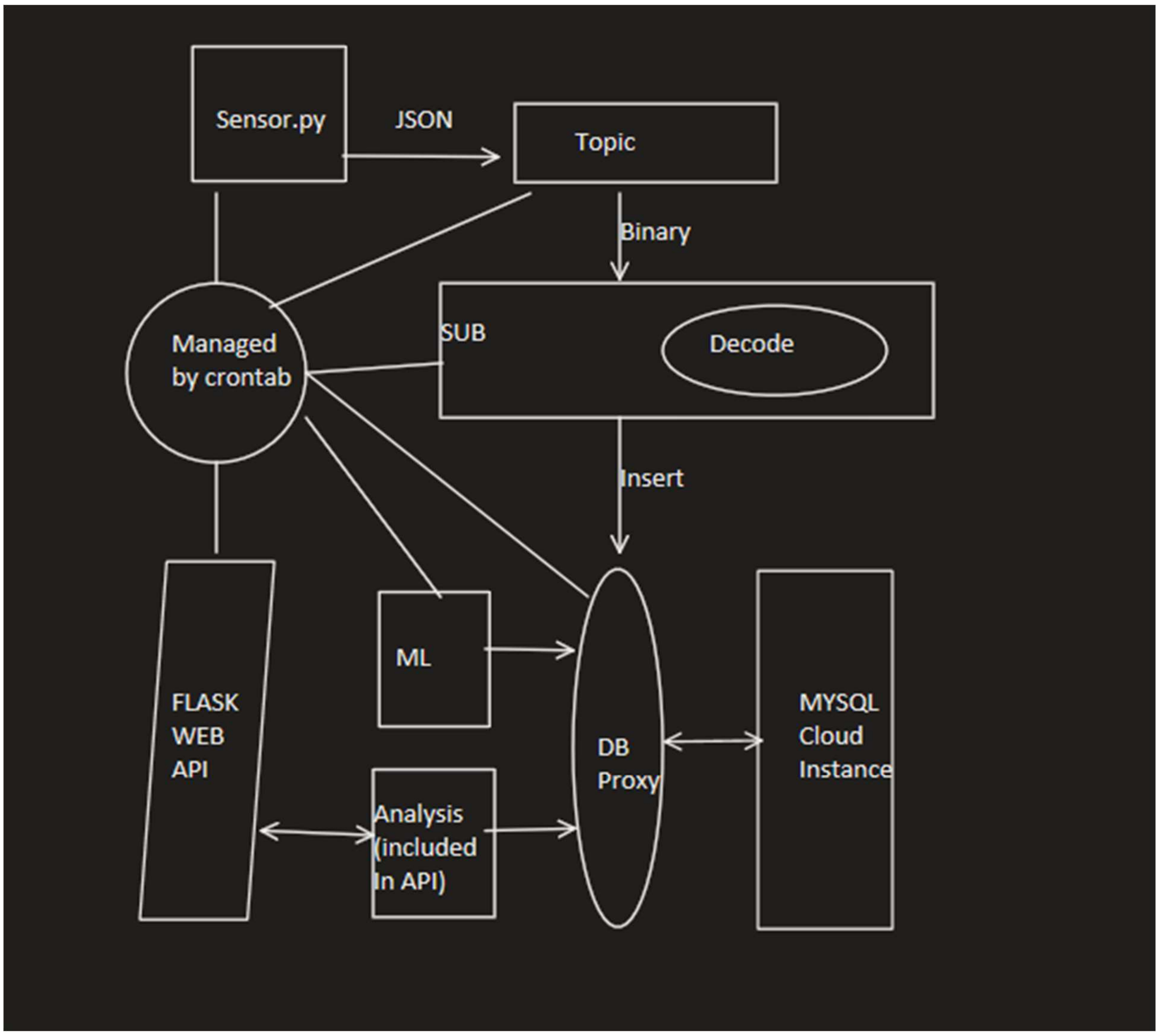
Python Data Engineer Project

1- introduction:

