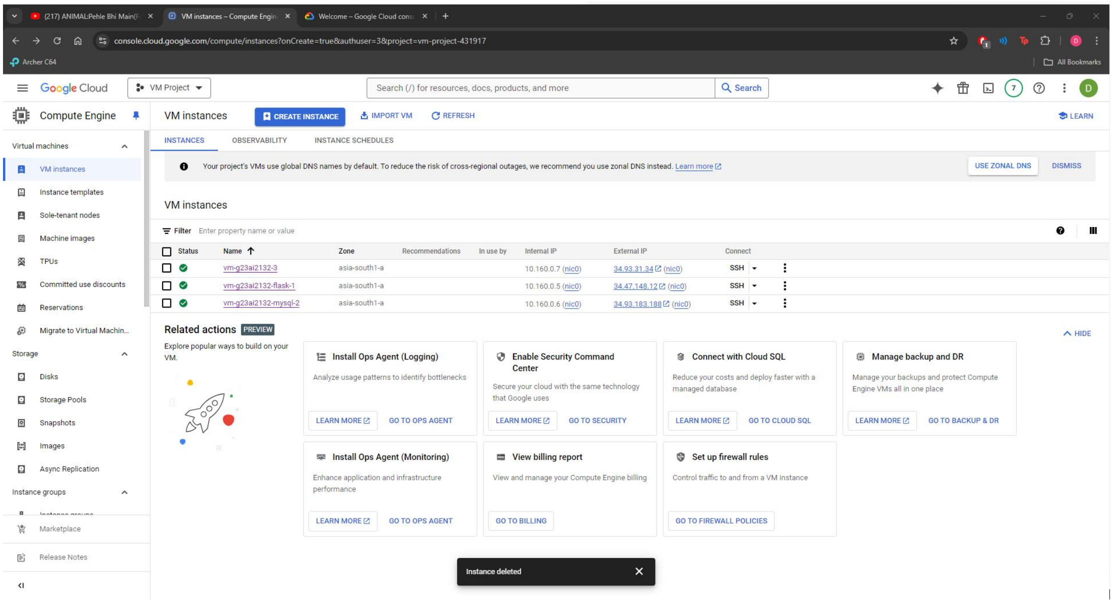
VCC ASSIGNMENT 2:-

Name:- Dhanshree Hood
Roll no: G23AI2132
Subject: Virtualization and Cloud Computing

To create a Flask-based REST API project using three Virtual Machines (VMs) on Google Cloud Platform (GCP):-

Set up three VMs on GCP:

VM1: This will host the Flask application.
VM2: This will host the MySQL database.
VM3: This will be used to test the API using Postman.
Configure the network settings so that VM1 can communicate with VM2 for database access and VM3 can access the API hosted on VM1.



Your project's VMs use global DNS names by default. To reduce the risk of cross-regional outages, we recommend you use zonal DNS instead. Learn more

VM instances

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
✓	vm-g23ai2132-3	asia-south1-a			10.160.0.7 (nic0)	34.93.31.34 (nic0)	SSH
✓	vm-g23ai2132-flask-1	asia-south1-a			10.160.0.5 (nic0)	34.47.148.12 (nic0)	SSH
✓	vm-g23ai2132-mysql-2	asia-south1-a			10.160.0.6 (nic0)	34.93.183.188 (nic0)	SSH

Related actions PREVIEW

Explore popular ways to build on your VM.

Install Ops Agent (Logging) — Analyze usage patterns to identify bottlenecks — LEARN MORE — GO TO OPS AGENT

Enable Security Command Center — Secure your cloud with the same technology that Google uses — LEARN MORE — GO TO SECURITY

Connect with Cloud SQL — Reduce your costs and deploy faster with a managed database — LEARN MORE — GO TO CLOUD SQL

Manage backup and DR — Manage your backups and protect Compute Engine VMs all in one place — LEARN MORE — GO TO BACKUP & DR

Install Ops Agent (Monitoring) — Enhance application and infrastructure performance — LEARN MORE — GO TO OPS AGENT

View billing report — View and manage your Compute Engine billing — GO TO BILLING

Set up firewall rules — Control traffic to and from a VM instance — GO TO FIREWALL POLICIES

Instance deleted

Deploy the Flask application on VM1.

Set up MySQL on VM2.

Test the API using Postman on VM3.

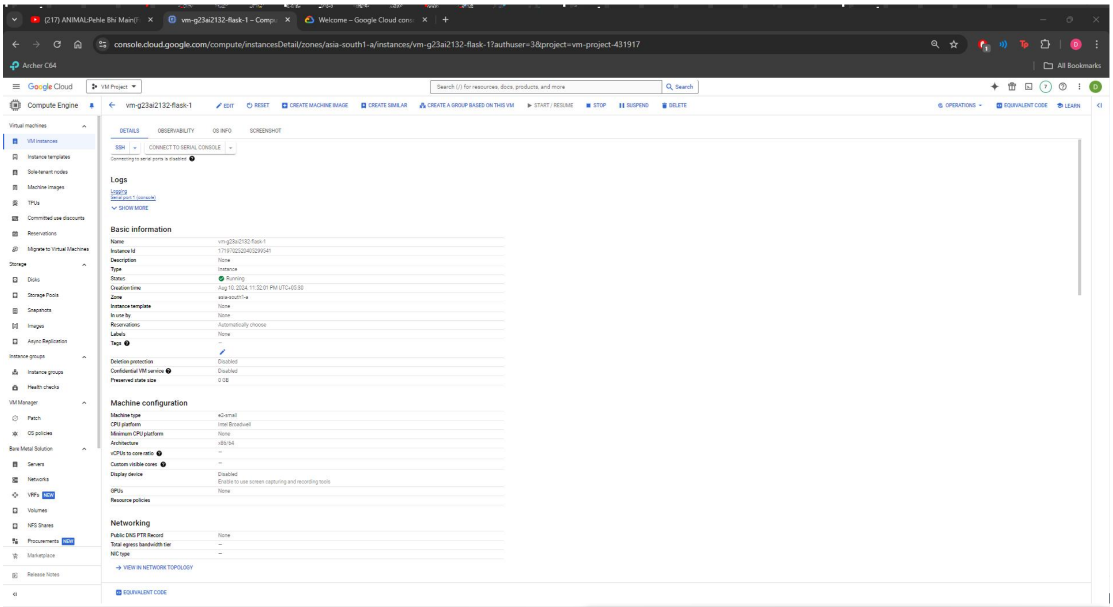
Step 1: Create Virtual Machines on GCP

We'll start by creating three VMs on Google Cloud Platform.

1.1. Create VM1 (Flask Application Server)

- ✓ Go to Google Cloud Console
- ✓ Create a New VM Instance:

- ✓ Navigate to Compute Engine > VM instances.
- ✓ Click Create Instance.
- ✓ Configure the instance:
- ✓ Name: vm-flask-app
- ✓ Region: Choose a region close to you (e.g., us-central1).
- ✓ Machine Type: n1-standard-1 (1 vCPU, 3.75 GB RAM).
- ✓ Boot Disk:
- ✓ OS: Ubuntu 22.04 LTS
- ✓ Size: 10 GB
- ✓ Firewall: Check Allow HTTP traffic.
- ✓ Click Create to launch the VM.



console.cloud.google.com/compute/instancesDetail/zones/asia-south1-a/instances/vm-g23ai2132-flask-1?authuser=3&project=vm-project-431917

Connecting to serial ports is disabled

Logs

Logging
Serial port 1 (console)

Basic information

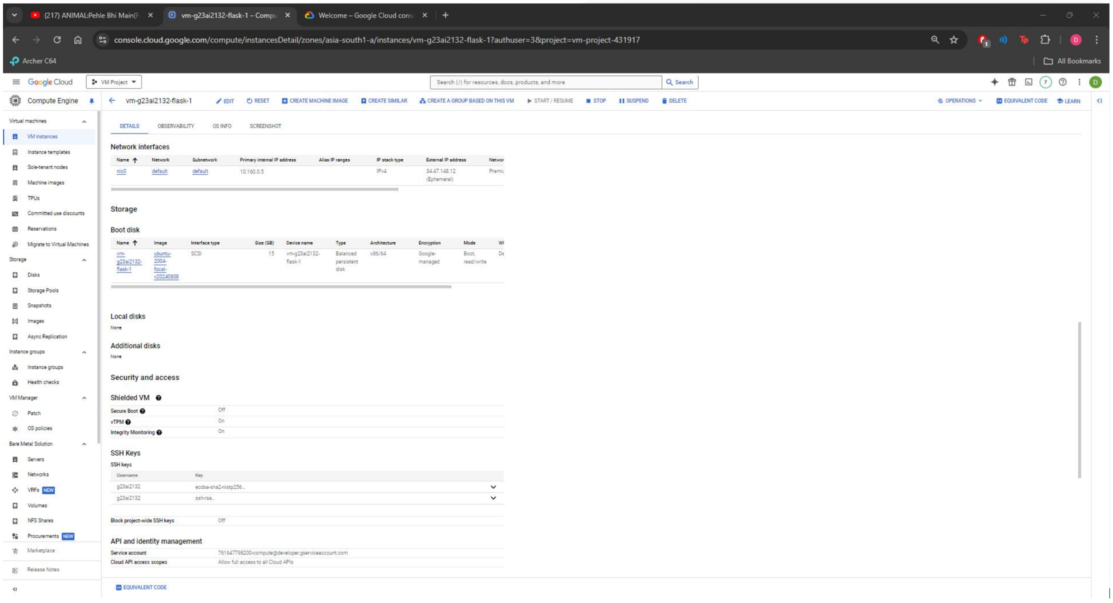
Name	vm-g23ai2132-flask-1
Instance Id	1719702520405299541
Description	None
Type	Instance
Status	Running
Creation time	Aug 10, 2024, 11:52:01 PM UTC+05:30
Zone	asia-south1-a
Instance template	None
In use by	None
Reservations	Automatically choose
Labels	None
Tags	–
Deletion protection	Disabled
Confidential VM service	Disabled
Preserved state size	0 GB

