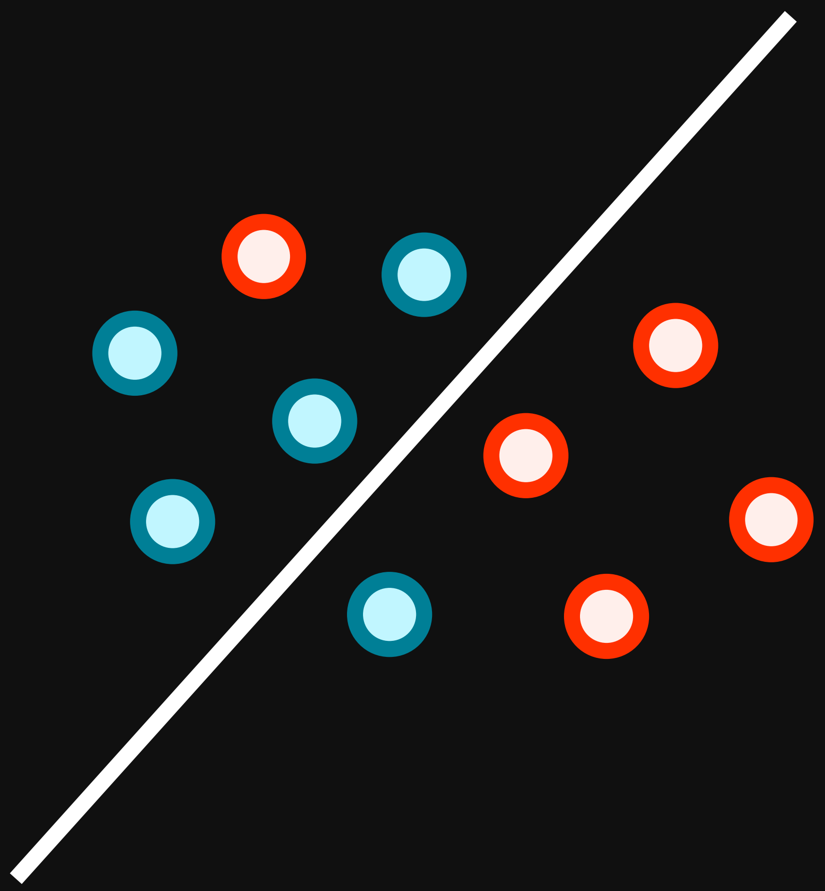
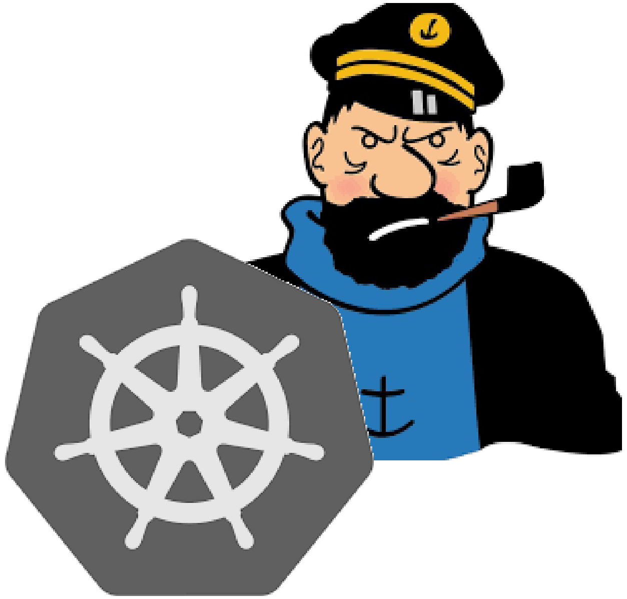
# Basics of Machine Learning

Dmitry Ryabokon, github.com/dryabokon



# Lesson 21
# Deployments in GCP



# Copying files cloudconsole ↔ VM

## Copy one file from cloudshell to VM

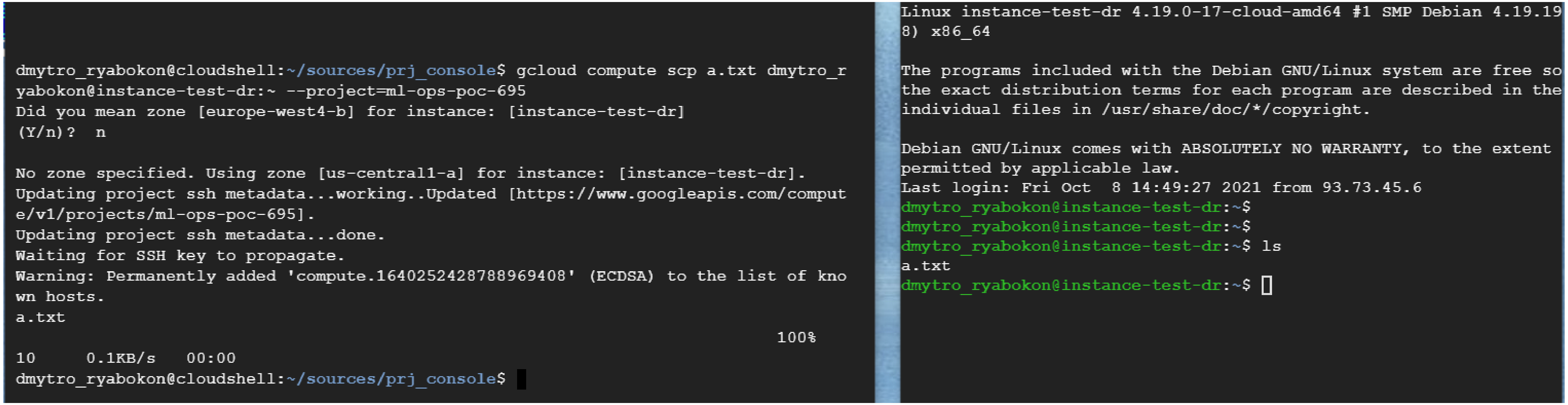
gcloud compute scp a.txt dmytro_ryabokon@instance-test-dr:~ --project=ml-ops-poc-695

## Copy one file from VM to cloudshell

gcloud compute scp dmytro_ryabokon@instance-test-dr:~/b.txt ./b.txt --project=ml-ops-poc-695

## Copy folder file from VM to cloudshell

gcloud compute scp --recurse dmytro_ryabokon@instance-test-dr:~/sources/prj_console/* ~/sources/prj_console/ --project=ml-ops-poc-695



```
dmytro_ryabokon@cloudshell:~/sources/prj_console$ gcloud compute scp a.txt dmytro_r
yabokon@instance-test-dr:~ --project=ml-ops-poc-695
Did you mean zone [europe-west4-b] for instance: [instance-test-dr]
(Y/n)?  n

No zone specified. Using zone [us-central1-a] for instance: [instance-test-dr].
Updating project ssh metadata...working..Updated [https://www.googleapis.com/comput
e/v1/projects/ml-ops-poc-695].
Updating project ssh metadata...done.
Waiting for SSH key to propagate.
Warning: Permanently added 'compute.1640252428788969408' (ECDSA) to the list of kno
wn hosts.
a.txt
                                                                        100%
10     0.1KB/s   00:00
dmytro_ryabokon@cloudshell:~/sources/prj_console$
```

```
Linux instance-test-dr 4.19.0-17-cloud-amd64 #1 SMP Debian 4.19.19
8) x86_64

The programs included with the Debian GNU/Linux system are free so
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct  8 14:49:27 2021 from 93.73.45.6
dmytro_ryabokon@instance-test-dr:~$
dmytro_ryabokon@instance-test-dr:~$
dmytro_ryabokon@instance-test-dr:~$ ls
a.txt
dmytro_ryabokon@instance-test-dr:~$
```

# Push docker image to GCR

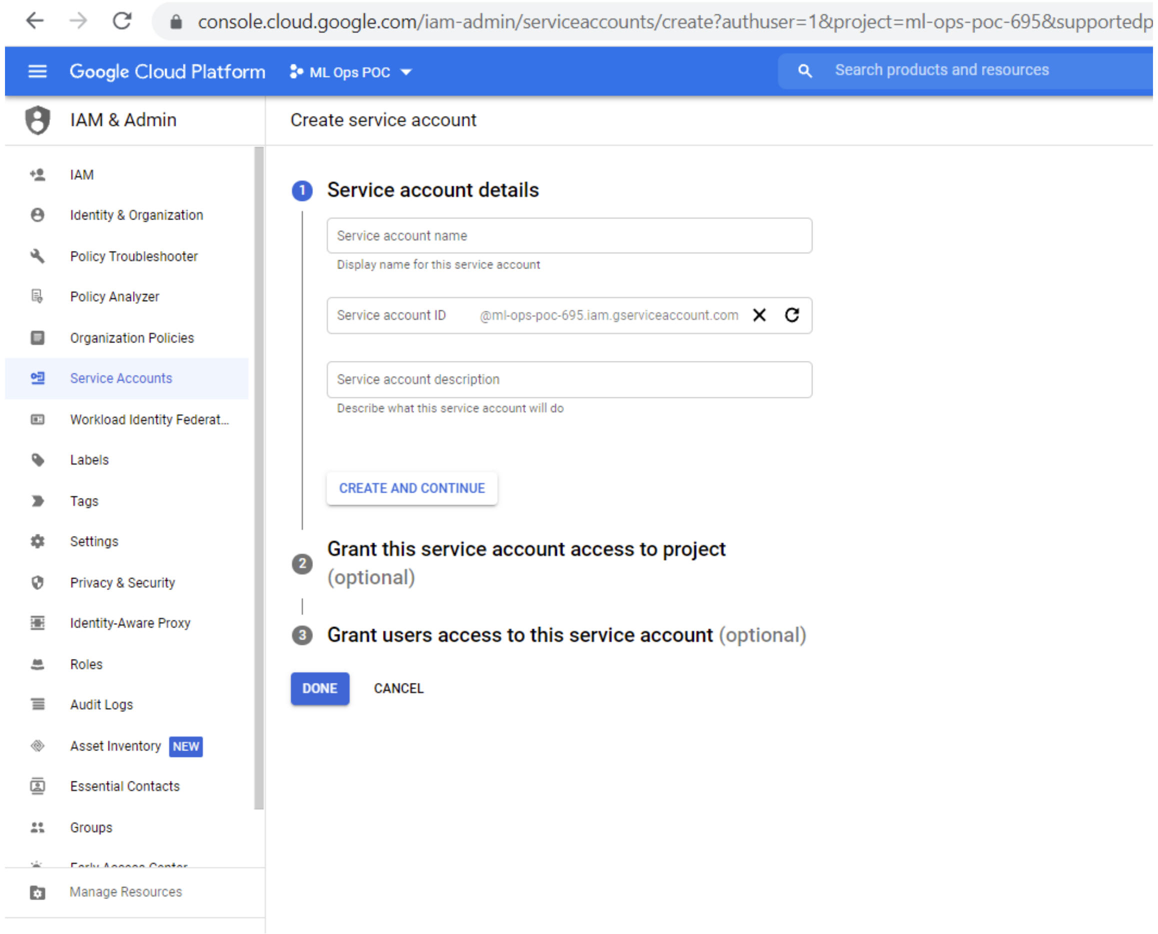
1) https://console.cloud.google.com/iam-admin/serviceaccount

