

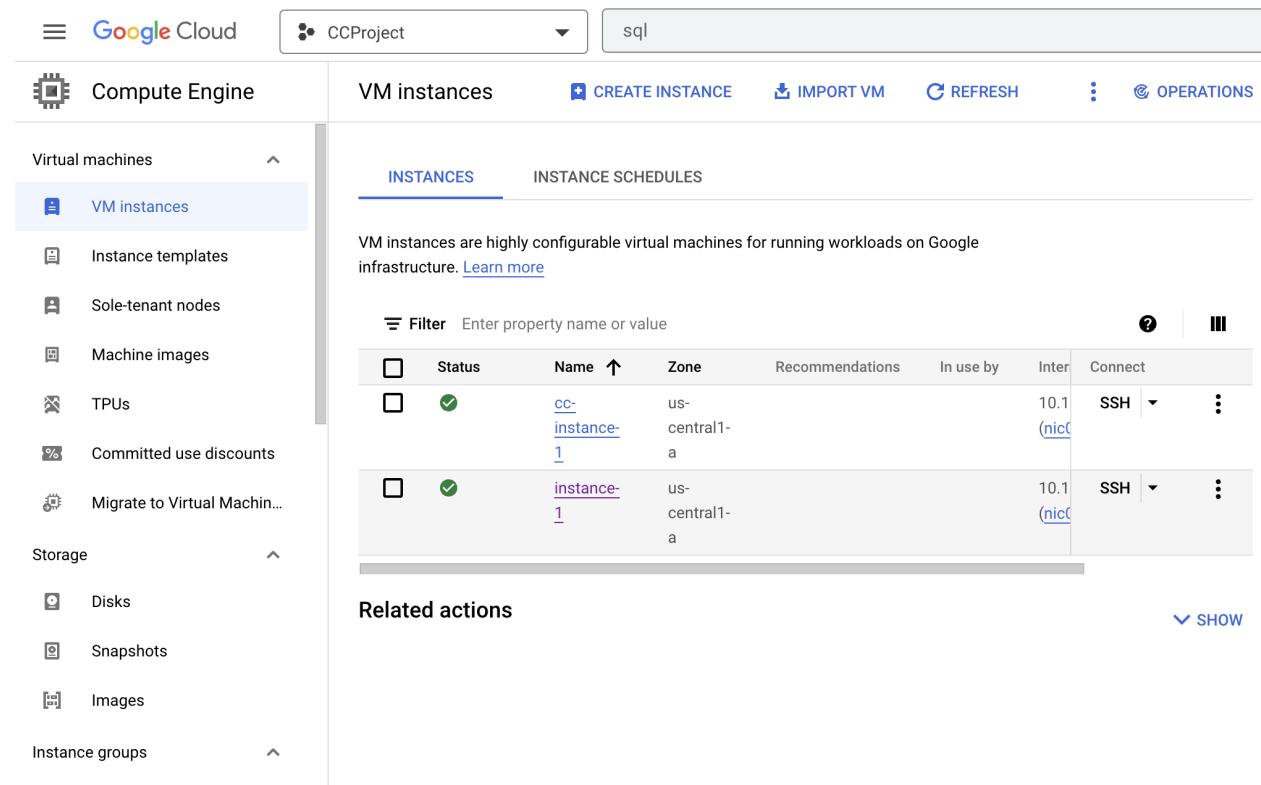
Cloud Computing - CSE 5333-001

Mid Term

Group -19
Priyank Gupta - 1001629364
Ravi Rajpurohit - 1002079916
Vedanti Ambulkar - 1001829121

As proposed, we are creating a web application to let the users sign up and login to the media collection for being able to upload the images and see the already available images.

1. Create a Virtual Machine instance



The screenshot shows the Google Cloud Compute Engine VM instances page. The left sidebar has sections for Virtual machines (VM instances selected), Storage (Disks, Snapshots, Images), and Instance groups. The main area shows a table of VM instances:

Status	Name	Zone	Recommendations	In use by	IP Address	Connect
Up	cc-instance-1	us-central1-a			10.1.24.123 (nic0)	SSH
Up	instance-1	us-central1-a			10.1.24.124 (nic0)	SSH

Related actions are shown at the bottom right.

We created a VM instance named instance-1 with the required configuration.