Machine configuration

Machine type	e2-small
CPU platform	Intel Broadwell
Minimum CPU platform	None
Architecture	x86/64
vCPUs to core ratio	–
Custom visible cores	–
Display device	Disabled Enable to use screen capturing and recording tools
GPUs	None
Resource policies	

Networking

Public DNS PTR Record	None
Total egress bandwidth tier	–
NIC type	–



console.cloud.google.com/compute/instancesDetail/zones/asia-south1-a/instances/vm-g23ai2132-flask-1?authuser=3&project=vm-project-431917

Network interfaces

Name	Network	Subnetwork	Primary internal IP address	Alias IP ranges	IP stack type	External IP address	Networ
nic0	default	default	10.160.0.5		IPv4	34.47.148.12 (Ephemeral)	Premiu

Storage

Boot disk

Name	Image	Interface type	Size (GB)	Device name	Type	Architecture	Encryption	Mode	Wi
vm-g23ai2132-flask-1	ubuntu-2004-focal-v20240808	SCSI	15	vm-g23ai2132-flask-1	Balanced persistent disk	x86/64	Google-managed	Boot, read/write	De

Local disks

None

Additional disks

None

Security and access

Shielded VM

Secure Boot	Off
vTPM	On
Integrity Monitoring	On

SSH Keys

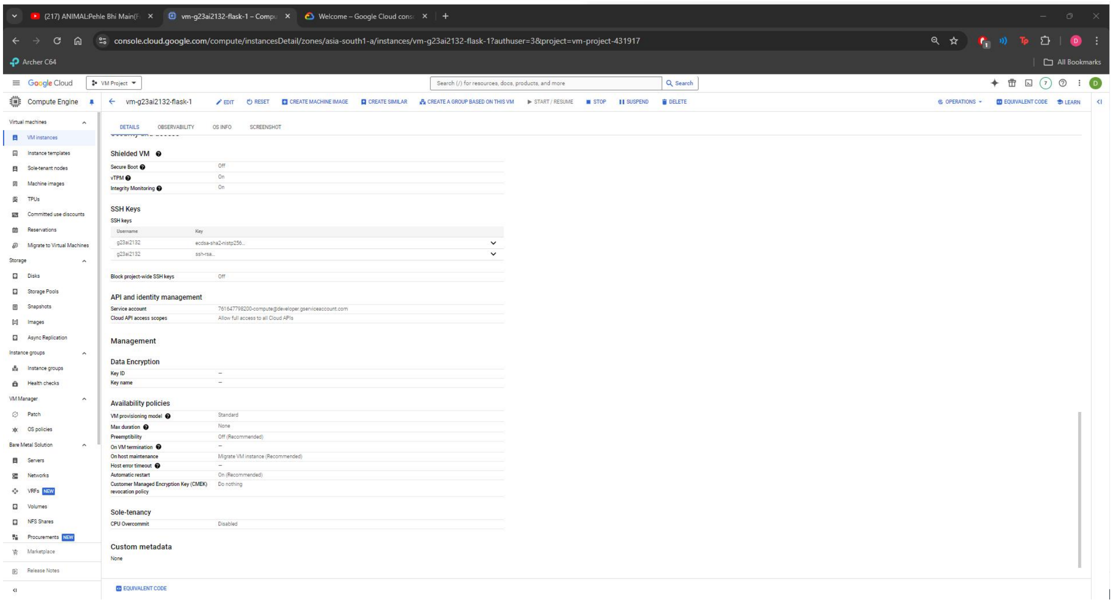
SSH keys

Username	Key
g23ai2132	ecdsa-sha2-nistp256...
g23ai2132	ssh-rsa...

Block project-wide SSH keys	Off

API and identity management

Service account	761647798200-compute@developer.gserviceaccount.com
Cloud API access scopes	Allow full access to all Cloud APIs



Shielded VM

Secure Boot	Off
vTPM	On
Integrity Monitoring	On

SSH Keys

SSH keys

Username	Key
g23ai2132	ecdsa-sha2-nistp256...
g23ai2132	ssh-rsa...

Block project-wide SSH keys	Off

API and identity management

Service account	761647798200-compute@developer.gserviceaccount.com
Cloud API access scopes	Allow full access to all Cloud APIs

Management

Data Encryption

Key ID	–
Key name	–

Availability policies

VM provisioning model	Standard
Max duration	None
Preemptibility	Off (Recommended)
On VM termination	–
On host maintenance	Migrate VM instance (Recommended)
Host error timeout	–
Automatic restart	On (Recommended)
Customer Managed Encryption Key (CMEK) revocation policy	Do nothing

Sole-tenancy

CPU Overcommit	Disabled

Custom metadata

None

1.2. Create VM2 (MySQL Database Server)

✓ Create a New VM Instance:

✓ Navigate to Compute Engine > VM instances.

✓ Click Create Instance.

✓ Configure the instance:

✓ Name: vm-mysql-db

✓ Region: Same as VM1.

✓ Machine Type: n1-standard-1 (1 vCPU, 3.75 GB RAM).

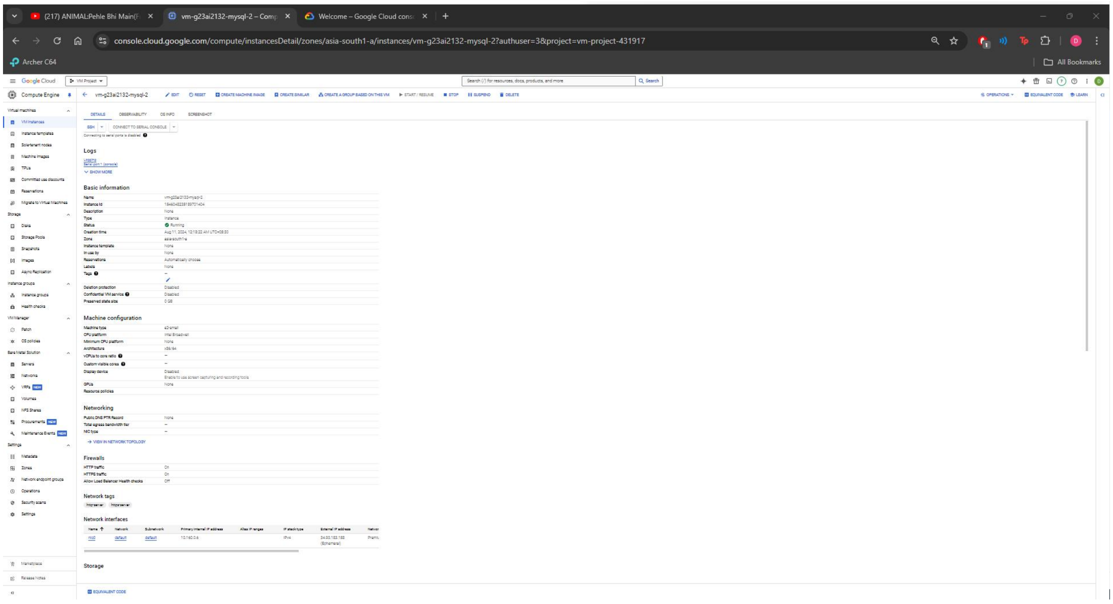
✓ Boot Disk:

✓ OS: Ubuntu 22.04 LTS

✓ Size: 10 GB

✓ Firewall: No special settings needed.

✓ Click Create to launch the VM.