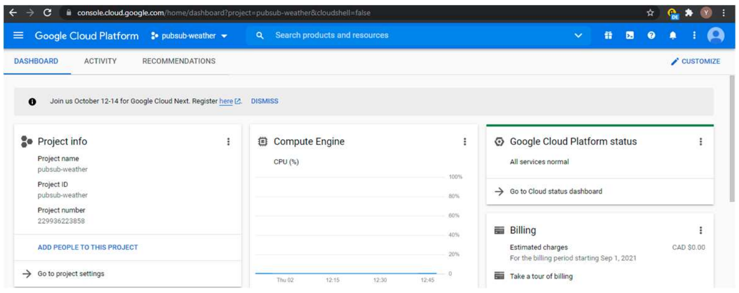
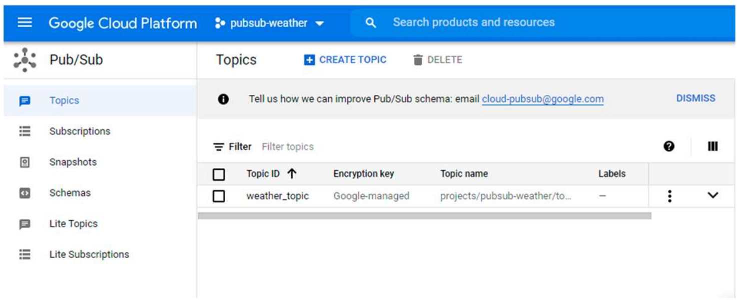
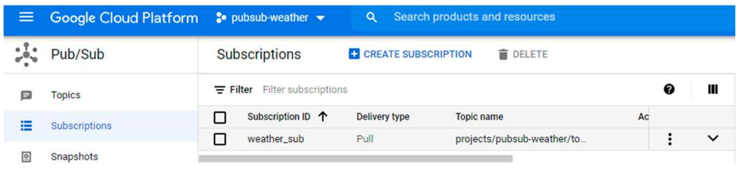
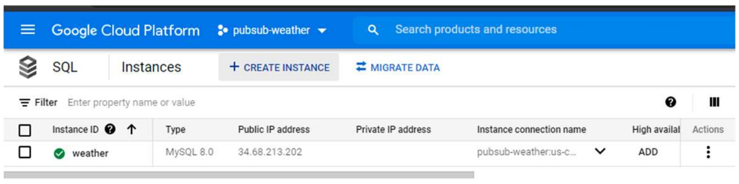
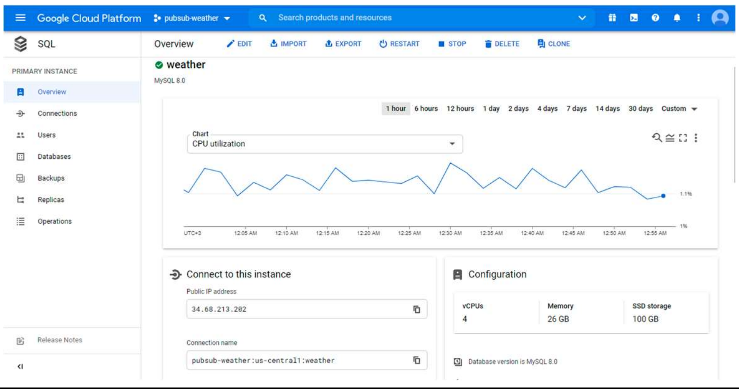
This project aims to simulate 3 IOTs which gathers the temperature each second and publish their output to pub/sub topic which is a messaging middleware for microservices communication. The subscriber pulls the data and store them in MYSQL database to be analyzed

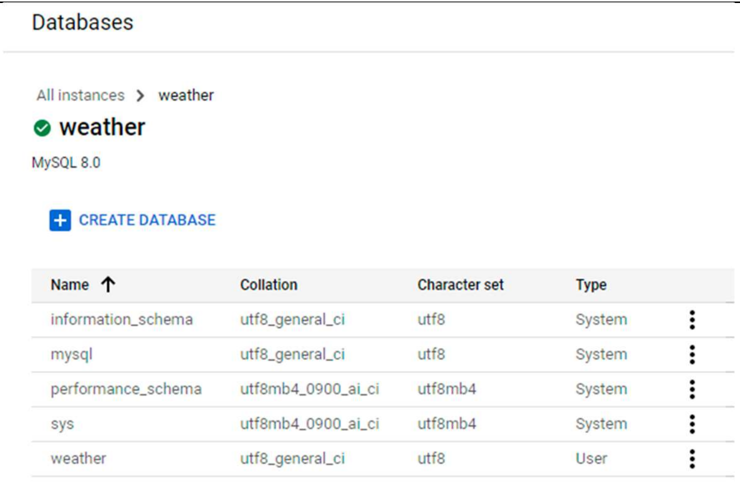
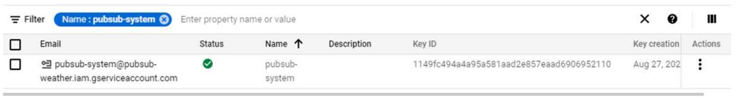
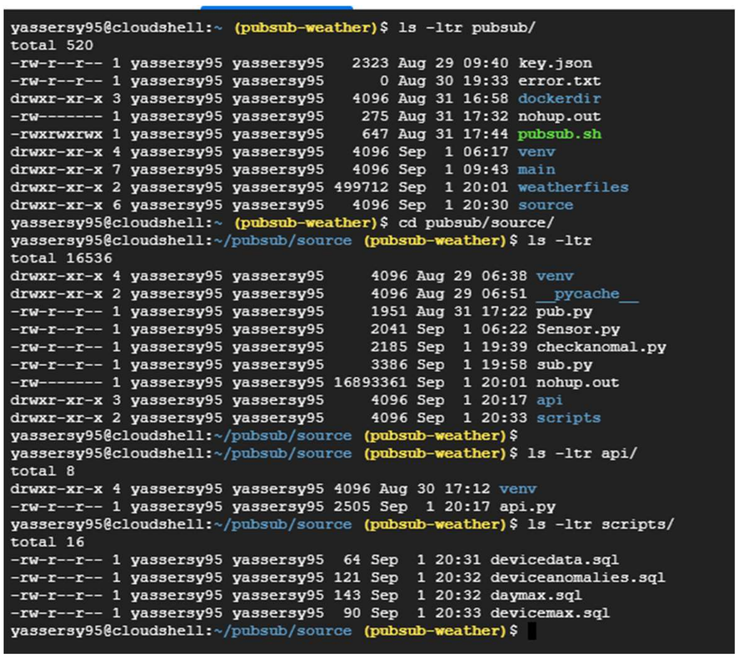
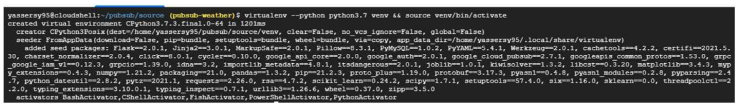
The database entries are analyzed using SQL and the anomalies are detected and flagged in the database. The analysis and anomalies can be accessed using a web API.

