# Set Up a Google Cloud Network: Challenge Lab

experimentLabschedule1 hour 30 minutesuniversal\_currency\_alt5 Creditsshow\_chartIntermediate

Rate Lab

infoThis lab may incorporate AI tools to support your learning.

## GSP314



## Overview

In a challenge lab you’re given a scenario and a set of tasks. Instead of following step-by-step instructions, you will use the skills learned from the labs in the course to figure out how to complete the tasks on your own! An automated scoring system (shown on this page) will provide feedback on whether you have completed your tasks correctly.

When you take a challenge lab, you will not be taught new Google Cloud concepts. You are expected to extend your learned skills, like changing default values and reading and researching error messages to fix your own mistakes.

To score 100% you must successfully complete all tasks within the time period!

This lab is recommended for students who have enrolled in the [Set up a Google Cloud Network](https://www.cloudskillsboost.google/course_templates/641) skill badge. Are you ready for the challenge?

## Setup

### Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

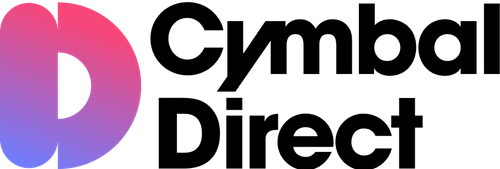
* Access to a standard internet browser (Chrome browser recommended).

**Note:** Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

* Time to complete the lab---remember, once you start, you cannot pause a lab.

**Note:** If you already have your own personal Google Cloud account or project, do not use it for this lab to avoid extra charges to your account.

## Challenge scenario



You are a cloud engineer working for Cymbal Direct, an online direct-to-consumer footwear and apparel retailer. Cymbal Direct has been scaling rapidly, and as part of their scaling strategy they have acquired a new company, Antern. Antern is an existing ecommerce shop with a large amount of data that, if properly integrated into your existing cloud environment, will be incredibly useful to further scale the business and reach more customers. As part of this acquisition, Cymbal Direct wants to move Antern’s existing workloads and infrastructure from on-prem, to Google Cloud.

Antern has the following resources that need to be migrated, copied, or recreated in Cymbal Direct’s existing cloud environment:

* A PostgreSQL database (running on a virtual machine) that needs to be migrated to Cloud SQL for PostgreSQL
* Containerized microservices application code to deploy on GKE (with reported reliability issues during testing that need to be troubleshooted)
* A VPC network with two subnetworks and firewalls that need to be created to connect new resources together
* IAM users across multiple projects that need to be granted the proper permissions and roles on specific resources

You are tasked with helping Cymbal Direct achieve these goals.

## Task 1: Migrate a stand-alone PostgreSQL database to a Cloud SQL for PostgreSQL instance

Antern has been using a PostgreSQL database running on an on-prem VM to store ecommerce shop orders. As part of the acquisition strategy, Cymbal has requested that this database be migrated to Cloud SQL for PostgreSQL using [Database Migration Service](https://cloud.google.com/database-migration). Having the database running on Cloud SQL allows Cymbal to get all the operational benefits of PostgreSQL with added enterprise availability, stability, and security.

In this task you must migrate the stand-alone PostgreSQL orders database running on the antern-postgresql-vm virtual machine to a Cloud SQL for PostgreSQL instance using a Database Migration Services continuous migration job and VPC Peering connectivity.

### Prepare the stand-alone PostgreSQL database for migration

**Note:** For the first task, you will need to log in to the **Antern Project** with the **Antern Owner** credentials.

In this sub-task you must prepare the stand-alone PostgreSQL database so that it satisfies the requirements for migration by Database Migration Services.

To complete this sub-task you must complete the following steps:

1. Enable the Google Cloud APIs required for Database Migration Services.

Database Migration Services require the **Database Migration API** and the **Service Networking API** to be enabled in order to function. You must enable these APIs for your project.

1. Upgrade the target databases on the antern-postgresql-vm virtual machine with the pglogical database extension.

You must install and configure the **pglogical** database extension on the stand-alone PostgreSQL database on the antern-postgresql-vm Compute Instance VM. The pglogical database extension package that you must install is named postgresql-13-pglogical.

To complete the configuration of the **pglogical** database extension you must edit the PostgreSQL configuration file /etc/postgresql/13/main/postgresql.conf to enable the **pglogical** database extension and you must edit the /etc/postgresql/13/main/pg\_hba.conf to allow access from all hosts.

