DBMS 2022

Innovative Project: Amazon RDS

What are we going to do?

Migrating to Cloud SQL from Amazon RDS for MySQL Using Database Migration Service

Overview:

Database Migration Service provides options for one-time and continuous jobs to migrate data to Cloud SQL using different
connectivity options, including IP allowlists, VPC peering, and reverse SSH tunnels.

In this lab, you migrate a MySQL database from an Amazon RDS instance for MySQL to Cloud SQL for MySQL using a one-time Database Migration Service job and an IP allowlist for connectivity. After you create and run the migration job, you will check that the database has been successfully migrated to your Cloud SQL for
MySQL instance.

Objectives:

In this project, we configure a one-time Database Migration Service job to migrate databases from a cloud-based MySQL instance to
Cloud SQL for MySQL.

Create a profile for a source connection to a cloud-based MySQL instance (Amazon RDS for MySQL).

Create and run a one-time migration job using Database Migration Service.

Verify that the migration job has completed successfully.

Setup:

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

01:30:00 Make sure you signed into Qwiklabs using an incognito window.

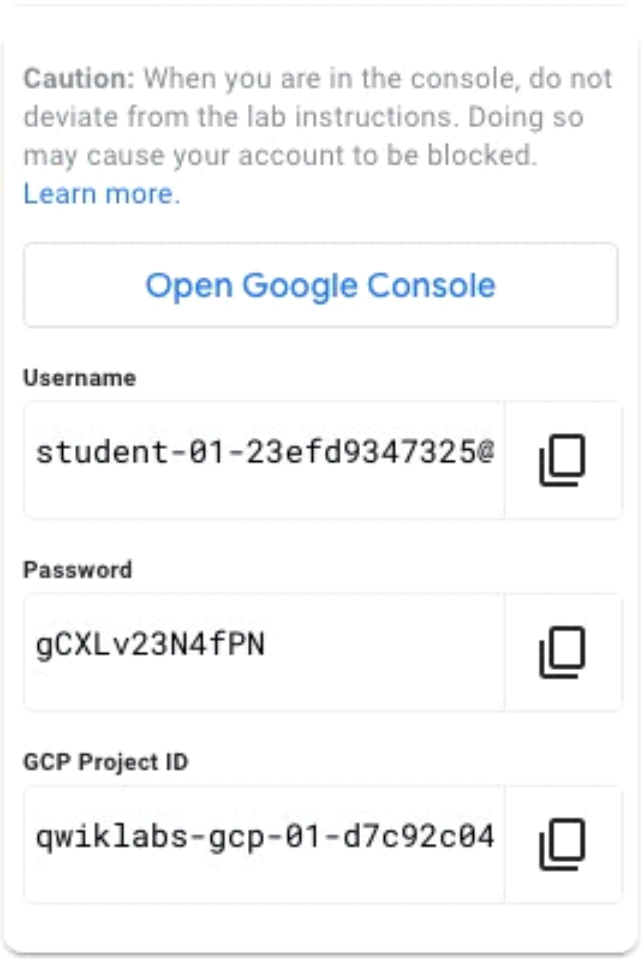
Note the lab's access time (for example, and make sure you can fin

There is no pause feature. You can restart if needed, but you have to start at the beginning.

When ready, click 

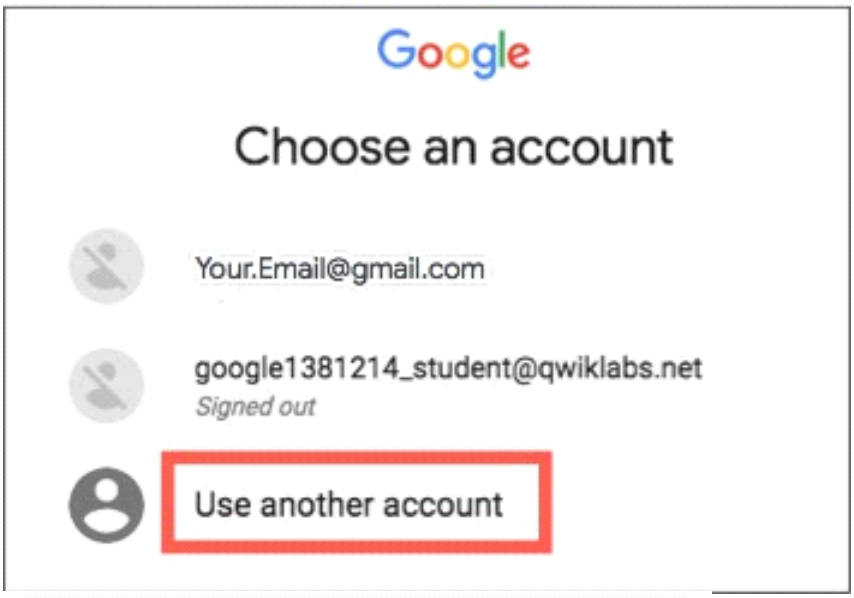
.

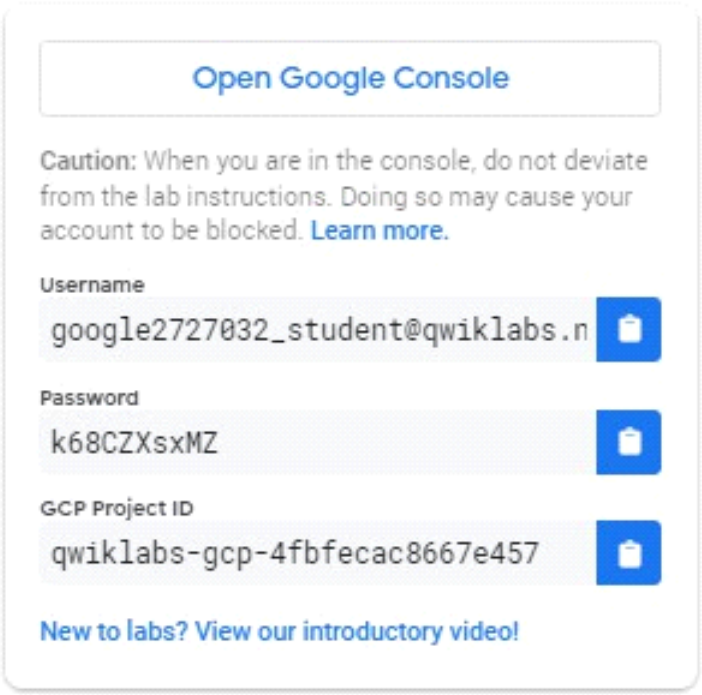
Note your lab credentials. You will use them to sign in to Cloud Platform Console.



Click Open Google Console.

Click Use another account and copy/paste credentials for this lab into the prompts.





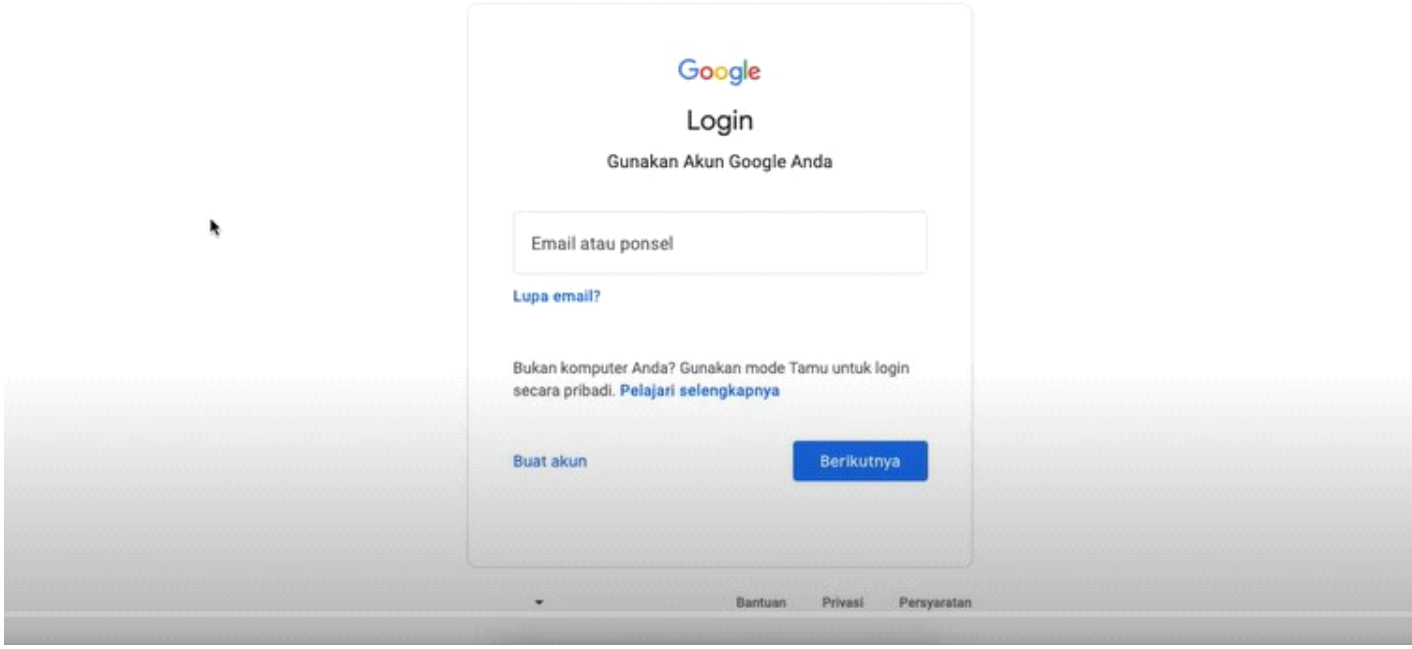
Accept the terms and skip the recovery resource page.

Do not click End Lab unless you are finished with the lab or want to restart it. This clears your work and removes the project.