The below sketch shows how the system should work:



2- <u>Usecase scenario:</u>

Step No.	Execution Steps	Expected Result
1	A new google cloud project was created. 'pubsub-weather'	 console.cloud.google.com/home/dashboard?project=pubsub-weather&cloudshell=false Google Cloud Platform · pubsub-weather · Search products and resources DASHBOARD · ACTIVITY · RECOMMENDATIONS · CUSTOMIZE Join us October 12-14 for Google Cloud Next. Register here. DISMISS Project info — Project name: pubsub-weather; Project ID: pubsub-weather; Project number: 229936223858; ADD PEOPLE TO THIS PROJECT; Go to project settings Compute Engine — CPU (%) Google Cloud Platform status — All services normal; Go to Cloud status dashboard Billing — Estimated charges CAD $0.00; For the billing period starting Sep 1, 2021; Take a tour of billing
2	A topic was created. 'weather_topic'	 Google Cloud Platform · pubsub-weather · Search products and resources Pub/Sub — Topics · CREATE TOPIC · DELETE Tell us how we can improve Pub/Sub schema: email cloud-pubsub@google.com DISMISS Topics, Subscriptions, Snapshots, Schemas, Lite Topics, Lite Subscriptions Filter: Filter topics Topic ID: weather_topic · Encryption key: Google-managed · Topic name: projects/pubsub-weather/to... · Labels: –
3	A pull subscriber was created. 'weather_sub'	 Google Cloud Platform · pubsub-weather · Search products and resources Pub/Sub — Subscriptions · CREATE SUBSCRIPTION · DELETE Topics, Subscriptions, Snapshots Filter: Filter subscriptions Subscription ID: weather_sub · Delivery type: Pull · Topic name: projects/pubsub-weather/to...
4	SQL instance was created. 'weather' Associated DB was created. 'weather'	 Google Cloud Platform · pubsub-weather · Search products and resources SQL — Instances · CREATE INSTANCE · MIGRATE DATA Filter: Enter property name or value Instance ID: weather · Type: MySQL 8.0 · Public IP address: 34.68.213.202 · Private IP address: · Instance connection name: pubsub-weather:us-c... · High availability: ADD  SQL — Overview · EDIT · IMPORT · EXPORT · RESTART · STOP · DELETE · CLONE PRIMARY INSTANCE: Overview, Connections, Users, Databases, Backups, Replicas, Operations weather — MySQL 8.0 Chart: CPU utilization Connect to this instance — Public IP address: 34.68.213.202; Connection name: pubsub-weather:us-central1:weather Configuration — vCPUs: 4; Memory: 26 GB; SSD storage: 100 GB; Database version is MySQL 8.0

		
5	Service account was created for the project to allow pub/sub and a key was generated.	
6	Open CloudShell. Select the project if not selected. 'gcloud config set project pubsub-weather' The associated screenshot shows the project directories and sources. (All the sources will be shared explicitly)	
7	Activate the virtual environment with python 3.7 along with all required libraries.	

Databases

All instances > weather

weather

MySQL 8.0

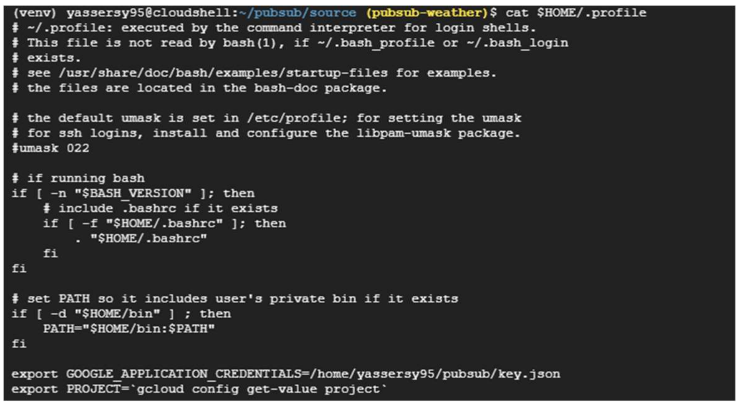
Name	Collation	Character set	Type
information_schema	utf8_general_ci	utf8	System
mysql	utf8_general_ci	utf8	System
performance_schema	utf8mb4_0900_ai_ci	utf8mb4	System
sys	utf8mb4_0900_ai_ci	utf8mb4	System
weather	utf8_general_ci	utf8	User