The screenshot shows the Google Cloud Compute Engine interface for managing VM instances. On the left, a sidebar lists categories like Virtual machines, Storage, and Instance groups. The main panel displays details for a VM named 'instance-1'. The 'DETAILS' tab is selected, showing basic information such as Name (instance-1), Instance Id (1727972578816024482), and Status (Running). It also includes sections for Logs, Basic information, and Firewall rules.

The details of the VM instance configuration are mentioned above.

This screenshot continues from the previous one, showing more detailed configuration for the same VM instance. The 'DETAILS' tab is still active. The 'Network interfaces' section shows a single interface named 'nic0' connected to the 'default' network. The 'Storage' section shows a 'Boot disk' named 'instance-1' with a size of 10 GB, using the 'ubuntu-1804-bionic-v20221018' image, and is connected via SCSI. There are also sections for 'Local disks' (None) and 'Additional disks' (None).

To mention a few, it is an Ubuntu-based instance with 10 GB ROM and x86 architecture.

2. Create a Cloud SQL instance

We created a MySQL instance named **cloud-project** for storing user credentials in database.

The screenshot shows the Google Cloud SQL Instances page. At the top, there are navigation links for Google Cloud, project selection, search bar, and user profile. Below the header, there are tabs for SQL and Instances, along with buttons for CREATE INSTANCE and MIGRATE DATA. A help assistant and info panel are also available. The main area displays a table of instances:

Instance ID	Type	Public IP address	Private IP address	Instance connection name	High availability	Location	Storage used	Labels	Actions
cloud-project	MySQL 8.0	34.67.38.201		ccproject-5333:us-central1:b	ENABLED	us-central1-b	1 GB of 100 GB		⋮
instance1	MySQL 8.0	35.225.27.12		ccproject-5333:us-central1:b	ADD	us-central1-b	1 GB of 100 GB		⋮
logindetails	MySQL 8.0	23.251.144.200		ccproject-5333:us-central1:a	ENABLED	us-central1-a	1 GB of 100 GB		⋮

The details of SQL instance are mentioned below.

The screenshot shows the Google Cloud SQL Overview page for the **cloud-project** instance. The left sidebar includes links for Overview, Query insights (NEW), Connections, Users, Databases, Backups, Replicas, and Operations. The main content area has tabs for Overview, EDIT, IMPORT, EXPORT, RESTART, STOP, DELETE, CLONE, and FAILOVER, along with a HELP ASSISTANT button. The Overview tab displays the following information:

- PRIMARY INSTANCE**: All instances > **cloud-project**, MySQL 8.0.
- Chart**: CPU utilization over the last 1 day, showing a fluctuating line between 1% and 2%.
- Go to Query insights for more in-depth info on queries and performance**.
- Connect to this instance**: Public IP address: 34.67.38.201, Connection name: ccproject-5333:us-central1:b:cloud-project.
- Configuration**: vCPUs: 4, Memory: 26 GB, SSD storage: 100 GB.
- Release Notes**.

The SQL database has to be configured for some parameters to allow the storage of data. In the following screenshots are the details of database parameters.

This screenshot shows the 'Connections' section of the Google Cloud SQL interface. On the left, a sidebar lists options like Overview, Query insights, Connections (which is selected), Users, Databases, Backups, Replicas, and Operations. The main area is titled 'Connections' and contains instructions for defining which networks can connect. It offers two options: 'Private IP' (disabled) and 'Public IP' (selected). A note says 'You can use the Cloud SQL Proxy for extra security with either option.' Below this is a section for 'Authorized networks' with a note about specifying CIDR ranges. A warning message states: 'You have added 0.0.0.0/0 as an allowed network. This prefix will allow any IPv4 client to pass the network firewall and make login attempts to your instance, including clients you did not intend to allow. Clients still need valid credentials to successfully log in to your instance.' A dropdown menu shows 'testing (0.0.0.0/0)' with an 'ADD NETWORK' button. At the bottom, there are 'SAVE' and 'DISCARD CHANGES' buttons.

The following users are made to test the database connection.

This screenshot shows the 'Users' section of the Google Cloud SQL interface. The sidebar shows 'Connections' is selected. The main area displays a list of users for the 'cloud-project' instance. It includes a header row with columns for User name, Host name, Authentication, and Password status. Two users are listed: 'ravi' and 'root'. Both users have '% (any host)' as their host name and 'Built-in' as their authentication method. The password status is 'N/A' for both. There is a '+ ADD USER ACCOUNT' button at the top of the user list table.