2) https://cloud.google.com/container-registry/docs/advanced-authentication
gcloud auth configure-docker

3) sudo docker tag hello_dima gcr.io/ml-ops-poc-695/hello_dima
4) sudo docker push gcr.io/ml-ops-poc-695/hello_dima:latest

# Push docker image to GCR

## Create Service Account



console.cloud.google.com/iam-admin/serviceaccounts/create?authuser=1&project=ml-ops-poc-695&supportedp

Create service account

1 Service account details

Service account name

Display name for this service account

Service account ID @ml-ops-poc-695.iam.gserviceaccount.com

Service account description

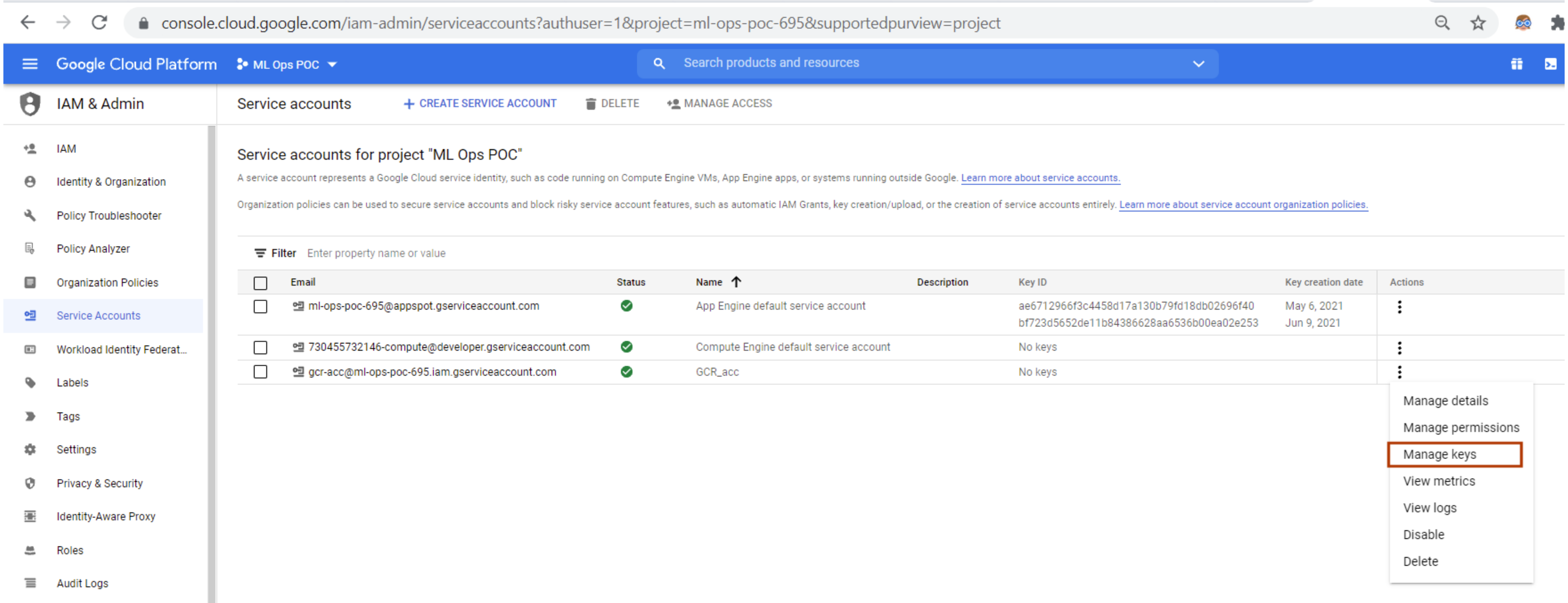
Describe what this service account will do

2 Grant this service account access to project (optional)

3 Grant users access to this service account (optional)

# Push docker image to GCR

## Create Service Account: add key, store it



Service accounts for project "ML Ops POC"

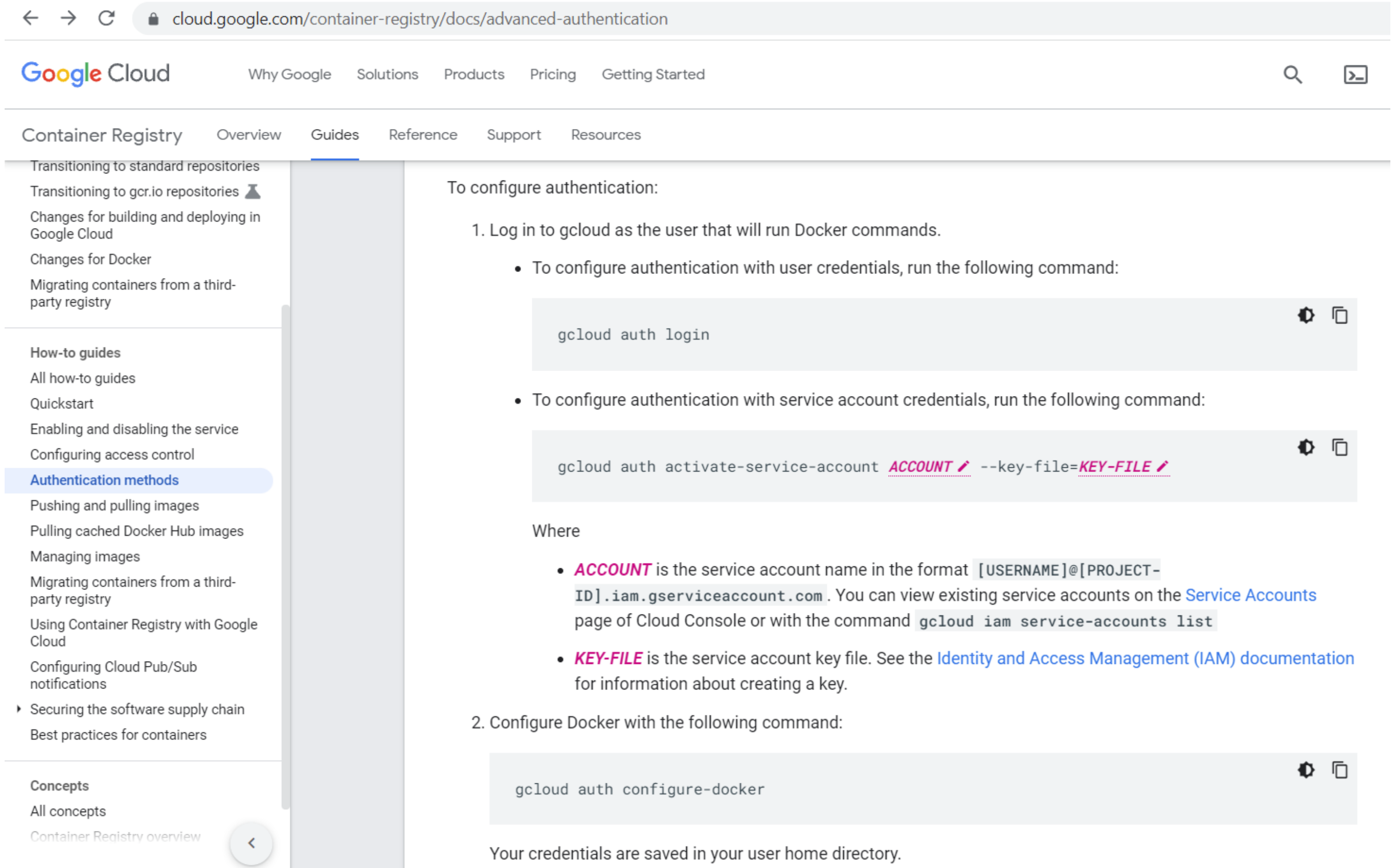
A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more about service accounts.

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. Learn more about service account organization policies.

| □ | Email | Status | Name ↑ | Description | Key ID | Key creation date | Actions |
|---|---|---|---|---|---|---|---|
| □ | ml-ops-poc-695@appspot.gserviceaccount.com | ✓ | App Engine default service account | | ae6712966f3c4458d17a130b79fd18db02696f40<br>bf723d5652de11b84386628aa6536b00ea02e253 | May 6, 2021<br>Jun 9, 2021 | ⋮ |
| □ | 730455732146-compute@developer.gserviceaccount.com | ✓ | Compute Engine default service account | | No keys | | ⋮ |
| □ | gcr-acc@ml-ops-poc-695.iam.gserviceaccount.com | ✓ | GCR_acc | | No keys | | ⋮ |

# Push docker image to GCR

## Configure authentication



cloud.google.com/container-registry/docs/advanced-authentication
Google Cloud
Why Google
Solutions
Products
Pricing
Getting Started
Container Registry
Overview
Guides
Reference
Support
Resources
Transitioning to standard repositories
Transitioning to gcr.io repositories
Changes for building and deploying in Google Cloud
Changes for Docker
Migrating containers from a third-party registry
How-to guides
All how-to guides
Quickstart
Enabling and disabling the service
Configuring access control
Authentication methods
Pushing and pulling images
Pulling cached Docker Hub images
Managing images
Migrating containers from a third-party registry
Using Container Registry with Google Cloud
Configuring Cloud Pub/Sub notifications
Securing the software supply chain
Best practices for containers
Concepts
All concepts
Container Registry overview

To configure authentication:

1. Log in to gcloud as the user that will run Docker commands.
   - To configure authentication with user credentials, run the following command:

     ```
     gcloud auth login
     ```

   - To configure authentication with service account credentials, run the following command:

     ```
     gcloud auth activate-service-account ACCOUNT --key-file=KEY-FILE
     ```

     Where

     - *ACCOUNT* is the service account name in the format `[USERNAME]@[PROJECT-ID].iam.gserviceaccount.com`. You can view existing service accounts on the Service Accounts page of Cloud Console or with the command `gcloud iam service-accounts list`
     - *KEY-FILE* is the service account key file. See the Identity and Access Management (IAM) documentation for information about creating a key.