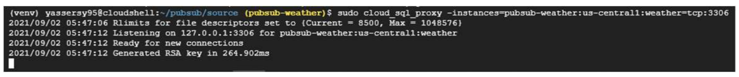
Filter Name : pubsub-system Enter property name or value

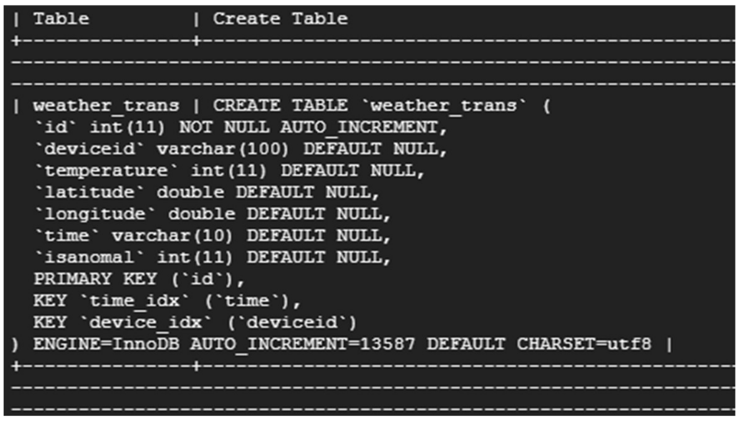
Email	Status	Name	Description	Key ID	Key creation	Actions
pubsub-system@pubsub-weather.iam.gserviceaccount.com	✓	pubsub-system		1149fc494a4a95a581aad2e857eaad6906952110	Aug 27, 202	⋮

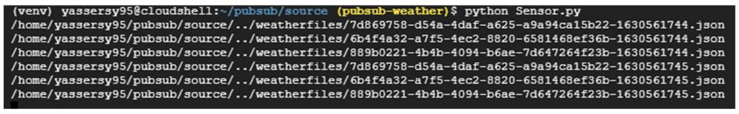
```
yassersy95@cloudshell:~ (pubsub-weather)$ ls -ltr pubsub/
total 520
-rw-r--r-- 1 yassersy95 yassersy95   2323 Aug 29 09:40 key.json
-rw-r--r-- 1 yassersy95 yassersy95      0 Aug 30 19:33 error.txt
drwxr-xr-x 3 yassersy95 yassersy95   4096 Aug 31 16:58 dockerdir
-rw------- 1 yassersy95 yassersy95    275 Aug 31 17:32 nohup.out
-rwxrwxrwx 1 yassersy95 yassersy95    647 Aug 31 17:44 pubsub.sh
drwxr-xr-x 4 yassersy95 yassersy95   4096 Sep  1 06:17 venv
drwxr-xr-x 7 yassersy95 yassersy95   4096 Sep  1 09:43 main
drwxr-xr-x 2 yassersy95 yassersy95 499712 Sep  1 20:01 weatherfiles
drwxr-xr-x 6 yassersy95 yassersy95   4096 Sep  1 20:30 source
yassersy95@cloudshell:~ (pubsub-weather)$ cd pubsub/source/
yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$ ls -ltr
total 16536
drwxr-xr-x 4 yassersy95 yassersy95      4096 Aug 29 06:38 venv
drwxr-xr-x 2 yassersy95 yassersy95      4096 Aug 29 06:51 __pycache__
-rw-r--r-- 1 yassersy95 yassersy95      1951 Aug 31 17:22 pub.py
-rw-r--r-- 1 yassersy95 yassersy95      2041 Sep  1 06:22 Sensor.py
-rw-r--r-- 1 yassersy95 yassersy95      2185 Sep  1 19:39 checkanomal.py
-rw-r--r-- 1 yassersy95 yassersy95      3386 Sep  1 19:58 sub.py
-rw------- 1 yassersy95 yassersy95 16893361 Sep  1 20:01 nohup.out
drwxr-xr-x 3 yassersy95 yassersy95      4096 Sep  1 20:17 api
drwxr-xr-x 2 yassersy95 yassersy95      4096 Sep  1 20:33 scripts
yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$
yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$ ls -ltr api/
total 8
drwxr-xr-x 4 yassersy95 yassersy95 4096 Aug 30 17:12 venv
-rw-r--r-- 1 yassersy95 yassersy95 2505 Sep  1 20:17 api.py
yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$ ls -ltr scripts/
total 16
-rw-r--r-- 1 yassersy95 yassersy95  64 Sep  1 20:31 devicedata.sql
-rw-r--r-- 1 yassersy95 yassersy95 121 Sep  1 20:32 deviceanomalies.sql
-rw-r--r-- 1 yassersy95 yassersy95 143 Sep  1 20:32 daymax.sql
-rw-r--r-- 1 yassersy95 yassersy95  90 Sep  1 20:33 devicemax.sql
yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$
```