(217) ANIMAL:Pehle Bhi Main(F
vm-g23ai2132-mysql-2 – Comp
Welcome – Google Cloud cons
console.cloud.google.com/compute/instancesDetail/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&project=vm-project-431917
Archer C64
All Bookmarks
Google Cloud
VM Project
Compute Engine
vm-g23ai2132-mysql-2
EDIT
RESET
CREATE MACHINE IMAGE
CREATE SIMILAR
CREATE A GROUP BASED ON THIS VM
START / RESUME
STOP
SUSPEND
DELETE
DETAILS
OBSERVABILITY
OS INFO
SCREENSHOT
SSH
CONNECT TO SERIAL CONSOLE

Connecting to serial ports is disabled

Logs

Logging

Serial port 1 (console)

SHOW MORE

Basic information

Name	vm-g23ai2132-mysql-2
Instance Id	1546048235189701404
Description	None
Type	Instance
Status	Running
Creation time	Aug 11, 2024, 12:18:22 AM UTC+05:30
Zone	asia-south1-a
Instance template	None
In use by	None
Reservations	Automatically choose
Labels	None
Tags	–
Deletion protection	Disabled
Confidential VM service	Disabled
Preserved state size	0 GB

Machine configuration

Machine type	e2-small
CPU platform	Intel Broadwell
Minimum CPU platform	None
Architecture	x86/64
vCPUs to core ratio	–
Custom visible cores	–
Display device	Disabled Enable to use screen capturing and recording tools
GPUs	None
Resource policies	

Networking

Public DNS PTR Record	None
Total egress bandwidth tier	–
NIC type	–

VIEW IN NETWORK TOPOLOGY

Firewalls

HTTP traffic	On
HTTPS traffic	On
Allow Load Balancer Health checks	Off

Network tags

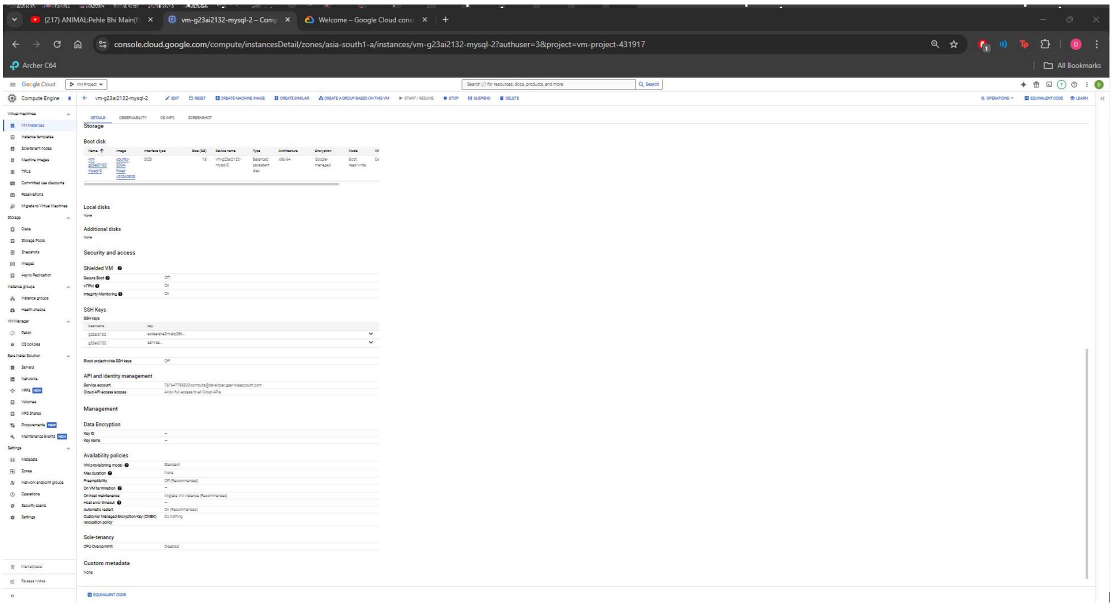
http-server https-server

Network interfaces

Name	Network	Subnetwork	Primary internal IP address	Alias IP ranges	IP stack type	External IP address	Networ
nic0	default	default	10.160.0.6		IPv4	34.93.183.188 (Ephemeral)	Premi

Storage

EQUIVALENT CODE


(217) ANIMAL:Pehle Bhi Main(F
vm-g23ai2132-mysql-2 – Comp
Welcome – Google Cloud conso
console.cloud.google.com/compute/instancesDetail/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&project=vm-project-431917
Archer C64
All Bookmarks
Google Cloud
VM Project
Search (/) for resources, docs, products, and more
Search
Compute Engine
vm-g23ai2132-mysql-2
EDIT
RESET
CREATE MACHINE IMAGE
CREATE SIMILAR
CREATE A GROUP BASED ON THIS VM
START / RESUME
STOP
SUSPEND
DELETE
DETAILS
OBSERVABILITY
OS INFO
SCREENSHOT
EQUIVALENT CODE

Storage

Boot disk

Name	Image	Interface type	Size (GB)	Device name	Type	Architecture	Encryption	Mode
vm-g23ai2132-mysql-2	ubuntu-2004-focal-v20240808	SCSI	15	vm-g23ai2132-mysql-2	Balanced persistent disk	x86/64	Google-managed	Boot, read/write

Local disks
None

Additional disks
None

Security and access

Shielded VM

Secure Boot	Off
vTPM	On
Integrity Monitoring	On

SSH Keys
SSH keys

Username	Key
g23ai2132	ecdsa-sha2-nistp256...
g23ai2132	ssh-rsa...

Block project-wide SSH keys	Off

API and identity management

Service account	761647798200-compute@developer.gserviceaccount.com
Cloud API access scopes	Allow full access to all Cloud APIs

Management

Data Encryption

Key ID	–
Key name	–

Availability policies

VM provisioning model	Standard
Max duration	None
Preemptibility	Off (Recommended)
On VM termination	–
On host maintenance	Migrate VM instance (Recommended)
Host error timeout	–
Automatic restart	On (Recommended)
Customer Managed Encryption Key (CMEK) revocation policy	Do nothing

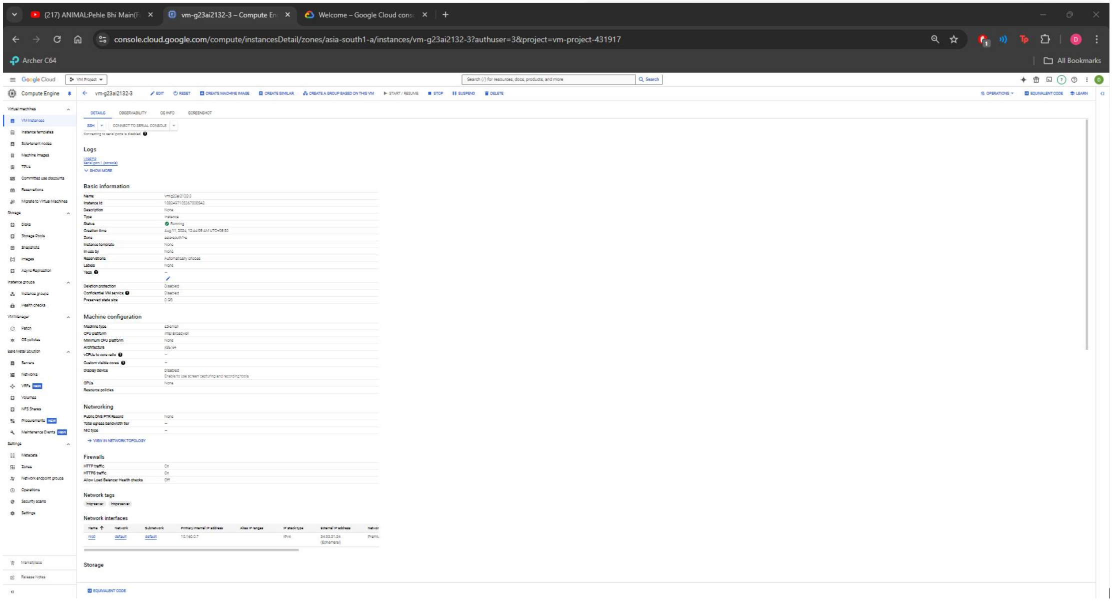
Sole-tenancy

CPU Overcommit	Disabled

Custom metadata
None

1.3. Create VM3 (Testing Environment)

✓ Create a New VM Instance:
✓ Navigate to Compute Engine > VM instances.
✓ Click Create Instance.
✓ Configure the instance:
✓ Name: vm-testing
✓ Region: Same as VM1 and VM2.
✓ Machine Type: n1-standard-1 (1 vCPU, 3.75 GB RAM).
✓ Boot Disk:
✓ OS: Ubuntu 22.04 LTS
✓ Size: 10 GB
✓ Firewall: No special settings needed.
✓ Click Create to launch the VM.
✓ Step 2: Set Up MySQL on VM2
✓ Let's configure VM2 to host the MySQL database.