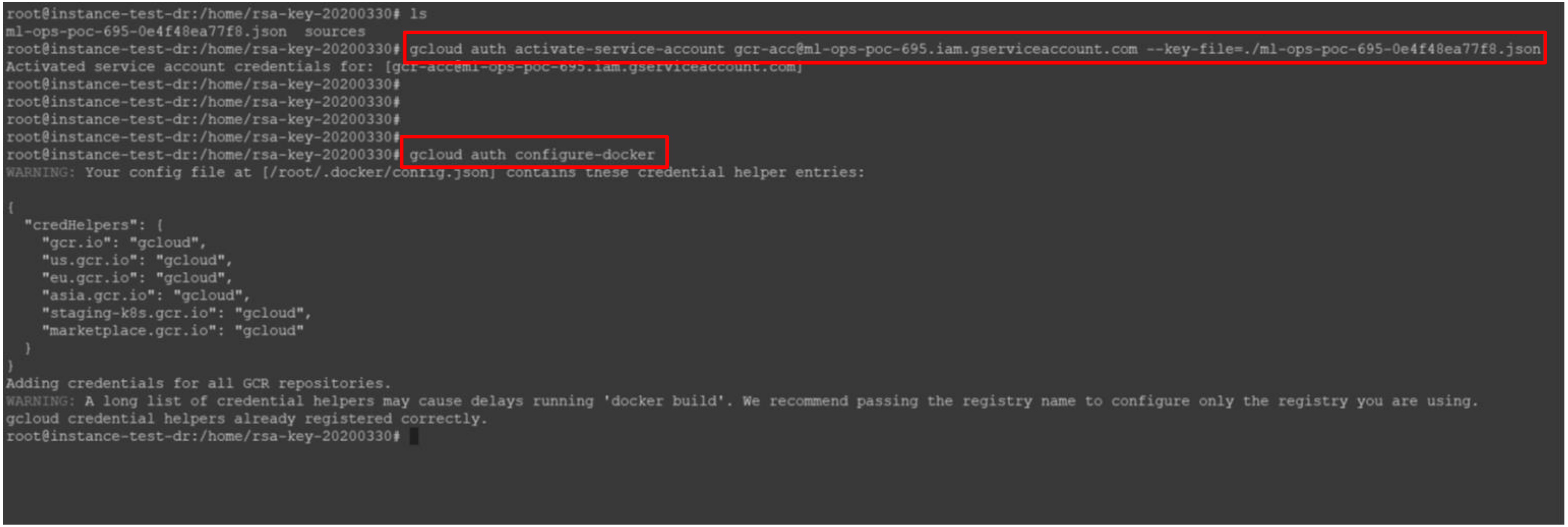
2. Configure Docker with the following command:

   ```
   gcloud auth configure-docker
   ```

   Your credentials are saved in your user home directory.

# Push docker image to GCR

## Configure authentication



```
root@instance-test-dr:/home/rsa-key-20200330# ls
ml-ops-poc-695-0e4f48ea77f8.json  sources
root@instance-test-dr:/home/rsa-key-20200330# gcloud auth activate-service-account gcr-acc@ml-ops-poc-695.iam.gserviceaccount.com --key-file=./ml-ops-poc-695-0e4f48ea77f8.json
Activated service account credentials for: [gcr-acc@ml-ops-poc-695.iam.gserviceaccount.com]
root@instance-test-dr:/home/rsa-key-20200330#
root@instance-test-dr:/home/rsa-key-20200330#
root@instance-test-dr:/home/rsa-key-20200330#
root@instance-test-dr:/home/rsa-key-20200330#
root@instance-test-dr:/home/rsa-key-20200330# gcloud auth configure-docker
WARNING: Your config file at [/root/.docker/config.json] contains these credential helper entries:

{
  "credHelpers": {
    "gcr.io": "gcloud",
    "us.gcr.io": "gcloud",
    "eu.gcr.io": "gcloud",
    "asia.gcr.io": "gcloud",
    "staging-k8s.gcr.io": "gcloud",
    "marketplace.gcr.io": "gcloud"
  }
}
Adding credentials for all GCR repositories.
WARNING: A long list of credential helpers may cause delays running 'docker build'. We recommend passing the registry name to configure only the registry you are using.
gcloud credential helpers already registered correctly.
root@instance-test-dr:/home/rsa-key-20200330# 
```

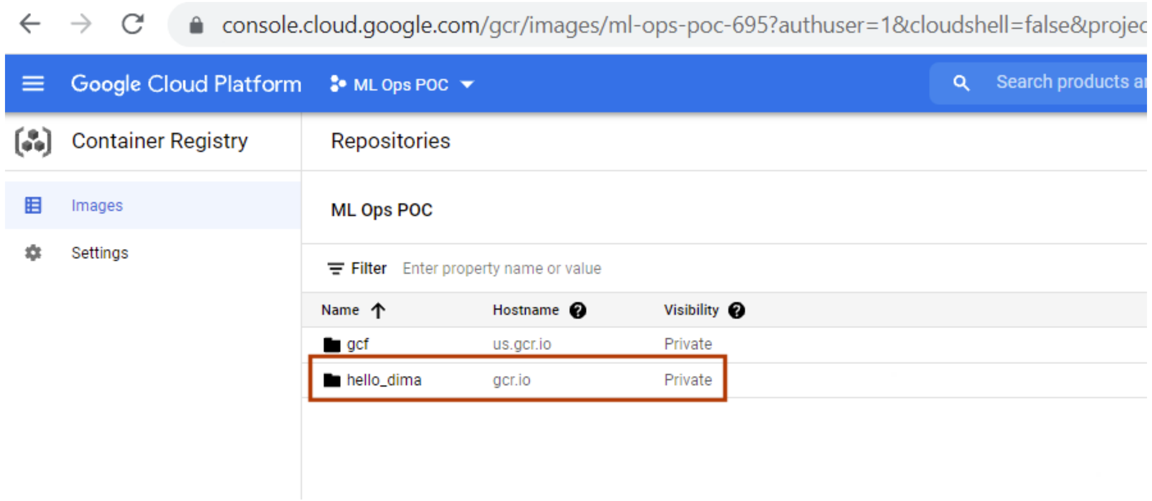
# Push docker image to GCR

## Push docker image

```
root@instance-test-dr:/home/rsa-key-20200330# sudo docker images
REPOSITORY                         TAG          IMAGE ID       CREATED        SIZE
prj_flask_nginx_nginx              latest       aac06a4109ae   10 hours ago   109MB
prj_flask_nginx_flask_app          latest       dfc41c0075e3   10 hours ago   928MB
hello_dima                         latest       032962ec3886   31 hours ago   5.6MB
gcr.io/ml-ops-poc-695/hello_dima   latest       032962ec3886   31 hours ago   5.6MB
ubuntu                             latest       597ce1600cf4   8 days ago     72.8MB
busybox                            latest       16ea53ea7c65   3 weeks ago    1.24MB
alpine                             latest       14119a10abf4   6 weeks ago    5.6MB
python                             3.6-jessie   890456b21ed5   2 years ago    703MB
nginx                              1.15.8       f09fe80eb0e7   2 years ago    109MB
python                             3.6.7        1ec4d11819ad   2 years ago    918MB
root@instance-test-dr:/home/rsa-key-20200330#
root@instance-test-dr:/home/rsa-key-20200330#
root@instance-test-dr:/home/rsa-key-20200330#
root@instance-test-dr:/home/rsa-key-20200330#
root@instance-test-dr:/home/rsa-key-20200330#
root@instance-test-dr:/home/rsa-key-20200330# sudo docker push gcr.io/ml-ops-poc-695/hello_dima:latest
The push refers to repository [gcr.io/ml-ops-poc-695/hello_dima]
e2eb06d8af82: Layer already exists
latest: digest: sha256:50f64478c42a993af03592591f1e7ba1435267ac8a1a25814ff71113545e31fd size: 528
root@instance-test-dr:/home/rsa-key-20200330#
```

# Push docker image to GCR

## Docker image appears at GCR



console.cloud.google.com/gcr/images/ml-ops-poc-695?authuser=1&cloudshell=false&projec
Google Cloud Platform
ML Ops POC
Search products a
Container Registry
Images
Settings
Repositories
ML Ops POC
Filter
Enter property name or value

| Name | Hostname | Visibility |
|---|---|---|
| gcf | us.gcr.io | Private |
| hello_dima | gcr.io | Private |

# Flask – nginx docker

https://towardsdatascience.com/how-to-deploy-ml-models-using-flask-gunicorn-nginx-docker-9b32055b3d0
https://github.com/ivanpanshin/flask_gunicorn_nginx_docker

# Flask – nginx docker

```python
#https://www.w3schools.com/tags/ref_httpmethods.asp
#https://towardsdatascience.com/how-to-deploy-ml-models-using-flask-gunicorn-nginx-docker-9b32055b3d0
#https://github.com/ivanpanshin/flask_gunicorn_nginx_docker
# ----------------------------------------------------------------------------------------------------------------------
from flask import Flask
from flask import request, jsonify
# ----------------------------------------------------------------------------------------------------------------------
server = Flask(__name__)
# ----------------------------------------------------------------------------------------------------------------------
#request data from a specified resource
def run_request_GET():
    return 'Get response OK\n'
# ----------------------------------------------------------------------------------------------------------------------
#send data to a server to create/update a resource
#POST method is called when you have to add a child resource
def run_request_POST():

    data_dct = request.json
    response = jsonify(data_dct)
    return response
# ----------------------------------------------------------------------------------------------------------------------
#send data to a server to create/update a resource
#calling the same PUT request multiple times will always produce the same result (PUT requests are idempotent)
#PUT method is called when you have to modify a single resource
def run_request_PUT():
    data_dct = request.json
    response = jsonify(data_dct)
    return response
# ----------------------------------------------------------------------------------------------------------------------
@server.route('/', methods=['GET', 'POST', 'PUT'])
def hello_world():
    if   request.method == 'GET': return run_request_GET()
    elif request.method == 'POST':return run_request_POST()
    elif request.method == 'PUT' :return run_request_PUT()
# ----------------------------------------------------------------------------------------------------------------------
if __name__ == "__main__":
    server.run(debug=True,port=8000)
    #curl -X GET  http://127.0.0.1:8000/
    #curl -X POST http://127.0.0.1:8000/ -d {\"key1\":\"value1\",\"key2\":\"value2\"} -H "Content-Type: application/json"

    #curl -w "\n" -s https://api.ipify.org
    #curl -X GET  http://34.122.156.46
    #curl -X POST http://34.122.156.46 -d {\"key1\":\"value1\",\"key2\":\"value2\"} -H "Content-Type: application/json"
```

```
FROM python:3.6.7
pip install flask gunicorn
COPY . .
```

```yaml
version: '3'

services:
  flask_app:
    container_name: flask_app
    restart: always
    build: ./flask_app
    ports:
      - "8000:8000"
    command: gunicorn -w 1 -b 0.0.0.0:8000 main:server

  nginx:
    container_name: nginx
    restart: always
    build: ./nginx
    ports:
      - "80:80"
    depends_on:
      - flask_app
```