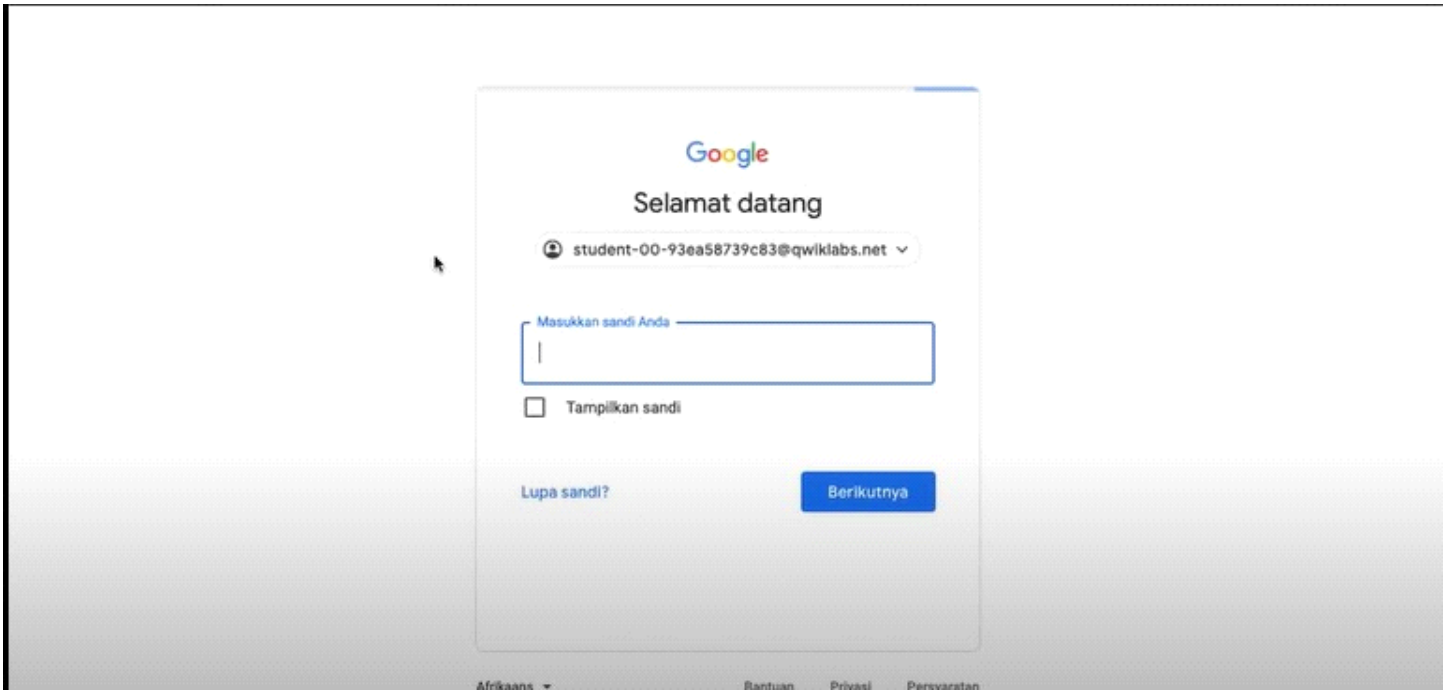
Copy the username, and then click Open Google Console. The lab spins up resources, and then opens another tab

that shows
the Choose an account page.

The Sign in page opens. Paste the username that you copied from the Connection Details panel. Then copy and paste the password.



In the same way, copy the password from previous page and paste it in the password section



Click through the subsequent pages:

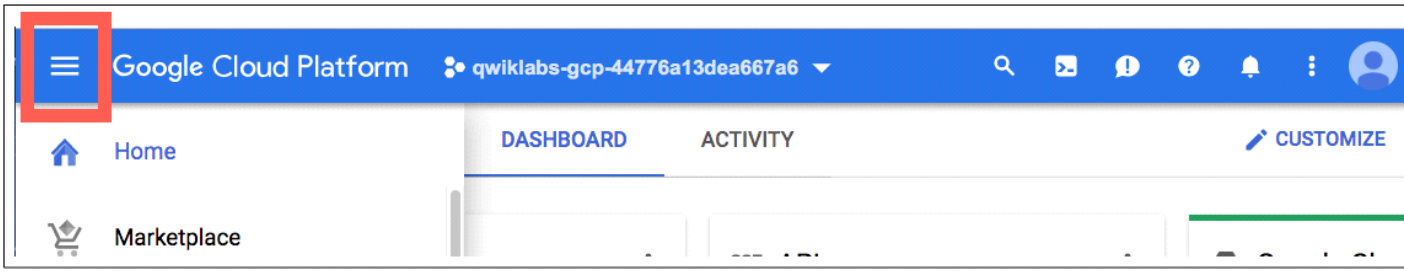
Accept the terms and conditions.

Do not add recovery options or two-factor authentication (because this is a temporary account).

Do not sign up for free trials.

After a few moments, the GCP console opens in this tab.

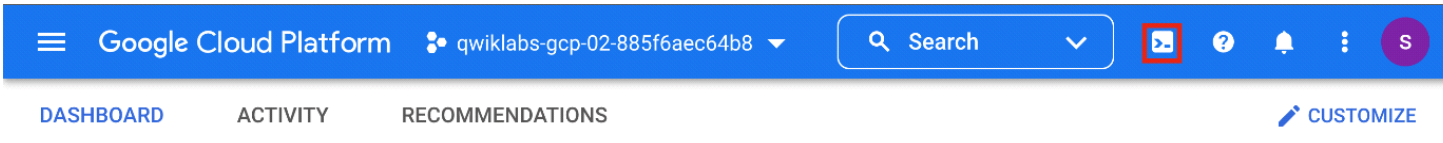
You can view the menu with a list of GCP Products and Services by clicking the Navigation menu at the top-left, next to “Google Cloud Platform”.



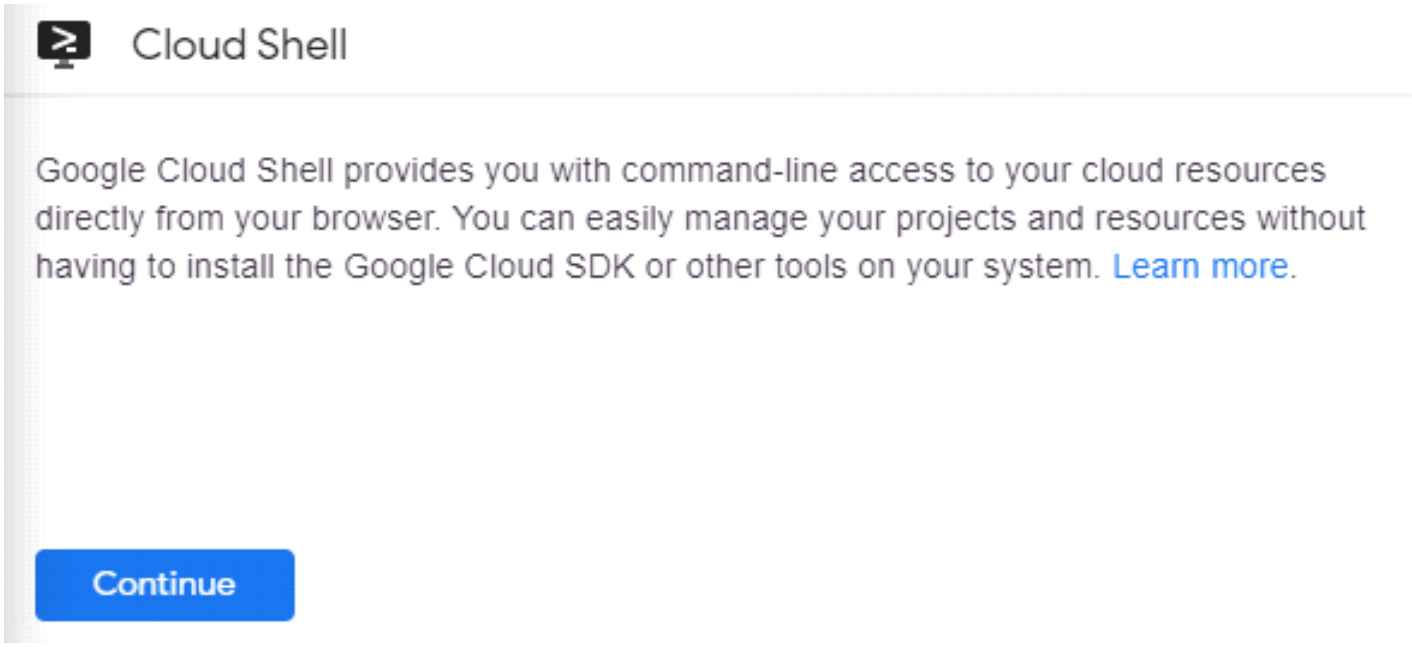
Activate Google Cloud Shell

Google Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Google Cloud Shell provides command- line access to your GCP resources.

In GCP console, on the top right toolbar, click the Open Cloud Shell button.

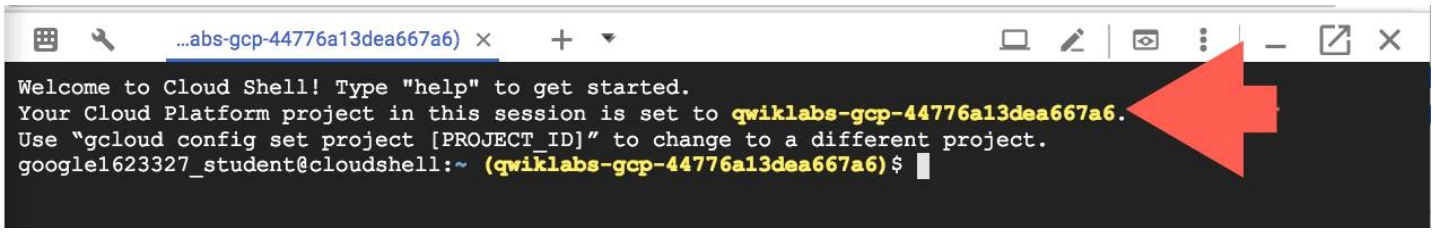


Click Continue.



It takes a few moments to provision and connect to the environment. When you are connected, you are already

authenticated, and the project is set to your PROJECT_ID. For example:



gcloud is the command-line tool for Google Cloud Platform. It comes pre-installed on Cloud Shell and supports tab-completion.