(217) ANIMAL:Pehle Bhi Main(F
vm-g23ai2132-3 – Compute En
Welcome – Google Cloud conso
console.cloud.google.com/compute/instancesDetail/zones/asia-south1-a/instances/vm-g23ai2132-3?authuser=3&project=vm-project-431917
Archer C64
All Bookmarks
Google Cloud
VM Project
Search (/) for resources, docs, products, and more
Search
Compute Engine
vm-g23ai2132-3
EDIT
RESET
CREATE MACHINE IMAGE
CREATE SIMILAR
CREATE A GROUP BASED ON THIS VM
START / RESUME
STOP
SUSPEND
DELETE
DETAILS
OBSERVABILITY
OS INFO
SCREENSHOT
SSH
CONNECT TO SERIAL CONSOLE
SHOW MORE
VIEW IN NETWORK TOPOLOGY
EQUIVALENT CODE

Connecting to serial ports is disabled

Logs
Logging
Serial port 1 (console)

Basic information

Name	vm-g23ai2132-3
Instance Id	1882497105367008542
Description	None
Type	Instance
Status	Running
Creation time	Aug 11, 2024, 12:44:05 AM UTC+05:30
Zone	asia-south1-a
Instance template	None
In use by	None
Reservations	Automatically choose
Labels	None
Tags	–
Deletion protection	Disabled
Confidential VM service	Disabled
Preserved state size	0 GB

Machine configuration

Machine type	e2-small
CPU platform	Intel Broadwell
Minimum CPU platform	None
Architecture	x86/64
vCPUs to core ratio	–
Custom visible cores	–
Display device	Disabled Enable to use screen capturing and recording tools
GPUs	None
Resource policies	

Networking

Public DNS PTR Record	None
Total egress bandwidth tier	–
NIC type	–

Firewalls

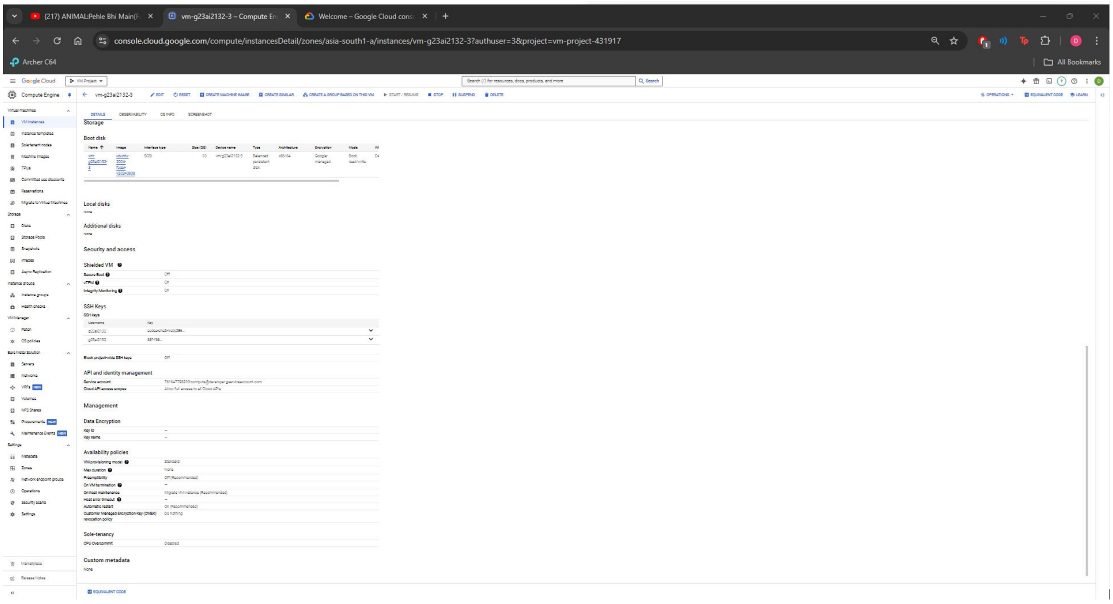
HTTP traffic	On
HTTPS traffic	On
Allow Load Balancer Health checks	Off

Network tags
http-server https-server

Network interfaces

Name	Network	Subnetwork	Primary internal IP address	Alias IP ranges	IP stack type	External IP address
nic0	default	default	10.160.0.7		IPv4	34.93.31.34 (Ephemeral)

Storage



console.cloud.google.com/compute/instancesDetail/zones/asia-south1-a/instances/vm-g23ai2132-3?authuser=3&project=vm-project-431917

vm-g23ai2132-3

Storage

Boot disk

Name	Image	Interface type	Size (GB)	Device name	Type	Architecture	Encryption	Mode
vm-g23ai2132-3	ubuntu-2004-focal-v20240808	SCSI	10	vm-g23ai2132-3	Balanced persistent disk	x86/64	Google-managed	Boot, read/write

Local disks

None

Additional disks

None

Security and access

Shielded VM

Secure Boot	Off
vTPM	On
Integrity Monitoring	On

SSH Keys

SSH keys

Username	Key
g23ai2132	ecdsa-sha2-nistp256...
g23ai2132	ssh-rsa...

Block project-wide SSH keys	Off

API and identity management

Service account	761647798200-compute@developer.gserviceaccount.com
Cloud API access scopes	Allow full access to all Cloud APIs

Management

Data Encryption

Key ID	–
Key name	–

Availability policies

VM provisioning model	Standard
Max duration	None
Preemptibility	Off (Recommended)
On VM termination	–
On host maintenance	Migrate VM instance (Recommended)
Host error timeout	–
Automatic restart	On (Recommended)
Customer Managed Encryption Key (CMEK) revocation policy	Do nothing

Sole-tenancy

CPU Overcommit	Disabled

Custom metadata

None

Step 2: SSH into VM2

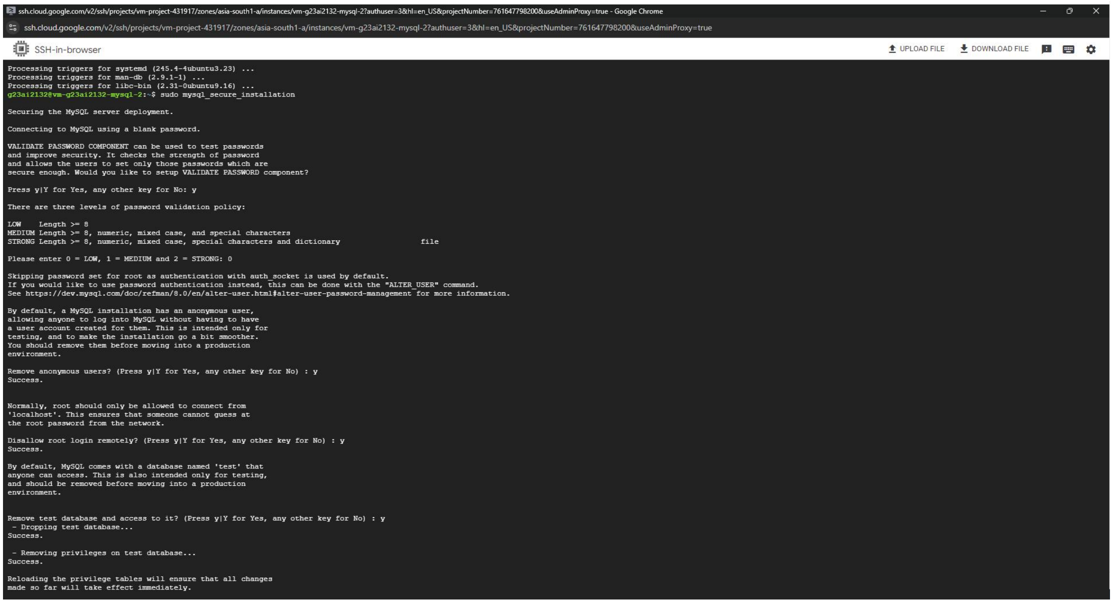
In the Google Cloud Console, navigate to Compute Engine > VM instances.
Find vm-mysql-db and click SSH to open a terminal.
Update Packages:

bash
sudo apt update && sudo apt upgrade -y
Install MySQL Server:

bash
sudo apt install mysql-server -y
Secure MySQL Installation:

bash
sudo mysql_secure_installation

Set a root password.
Remove anonymous users.
Disallow root login remotely.
Remove test databases.
Reload privilege tables.


ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
Processing triggers for systemd (245.4-4ubuntu3.23) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.16) ...
g23ai2132@vm-g23ai2132-mysql-2:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary                  file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
 - Dropping test database...
Success.