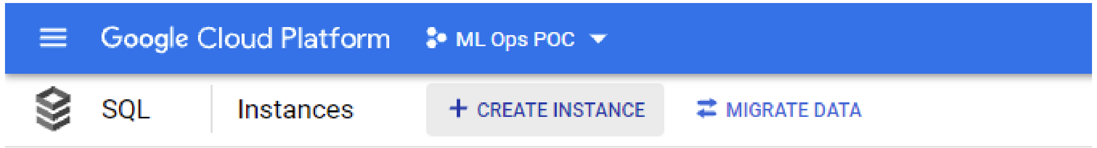
# Dataproc, BigQuery, and Apache Spark for ML

https://cloud.google.com/dataproc/docs/tutorials/bigquery-sparkml#spark-ml-tutorial_regression-console

# Dataproc, BigQuery, and Apache Spark for ML

## 1. Create Cloud MySQL instance

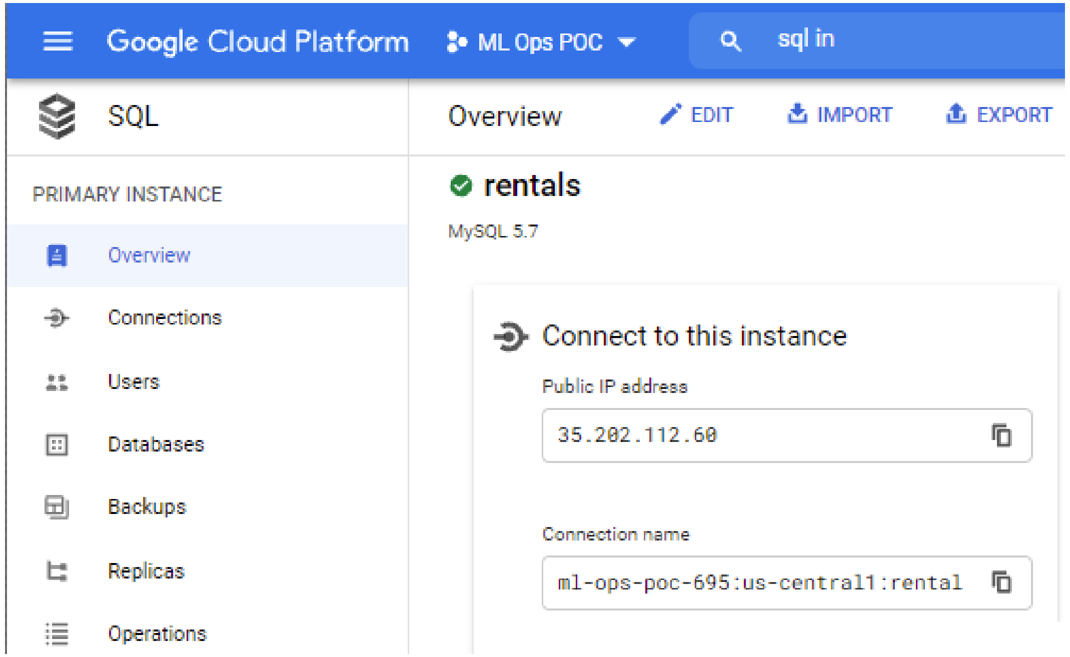
```
#sudo apt-get install mysql\*
#sudo gcloud sql connect rentals
```



# Dataproc, BigQuery, and Apache Spark for ML

## 2. Create Tables in Cloud MySQL

```
MySQL [(none)]> CREATE DATABASE IF NOT EXISTS
recommendation_spark;
DROP TABLE IF EXISTS Accommodation;
CREATE TABLE IF NOT EXISTS Accommodation
(
  id varchar(255),
  title varchar(255),
  location varchar(255),
  price int,
  rooms int,
  rating float,
  type varchar(255),
  PRIMARY KEY (ID)
);
CREATE TABLE  IF NOT EXISTS Rating
(
  userId varchar(255),
  accoId varchar(255),
  rating int,
  PRIMARY KEY(accoId, userId),
  FOREIGN KEY (accoId)
    REFERENCES Accommodation(id)
);
CREATE TABLE  IF NOT EXISTS Recommendation
(
  userId varchar(255),
  accoId varchar(255),
  prediction float,
  PRIMARY KEY(userId, accoId),
  FOREIGN KEY (accoId)
    REFERENCES Accommodation(id)
);
```



rentals

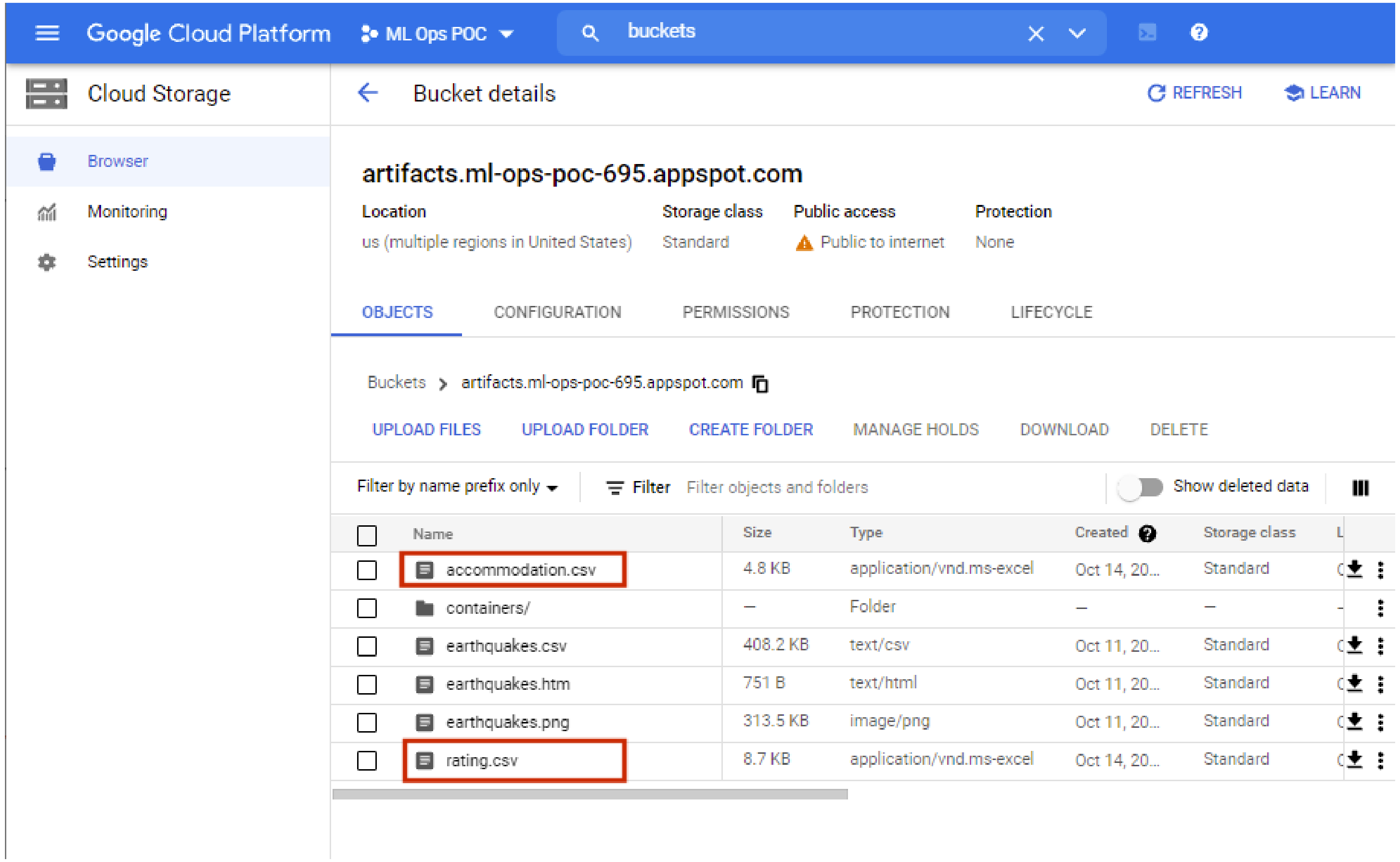
MySQL 5.7

Public IP address: 35.202.112.60

Connection name: ml-ops-poc-695:us-central1:rental

# Dataproc, BigQuery, and Apache Spark for ML

## 3. Export data to Cloud Storage Bucket



artifacts.ml-ops-poc-695.appspot.com

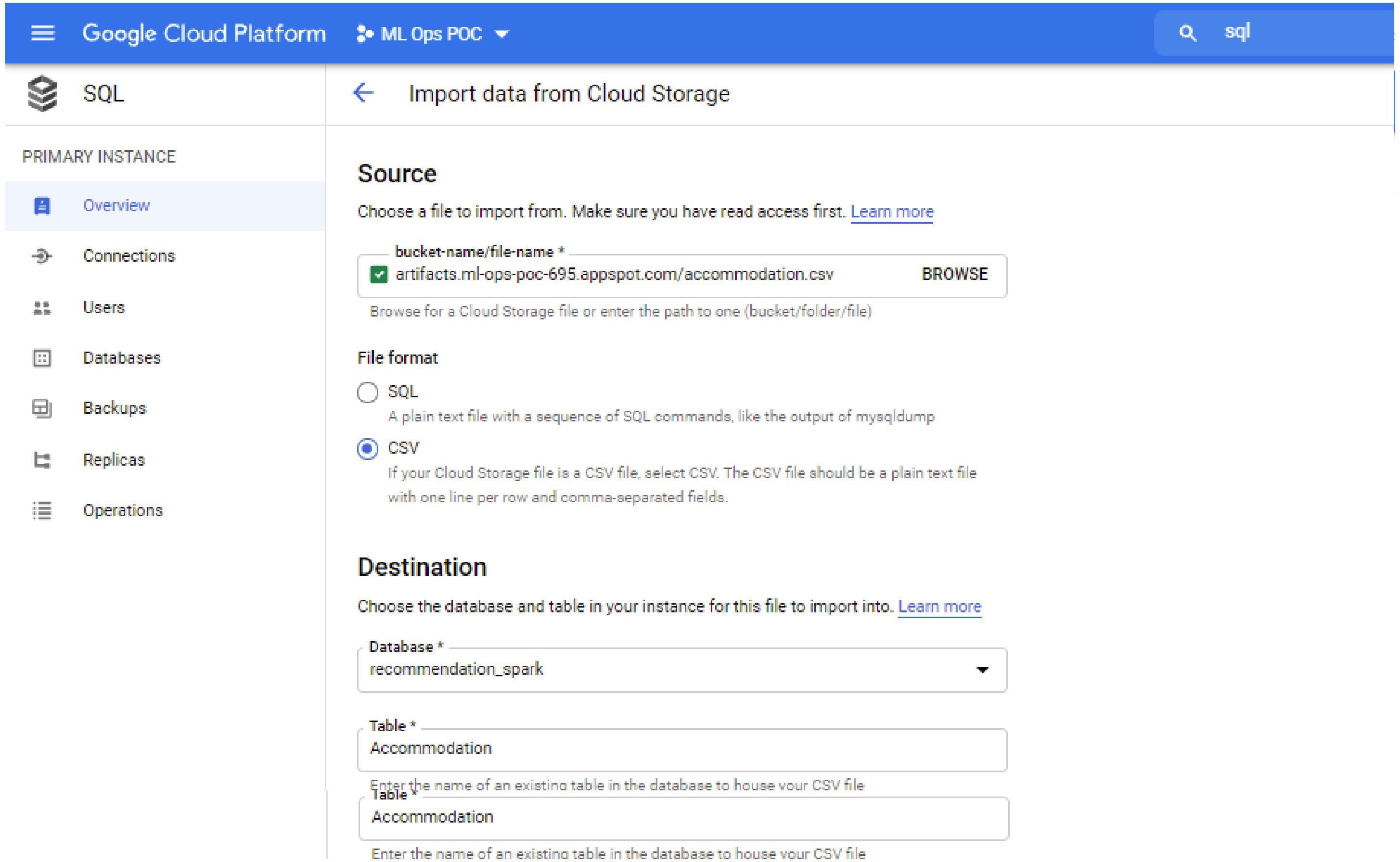
| Location | Storage class | Public access | Protection |
|---|---|---|---|
| us (multiple regions in United States) | Standard | ⚠ Public to internet | None |