Created a database object called **cc** to store the user credentials.

The screenshot shows the Google Cloud SQL Databases page for a primary instance named 'cloud-project'. The 'Databases' section lists several databases, including 'cc' (which is circled in red), 'information_schema', 'mysql', 'performance_schema', and 'sys'. The 'cc' database is highlighted with a red circle. The page includes a sidebar with links for Overview, Query insights, Connections, Users, Databases (which is selected and highlighted in blue), Backups, Replicas, and Operations. A 'CREATE DATABASE' button is visible at the top right of the database list. The bottom of the page has a 'Release Notes' section and a navigation bar.

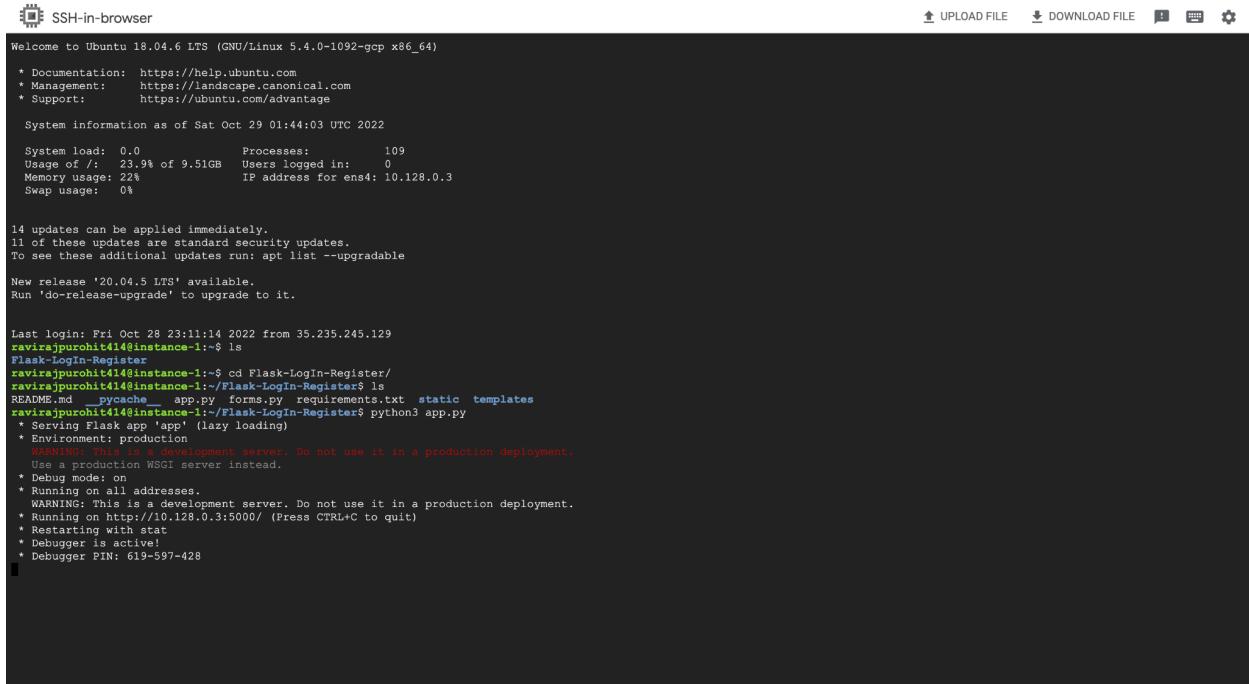
3. Allow incoming traffic for the web application hosted on VM

Add a firewall rule to allow users to call the instance.

The screenshot shows the Google Cloud Firewall rules page for a VPC network. The 'Firewall' section is selected. It displays a list of existing firewall rules, including 'default-allow-http', 'default-allow-https', 'flask', 'default-allow-internal', 'default-allow-rdp', and 'default-allow-icmp'. A new rule, 'flask', was just created and is highlighted with a red oval. A success message box is overlaid on the table, stating 'Successfully created firewall rule "flask."'. The sidebar on the left lists VPC networks, IP addresses, Bring your own IP, Firewall (selected and highlighted in blue), Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring. The top navigation bar shows the project name 'CCProject' and a search bar.