 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.
```

Edit the MySQL configuration file:

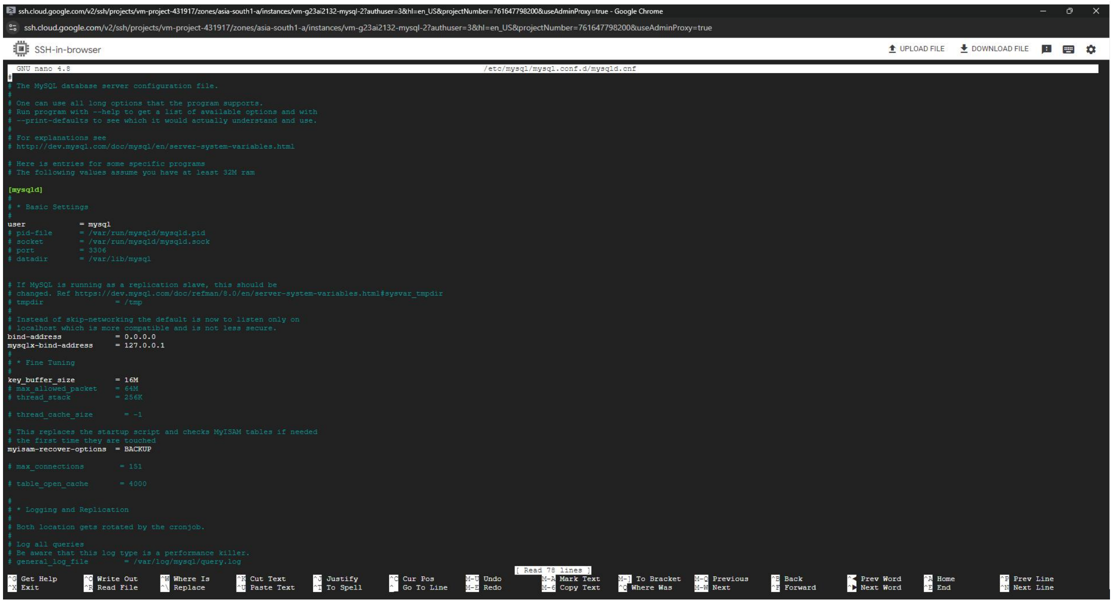
bash

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

Find the line bind-address = 127.0.0.1 and change it to:

bind-address = 0.0.0.0

Save and exit (CTRL+X, Y, Enter).


ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
  GNU nano 4.8                                                              /etc/mysql/mysql.conf.d/mysqld.cnf
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user            = mysql
# pid-file      = /var/run/mysqld/mysqld.pid
# socket        = /var/run/mysqld/mysqld.sock
# port          = 3306
# datadir       = /var/lib/mysql


# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir                = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
mysqlx-bind-address     = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size         = 16M
# max_allowed_packet    = 64M
# thread_stack          = 256K

# thread_cache_size       = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options  = BACKUP

# max_connections        = 151

# table_open_cache       = 4000

#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
# general_log_file        = /var/log/mysql/query.log
                                                              [ Read 78 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text  M-] To Bracket M-Q Previous   ^B Back        ^◀ Prev Word   ^A Home        ^P Prev Line
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text  ^Q Where Was   M-W Next       ^F Forward     ^▶ Next Word   ^E End         ^N Next Line
```

Configure MySQL to Accept Remote Connections:

Restart MySQL Service:

bash

sudo systemctl restart mysql

Create a Database and User for Flask App:

Log in to MySQL:

bash

sudo mysql -u root -p

Create the database and user:

sql
CREATE DATABASE mydb;
CREATE USER 'dhood'@'%' IDENTIFIED BY 'dhanshree';
GRANT ALL PRIVILEGES ON mydb.* TO 'dhood'@'%';
FLUSH PRIVILEGES;

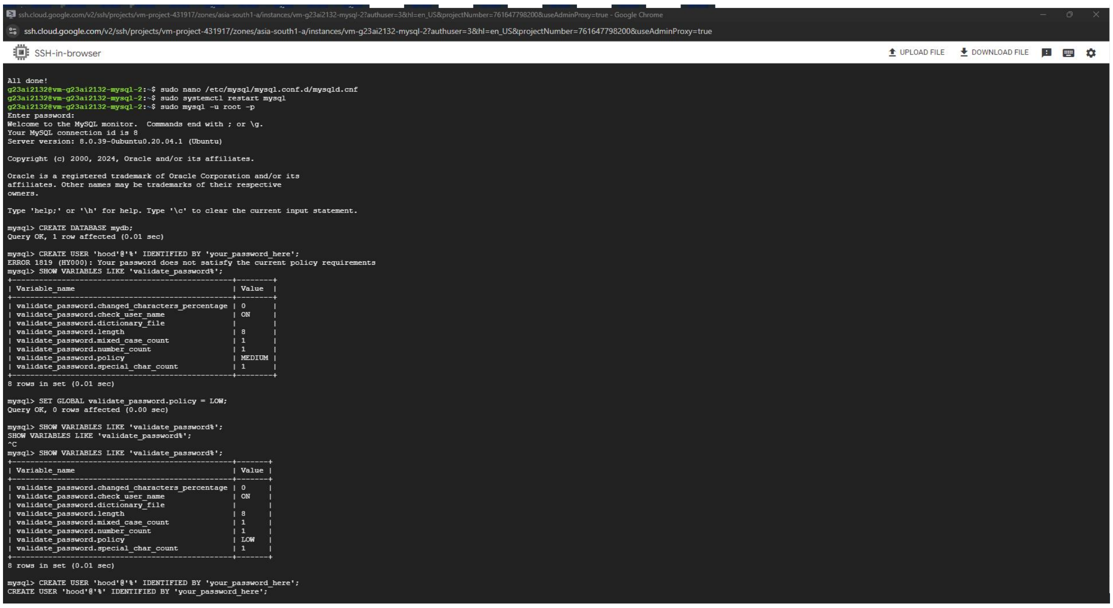
Add data manually into mysql mydb database table employess:
INSERT INTO employees (name, position, salary) VALUES ('John Doe', 'Software Engineer', 75000.00);

INSERT INTO employees (name, position, salary) VALUES ('Dhanshree Hood', 'AI Engineer', 80000.00);

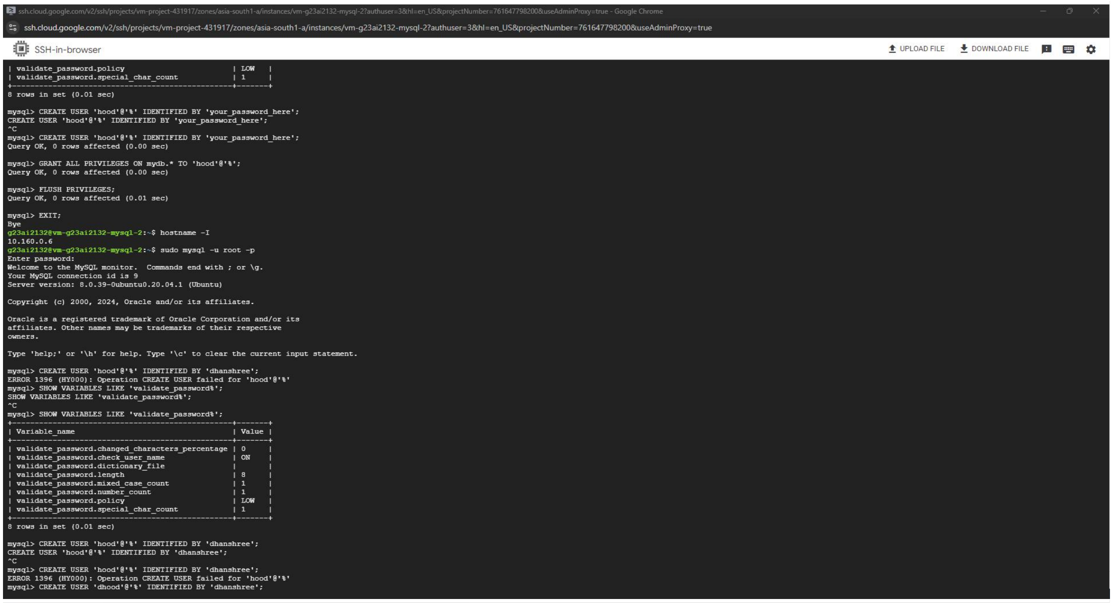
INSERT INTO employees (name, position, salary) VALUES ('Vanita Hood', 'HR', 90000.00);

INSERT INTO employees (name, position, salary) VALUES ('Gursharan Singh', 'Buisness Analyst', 60000.00);

