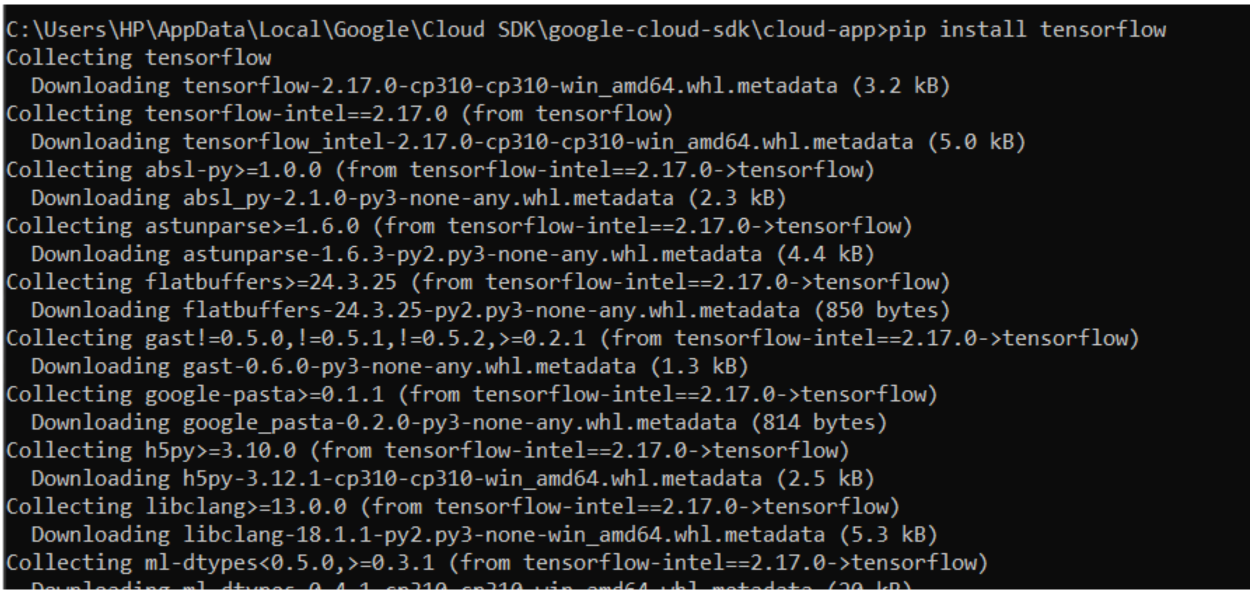


```
C:\Users\HP\AppData\Local\Google\Cloud SDK\google-cloud-sdk\cloud-app>python connector.py

C:\Users\HP\AppData\Local\Google\Cloud SDK\google-cloud-sdk\cloud-app>
```

Integrating machine learning with google cloud

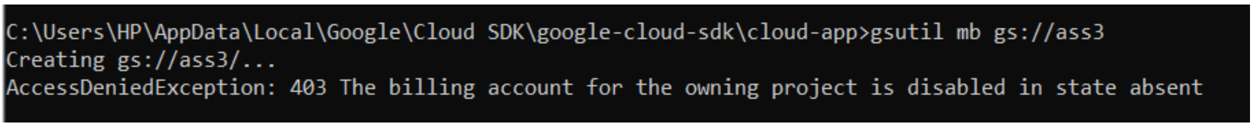
Train and deploy a machine learning model using google cloud ai platform

Installed tensorflow (machine learning library developed by google)