1. Create a dedicated user for database migration on the stand-alone database.

The new user that you create on the stand-alone PostgreSQL installation on the antern-postgresql-vm virtual machine must be configured using the following user name and password:

* **Migration user name** : import\_user
* **Migration user password** : DMS\_1s\_cool!

1. Grant that user the required privileges and permissions for databases to be migrated.

Database Migration Services require that the migration user have privileges to specific schemata and relations of the target databases for migration, in this case that is the orders and postgres databases.

1. The Database Migration Service requires that all tables to be migrated have a primary key. Once you have granted the user the required privileges, run the following to add a primary key to the inventory\_items table and exit psql.

ALTER TABLE public.inventory\_items ADD PRIMARY KEY(id);

\q

exit

Copied!

content\_copy

**Note:** The detailed prerequisites for migrating a stand-alone PostgreSQL database to Cloud SQL for PostgreSQL are provided in the suggestion links in the Cloud Console GUI for Database Migration Services. Should you need some help on the detailed steps you must take, refer to that documentation, or you can look at the detailed steps in the [Migration Lab](https://www.cloudskillsboost.google/authoring/labs/4198) that is part of this skill badge.

Click **Check my progress** to verify the objective.

Success: Source database prepared for migration.

Prepare a stand-alone PostgreSQL database for migration to Cloud SQL for PostgreSQL.

Check my progress

*Success: Source database prepared for migration.*

### Migrate the stand-alone PostgreSQL database to a Cloud SQL for PostgreSQL instance

In this sub-task you must perform the migration using Database Migration Services.

To complete this sub-task you must complete the following steps:

1. Create a new Database Migration Service connection profile for the stand-alone PostgreSQL database, using the credentials of the import\_user migration user you created earlier.

* **Username** : import\_user
* **Password** : DMS\_1s\_cool!
* **Region** : us-central1

You must configure the connection profile using the internal ip-address of the source compute instance.

1. Create a new **continuous** Database Migration Service job.

* For **Migration job name**, enter b2b-postgres34
* For **Source database engine** select PostgreSQL.
* For **Destination database engine** select Cloud SQL for PostgreSQL.
* **Destination region** : us-central1

As part of the migration job configuration, make sure that you specify the following properties for the destination Cloud SQL instance:

* The **Destination Instance ID** must be set to b2b-postgres34
* The **Password** for the migrated instance must be set to supersecret!
* **Database version** must be set to **Cloud SQL for PostgreSQL 13**
* Choose Cloud SQL edition **Enterprise** only.
* **Region** must be set to us-central1
* For **Connections** both **Public IP** and **Private IP** must be set.
* For Private IP use an automatically allocated IP range and click **Allocate & Connect**.
* Select a standard machine shape with **4 vCPU, 16 GB**
* For **Storage type**, use **SSD**
* Set the storage capacity to **10 GB**

For the **Connectivity Method**, you must use **VPC peering** with the **default** VPC network.

1. Test and then start the continuous migration job.

**Note:**If you do not correctly prepare the source PostgreSQL environment, the migration might fail completely, or it might fail to migrate some individual tables. If some tables are missing, even though the migration appears to be working otherwise, check that you have correctly configured all of the source database tables.

Click **Check my progress** to verify the objective.

Success: Continuous migration PostgreSQL database to Cloud SQL was successful.

Migrate a stand-alone PostgreSQL database to a Cloud SQL for PostgreSQL instance.

Check my progress

*Success: Continuous migration PostgreSQL database to Cloud SQL was successful.*

### Promote a Cloud SQL to be a stand-alone instance for reading and writing data

* In this task, you must complete the migration by promoting the Cloud SQL for PostgreSQL instance to a stand-alone instance.
* When the promotion is complete, the status of the job updates to Completed.

Click **Check my progress** to verify the objective.

Step Completed!

Promote a Cloud SQL Migration Replica to a stand-alone instance.

Check my progress

*Step Completed!*

**Note:** For the lab's purposes, you don't need to wait for the migration to be done to promote it to a stand-alone instance. If the action is available in the Console you can run it.