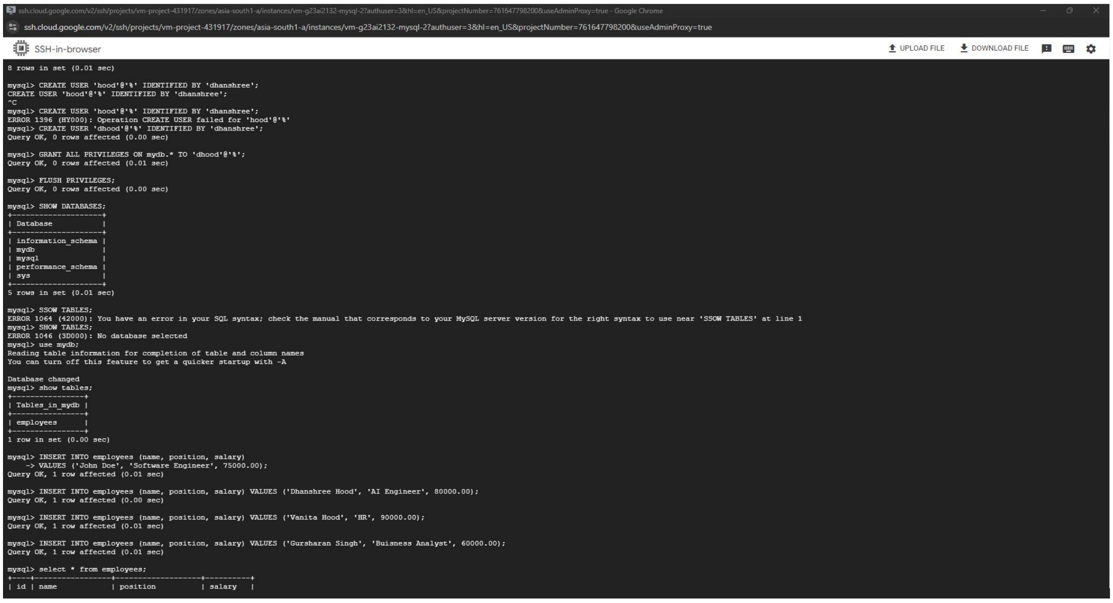
EXIT;


ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
All done!
g23ai2132@vm-g23ai2132-mysql-2:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
g23ai2132@vm-g23ai2132-mysql-2:~$ sudo systemctl restart mysql
g23ai2132@vm-g23ai2132-mysql-2:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.39-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CREATE DATABASE mydb;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER 'hood'@'%' IDENTIFIED BY 'your_password_here';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> SHOW VARIABLES LIKE 'validate_password%';
+-------------------------------------------------+--------+
| Variable_name                                   | Value  |
+-------------------------------------------------+--------+
| validate_password.changed_characters_percentage | 0      |
| validate_password.check_user_name               | ON     |
| validate_password.dictionary_file               |        |
| validate_password.length                        | 8      |
| validate_password.mixed_case_count              | 1      |
| validate_password.number_count                  | 1      |
| validate_password.policy                        | MEDIUM |
| validate_password.special_char_count            | 1      |
+-------------------------------------------------+--------+
8 rows in set (0.01 sec)

mysql> SET GLOBAL validate_password.policy = LOW;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW VARIABLES LIKE 'validate_password%';
SHOW VARIABLES LIKE 'validate_password%';
^C
mysql> SHOW VARIABLES LIKE 'validate_password%';
+-------------------------------------------------+-------+
| Variable_name                                   | Value |
+-------------------------------------------------+-------+
| validate_password.changed_characters_percentage | 0     |
| validate_password.check_user_name               | ON    |
| validate_password.dictionary_file               |       |
| validate_password.length                        | 8     |
| validate_password.mixed_case_count              | 1     |
| validate_password.number_count                  | 1     |
| validate_password.policy                        | LOW   |
| validate_password.special_char_count            | 1     |
+-------------------------------------------------+-------+
8 rows in set (0.01 sec)

mysql> CREATE USER 'hood'@'%' IDENTIFIED BY 'your_password_here';
CREATE USER 'hood'@'%' IDENTIFIED BY 'your_password_here';
```


ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
| validate_password.policy                       | LOW   |
| validate_password.special_char_count           | 1     |
+------------------------------------------------+-------+
8 rows in set (0.01 sec)

mysql> CREATE USER 'hood'@'%' IDENTIFIED BY 'your_password_here';
CREATE USER 'hood'@'%' IDENTIFIED BY 'your_password_here';
^C
mysql> CREATE USER 'hood'@'%' IDENTIFIED BY 'your_password_here';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON mydb.* TO 'hood'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> EXIT;
Bye
g23ai2132@vm-g23ai2132-mysql-2:~$ hostname -I
10.160.0.6
g23ai2132@vm-g23ai2132-mysql-2:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.39-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CREATE USER 'hood'@'%' IDENTIFIED BY 'dhanshree';
ERROR 1396 (HY000): Operation CREATE USER failed for 'hood'@'%'
mysql> SHOW VARIABLES LIKE 'validate_password%';
SHOW VARIABLES LIKE 'validate_password%';
^C
mysql> SHOW VARIABLES LIKE 'validate_password%';
+------------------------------------------------+-------+
| Variable_name                                  | Value |
+------------------------------------------------+-------+
| validate_password.changed_characters_percentage | 0     |
| validate_password.check_user_name              | ON    |
| validate_password.dictionary_file              |       |
| validate_password.length                       | 8     |
| validate_password.mixed_case_count             | 1     |
| validate_password.number_count                 | 1     |
| validate_password.policy                       | LOW   |
| validate_password.special_char_count           | 1     |
+------------------------------------------------+-------+
8 rows in set (0.01 sec)

mysql> CREATE USER 'hood'@'%' IDENTIFIED BY 'dhanshree';
CREATE USER 'hood'@'%' IDENTIFIED BY 'dhanshree';
^C
mysql> CREATE USER 'hood'@'%' IDENTIFIED BY 'dhanshree';
ERROR 1396 (HY000): Operation CREATE USER failed for 'hood'@'%'
mysql> CREATE USER 'dhood'@'%' IDENTIFIED BY 'dhanshree';
```


ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-mysql-2?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
8 rows in set (0.01 sec)

mysql> CREATE USER 'hood'@'%' IDENTIFIED BY 'dhanshree';
CREATE USER 'hood'@'%' IDENTIFIED BY 'dhanshree';
^C
mysql> CREATE USER 'hood'@'%' IDENTIFIED BY 'dhanshree';
ERROR 1396 (HY000): Operation CREATE USER failed for 'hood'@'%'
mysql> CREATE USER 'dhood'@'%' IDENTIFIED BY 'dhanshree';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON mydb.* TO 'dhood'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mydb               |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)

mysql> SSOW TABLES;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'SSOW TABLES' at line 1
mysql> SHOW TABLES;
ERROR 1046 (3D000): No database selected
mysql> use mydb;
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> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| employees      |
+----------------+
1 row in set (0.00 sec)

mysql> INSERT INTO employees (name, position, salary)
    -> VALUES ('John Doe', 'Software Engineer', 75000.00);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO employees (name, position, salary) VALUES ('Dhanshree Hood', 'AI Engineer', 80000.00);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employees (name, position, salary) VALUES ('Vanita Hood', 'HR', 90000.00);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO employees (name, position, salary) VALUES ('Gursharan Singh', 'Buisness Analyst', 60000.00);
Query OK, 1 row affected (0.01 sec)

mysql> select * from employees;
+----+-----------------+-------------------+----------+
| id | name            | position          | salary   |
```

Step 3: Deploy Flask Application on VM1

SSH into VM1:

In the Google Cloud Console, navigate to Compute Engine > VM instances.
Find vm-flask-app and click SSH to open a terminal.

Update Packages:

bash

sudo apt update && sudo apt upgrade -y

Install Python and Pip:

bash

sudo apt install python3 python3-pip -y

Install Required Python Packages:

bash

```
pip3 install Flask Flask-SQLAlchemy Flask-RESTful pymysql
```

Install MySQL Client and Libraries:

bash

```
sudo apt install libmysqlclient-dev -y
```

Create the Flask Application:-

Create a directory for your app:

bash

```
mkdir ~/flask_api && cd ~/flask_api
```

Create the app.py file:

bash

```
nano app.py
```

Python code:

```
from flask import Flask, request, jsonify
from flask_sqlalchemy import SQLAlchemy
from flask_restful import Api, Resource

app = Flask(__name__)
api = Api(app)

# Database configuration
app.config['SQLALCHEMY_DATABASE_URI'] =
'mysql+pymysql://hood:your_password_here@<VM2_Internal_IP>:3306/mydb'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

db = SQLAlchemy(app)

# Define the Employee model
class Employee(db.Model):
    __tablename__ = 'employees'
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    position = db.Column(db.String(50), nullable=False)
    salary = db.Column(db.Numeric(10, 2), nullable=False)

    def to_dict(self):
        return {
            'id': self.id,
            'name': self.name,
            'position': self.position,
            'salary': str(self.salary)
        }