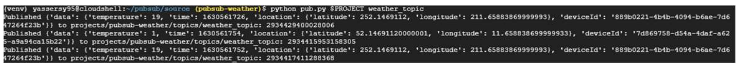
```
yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$ virtualenv --python python3.7 venv && source venv/bin/activate
created virtual environment CPython3.7.3.final.0-64 in 1201ms
  creator CPython3Posix(dest=/home/yassersy95/pubsub/source/venv, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/yassersy95/.local/share/virtualenv)
    added seed packages: Flask==2.0.1, Jinja2==3.0.1, MarkupSafe==2.0.1, Pillow==8.3.1, PyMySQL==1.0.2, PyYAML==5.4.1, Werkzeug==2.0.1, cachetools==4.2.2, certifi==2021.5.
30, charset_normalizer==2.0.4, click==8.0.1, cycler==0.10.0, google_api_core==2.0.0, google_auth==2.0.1, google_cloud_pubsub==2.7.1, googleapis_common_protos==1.53.0, grpc
_google_iam_v1==0.12.3, grpcio==1.39.0, idna==3.2, importlib_metadata==4.8.1, itsdangerous==2.0.1, joblib==1.0.1, kiwisolver==1.3.2, libcst==0.3.20, matplotlib==3.4.3, myp
y_extensions==0.4.3, numpy==1.21.2, packaging==21.0, pandas==1.3.2, pip==21.2.3, proto_plus==1.19.0, protobuf==3.17.3, pyasn1==0.4.8, pyasn1_modules==0.2.8, pyparsing==2.4
.7, python_dateutil==2.8.2, pytz==2021.1, requests==2.26.0, rsa==4.7.2, scikit_learn==0.24.2, scipy==1.7.1, setuptools==57.4.0, six==1.16.0, sklearn==0.0, threadpoolctl==2
.2.0, typing_extensions==3.10.0.1, typing_inspect==0.7.1, urllib3==1.26.6, wheel==0.37.0, zipp==3.5.0
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator
```

8	variables related to the project were add to '.profile' as Environment variables	
		`(venv) yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$ cat $HOME/.profile`
		`# ~/.profile: executed by the command interpreter for login shells.`
		`# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login`
		`# exists.`
		`# see /usr/share/doc/bash/examples/startup-files for examples.`
		`# the files are located in the bash-doc package.`
		
		`# the default umask is set in /etc/profile; for setting the umask`
		`# for ssh logins, install and configure the libpam-umask package.`
		`#umask 022`
		
		`# if running bash`
		`if [ -n "$BASH_VERSION" ]; then`
		`    # include .bashrc if it exists`
		`    if [ -f "$HOME/.bashrc" ]; then`
		`        . "$HOME/.bashrc"`
		`    fi`
		`fi`
		
		`# set PATH so it includes user's private bin if it exists`
		`if [ -d "$HOME/bin" ] ; then`
		`    PATH="$HOME/bin:$PATH"`
		`fi`
		
		`export GOOGLE_APPLICATION_CREDENTIALS=/home/yassersy95/pubsub/key.json`
		``export PROJECT=`gcloud config get-value project` ``

9	Launch the SQL Cloud instance proxy via root user.
	'sudo cloud_sql_proxy -instances=pubsub-weather:us-central1:weather=tcp:3306'
	It will allows the system to connect to the cloud database via a proxy with the IP 127.0.0.1:3306
		
		`(venv) yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$ sudo cloud_sql_proxy -instances=pubsub-weather:us-central1:weather=tcp:3306`
		`2021/09/02 05:47:06 Rlimits for file descriptors set to {Current = 8500, Max = 1048576}`
		`2021/09/02 05:47:12 Listening on 127.0.0.1:3306 for pubsub-weather:us-central1:weather`
		`2021/09/02 05:47:12 Ready for new connections`
		`2021/09/02 05:47:12 Generated RSA key in 264.902ms`

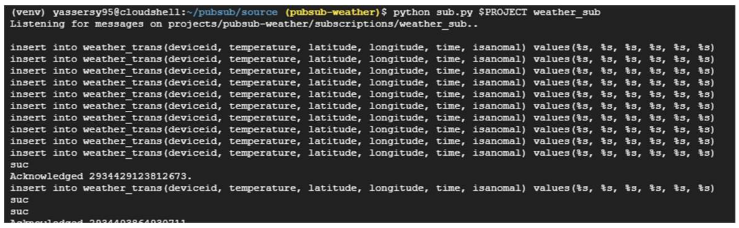
10	User 'weather' and Table 'weather_trans' were created	
		`| Table         | Create Table`
		`+---------------+----------------------------------------`
		`--------------------------------------------------------`
		`--------------------------------------------------------`
		``| weather_trans | CREATE TABLE `weather_trans` (``
		`` `id` int(11) NOT NULL AUTO_INCREMENT,``
		`` `deviceid` varchar(100) DEFAULT NULL,``
		`` `temperature` int(11) DEFAULT NULL,``
		`` `latitude` double DEFAULT NULL,``
		`` `longitude` double DEFAULT NULL,``
		`` `time` varchar(10) DEFAULT NULL,``
		`` `isanomal` int(11) DEFAULT NULL,``
		``  PRIMARY KEY (`id`),``
		``  KEY `time_idx` (`time`),``
		``  KEY `device_idx` (`deviceid`)``
		`) ENGINE=InnoDB AUTO_INCREMENT=13587 DEFAULT CHARSET=utf8 |`
		`+---------------+----------------------------------------`
		`--------------------------------------------------------`
		`--------------------------------------------------------`