4. Deploy the web application on VM instance

Add manual SSH key and add the web application on the virtual machine instance



```
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1092-gcp x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Sat Oct 29 01:44:03 UTC 2022

System load: 0.0          Processes:           109
Usage of /: 23.9% of 9.51GB  Users logged in: 0
Memory usage: 22%          IP address for ens4: 10.128.0.3
Swap usage: 0%

14 updates can be applied immediately.
11 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Oct 28 23:11:14 2022 from 35.235.245.129
ravirajpurushit@149instance-1:~$ ls
Flask-Login-Register/
ravirajpurushit@149instance-1:~/Flask-Login-Register$ ls
README.md  __pycache__  app.py  forms.py  requirements.txt  static  templates
ravirajpurushit@149instance-1:~/Flask-Login-Register$ python3 app.py
* Serving Flask app 'app' (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 all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on 0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 619-597-428
```

Running the deployed application on the VM instance opens the web application interface which allows users to either login or signup.



A user has to register first to be able to sign in and access the portal. The registration page asks for a username, an email address and a password and looks like below. And these information is getting stored in the CloudSQL in the backend.

The screenshot shows a web browser window with the URL `35.225.21.31:5000/register/`. The page title is "Register". The form fields are as follows:

- Ad**: A text input field containing "123".
- Username**: A text input field containing "ravi".
- Email**: A text input field containing "ravi@gmail.com".
- Password**: A text input field containing "....".
- Confirm Password**: A text input field containing "....".

Below the form is a blue "Register" button and a link "Login into existing account!".

Once registered, the interface lets you login to the application after verifying the user credentials from the database.

The screenshot shows a web browser window with the URL `35.225.21.31:5000`. The top navigation bar includes links for "Logout" and "Welcome, ravi".

The user information is stored in the database after encrypting the password. The table looks like below.

DBeaver 21.3.5 - user

Database Navigator Projects 34.67.38.201 - 34.67.38.201:3306 <jxb7386> <jxb7386> <jxb7386> vaccine address unit people <localhost> ... users accounts user

Enter a part of object name here

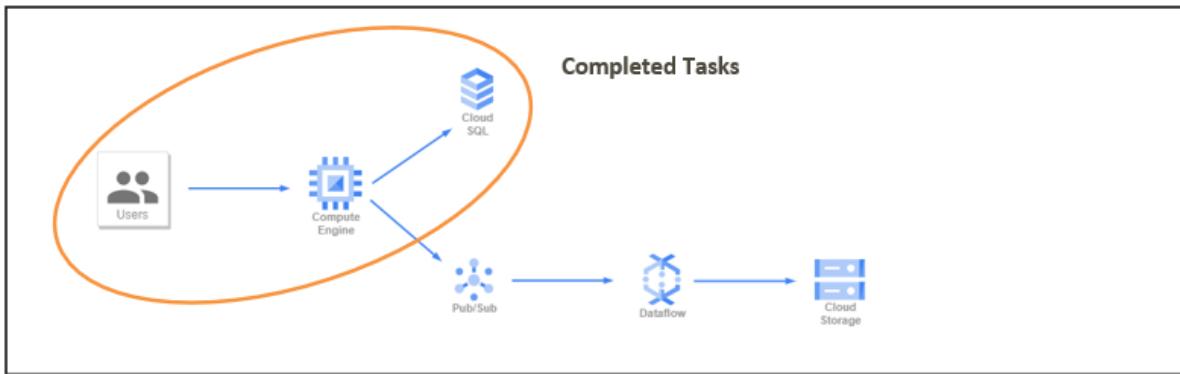
Databases cc Tables user

Grid Test

	id	name	username	email	password
1	1	Ravi	ravi	ravi@gmail.com	sha256\$pgsZ4zoJgrpR5qaT\$a6fac707d2c951be6ac5a770c9717681b40b7fd304ad36e9513c44406e6b27

Tasks Completed:

1. Developed a flask application for login and registration of users
2. Deployed the Login Web Page on the Compute Engine VM
3. Create the database in Cloud SQL using MySQL for Login Credentials
4. Successfully tested the application to verify if the user is able to register and sign in



Tasks Todo:

1. Develop a web API for uploading posters
2. Setting up a Pub/Sub topic to track the addition/deletion of objects
3. Create a Cloud Storage bucket for storing Posters/images
4. Create a dataflow template to transfer the image from the web application to the cloud storage bucket