- You can list the active account name with this command: gcloud auth list

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-04-778211046e35.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
student_00_93ea58739c83@cloudshell:~ (qwiklabs-gcp-04-778211046e35)$ gcloud auth list
Credentialed Accounts

ACTIVE: *
ACCOUNT: student-00-93ea58739c83@qwiklabs.net

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

student_00_93ea58739c83@cloudshell:~ (qwiklabs-gcp-04-778211046e35)$
```

Output:

- You can list the project ID with this command: gcloud config list project

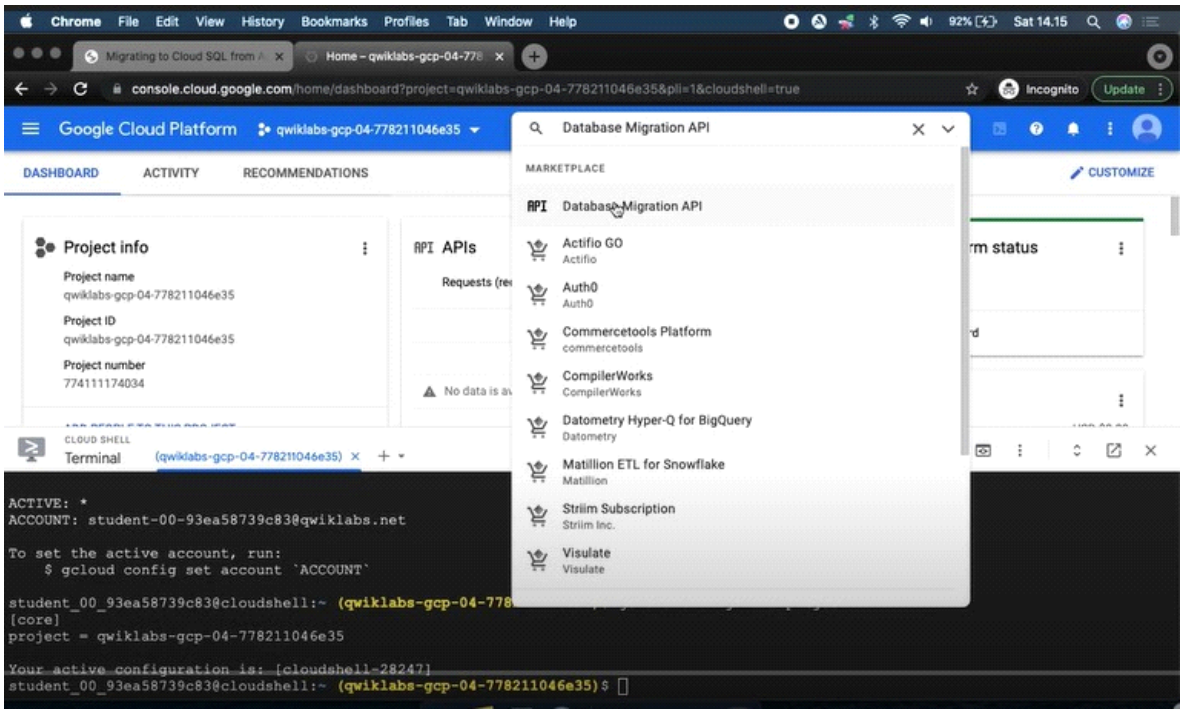
Output:

```
student_00_93ea58739c83@cloudshell:~ (qwiklabs-gcp-04-778211046e35)$ gcloud config
[core]
project = qwiklabs-gcp-04-778211046e35