11	run the sensor.py.
	It will generate Json files with the name of '[deviceid][timestamp].json' in the directory: $HOME/pubsub/weatherfiles
		
		`(venv) yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$ python Sensor.py`
		`/home/yassersy95/pubsub/source/../weatherfiles/7d869758-d54a-4daf-a625-a9a94ca15b22-1630561744.json`
		`/home/yassersy95/pubsub/source/../weatherfiles/6b4f4a32-a7f5-4ec2-8820-6581468ef36b-1630561744.json`
		`/home/yassersy95/pubsub/source/../weatherfiles/889b0221-4b4b-4094-b6ae-7d647264f23b-1630561744.json`
		`/home/yassersy95/pubsub/source/../weatherfiles/7d869758-d54a-4daf-a625-a9a94ca15b22-1630561745.json`
		`/home/yassersy95/pubsub/source/../weatherfiles/6b4f4a32-a7f5-4ec2-8820-6581468ef36b-1630561745.json`
		`/home/yassersy95/pubsub/source/../weatherfiles/889b0221-4b4b-4094-b6ae-7d647264f23b-1630561745.json`

12	Run pub.sy for the selected project and topic, it will transfer the json file to binary and send them to the messaging queue to be pulled by the subscriber. The json file will be deleted after being sent.	
		`(venv) yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$ python pub.py $PROJECT weather_topic`
		`Published {'data': {'temperature': 19, 'time': 1630561726, 'location': {'latitude': 252.1469112, 'longitude': 211.65883869999993}, 'deviceId': '889b0221-4b4b-4094-b6ae-7d6`
		`47264f23b'}} to projects/pubsub-weather/topics/weather_topic: 2934429400028006`
		`Published {'data': {'temperature': 1, 'time': 1630561754, 'location': {'latitude': 52.14691120000001, 'longitude': 11.658838699999933}, 'deviceId': '7d869758-d54a-4daf-a62`
		`5-a9a94ca15b22'}} to projects/pubsub-weather/topics/weather_topic: 2934415953158305`
		`Published {'data': {'temperature': 19, 'time': 1630561752, 'location': {'latitude': 252.1469112, 'longitude': 211.65883869999993}, 'deviceId': '889b0221-4b4b-4094-b6ae-7d6`
		`47264f23b'}} to projects/pubsub-weather/topics/weather_topic: 2934417411288368`

13

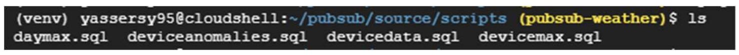
Run sub.py for the selected project. It will pull the binary messages from the messaging queue and convert them to python to be inserted in the database.



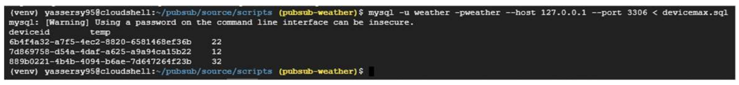
```
(venv) yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$ python sub.py $PROJECT weather_sub
Listening for messages on projects/pubsub-weather/subscriptions/weather_sub..

insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
suc
Acknowledged 2934429123812673.
insert into weather_trans(deviceid, temperature, latitude, longitude, time, isanomal) values(%s, %s, %s, %s, %s, %s)
suc
suc
```

14

The SQL scripts were created under source/scripts directory in case the need to be run manually.



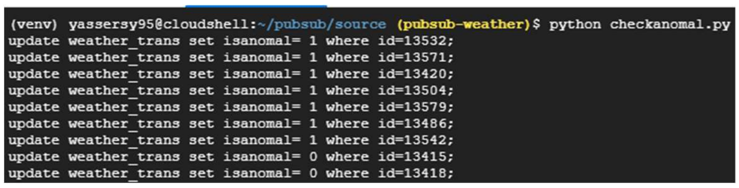
```
(venv) yassersy95@cloudshell:~/pubsub/source/scripts (pubsub-weather)$ ls
daymax.sql  deviceanomalies.sql  devicedata.sql  devicemax.sql
```



```
(venv) yassersy95@cloudshell:~/pubsub/source/scripts (pubsub-weather)$ mysql -u weather -pweather --host 127.0.0.1 --port 3306 < devicemax.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
deviceid        temp
6b4f4a32-a7f5-4ec2-8820-6581468ef36b    22
7d869758-d54a-4daf-a625-a9a94ca15b22    12
889b0221-4b4b-4094-b6ae-7d647264f23b    32
(venv) yassersy95@cloudshell:~/pubsub/source/scripts (pubsub-weather)$
```