## Task 2: Update permissions and add IAM roles to users

Now that the database has been migrated to a Cloud SQL for PostgreSQL instance, you will need to update user roles via IAM for different members on the Antern and Cymbal teams. Specifically, you want to grant the **Antern Editor** user access to the Cloud SQL database, the **Cymbal Owner** admin access to have full control of Cloud SQL resources, and the **Cymbal Editor** to have editor permissions on the project.

**Note:** For this task, you will need to log in to the **Antern Project** with the **Antern Owner** credentials.**Hint:** Grant the desired roles to the required users using **Cloud IAM**.

1. Grant the **Antern Editor** user the **Cloud SQL Instance User** role for the CloudSQL database. Their username is: student-04-09b8a3c0447d@qwiklabs.net.
   * Navigate to the Cloud SQL database you just created. In the **Users** section, add the **Antern Editor** user account to the database you created. Use **Cloud IAM authentication** and for the principal use their username above.
2. Grant the **Cymbal Owner** user the **Cloud SQL Admin** role for the CloudSQL database. Their username is: student-02-e7544f81bee8@qwiklabs.net.
   * Navigate to the Cloud SQL database you just created. In the **Users** section, add the **Cymbal Owner** user account to the database you created. Use **Cloud IAM authentication** and for the principal use their username above.
3. Change the **Cymbal Editor** user role from **Viewer** to **Editor**. Their username is student-02-02a29db28cc4@qwiklabs.net.

Click **Check my progress** to verify the objective.

Assessment Completed!

Update permissions and add IAM roles to users

Check my progress

*Assessment Completed!*

## Task 3: Create networks and firewalls

As part of the acquisition strategy, a VPC network to connect resources internally needs to be recreated in the Cymbal project. Specifically, you will need to create a VPC network with two subnets and firewalls to open connections between resources. Additionally, on this network your team will need to be able to connect to Linux and Windows machines using SSH and RDP, as well as diagnose network communication issues via ICMP.

In this task, you will create a VPC network and firewall rules that satisfy these requirements.

**Note:** For this task, you will need to log in to the **Cymbal Project** with the **Cymbal Owner** credentials.

### Create a VPC network with two subnetworks

1. Create a VPC network named vpc-network-n2yg with two subnets: subnet-a-9j21 and subnet-b-5yw3. Use a **Regional** dynamic routing mode.
2. For subnet-a-9j21 set the region to us-west1.
   * Set the **IP stack type** to **IPv4 (single-stack)**
   * Set IPv4 range to 10.10.10.0/24
3. For subnet-b-5yw3 set the region to us-east4.
   * Set the **IP stack type** to **IPv4 (single-stack)**
   * Set IPv4 range to 10.10.20.0/24

### Create firewall rules for the VPC network

1. Create a firewall rule named hhij-firewall-ssh.
   * For the network, use vpc-network-n2yg.
   * Set the priority to **65535**, the traffic to **Ingress** and action to **Allow**
   * The targets should be set to all instances in the network and the IPv4 ranges to 0.0.0.0/24
   * Set the Protocol to **TCP** and port to 22
2. Create a firewall rule named mspz-firewall-rdp.
   * For the network, use vpc-network-n2yg.
   * Set the priority to **65535**, the traffic to **Ingress** and action to **Allow**
   * The targets should be set to all instances in the network and the IPv4 ranges to 0.0.0.0/24
   * Set the Protocol to **TCP** and port to 3389
3. Create a firewall rule named ricg-firewall-icmp.
   * For the network, use vpc-network-n2yg.
   * Set the priority to **65535**, the traffic to **Ingress** and action to **Allow**
   * The targets should be set to all instances in the network and the IPv4 ranges to 0.0.0.0/24
   * Set the Protocol to **icmp**

Click **Check my progress** to verify the objective.

Assessment Completed!

Create networks and firewalls

Check my progress

*Assessment Completed!*

## Task 4: Troubleshoot and fix a broken GKE cluster

**Note:** For this task, you will need to log in to the **Cymbal Project** with the **Cymbal Owner** credentials.

After deploying the e-commerce website GKE cluster, your team has notified you that there are a few known issues with the GKE cluster that need to be addressed. They have found three bugs that need to be fixed:

* **Bug #1**: Too much latency of the frontend service
* **Bug #2**: Ratings have become stale
* **Bug #3** Crashing bug in the recommendation service

As part of the acquisition strategy, you have been assigned to fix Bug #2. Other engineers working on your team have provided some additional information for each of the issues they have found, which you can use to troubleshoot the issue.