Buckets > artifacts.ml-ops-poc-695.appspot.com

| □ | Name | Size | Type | Created | Storage class | L |
|---|---|---|---|---|---|---|
| □ | accommodation.csv | 4.8 KB | application/vnd.ms-excel | Oct 14, 20... | Standard | C |
| □ | containers/ | — | Folder | — | — | - |
| □ | earthquakes.csv | 408.2 KB | text/csv | Oct 11, 20... | Standard | C |
| □ | earthquakes.htm | 751 B | text/html | Oct 11, 20... | Standard | C |
| □ | earthquakes.png | 313.5 KB | image/png | Oct 11, 20... | Standard | C |
| □ | rating.csv | 8.7 KB | application/vnd.ms-excel | Oct 14, 20... | Standard | C |

# Dataproc, BigQuery, and Apache Spark for ML

## 4. Import data from Bucket to Cloud SQL



Import data from Cloud Storage

### Source

Choose a file to import from. Make sure you have read access first. Learn more

bucket-name/file-name *
☑ artifacts.ml-ops-poc-695.appspot.com/accommodation.csv BROWSE
Browse for a Cloud Storage file or enter the path to one (bucket/folder/file)

File format

○ SQL
A plain text file with a sequence of SQL commands, like the output of mysqldump

● CSV
If your Cloud Storage file is a CSV file, select CSV. The CSV file should be a plain text file with one line per row and comma-separated fields.

### Destination

Choose the database and table in your instance for this file to import into. Learn more

Database *
recommendation_spark

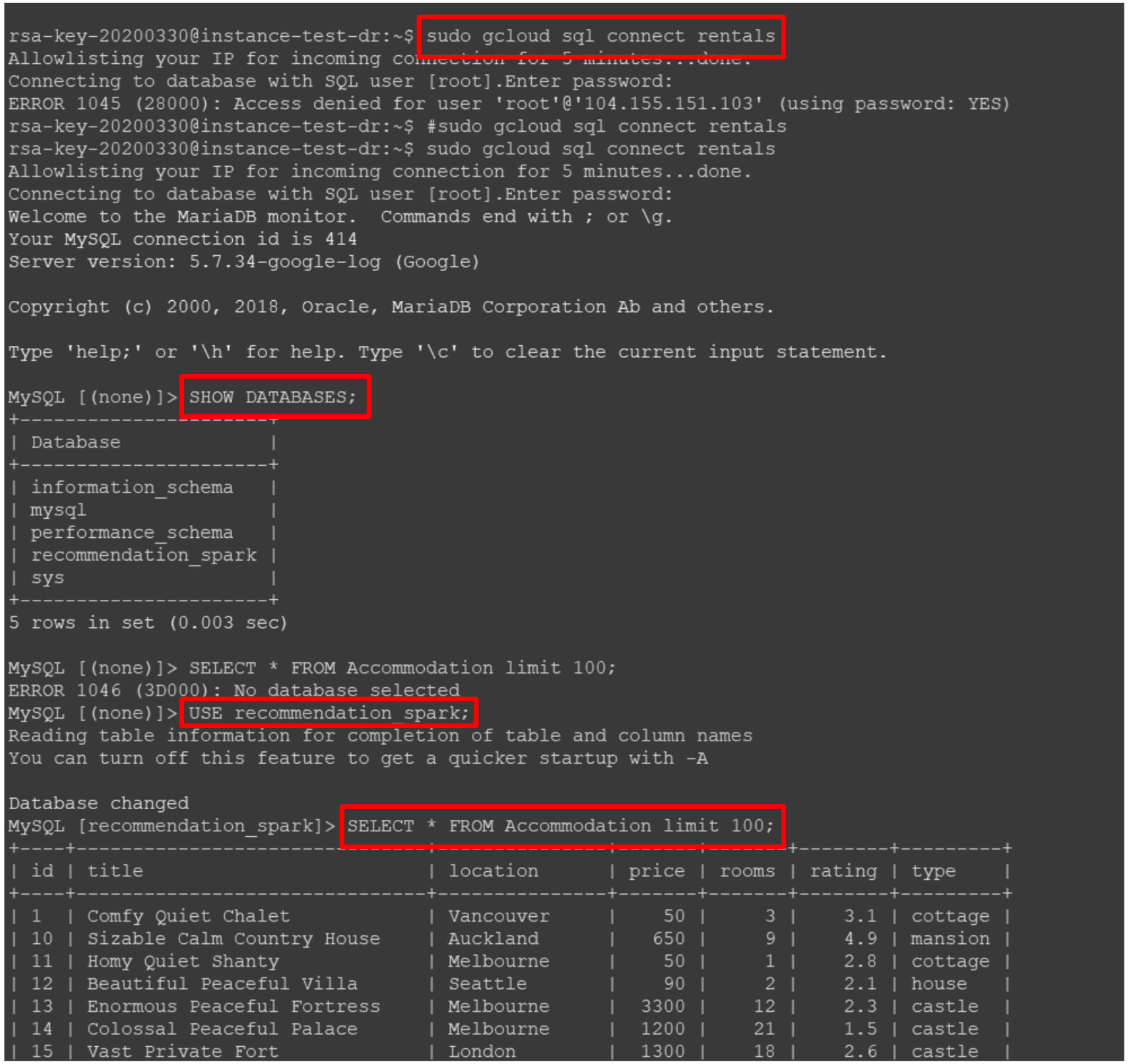
Table *
Accommodation
Enter the name of an existing table in the database to house your CSV file

Table *
Accommodation
Enter the name of an existing table in the database to house your CSV file

# Dataproc, BigQuery, and Apache Spark for ML

## 5. Explore Cloud SQL data

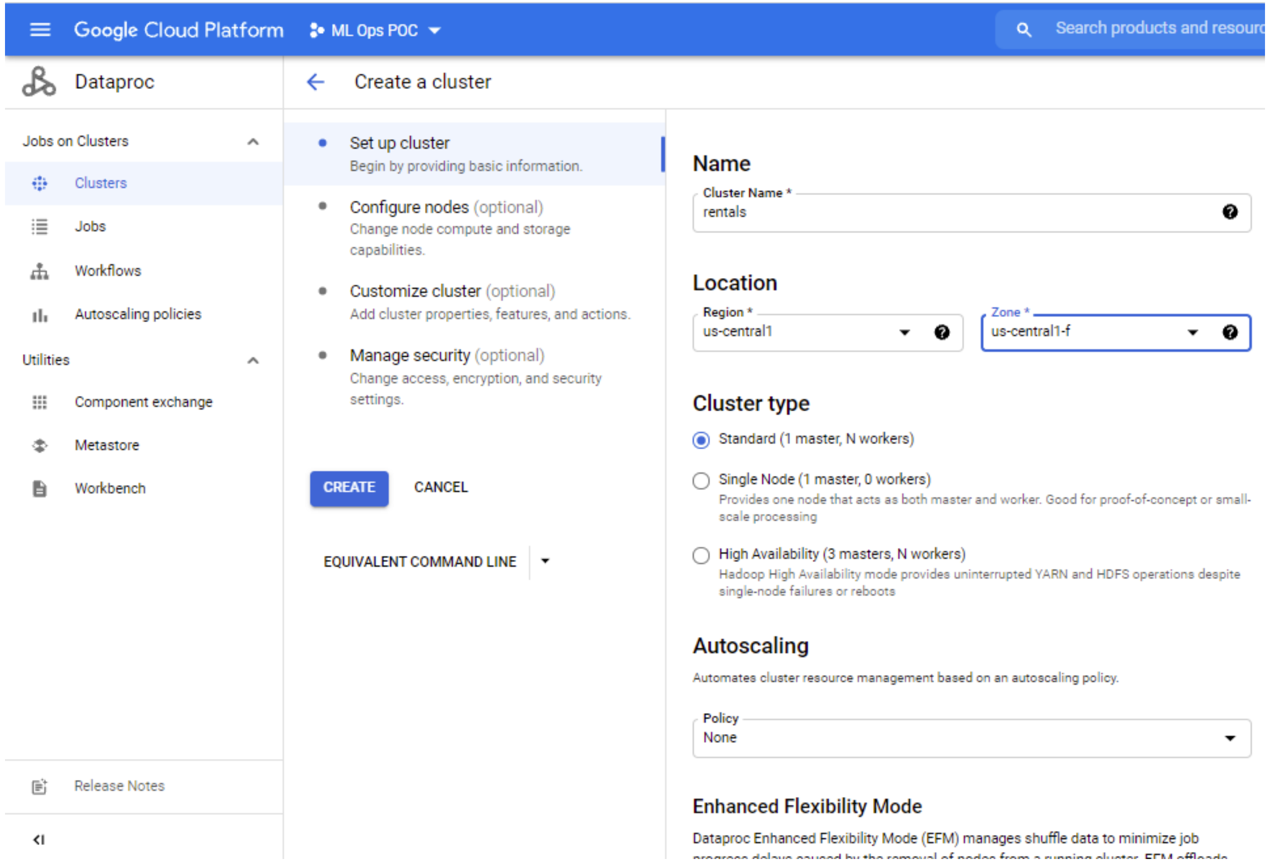
```
sudo gcloud sql connect rentals
SHOW DATABASES;
USE recommendation_spark;
SELECT * FROM Accommodation limit 100;
```



```
rsa-key-20200330@instance-test-dr:~$ sudo gcloud sql connect rentals
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'104.155.151.103' (using password: YES)
rsa-key-20200330@instance-test-dr:~$ #sudo gcloud sql connect rentals
rsa-key-20200330@instance-test-dr:~$ sudo gcloud sql connect rentals
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 414
Server version: 5.7.34-google-log (Google)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> SHOW DATABASES;
+----------------------+
| Database             |
+----------------------+
| information_schema   |
| mysql                |
| performance_schema   |
| recommendation_spark |
| sys                  |
+----------------------+
5 rows in set (0.003 sec)

MySQL [(none)]> SELECT * FROM Accommodation limit 100;
ERROR 1046 (3D000): No database selected
MySQL [(none)]> USE recommendation_spark;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [recommendation_spark]> SELECT * FROM Accommodation limit 100;
+----+--------------------------------+---------------+-------+-------+--------+---------+
| id | title                          | location      | price | rooms | rating | type    |
+----+--------------------------------+---------------+-------+-------+--------+---------+
| 1  | Comfy Quiet Chalet             | Vancouver     |    50 |     3 |    3.1 | cottage |
| 10 | Sizable Calm Country House     | Auckland      |   650 |     9 |    4.9 | mansion |
| 11 | Homy Quiet Shanty              | Melbourne     |    50 |     1 |    2.8 | cottage |
| 12 | Beautiful Peaceful Villa       | Seattle       |    90 |     2 |    2.1 | house   |
| 13 | Enormous Peaceful Fortress     | Melbourne     |  3300 |    12 |    2.3 | castle  |
| 14 | Colossal Peaceful Palace       | Melbourne     |  1200 |    21 |    1.5 | castle  |
| 15 | Vast Private Fort              | London        |  1300 |    18 |    2.6 | castle  |
```