15

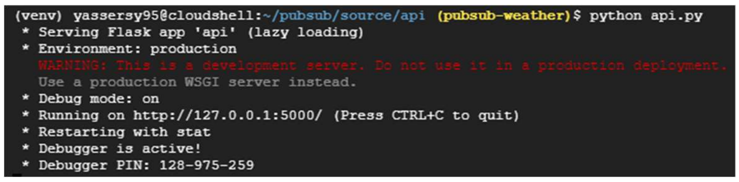
Running the process 'checkanomal.py' which will check all new entries and flag them as anomaly or not



```
(venv) yassersy95@cloudshell:~/pubsub/source (pubsub-weather)$ python checkanomal.py
update weather_trans set isanomal= 1 where id=13532;
update weather_trans set isanomal= 1 where id=13571;
update weather_trans set isanomal= 1 where id=13420;
update weather_trans set isanomal= 1 where id=13504;
update weather_trans set isanomal= 1 where id=13579;
update weather_trans set isanomal= 1 where id=13486;
update weather_trans set isanomal= 1 where id=13542;
update weather_trans set isanomal= 0 where id=13415;
update weather_trans set isanomal= 0 where id=13418;
```

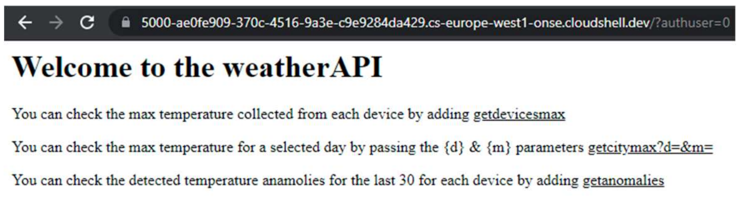
16

Run the flask web API which returns the result of the data analysis and anomaly detection.



```
(venv) yassersy95@cloudshell:~/pubsub/source/api (pubsub-weather)$ python api.py
 * Serving Flask app 'api' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 128-975-259
```

Home page:



5000-ae0fe909-370c-4516-9a3e-c9e9284da429.cs-europe-west1-onse.cloudshell.dev/?authuser=0

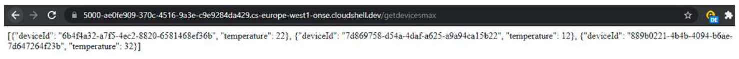
Welcome to the weatherAPI

You can check the max temperature collected from each device by adding getdevicesmax

You can check the max temperature for a selected day by passing the {d} & {m} parameters getcitymax?d=&m=

You can check the detected temperature anamolies for the last 30 for each device by adding getanomalies

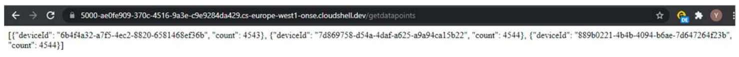
Get max temperature for each device:



5000-ae0fe909-370c-4516-9a3e-c9e9284da429.cs-europe-west1-onse.cloudshell.dev/getdevicesmax

```
[{"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": 22}, {"deviceId": "7d869758-d54a-4daf-a625-a9a94ca15b22", "temperature": 12}, {"deviceId": "889b0221-4b4b-4094-b6ae-7d647264f23b", "temperature": 32}]
```

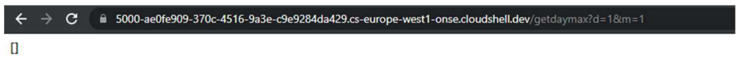
Get the aggregated count by device:



5000-ae0fe909-370c-4516-9a3e-c9e9284da429.cs-europe-west1-onse.cloudshell.dev/getdatapoints

```
[{"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "count": 4543}, {"deviceId": "7d869758-d54a-4daf-a625-a9a94ca15b22", "count": 4544}, {"deviceId": "889b0221-4b4b-4094-b6ae-7d647264f23b", "count": 4544}]
```

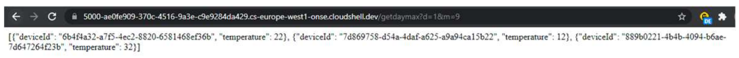
Check by day and month for the max temperature:
(No data found)



5000-ae0fe909-370c-4516-9a3e-c9e9284da429.cs-europe-west1-onse.cloudshell.dev/getdaymax?d=1&m=1