Your active configuration is: [cloudshell-28247]
student_00_93ea58739c83@cloudshell:~ (qwiklabs-gcp-04-778211046e35)$
```

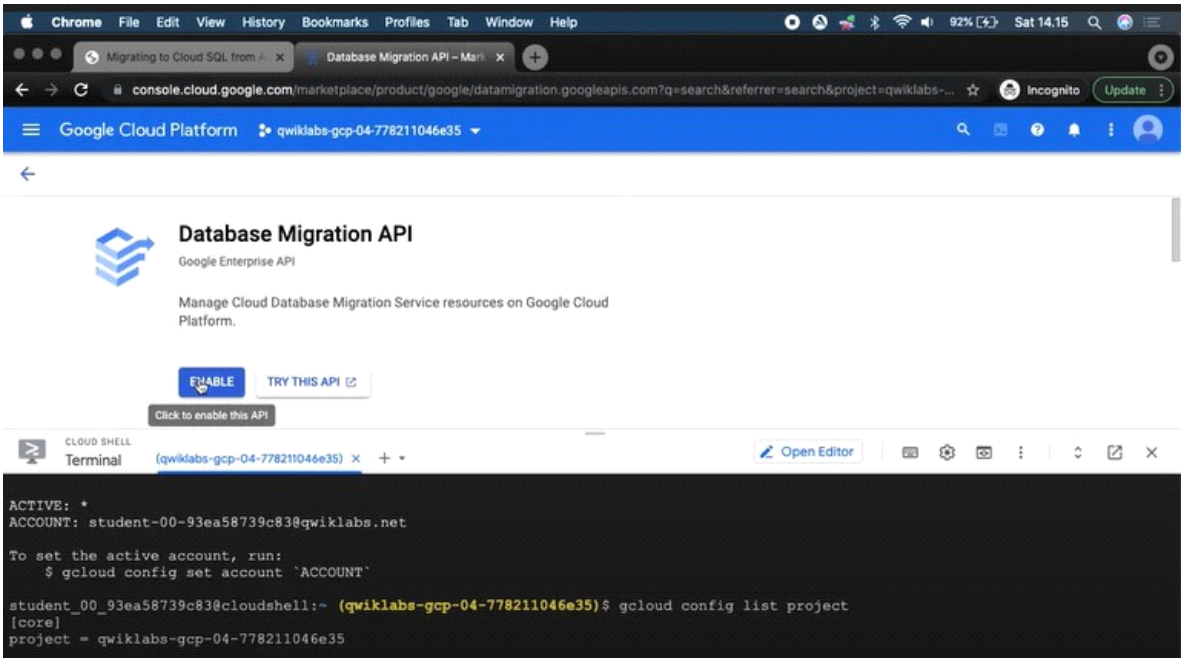
Verify that the Database Migration API is enabled

In the Google Cloud Console, enter Database Migration API in the top search bar.



Click on the result for Database Migration API. This page will either show status information or give you the option to enable the API.

If necessary, enable the API.



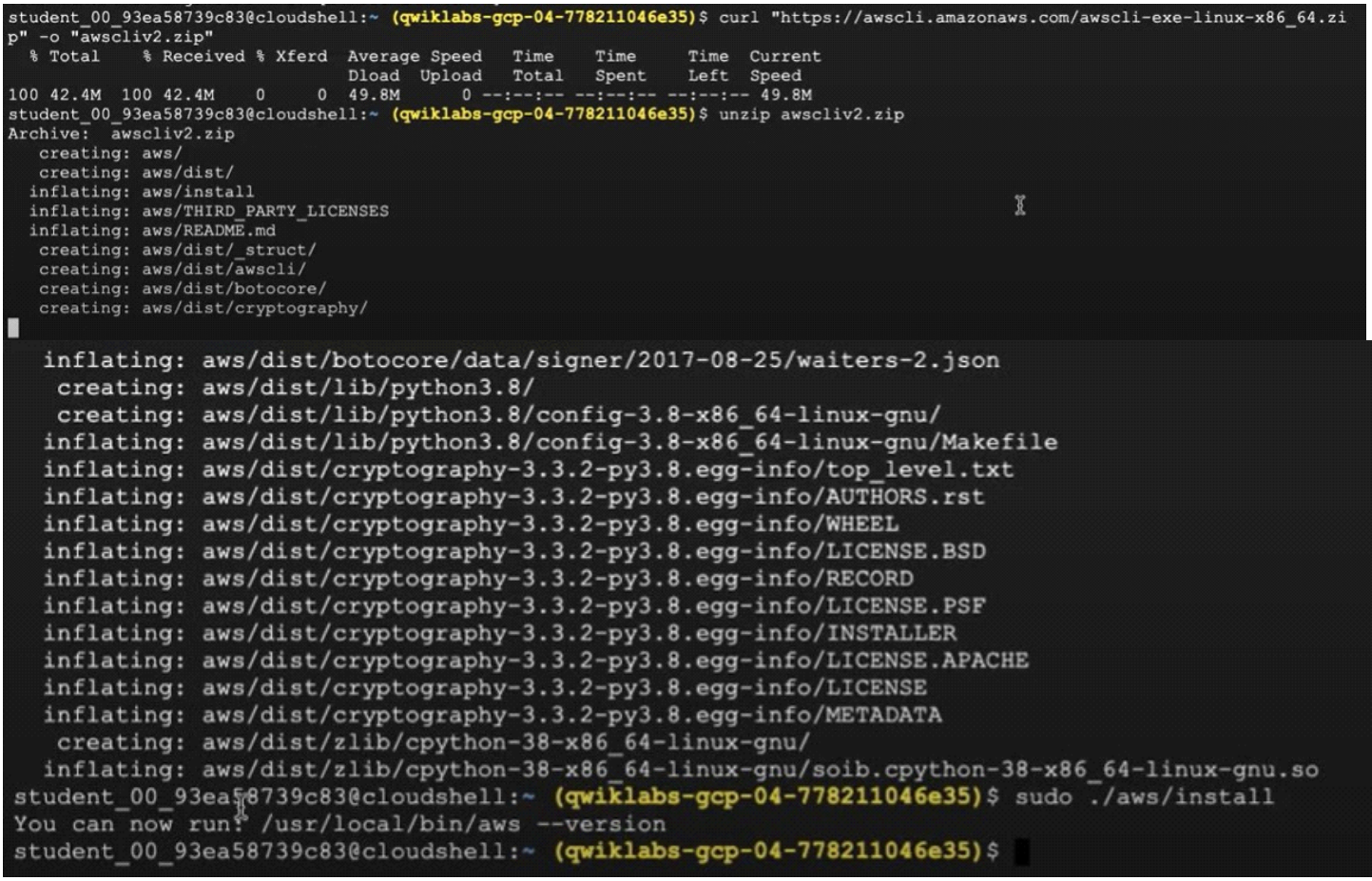
Task 1. Install and configure the AWS CLI tool in Cloud Shell

Although AWS configuration tasks can be completed in the AWS console, this lab uses the AWS CLI to complete these tasks in Cloud Shell. In this task, you install the AWS CLI tool in Cloud Shell to access the AWS resources from Google Cloud.

To install the AWS CLI tool in Cloud Shell, run the following commands:

- curl "<https://awscli.amazonaws.com/awscli-exe-linux- x86_64.zip>" -o "awscliv2.zip"
  unzip awscliv2.zip
  sudo ./aws/ install

Output:



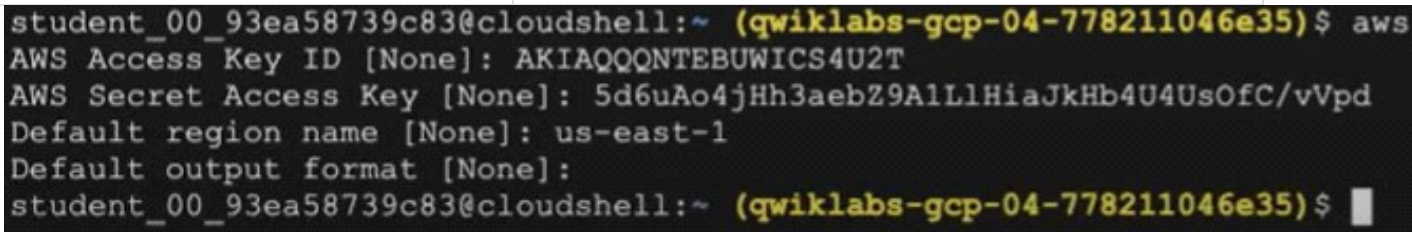
- To configure the AWS CLI tool in Cloud Shell, run the following command:

aws configure

Enter the required information to configure the AWS CLI tool:

Property	Value
AWS Access Key ID [None]	enter the provided value for AWS Access Key
AWS Secret Access Key [None]	enter the provided value for AWS Secret Key

Default region name [None]	us-east-1
Default output format [None]	to accept the default, do not enter a value

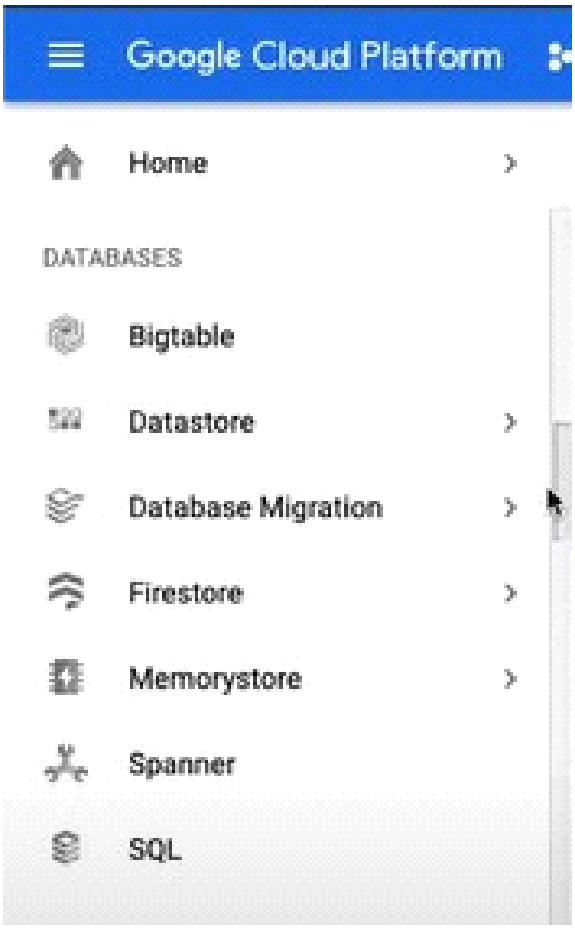


You have now configured the AWS CLI tool.

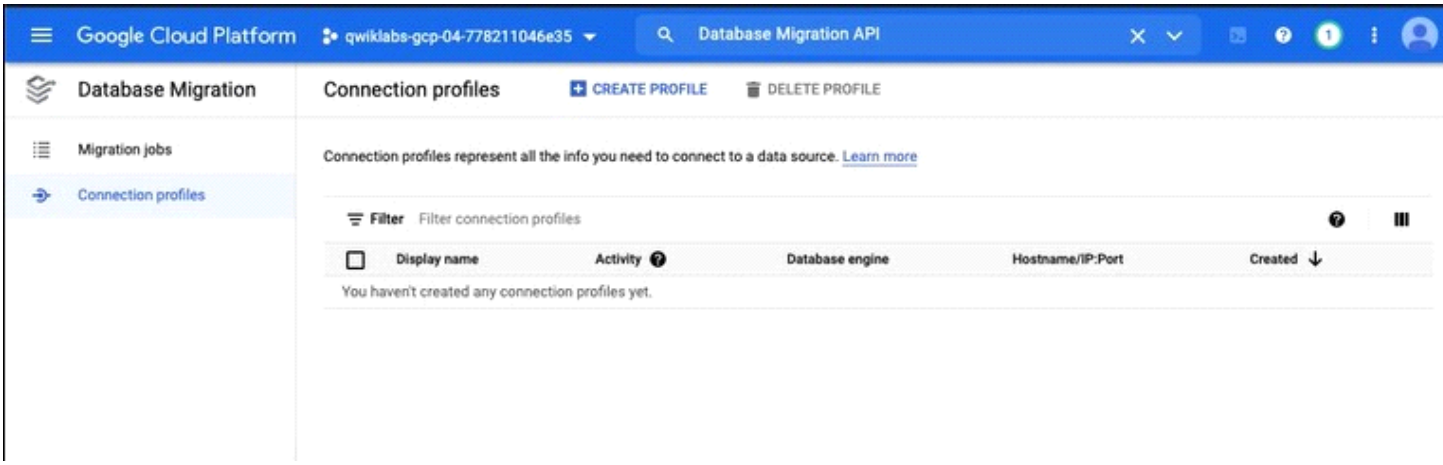