# Dataproc, BigQuery, and Apache Spark for ML

## 6. Dataproc cluster setup



Create a cluster

- Set up cluster
  Begin by providing basic information.
- Configure nodes (optional)
  Change node compute and storage capabilities.
- Customize cluster (optional)
  Add cluster properties, features, and actions.
- Manage security (optional)
  Change access, encryption, and security settings.

**Name**

Cluster Name *: rentals

**Location**

Region *: us-central1

Zone *: us-central1-f

**Cluster type**

● Standard (1 master, N workers)

○ Single Node (1 master, 0 workers)
Provides one node that acts as both master and worker. Good for proof-of-concept or small-scale processing

○ High Availability (3 masters, N workers)
Hadoop High Availability mode provides uninterrupted YARN and HDFS operations despite single-node failures or reboots

**Autoscaling**

Automates cluster resource management based on an autoscaling policy.

Policy: None

**Enhanced Flexibility Mode**

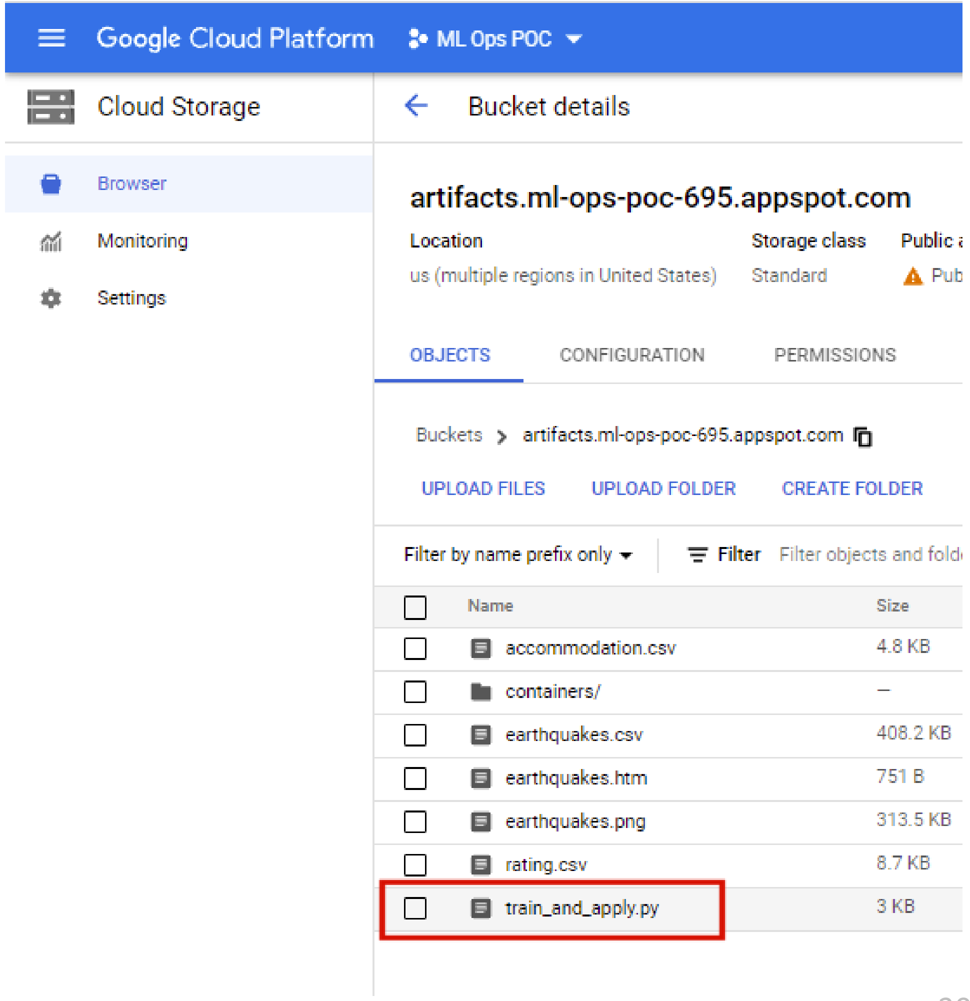
Dataproc Enhanced Flexibility Mode (EFM) manages shuffle data to minimize job progress delays caused by the removal of nodes from a running cluster. EFM offloads

```
echo "Authorizing Cloud Dataproc to connect with Cloud SQL"
CLUSTER=rentals
CLOUDSQL=rentals
ZONE=us-central1-f
NWORKERS=2
machines="$CLUSTER-m"
for w in `seq 0 $(($NWORKERS - 1))`; do
   machines="$machines $CLUSTER-w-$w"
done
echo "Machines to authorize: $machines in $ZONE ... finding
their IP addresses"
ips=""
for machine in $machines; do
    IP_ADDRESS=$(gcloud compute instances describe $machine -
-zone=$ZONE --
format='value(networkInterfaces.accessConfigs[].natIP)' | sed
"s/\['//g" | sed "s/'\]//g" )/32
    echo "IP address of $machine is $IP_ADDRESS"
    if [ -z  $ips ]; then
       ips=$IP_ADDRESS
    else
       ips="$ips,$IP_ADDRESS"
    fi
done
echo "Authorizing [$ips] to access cloudsql=$CLOUDSQL"
gcloud sql instances patch $CLOUDSQL --authorized-networks
$ips
```

# Dataproc, BigQuery, and Apache Spark for ML

## 7. Prepare PySpark script at GS

```
gsutil cp gs://cloud-training/bdml/v2.0/model/train_and_apply.py train_and_apply.py

#patch file with credentials
#cloudshell edit train_and_apply.py

gsutil cp train_and_apply.py gs://$DEVSHELL_PROJECT_ID
```



artifacts.ml-ops-poc-695.appspot.com

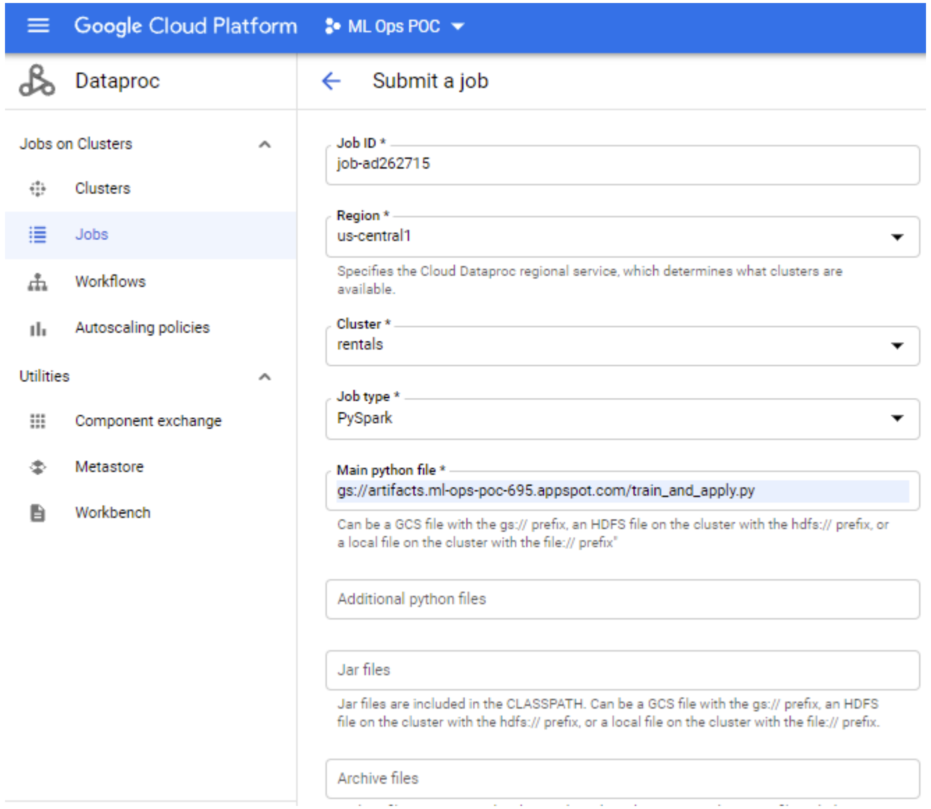
| Location | Storage class | Public a |
|---|---|---|
| us (multiple regions in United States) | Standard | ⚠ Pub |

Buckets > artifacts.ml-ops-poc-695.appspot.com

| | Name | Size |
|---|---|---|
| □ | accommodation.csv | 4.8 KB |
| □ | containers/ | – |
| □ | earthquakes.csv | 408.2 KB |
| □ | earthquakes.htm | 751 B |
| □ | earthquakes.png | 313.5 KB |
| □ | rating.csv | 8.7 KB |
| □ | train_and_apply.py | 3 KB |

# Dataproc, BigQuery, and Apache Spark for ML

## 8. Run PySpak script



Submit a job

Job ID *
job-ad262715

Region *
us-central1

Specifies the Cloud Dataproc regional service, which determines what clusters are available.