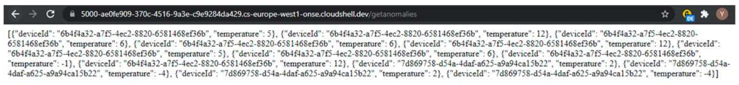
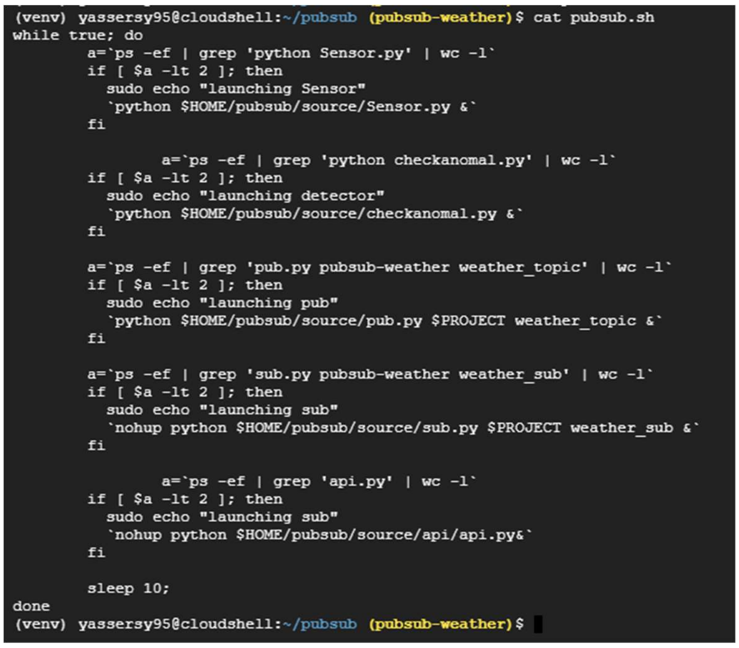
```
[]
```

(Data exists)



5000-ae0fe909-370c-4516-9a3e-c9e9284da429.cs-europe-west1-onse.cloudshell.dev/getdaymax?d=1&m=9

```
[{"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": 22}, {"deviceId": "7d869758-d54a-4daf-a625-a9a94ca15b22", "temperature": 12}, {"deviceId": "889b0221-4b4b-4094-b6ae-7d647264f23b", "temperature": 32}]
```

		anomalies for each device in the last 30 minutes: 
17	A shell script was created to ensure the process can work as a flow. It can be managed in the crontab to the crontab.	

```
5000-ae0fe909-370c-4516-9a3e-c9e9284da429.cs-europe-west1-onse.cloudshell.dev/getanomalies
```

[{"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": 5}, {"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": 12}, {"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": 6}, {"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": 6}, {"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": 12}, {"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": 5}, {"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": 6}, {"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": -1}, {"deviceId": "6b4f4a32-a7f5-4ec2-8820-6581468ef36b", "temperature": 12}, {"deviceId": "7d869758-d54a-4daf-a625-a9a94ca15b22", "temperature": 2}, {"deviceId": "7d869758-d54a-4daf-a625-a9a94ca15b22", "temperature": -4}, {"deviceId": "7d869758-d54a-4daf-a625-a9a94ca15b22", "temperature": 2}, {"deviceId": "7d869758-d54a-4daf-a625-a9a94ca15b22", "temperature": -4}]

```
(venv) yassersy95@cloudshell:~/pubsub (pubsub-weather)$ cat pubsub.sh
while true; do
        a=`ps -ef | grep 'python Sensor.py' | wc -l`
        if [ $a -lt 2 ]; then
          sudo echo "launching Sensor"
          `python $HOME/pubsub/source/Sensor.py &`
        fi

                a=`ps -ef | grep 'python checkanomal.py' | wc -l`
        if [ $a -lt 2 ]; then
          sudo echo "launching detector"
          `python $HOME/pubsub/source/checkanomal.py &`
        fi

        a=`ps -ef | grep 'pub.py pubsub-weather weather_topic' | wc -l`
        if [ $a -lt 2 ]; then
          sudo echo "launching pub"
          `python $HOME/pubsub/source/pub.py $PROJECT weather_topic &`
        fi

        a=`ps -ef | grep 'sub.py pubsub-weather weather_sub' | wc -l`
        if [ $a -lt 2 ]; then
          sudo echo "launching sub"
          `nohup python $HOME/pubsub/source/sub.py $PROJECT weather_sub &`
        fi

                a=`ps -ef | grep 'api.py' | wc -l`
        if [ $a -lt 2 ]; then
          sudo echo "launching sub"
          `nohup python $HOME/pubsub/source/api/api.py&`
        fi

        sleep 10;
done
(venv) yassersy95@cloudshell:~/pubsub (pubsub-weather)$
```

3- Notes:

a. The project was created on the cloudshell which provide 5GB storage space without using any VM compute instances or buckets.

b. The three sensors are considered in three different places with different climate, and the data generated is illogical with a second-by-second change in the temperature. This noise was created to ensure having enough noise for the machine learning process

c. The complete ETL process along with the web API and the ML process can be automated to run without the user intervention by adding the pubsub.sh to a crontab.

d. The use of docker container is a planned step.

e. K-means algorithm was used to detect the anomaly for the temperatures in one minute. Here, each minute is separate from the others and the process works on it separately. Other algorithms can be used such as LSTM to predict what the next temperature would be in a series with a specified margin of error and check whether the next input matches the predicted or not. Implementing such approach requires the data to be more stable and less random and real which is the reason for choosing the K-mean algorithm.