Task 2. Create a new connection profile for the Amazon RDS instance for MySQL

A connection profile stores information about the source database instance (such as Amazon RDS for MySQL) and is used by Database Migration Service to migrate data from the source to your destination Cloud SQL database instance. After you create a connection profile, it can be reused across migration jobs.

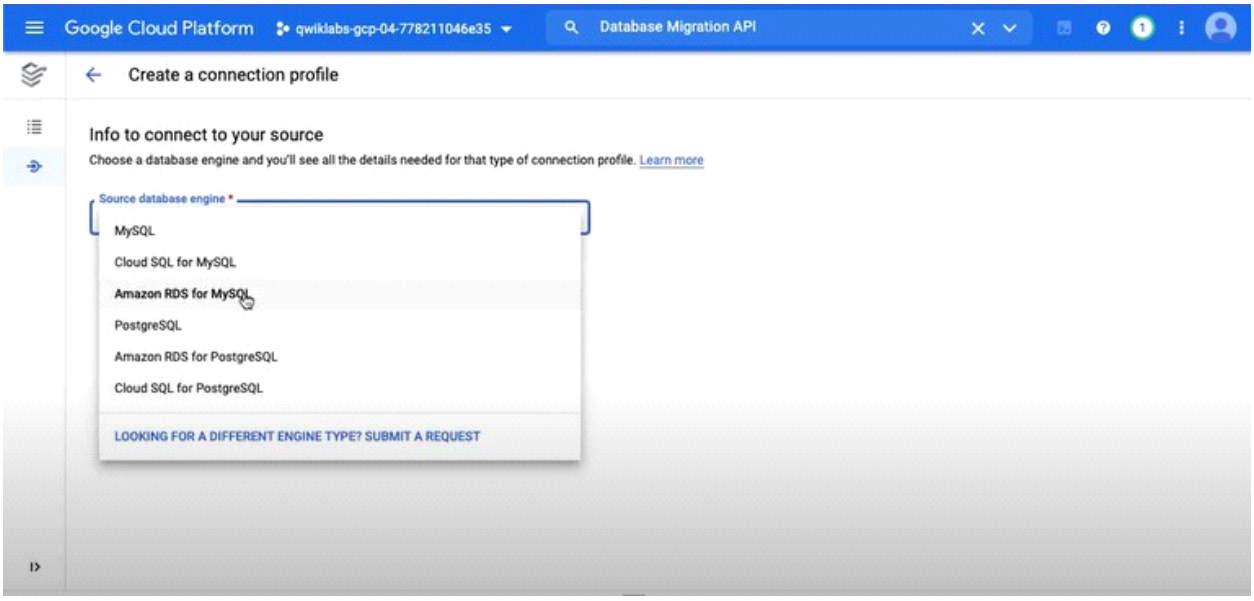
In the Google Cloud Console, on the Navigation menu ( ), click Database Migration > Connection profiles.



Click Create Profile.

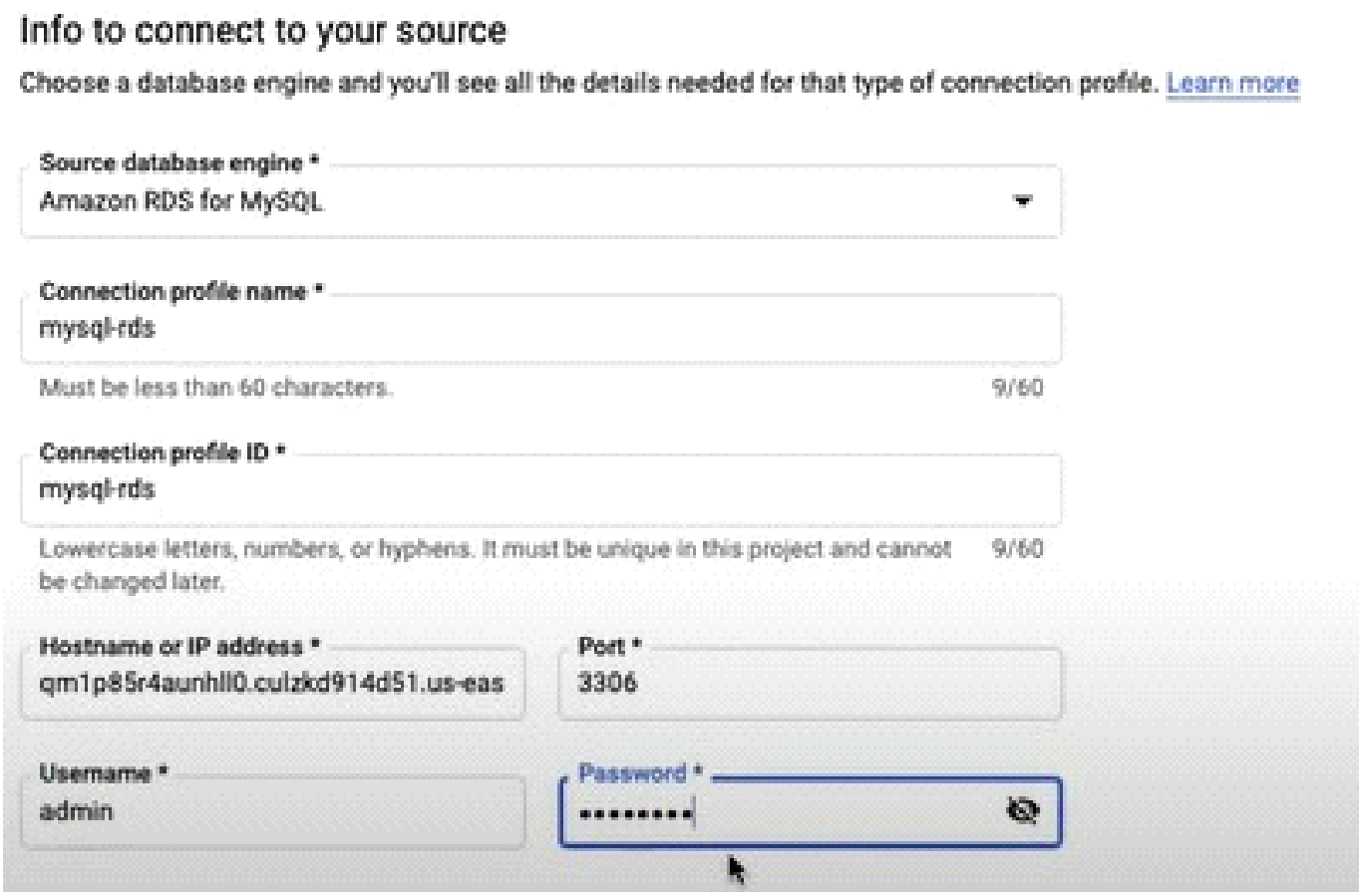


For Source database engine, select Amazon RDS for MySQL.



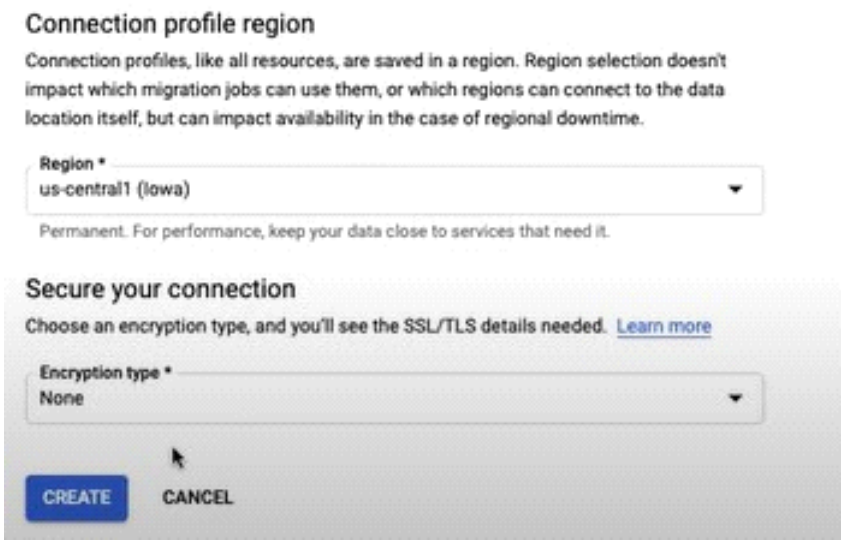
Enter the required information for a connection profile:

Property	Value
Connection profile name	mysql-rds
Connection profile ID	keep the auto-generated value
Hostname or IP address	enter the IP address you identified for the Amazon RDS instance in the Setup tasks (such as 54.84.181.60)
Port	3306
Username	admin
Password	changeme

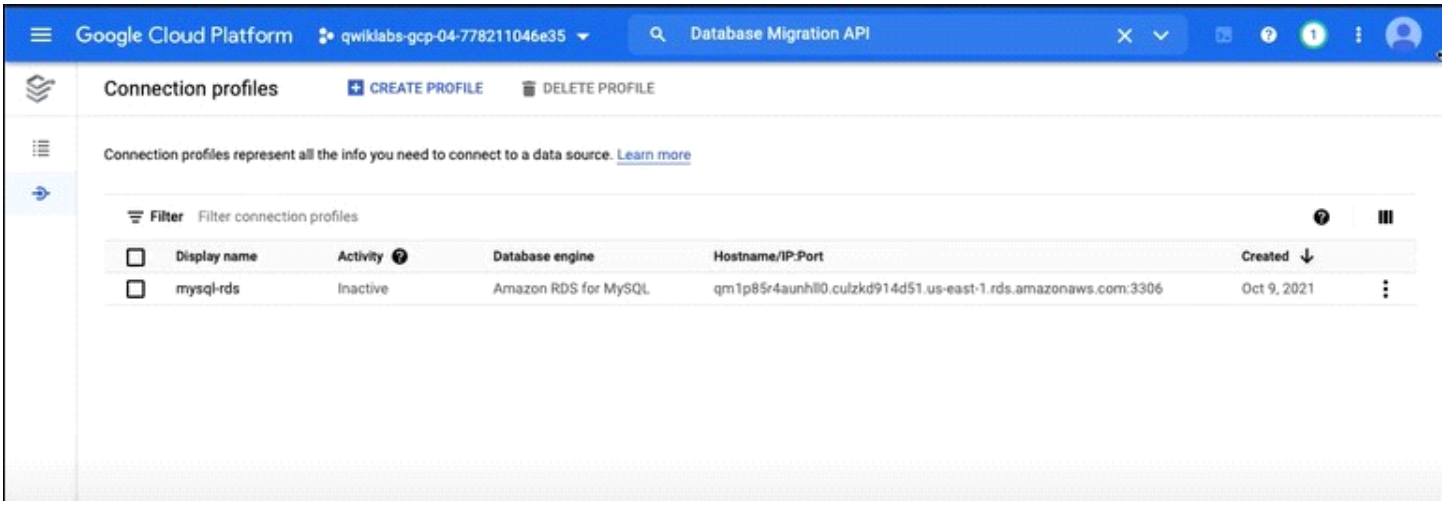


Because you will use an IP allowlist as the connectivity option, you are providing a public address for the source.

For Region, select us-central1 (Iowa). For the Encryption Type, select None. Click Create.



A new connection profile named mysql-rds will appear in the Connections profile list.



Task 3. Create a one-time migration job

When you create a new migration job, you first define the source database instance using a previously created connection profile.
Then, you create a new destination database instance and configure connectivity between the source and destination instances.