Cluster *
rentals

Job type *
PySpark

Main python file *
gs://artifacts.ml-ops-poc-695.appspot.com/train_and_apply.py

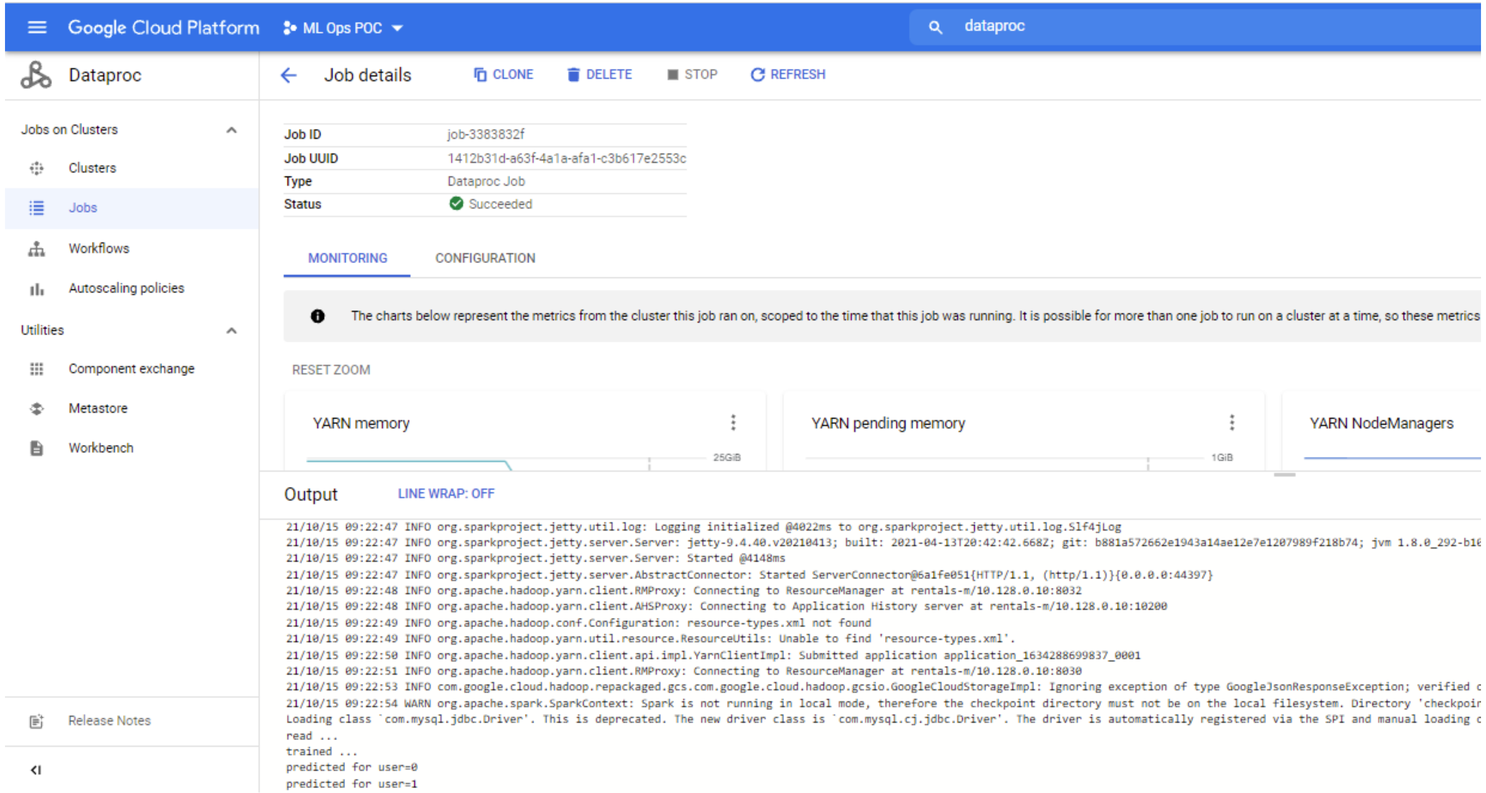
Can be a GCS file with the gs:// prefix, an HDFS file on the cluster with the hdfs:// prefix, or a local file on the cluster with the file:// prefix"

Additional python files

Jar files

Jar files are included in the CLASSPATH. Can be a GCS file with the gs:// prefix, an HDFS file on the cluster with the hdfs:// prefix, or a local file on the cluster with the file:// prefix.

Archive files



Job details

| | |
|---|---|
| Job ID | job-3383832f |
| Job UUID | 1412b31d-a63f-4a1a-afa1-c3b617e2553c |
| Type | Dataproc Job |
| Status | ✓ Succeeded |

The charts below represent the metrics from the cluster this job ran on, scoped to the time that this job was running. It is possible for more than one job to run on a cluster at a time, so these metrics

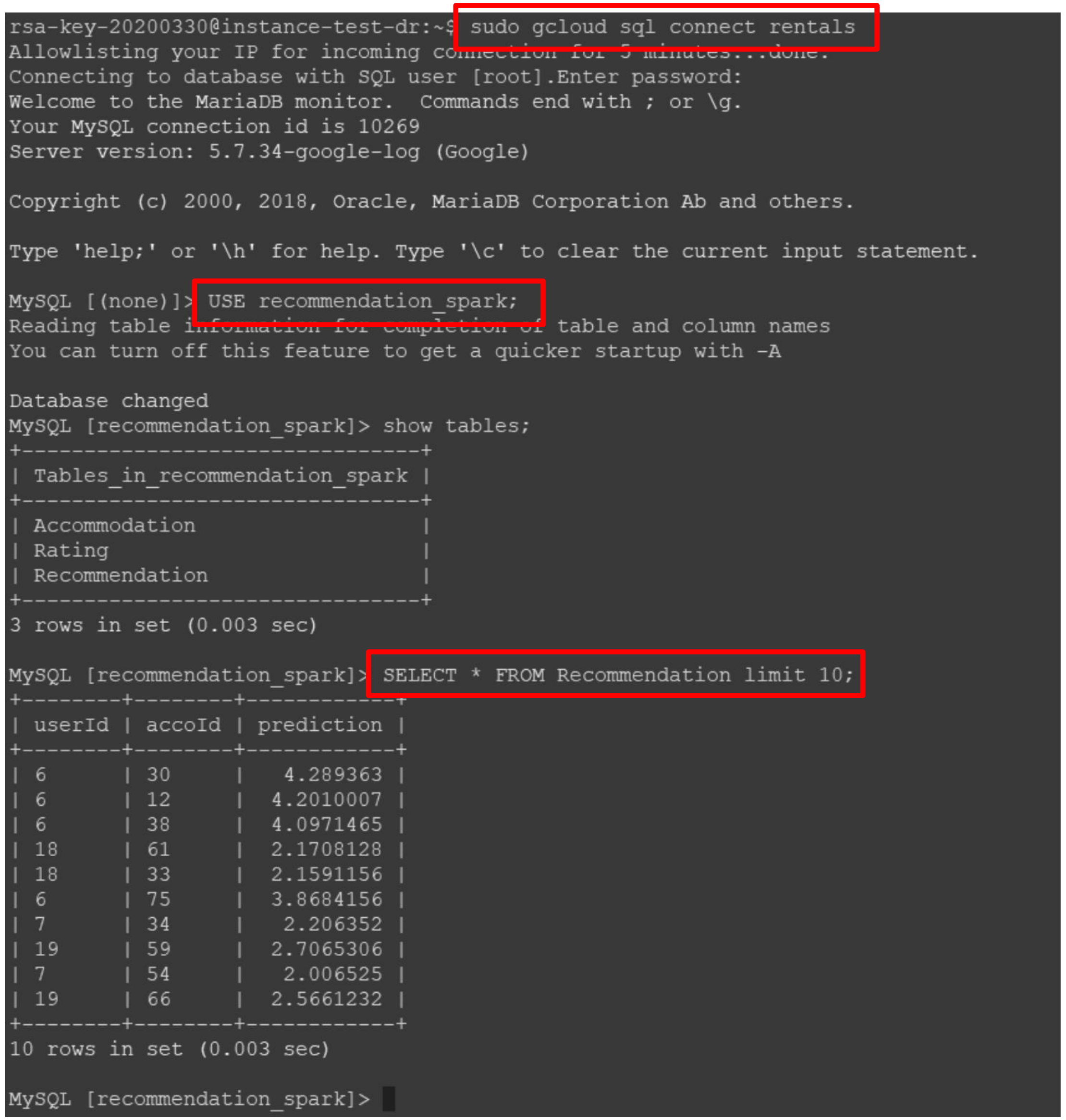
Output

```
21/10/15 09:22:47 INFO org.sparkproject.jetty.util.log: Logging initialized @4022ms to org.sparkproject.jetty.util.log.Slf4jLog
21/10/15 09:22:47 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; built: 2021-04-13T20:42:42.668Z; git: b881a572662e1943a14ae12e7e1207989f218b74; jvm 1.8.0_292-b1
21/10/15 09:22:47 INFO org.sparkproject.jetty.server.Server: Started @4148ms
21/10/15 09:22:47 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerConnector@6a1fe051{HTTP/1.1, (http/1.1)}{0.0.0.0:44397}
21/10/15 09:22:48 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at rentals-m/10.128.0.10:8032
21/10/15 09:22:48 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at rentals-m/10.128.0.10:10200
21/10/15 09:22:49 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
21/10/15 09:22:49 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
21/10/15 09:22:50 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1634288699837_0001
21/10/15 09:22:51 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at rentals-m/10.128.0.10:8030
21/10/15 09:22:53 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified c
21/10/15 09:22:54 WARN org.apache.spark.SparkContext: Spark is not running in local mode, therefore the checkpoint directory must not be on the local filesystem. Directory 'checkpoi
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading c
read ...
trained ...
predicted for user=0
predicted for user=1
```

# Dataproc, BigQuery, and Apache Spark for ML

## 9. Explore results

```
sudo gcloud sql connect rentals
SHOW DATABASES;
USE recommendation_spark;
SELECT * FROM Recommendation limit 10;
```