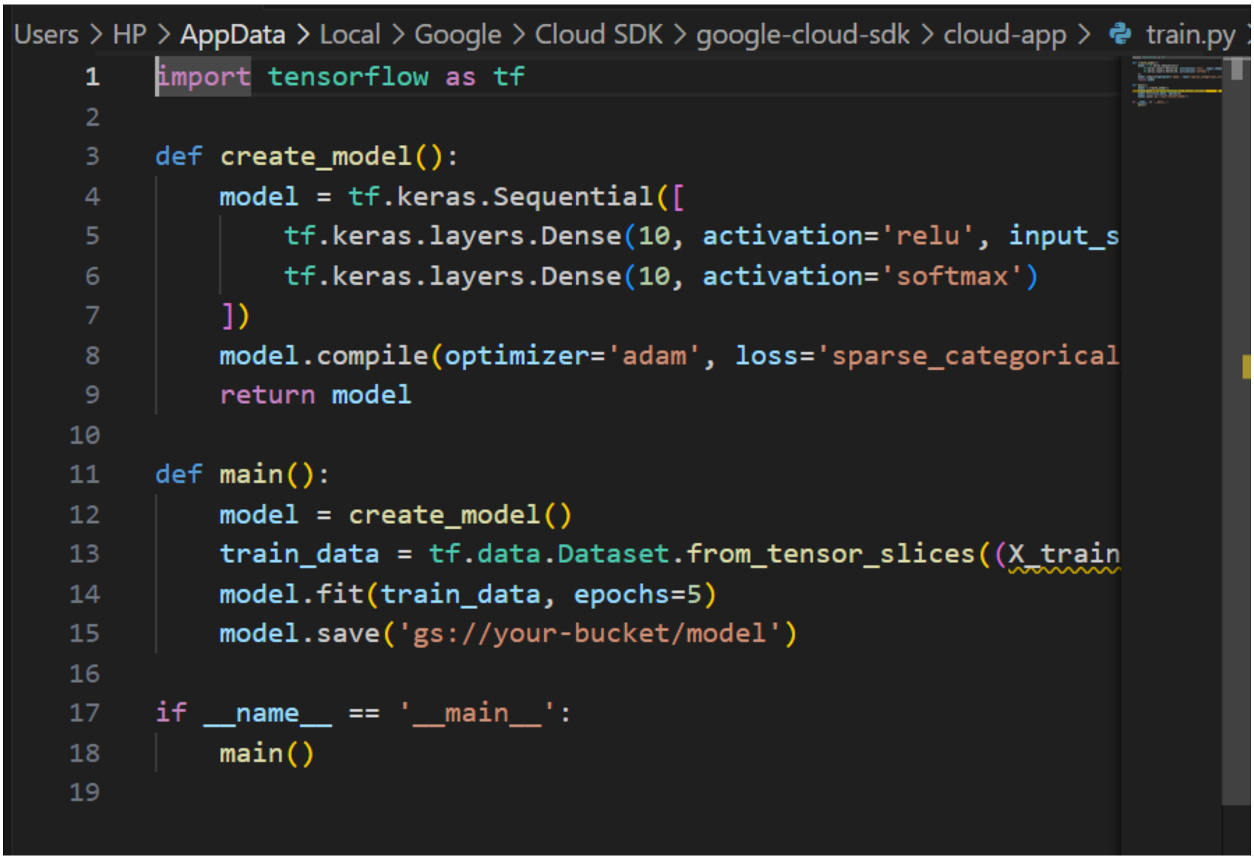
```
C:\Users\HP\AppData\Local\Google\Cloud SDK\google-cloud-sdk\cloud-app>pip install tensorflow
Collecting tensorflow
  Downloading tensorflow-2.17.0-cp310-cp310-win_amd64.whl.metadata (3.2 kB)
Collecting tensorflow-intel==2.17.0 (from tensorflow)
  Downloading tensorflow_intel-2.17.0-cp310-cp310-win_amd64.whl.metadata (5.0 kB)
Collecting absl-py>=1.0.0 (from tensorflow-intel==2.17.0->tensorflow)
  Downloading absl_py-2.1.0-py3-none-any.whl.metadata (2.3 kB)
Collecting astunparse>=1.6.0 (from tensorflow-intel==2.17.0->tensorflow)
  Downloading astunparse-1.6.3-py2.py3-none-any.whl.metadata (4.4 kB)
Collecting flatbuffers>=24.3.25 (from tensorflow-intel==2.17.0->tensorflow)
  Downloading flatbuffers-24.3.25-py2.py3-none-any.whl.metadata (850 bytes)
Collecting gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 (from tensorflow-intel==2.17.0->tensorflow)
  Downloading gast-0.6.0-py3-none-any.whl.metadata (1.3 kB)
Collecting google-pasta>=0.1.1 (from tensorflow-intel==2.17.0->tensorflow)
  Downloading google_pasta-0.2.0-py3-none-any.whl.metadata (814 bytes)
Collecting h5py>=3.10.0 (from tensorflow-intel==2.17.0->tensorflow)
  Downloading h5py-3.12.1-cp310-cp310-win_amd64.whl.metadata (2.5 kB)
Collecting libclang>=13.0.0 (from tensorflow-intel==2.17.0->tensorflow)
  Downloading libclang-18.1.1-py2.py3-none-win_amd64.whl.metadata (5.3 kB)
Collecting ml-dtypes<0.5.0,>=0.3.1 (from tensorflow-intel==2.17.0->tensorflow)
  Downloading ml_dtypes-0.4.1-cp310-cp310-win_amd64.whl.metadata (20 kB)
```