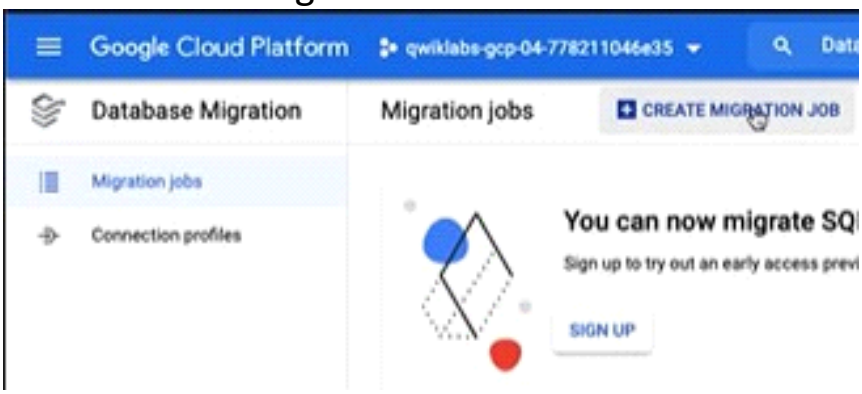
In this task, you will use the migration job interface to

create a new Cloud SQL for MySQL instance and set it as the destination for the one-time migration job from the source Amazon RDS for MySQL instance.

- Create a new migration job

In the Google Cloud Console, on the Navigation menu ( ), click Database Migratio

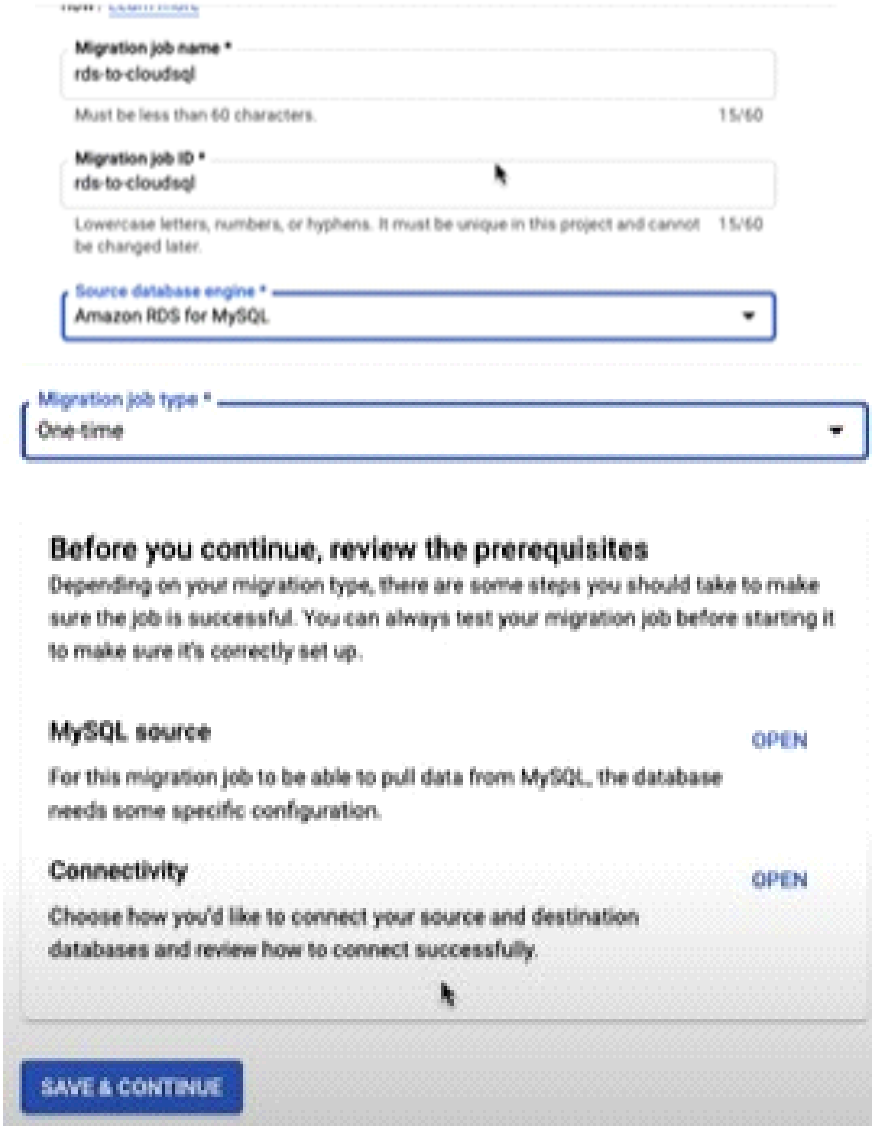
Click Create Migration Job.



For Create a migration job, on the Get Started tab, use the following values:

Property	Value
Migration job name	rds-to-cloudsql
Migration job ID	keep the auto-generated value
Source database engine	Amazon RDS for MySQL
Destination region	us-central1 (Iowa)
Migration job type	One-time

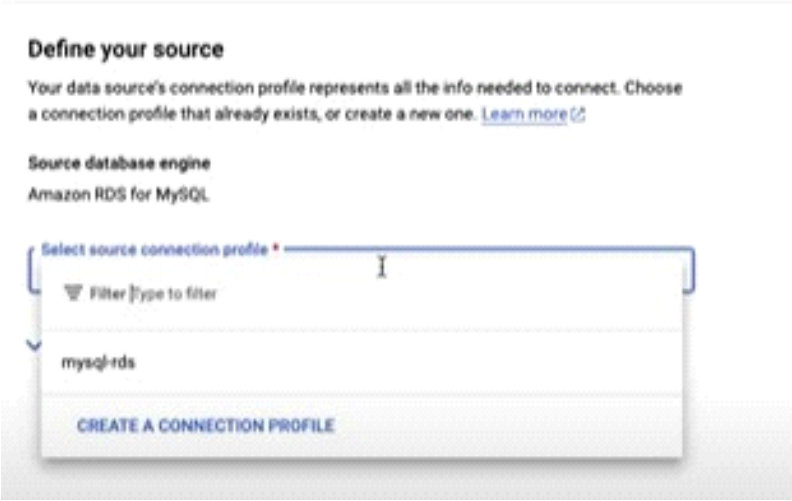
Leave all other settings as default.



Click Save & Continue.

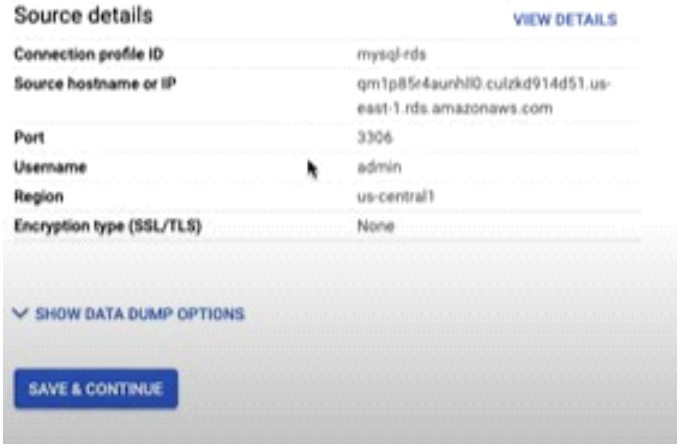
- Define the source instance

For source connection profile, select mysql-rds. Leave the defaults for the other settings.



After you select the source connection profile, you can see its configuration details, including source hostname or IP address, port, username, and encryption type.

Click Save & Continue.

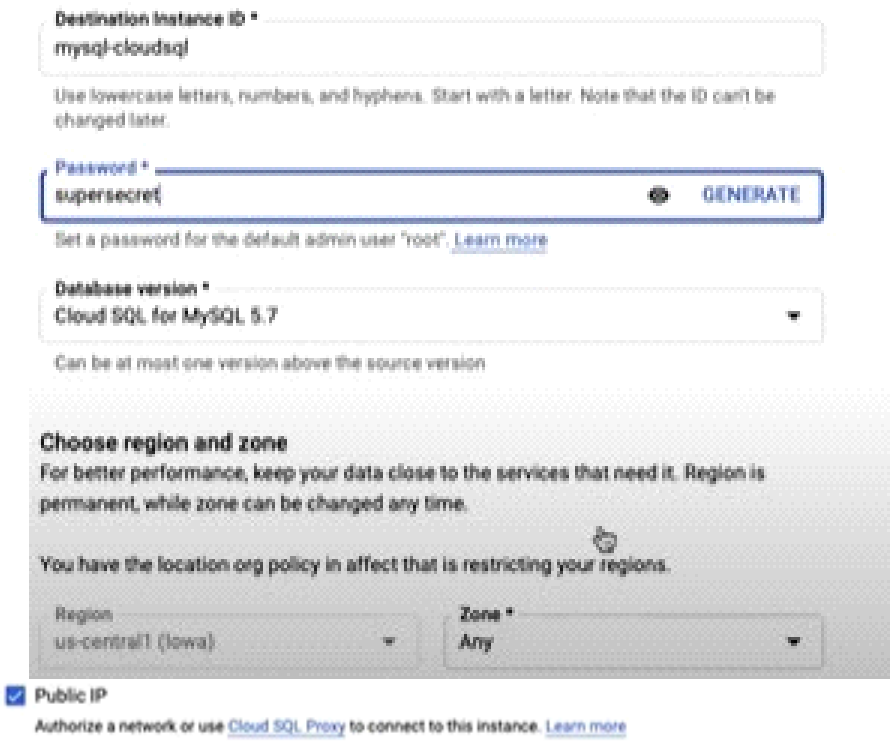


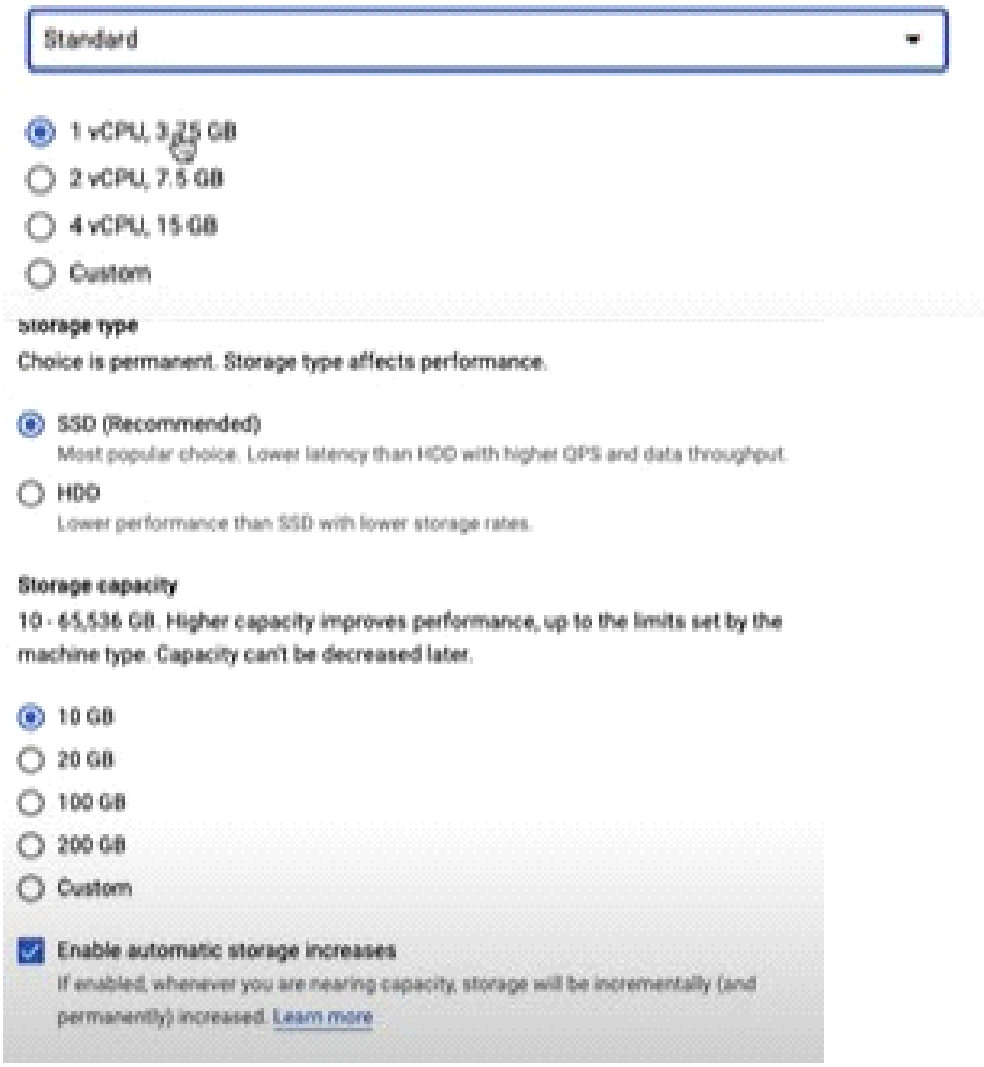
- Create the destination instance

Enter the required information to create the destination instance on Cloud SQL:

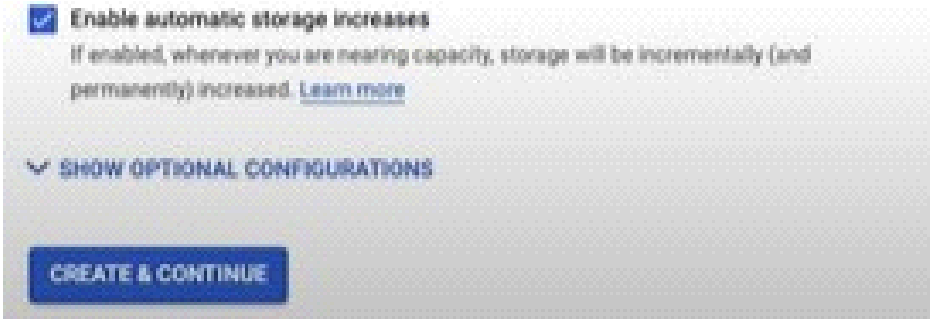
Property	Value
Destination Instance ID	mysql-cloudsql

Root password	supersecret
Database version	Cloud SQL for MySQL 5.7
Zone	Any
Connections	Public IP
Machine type	Standard (1 vCPU, 3.75 GB)
Storage type	SSD
Storage capacity	10