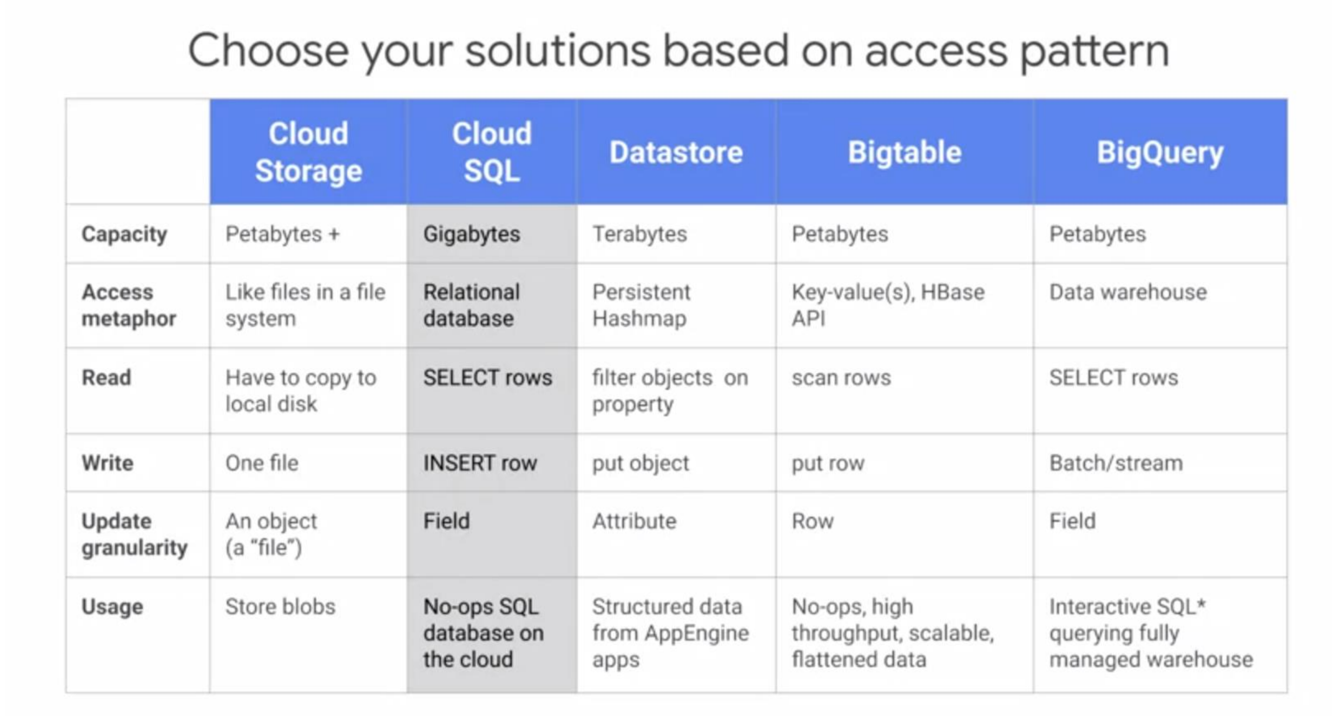


```
rsa-key-20200330@instance-test-dr:~$ sudo gcloud sql connect rentals
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 10269
Server version: 5.7.34-google-log (Google)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> USE recommendation_spark;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [recommendation_spark]> show tables;
+-------------------------------+
| Tables_in_recommendation_spark |
+-------------------------------+
| Accommodation                 |
| Rating                        |
| Recommendation                |
+-------------------------------+
3 rows in set (0.003 sec)

MySQL [recommendation_spark]> SELECT * FROM Recommendation limit 10;
+--------+--------+------------+
| userId | accoId | prediction |
+--------+--------+------------+
| 6      | 30     |   4.289363 |
| 6      | 12     |  4.2010007 |
| 6      | 38     |  4.0971465 |
| 18     | 61     |  2.1708128 |
| 18     | 33     |  2.1591156 |
| 6      | 75     |  3.8684156 |
| 7      | 34     |   2.206352 |
| 19     | 59     |  2.7065306 |
| 7      | 54     |   2.006525 |
| 19     | 66     |  2.5661232 |
+--------+--------+------------+
10 rows in set (0.003 sec)

MySQL [recommendation_spark]>
```

# References

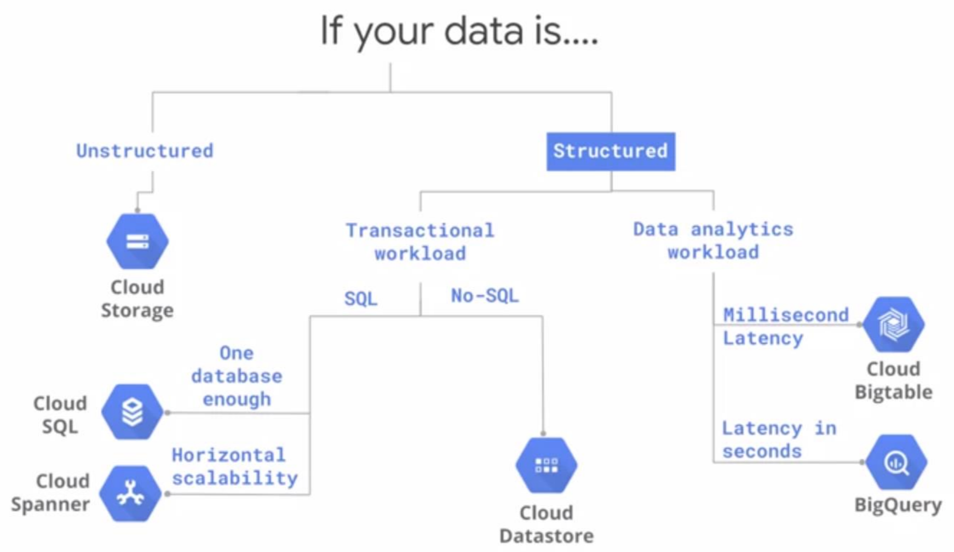
https://cloud.google.com/dataproc/docs/tutorials/bigquery-sparkml#spark-ml-tutorial_regression-console

# Dataproc, BigQuery, and Apache Spark for ML

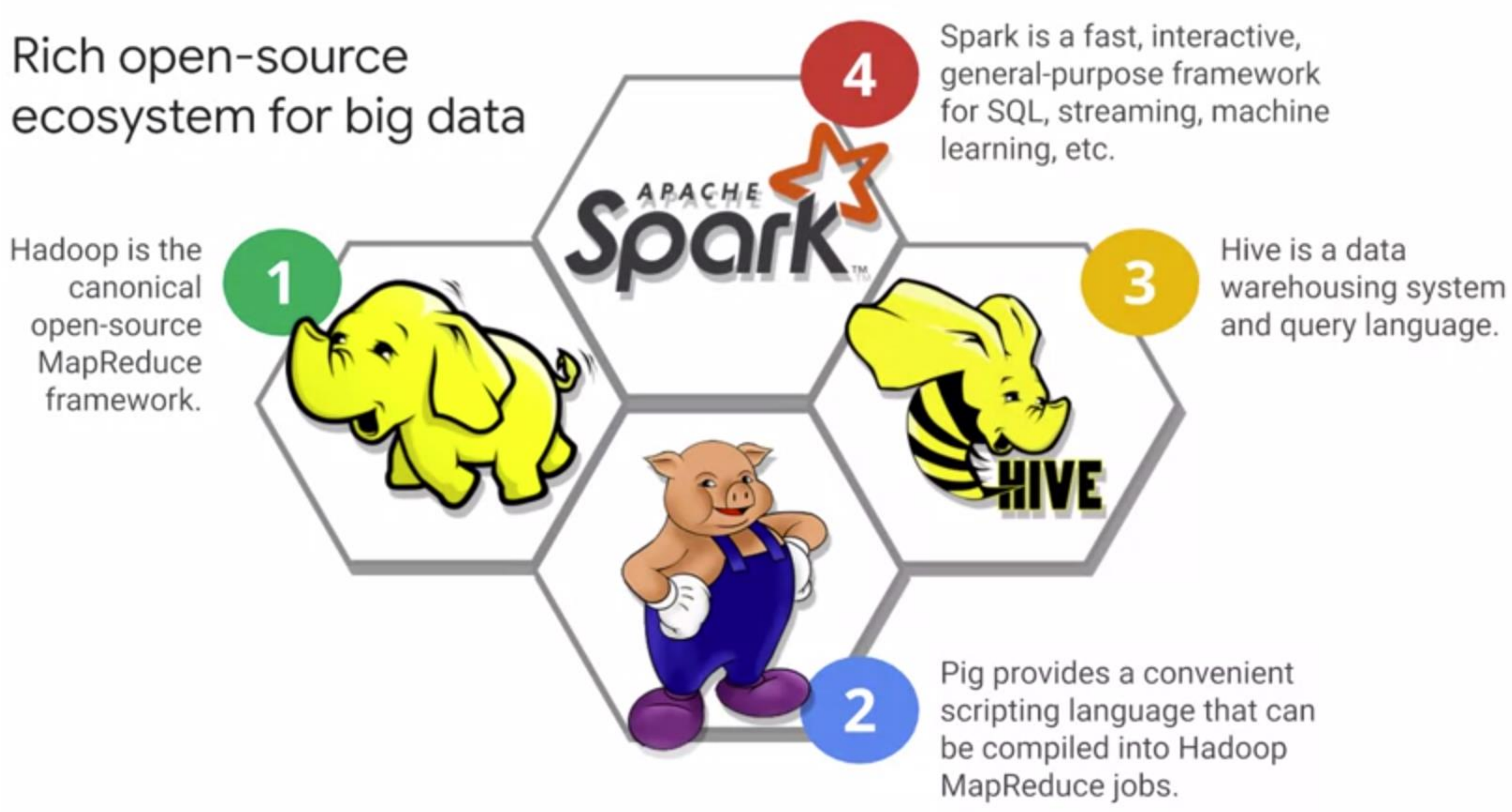


## Choose your solutions based on access pattern

| | Cloud Storage | Cloud SQL | Datastore | Bigtable | BigQuery |
|---|---|---|---|---|---|
| **Capacity** | Petabytes + | Gigabytes | Terabytes | Petabytes | Petabytes |
| **Access metaphor** | Like files in a file system | Relational database | Persistent Hashmap | Key-value(s), HBase API | Data warehouse |
| **Read** | Have to copy to local disk | SELECT rows | filter objects on property | scan rows | SELECT rows |
| **Write** | One file | INSERT row | put object | put row | Batch/stream |
| **Update granularity** | An object (a "file") | Field | Attribute | Row | Field |
| **Usage** | Store blobs | No-ops SQL database on the cloud | Structured data from AppEngine apps | No-ops, high throughput, scalable, flattened data | Interactive SQL* querying fully managed warehouse |



# Dataproc, BigQuery, and Apache Spark for ML



## Rich open-source ecosystem for big data

1 Hadoop is the canonical open-source MapReduce framework.

2 Pig provides a convenient scripting language that can be compiled into Hadoop MapReduce jobs.

3 Hive is a data warehousing system and query language.

4 Spark is a fast, interactive, general-purpose framework for SQL, streaming, machine learning, etc.