Created a bucket



```
C:\Users\HP\AppData\Local\Google\Cloud SDK\google-cloud-sdk\cloud-app>gsutil mb gs://ass3
Creating gs://ass3/...
AccessDeniedException: 403 The billing account for the owning project is disabled in state absent
```

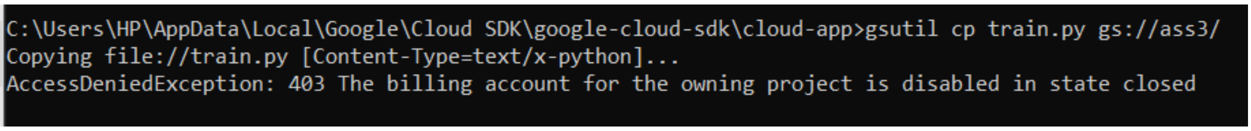
train.py – used to create a machine learning model and is saved in google cloud storage for further deployment on google cloud ai platform.



Users > HP > AppData > Local > Google > Cloud SDK > google-cloud-sdk > cloud-app > train.py >

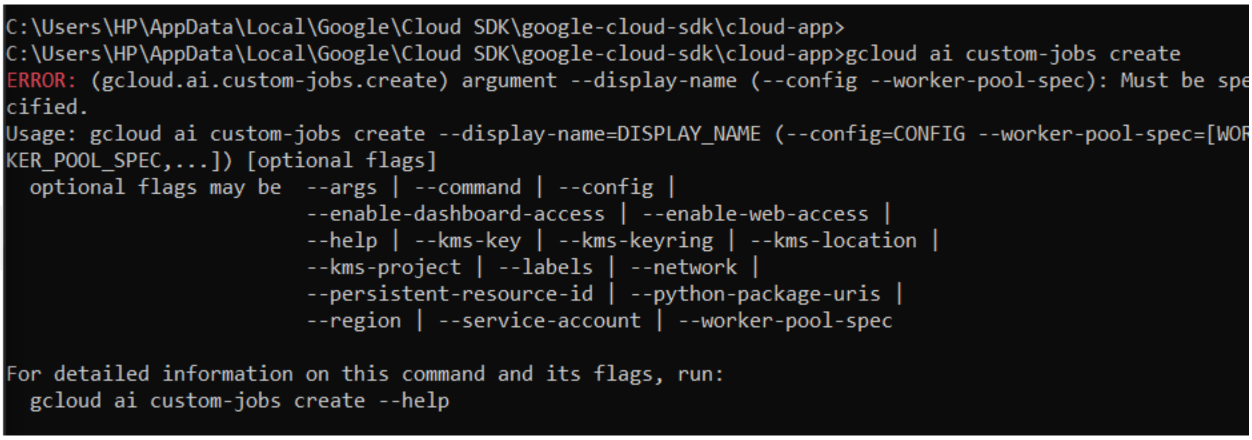
```python
import tensorflow as tf

def create_model():
    model = tf.keras.Sequential([
        tf.keras.layers.Dense(10, activation='relu', input_s
        tf.keras.layers.Dense(10, activation='softmax')
    ])
    model.compile(optimizer='adam', loss='sparse_categorical
    return model

def main():
    model = create_model()
    train_data = tf.data.Dataset.from_tensor_slices((X_train
    model.fit(train_data, epochs=5)
    model.save('gs://your-bucket/model')

if __name__ == '__main__':
    main()

```

gsutil cp train.py gs://ass3/ - uploads the file to the bucket.