Click Create & Continue.

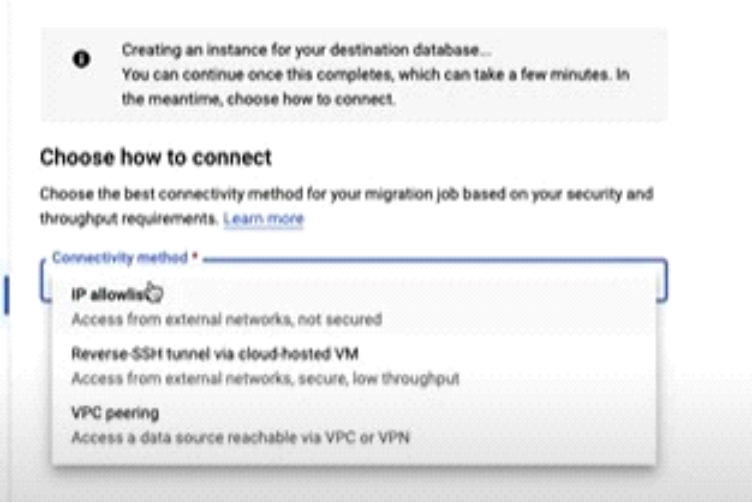


When prompted to confirm, click Create Destination & Continue.

- Define the connectivity method

A message will state that your destination database instance is being created. Continue to step 1 while you wait.

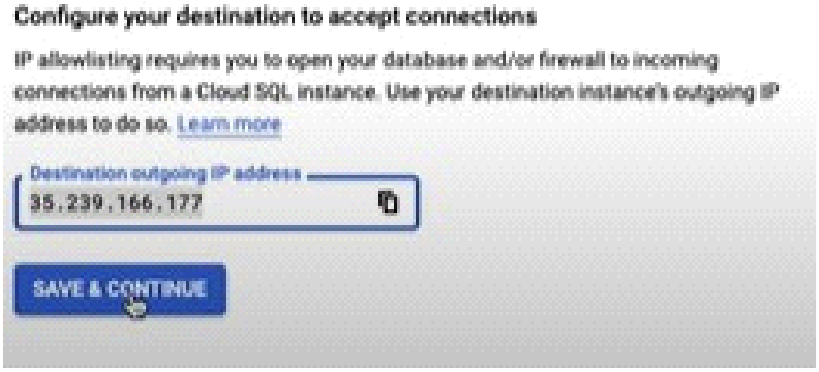
For Connectivity method, select IP allowlist.



When the destination database (Cloud SQL for MySQL) is created, the IP address will be available.

Copy the Destination outgoing IP address (such as 35.239.140.158) to configure the IP allowlist on the Amazon RDS instance.

Click Save & Continue.



Leave this window open. You will modify the IP allowlist on the Amazon RDS instance in the next task.

Task 4. Configure the IP allowlist on source instance

To allow connections between the source and destination instances, you need to modify the IP allowlist on the source. For Amazon RDS,
you accomplish this by modifying a Database Security Group with the public IP address of the destination instance (such as Cloud SQL).

Although this task can be completed in the AWS console, you will use the AWS CLI to complete this task in Cloud Shell.

To modify the IP allowlist on the Amazon RDS instance, run the following command in Cloud Shell:

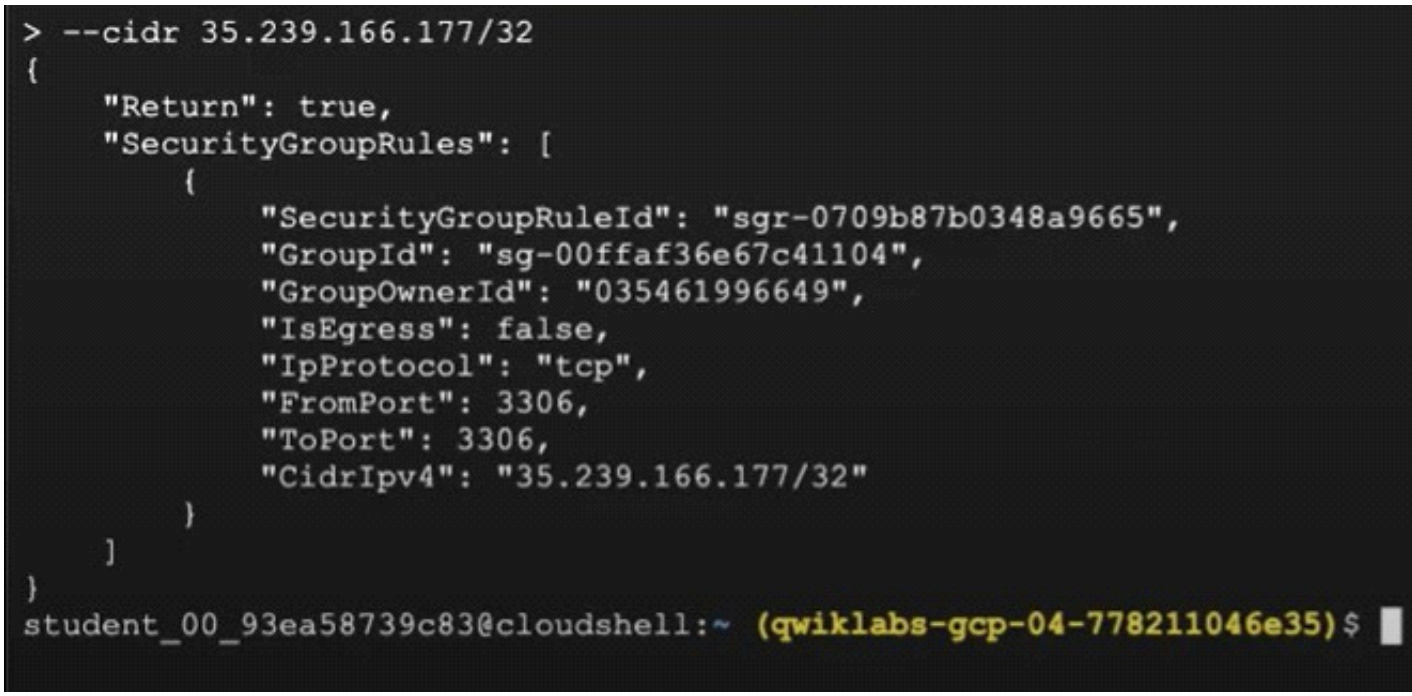
Replace <sg_id> with the provided value for AWS RDS Database Security Group (such as sg-06700713f70076ad8) on this page, and replace <a.b.c.d> with the Destination outgoing IP address of your Cloud SQL instance (such as 35.239.140.158).

- aws ec2 authorize-security-group-ingress \
  --group-id <sg-id> \
  --protocol tcp \
  --port 3306 \