### Hints:

* **Bug #1**: Visit the external IP of the demo application to see if there are any visible changes. You can also use monitoring dashboards to see metrics associated with each service.
* **Bug #2**: Product ratings are managed by the "rating service", hosted on Google AppEngine. The rating data is kept up-to-date by periodically calling an API endpoint that collects all recently sent new rating scores for each product and calculates the new rating. Try to check if the rating service operates normally by inspecting the logs from the AppEngine service. Another team member mentioned this might have to do with an issue in the main.py file.
* **Bug #3**: Browse your website until you encounter an issue, and use Cloud Logging to view logs exported by each service. Another team member mentioned this crashing bug might be due to an integer conversion stage in the service.

### Create a BigQuery log sink

Before fixing the underlying issue, you have been requested to create a log sink to send out the errors associated with the broken service. You will then need to use IAM to give users from Antern different levels of access to BigQuery so they can view and interact with the dataset.

1. Use the **Logs Explorer** to investigate your running GKE app and investigate the service that has errors. Hint: you should be looking for logs with severity ERROR.
2. Once you have identified the service error logs, create a sink to send the logs out to BigQuery.
   * Name the sink ratings-service-error-sink
   * For the destination, create a BigQuery dataset named gke\_app\_errors\_sink with a location of **us (multiple regions in United States)**.
   * In your **inclusion filter**, make sure to include: resource.type, resource.labels.module\_id, and severity.

**Note:** If you are getting any permission error, so make sure you are log in to the **Cymbal Project** with the **Cymbal Owner** credentials.

1. Grant the **Antern Editor** user the **BigQuery Data Viewer** role for this project. Their username is: student-04-09b8a3c0447d@qwiklabs.net.
2. Grant the **Antern Owner** user the **BigQuery Admin** role for this project. Their username is: student-04-28c2e0f9f813@qwiklabs.net.

Click **Check my progress** to verify the objective.

Assessment Completed!

Create a BigQuery log sink

Check my progress

*Assessment Completed!*

### Fix the GKE cluster

Now that you have created a log sink in BigQuery for the errors in the service, some engineers on your team took a look and figured out the correct steps to fix the issue. In this task you will download the solution code and run it to fix the service in your GKE cluster.

1. Connect to the **cloud-ops-sandbox** GKE cluster and run the following commands to remediate the issue. Answer the verification questions when prompted.

git clone --depth 1 --branch csb\_1220 https://github.com/GoogleCloudPlatform/cloud-ops-sandbox.git

cd cloud-ops-sandbox/sre-recipes

./sandboxctl sre-recipes restore recipe2

./sandboxctl sre-recipes verify recipe2

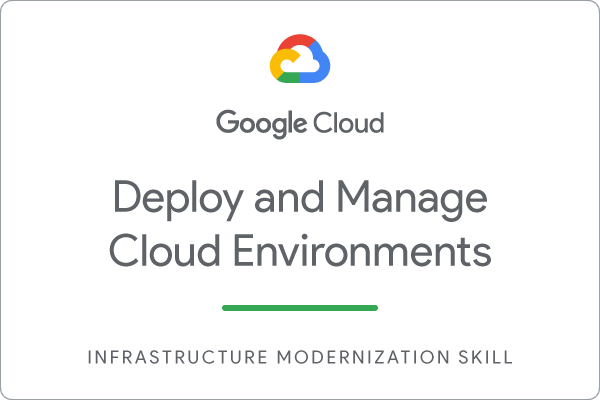
Copied!

content\_copy

1. Verify the e-commerce shop is properly working.

## Congratulations!

You have completed a days work as a Cloud Engineer! You migrated a PostgreSQL database to a Cloud SQL for PostgreSQL instance, created a VPC network with subnetworks and firewalls, troubleshooted issues with a GKE cluster, and granted IAM users permission to across multiple projects.



### Earn your next skill badge

This self-paced lab is part of the [Set up a Google Cloud Network](https://www.cloudskillsboost.google/course_templates/641) skill badge. Completing this skill badge earns you the badge above, to recognize your achievement. Share your badge on your resume and social platforms, and announce your accomplishment using #GoogleCloudBadge.