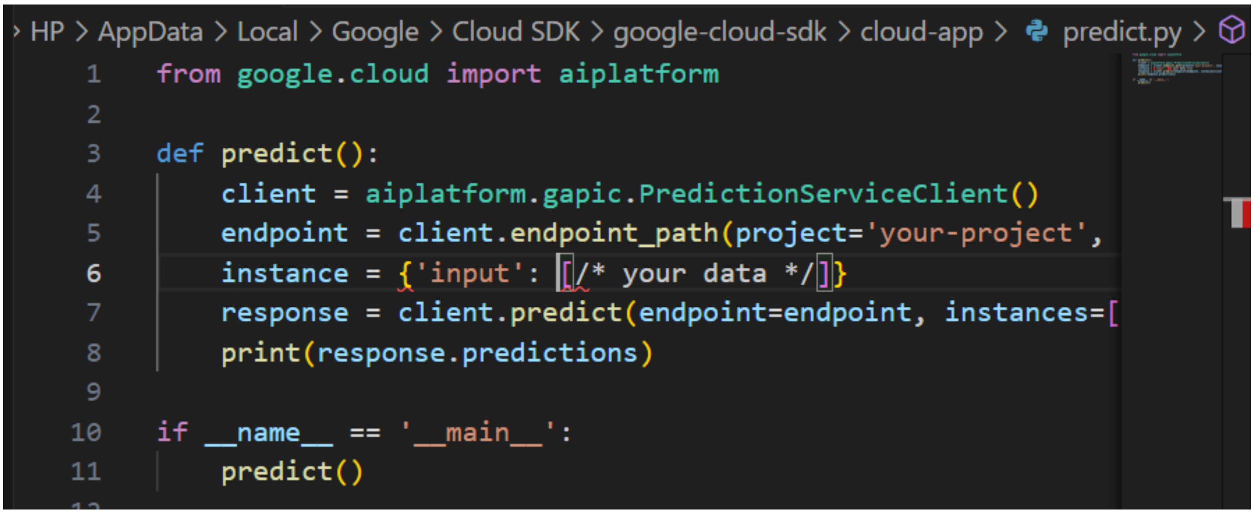
```
C:\Users\HP\AppData\Local\Google\Cloud SDK\google-cloud-sdk\cloud-app>gsutil cp train.py gs://ass3/
Copying file://train.py [Content-Type=text/x-python]...
AccessDeniedException: 403 The billing account for the owning project is disabled in state closed
```

gcloud ai custom-jobs create --region=your-region --display-name=ml-job --python-package-uris=gs://your-bucket/train.py --python-module=train --container-image-uri=gcr.io/cloud-aiplatform/training/tf-cpu.2-4:latest – deploy the model in the google cloud ai platform



```
C:\Users\HP\AppData\Local\Google\Cloud SDK\google-cloud-sdk\cloud-app>
C:\Users\HP\AppData\Local\Google\Cloud SDK\google-cloud-sdk\cloud-app>gcloud ai custom-jobs create
ERROR: (gcloud.ai.custom-jobs.create) argument --display-name (--config --worker-pool-spec): Must be spe
cified.
Usage: gcloud ai custom-jobs create --display-name=DISPLAY_NAME (--config=CONFIG --worker-pool-spec=[WOR
KER_POOL_SPEC,...]) [optional flags]
  optional flags may be  --args | --command | --config |
                         --enable-dashboard-access | --enable-web-access |
                         --help | --kms-key | --kms-keyring | --kms-location |
                         --kms-project | --labels | --network |
                         --persistent-resource-id | --python-package-uris |
                         --region | --service-account | --worker-pool-spec

For detailed information on this command and its flags, run:
  gcloud ai custom-jobs create --help
```

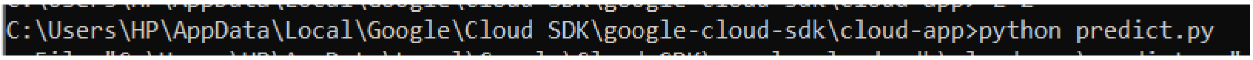
predict.py – script for prediction to test the deployed model.



› HP › AppData › Local › Google › Cloud SDK › google-cloud-sdk › cloud-app › predict.py ›

```python
from google.cloud import aiplatform

def predict():
    client = aiplatform.gapic.PredictionServiceClient()
    endpoint = client.endpoint_path(project='your-project',
    instance = {'input': [/* your data */]}
    response = client.predict(endpoint=endpoint, instances=[
    print(response.predictions)

if __name__ == '__main__':
    predict()
```

python predict.py – running the script



```
C:\Users\HP\AppData\Local\Google\Cloud SDK\google-cloud-sdk\cloud-app>python predict.py
```