```
student_00_93ea58739c83@cloudshell:~ (qwiklabs-gcp-04-778211046e35)$ aws ec2 authorize-se
> --group-id sg-00ffaf36e67c41104 \
> --protocol tcp \
>     --port 3306 \
> --cidr 35.239.166.177/32
```

  --cidr <a.b.c.d>/32

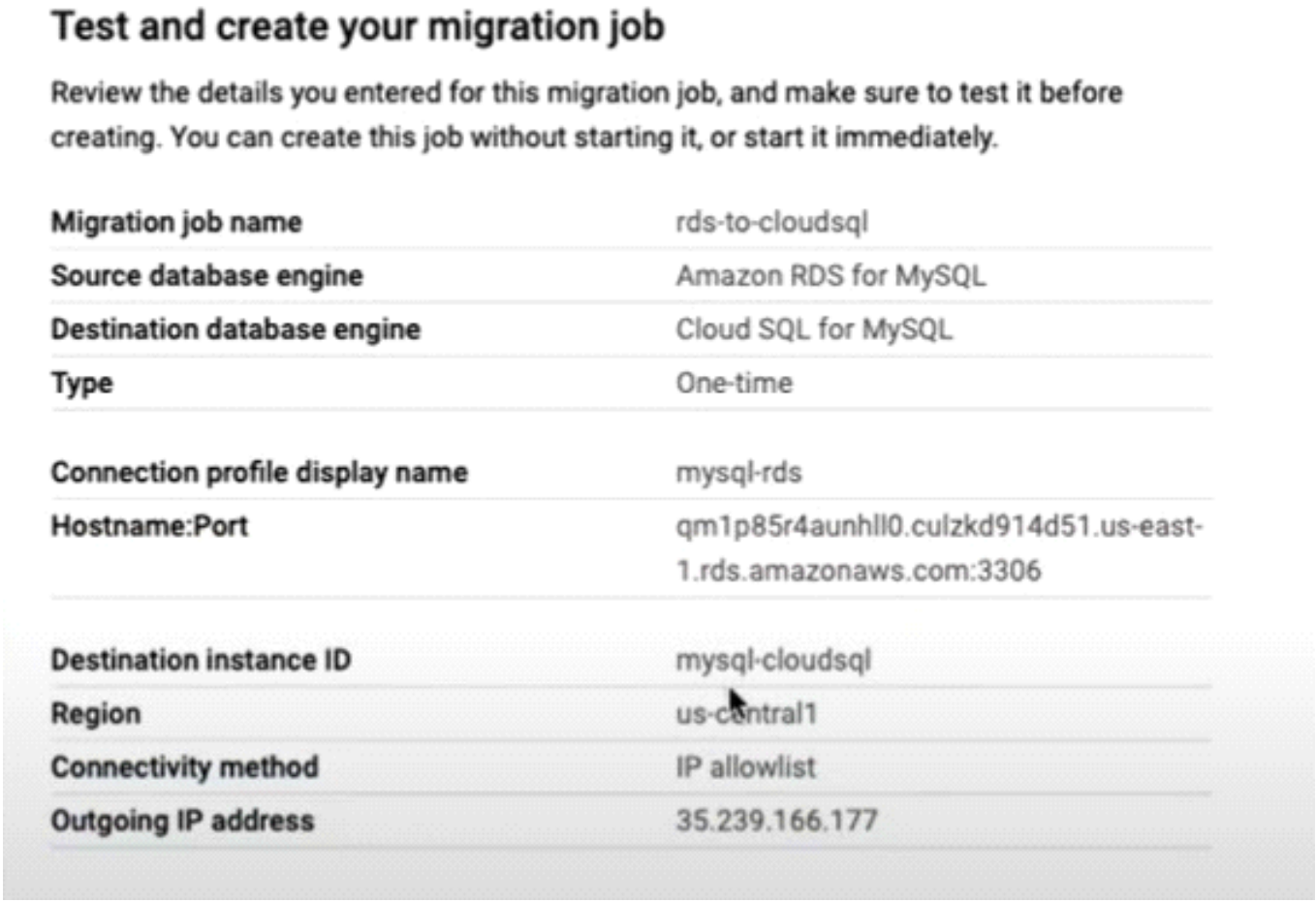
Output:



The IP address for the Cloud SQL instance has now been added to the IP allowlist on the Amazon RDS instance.

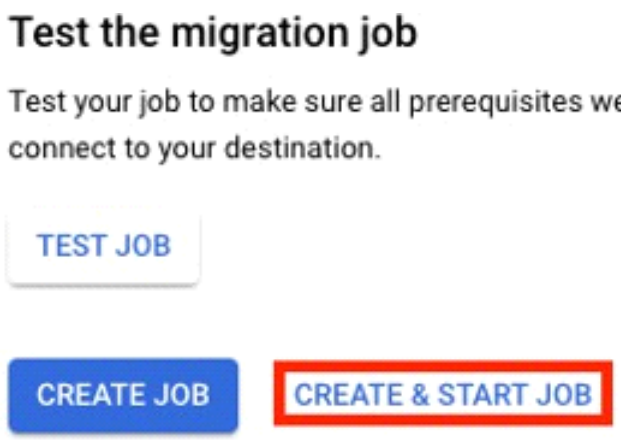
Task 5. Test and run a one-time migration job

Return to the migration job window and review the summary.



Click Test Job.

After a successful test, click Create & Start Job.



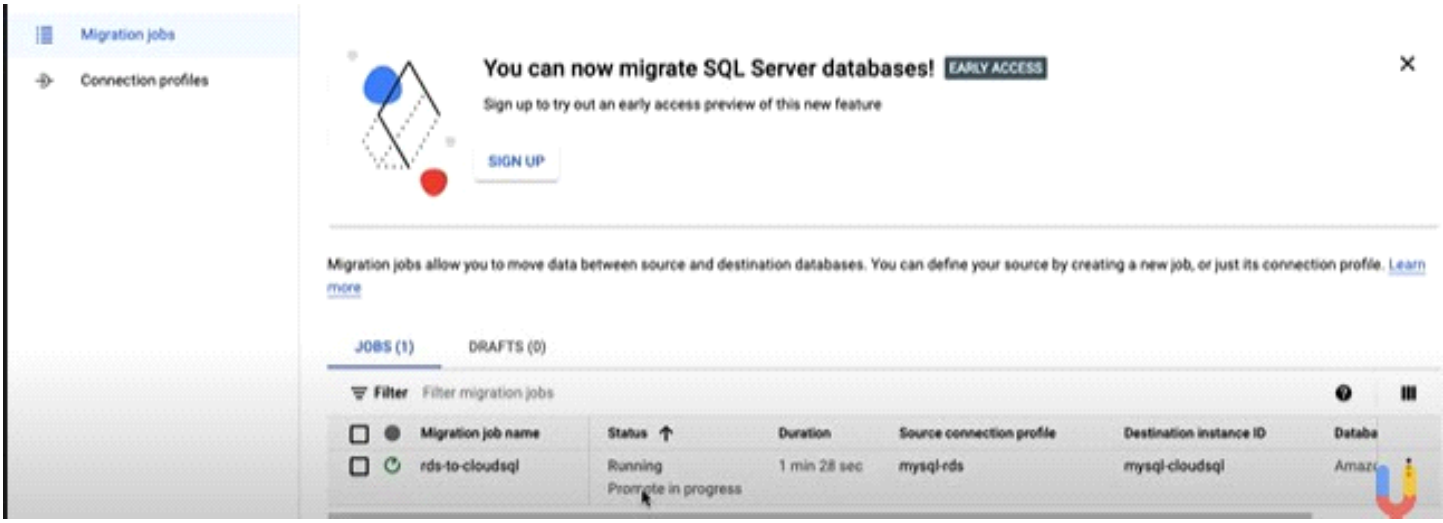
Note: Be sure to click on the button for CREATE & START

JOB to ensure that the job is successfully started.

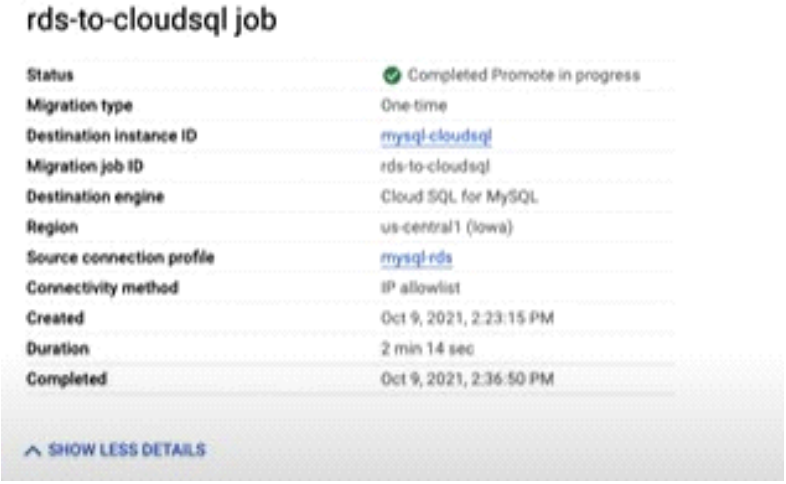
If prompted to confirm, click Create & Start.

Task 6. Review status of the one-time migration job in DMS

In the Google Cloud Console, on the Navigation menu ( ), click Database Migratio



Click the migration job rds-to-cloudsql to see the details page. Review the migration job status.



If you have not started the job, the status will show as Not started. You can choose to start or delete the job.

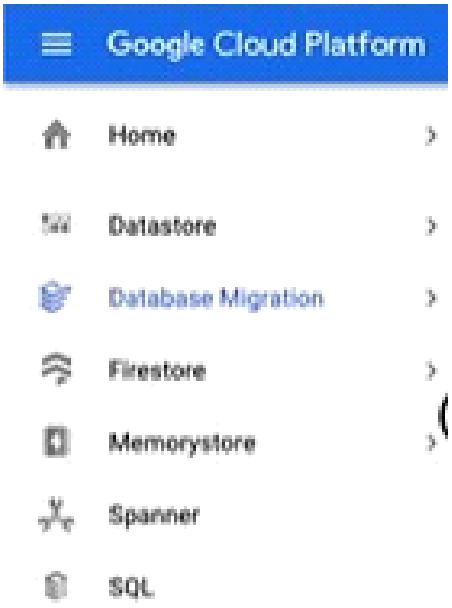
After the job has started, the status will show as Starting and then transition to Running.

When the job status changes to Completed, the migration job has completed successfully, and you can move on to the next task.

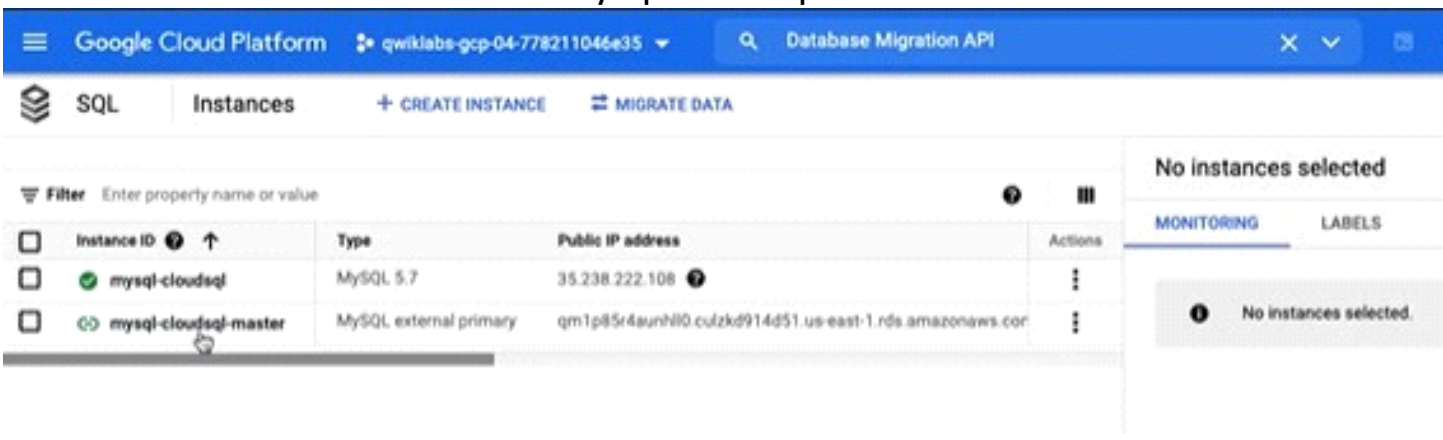
Task 7. Confirm the data in Cloud SQL for MySQL

Check MySQL databases in Cloud SQL

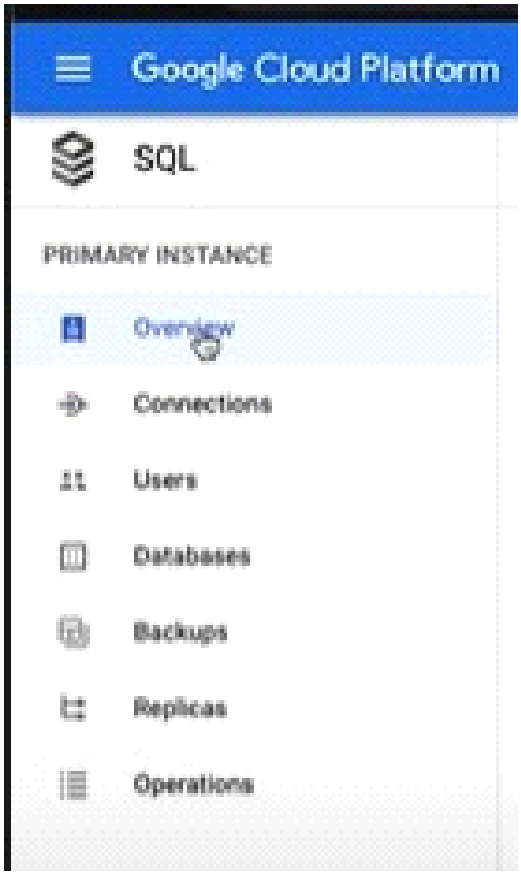
In the Google Cloud Console, on the Navigation menu ( ), click Databases > SQL.



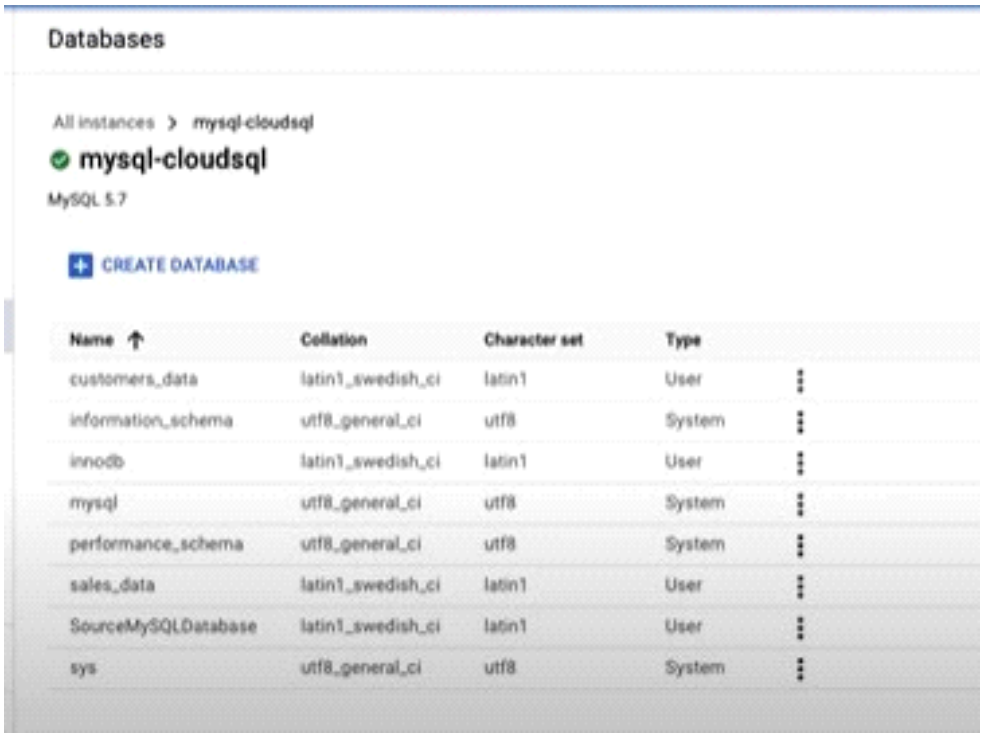
Click on the instance ID called mysql-cloudsql.



In the Primary Instance menu, click Databases.



Notice that the databases called customers_data and sales_data have been migrated to Cloud SQL.



Connect to MySQL database

In the Primary Instance menu, click Overview.

In Connect to this instance panel, click on Open cloud shell button.

The command to connect to MySQL will pre-populate in Cloud Shell:

- Run the above command.

gcloud sql connect mysql-cloudsql
--user=root --quiet If prompted, click
Authorize for the API.
When prompted for a password, which you previously set,
enter: supersecret

```
student_00_93ea58739c83@cloudshell:~ (qwiklabs-gcp-04-778211046e35)$ gcloud sql connect mysql-cloudsql --us
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 5.7.34-google-log (Google)

Copyright (c) 2000, 2021, 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>
```

You have now activated the MySQL interactive console. Review data in Cloud SQL for MySQL database

To select the database in the MySQL interactive console, run the following command:

```
mysql> use customers_data;
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> select count(*) from customers;
+-----------+
| count(*)  |
+-----------+
|      5030 |
+-----------+
1 row in set (0.20 sec)

mysql>
```

There are 5,030 records in the customers table that was migrated from the Amazon RDS instance for MySQL.

Exit the MySQL interactive console:

exit

```
mysql> exit
Bye
student_00_93ea58739c83@cloudshell:~ (qwiklabs-gcp-04-778211046e35)$
```

End Lab

End the Lab in the website where we started.