# RESTful resource for Employee
class EmployeeResource(Resource):
    def get(self, emp_id=None):
        if emp_id:
            employee = Employee.query.get(emp_id)
            if not employee:
                return {'message': 'Employee not found'}, 404
            return employee.to_dict()
        else:
            employees = Employee.query.all()
            return [emp.to_dict() for emp in employees]

    def post(self):
        data = request.json
        new_employee = Employee(
            name=data['name'],
            position=data['position'],
            salary=data['salary']
        )
        db.session.add(new_employee)
        db.session.commit()
        return {'message': 'Employee created successfully'}, 201

    def put(self, emp_id):
        data = request.json
        employee = Employee.query.get(emp_id)
        if not employee:
            return {'message': 'Employee not found'}, 404

        employee.name = data.get('name', employee.name)
        employee.position = data.get('position', employee.position)
        employee.salary = data.get('salary', employee.salary)
        db.session.commit()
        return {'message': 'Employee updated successfully'}

    def delete(self, emp_id):
        employee = Employee.query.get(emp_id)
        if not employee:
            return {'message': 'Employee not found'}, 404
        db.session.delete(employee)
        db.session.commit()
        return {'message': 'Employee deleted successfully'}

# Add API routes
api.add_resource(EmployeeResource, '/employees', '/employees/<int:emp_id>')

if __name__ == '__main__':
    # Create the database tables
    with app.app_context():
        db.create_all()

    app.run(host='0.0.0.0', port=5000, debug=True)
```

Replace <VM2_Internal_IP> with the internal IP address of VM2. You can find this in the VM instances list under Internal IP.

Save and exit the editor: Press CTRL+X, then Y, and Enter.

Run the Flask Application:
bash

```
python3 app.py
```

You should see output indicating the app is running:

plaintext

```
* Running on http://0.0.0.0:
```

Step 3: Run CRUD operations from vm-g23ai2132-3

Install Necessary Tools on VM 3

Update Package List:
Run the following command to update the package list:
bash

```
sudo apt update
```

Install curl:

Install curl which will be used to send HTTP requests:
bash

```
sudo apt install curl
```

Perform CRUD Operations from VM 3:

Create (Insert) a New Employee:
Use the following curl command to insert a new employee:
bash

```
curl -X POST http://10.160.0.5:3306/employees -H "Content-Type: application/json"
-d '{
   "name": "John Doe",
   "position": "Software Engineer",
   "salary": 75000.00
}'
```

Retrieve all employee records:
bash

```
curl -X GET http://10.160.0.5:3306/employees
```

Retrieve a specific employee by ID:
bash

```
curl -X GET http://10.160.0.5:3306/employees/1
```

Update an Employee Record
Update an existing employee record:
bash

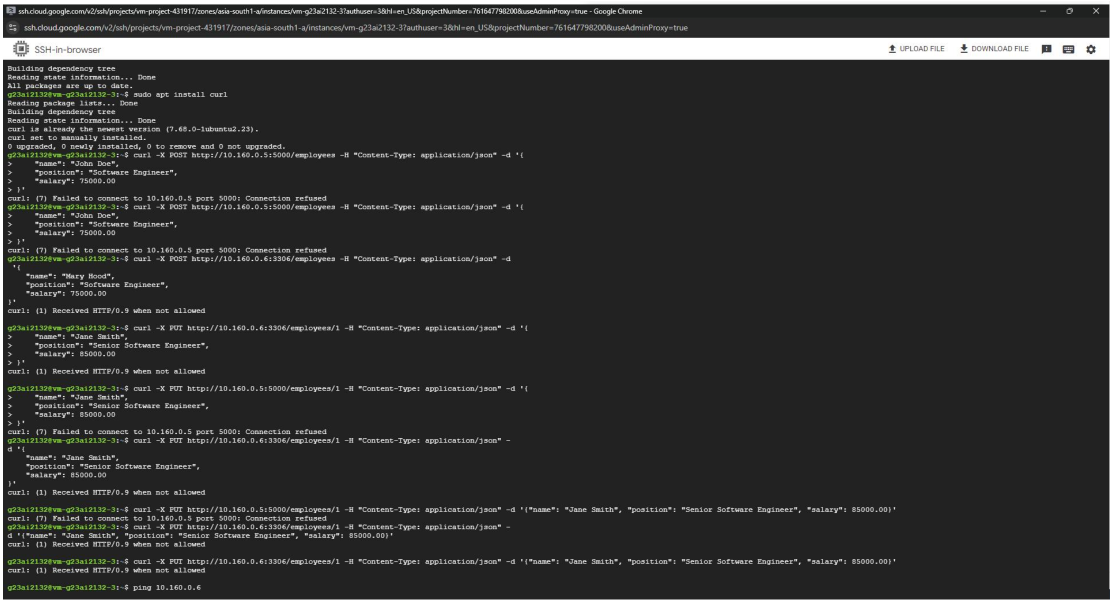
```
curl -X PUT http://10.160.0.5:3306/employees/1 -H "Content-Type: application/json"
-d '{
    "name": "Jane Smith",
    "position": "Senior Software Engineer",
    "salary": 85000.00
}'
```

Delete an Employee Record

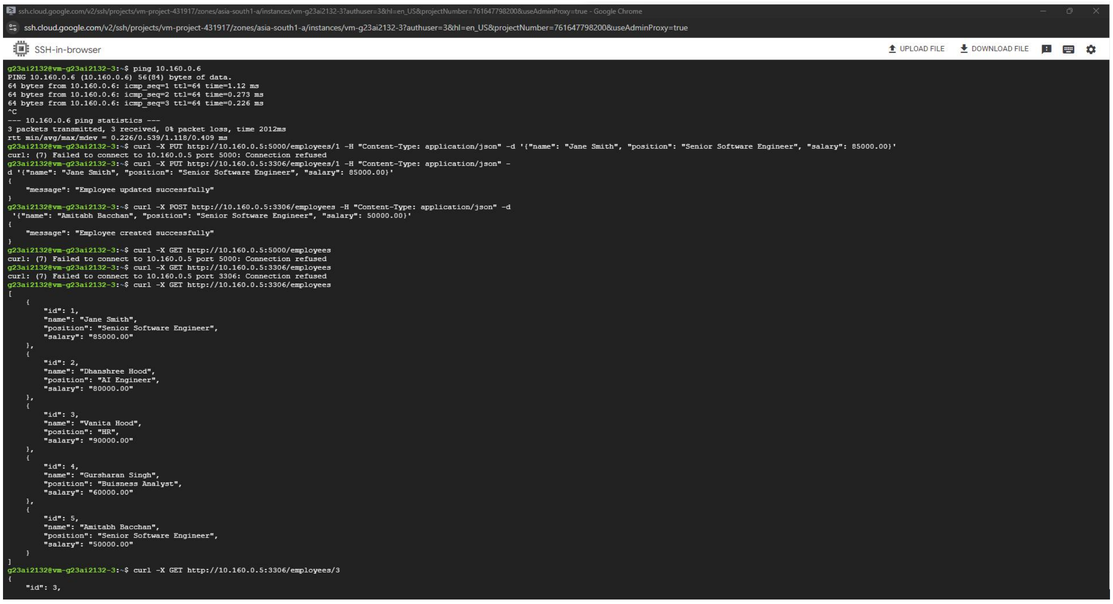
Delete an employee record by ID:

bash

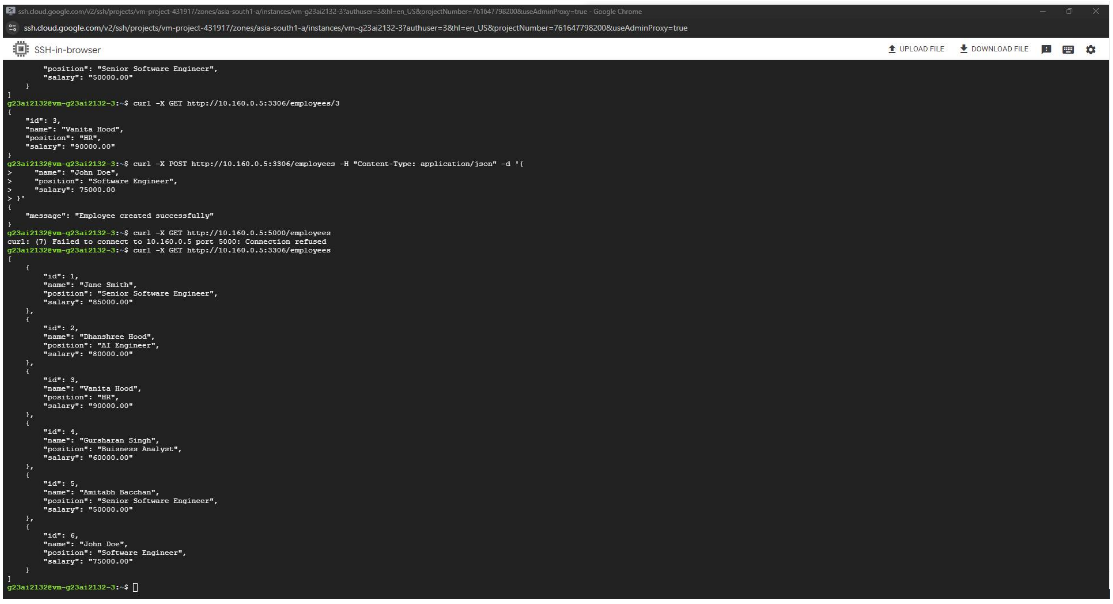
```
curl -X DELETE http://10.160.0.5:3306/employees/1
```


ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-3?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-3?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
Building dependency tree
Reading state information... Done
All packages are up to date.
g23ai2132@vm-g23ai2132-3:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.68.0-1ubuntu2.23).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
g23ai2132@vm-g23ai2132-3:~$ curl -X POST http://10.160.0.5:5000/employees -H "Content-Type: application/json" -d '{
>     "name": "John Doe",
>     "position": "Software Engineer",
>     "salary": 75000.00
> }'
curl: (7) Failed to connect to 10.160.0.5 port 5000: Connection refused
g23ai2132@vm-g23ai2132-3:~$ curl -X POST http://10.160.0.5:5000/employees -H "Content-Type: application/json" -d '{
>     "name": "John Doe",
>     "position": "Software Engineer",
>     "salary": 75000.00
> }'
curl: (7) Failed to connect to 10.160.0.5 port 5000: Connection refused
g23ai2132@vm-g23ai2132-3:~$ curl -X POST http://10.160.0.6:3306/employees -H "Content-Type: application/json" -d
 '{
    "name": "Mary Hood",
    "position": "Software Engineer",
    "salary": 75000.00
}'
curl: (1) Received HTTP/0.9 when not allowed

g23ai2132@vm-g23ai2132-3:~$ curl -X PUT http://10.160.0.6:3306/employees/1 -H "Content-Type: application/json" -d '{
>     "name": "Jane Smith",
>     "position": "Senior Software Engineer",
>     "salary": 85000.00
> }'
curl: (1) Received HTTP/0.9 when not allowed

g23ai2132@vm-g23ai2132-3:~$ curl -X PUT http://10.160.0.5:5000/employees/1 -H "Content-Type: application/json" -d '{
>     "name": "Jane Smith",
>     "position": "Senior Software Engineer",
>     "salary": 85000.00
> }'
curl: (7) Failed to connect to 10.160.0.5 port 5000: Connection refused
g23ai2132@vm-g23ai2132-3:~$ curl -X PUT http://10.160.0.6:3306/employees/1 -H "Content-Type: application/json" -
d '{
    "name": "Jane Smith",
    "position": "Senior Software Engineer",
    "salary": 85000.00
}'
curl: (1) Received HTTP/0.9 when not allowed

g23ai2132@vm-g23ai2132-3:~$ curl -X PUT http://10.160.0.5:5000/employees/1 -H "Content-Type: application/json" -d '{"name": "Jane Smith", "position": "Senior Software Engineer", "salary": 85000.00}'
curl: (7) Failed to connect to 10.160.0.5 port 5000: Connection refused
g23ai2132@vm-g23ai2132-3:~$ curl -X PUT http://10.160.0.6:3306/employees/1 -H "Content-Type: application/json" -
d '{"name": "Jane Smith", "position": "Senior Software Engineer", "salary": 85000.00}'
curl: (1) Received HTTP/0.9 when not allowed

g23ai2132@vm-g23ai2132-3:~$ curl -X PUT http://10.160.0.6:3306/employees/1 -H "Content-Type: application/json" -d '{"name": "Jane Smith", "position": "Senior Software Engineer", "salary": 85000.00}'
curl: (1) Received HTTP/0.9 when not allowed

g23ai2132@vm-g23ai2132-3:~$ ping 10.160.0.6
```


ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-3?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-3?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
g23ai2132@vm-g23ai2132-3:~$ ping 10.160.0.6
PING 10.160.0.6 (10.160.0.6) 56(84) bytes of data.
64 bytes from 10.160.0.6: icmp_seq=1 ttl=64 time=1.12 ms
64 bytes from 10.160.0.6: icmp_seq=2 ttl=64 time=0.273 ms
64 bytes from 10.160.0.6: icmp_seq=3 ttl=64 time=0.226 ms
^C
--- 10.160.0.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2012ms
rtt min/avg/max/mdev = 0.226/0.539/1.118/0.409 ms
g23ai2132@vm-g23ai2132-3:~$ curl -X PUT http://10.160.0.5:5000/employees/1 -H "Content-Type: application/json" -d '{"name": "Jane Smith", "position": "Senior Software Engineer", "salary": 85000.00}'
curl: (7) Failed to connect to 10.160.0.5 port 5000: Connection refused
g23ai2132@vm-g23ai2132-3:~$ curl -X PUT http://10.160.0.5:3306/employees/1 -H "Content-Type: application/json" -
d '{"name": "Jane Smith", "position": "Senior Software Engineer", "salary": 85000.00}'
{
    "message": "Employee updated successfully"
}
g23ai2132@vm-g23ai2132-3:~$ curl -X POST http://10.160.0.5:3306/employees -H "Content-Type: application/json" -d
 '{"name": "Amitabh Bacchan", "position": "Senior Software Engineer", "salary": 50000.00}'
{
    "message": "Employee created successfully"
}
g23ai2132@vm-g23ai2132-3:~$ curl -X GET http://10.160.0.5:5000/employees
curl: (7) Failed to connect to 10.160.0.5 port 5000: Connection refused
g23ai2132@vm-g23ai2132-3:~$ curl -X GET http://10.160.0.5:3306/employees
curl: (7) Failed to connect to 10.160.0.5 port 3306: Connection refused
g23ai2132@vm-g23ai2132-3:~$ curl -X GET http://10.160.0.5:3306/employees
[
    {
        "id": 1,
        "name": "Jane Smith",
        "position": "Senior Software Engineer",
        "salary": "85000.00"
    },
    {
        "id": 2,
        "name": "Dhanshree Hood",
        "position": "AI Engineer",
        "salary": "80000.00"
    },
    {
        "id": 3,
        "name": "Vanita Hood",
        "position": "HR",
        "salary": "90000.00"
    },
    {
        "id": 4,
        "name": "Gursharan Singh",
        "position": "Buisness Analyst",
        "salary": "60000.00"
    },
    {
        "id": 5,
        "name": "Amitabh Bacchan",
        "position": "Senior Software Engineer",
        "salary": "50000.00"
    }
]
g23ai2132@vm-g23ai2132-3:~$ curl -X GET http://10.160.0.5:3306/employees/3
{
    "id": 3,
```



ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-3?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/vm-project-431917/zones/asia-south1-a/instances/vm-g23ai2132-3?authuser=3&hl=en_US&projectNumber=761647798200&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
        "position": "Senior Software Engineer",
        "salary": "50000.00"
    }
]
g23ai2132@vm-g23ai2132-3:~$ curl -X GET http://10.160.0.5:3306/employees/3
{
    "id": 3,
    "name": "Vanita Hood",
    "position": "HR",
    "salary": "90000.00"
}
g23ai2132@vm-g23ai2132-3:~$ curl -X POST http://10.160.0.5:3306/employees -H "Content-Type: application/json" -d '{
>     "name": "John Doe",
>     "position": "Software Engineer",
>     "salary": 75000.00
> }'
{
    "message": "Employee created successfully"
}
g23ai2132@vm-g23ai2132-3:~$ curl -X GET http://10.160.0.5:5000/employees
curl: (7) Failed to connect to 10.160.0.5 port 5000: Connection refused
g23ai2132@vm-g23ai2132-3:~$ curl -X GET http://10.160.0.5:3306/employees
[
    {
        "id": 1,
        "name": "Jane Smith",
        "position": "Senior Software Engineer",
        "salary": "85000.00"
    },
    {
        "id": 2,
        "name": "Dhanshree Hood",
        "position": "AI Engineer",
        "salary": "80000.00"
    },
    {
        "id": 3,
        "name": "Vanita Hood",
        "position": "HR",
        "salary": "90000.00"
    },
    {
        "id": 4,
        "name": "Gursharan Singh",
        "position": "Buisness Analyst",
        "salary": "60000.00"
    },
    {
        "id": 5,
        "name": "Amitabh Bacchan",
        "position": "Senior Software Engineer",
        "salary": "50000.00"
    },
    {
        "id": 6,
        "name": "John Doe",
        "position": "Software Engineer",
        "salary": "75000.00"
    }
]
g23ai2132@vm-g23ai2132-3:~$ 
```

Conclusion:

In this project, you successfully developed a RESTful API using Flask and SQLAlchemy, allowing CRUD operations on an employee database stored in MySQL. You effectively deployed the Flask application across multiple virtual machines on Google Cloud Platform (GCP), demonstrating an understanding of cloud infrastructure and the importance of proper IP configuration for seamless communication between VMs.

By overcoming various challenges, such as configuring MySQL for remote access, resolving connection issues, and executing API requests through curl, I have gained hands-on experience with both backend development and cloud deployment. This project not only highlights my ability to build and deploy scalable applications but also showcases my problem-solving skills in real-world scenarios. My perseverance and technical proficiency have culminated in a robust, cloud-based application that could serve as a solid foundation for more complex systems in the future.