Securing Compute Engine Applications with BeyondCorp Enterprise

1 hour 15 minutes No cost

Overview

In this lab, you secure Compute Engine workloads using BeyondCorp Enterprise’s Identity-Aware Proxy (IAP) to restrict traffic based on identity.

IAP is a feature of BeyondCorp Enterprise, Google Cloud's zero-trust solution that enables an organization's workforce to access web applications securely from anywhere, without the need for VPN and without fear of malware, phishing, and data loss.

This lab provisions a web-based integrated development environment (IDE) that you will restrict access to by enabling zero-trust configuration.

Objectives

In this lab, you learn how to perform the following tasks:

- Configure OAuth Consent.
- Set up OAuth access credentials.
- Set up IAP access for the deployed application.
- Use IAP to restrict access to the application.

Setup and requirements

Before you click the Start Lab button

Note: 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 Qwiklabs 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.

What you need

To complete this lab, you need:

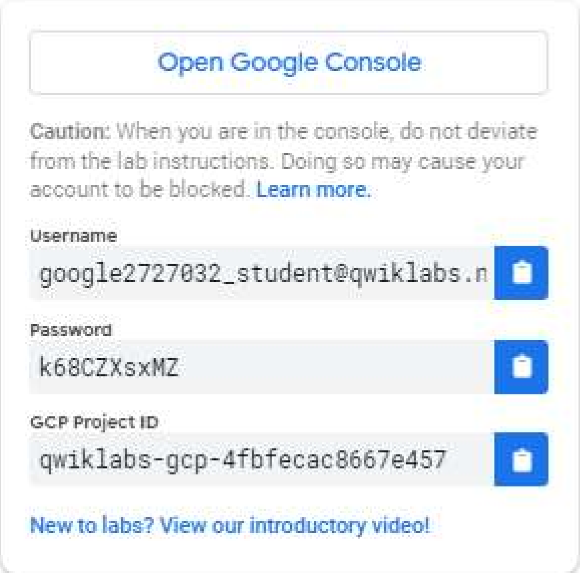
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

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

Note: If you are using a Pixelbook, open an Incognito window to run this lab.

How to start your lab and sign in to the Console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Username
google2727032_student@qwiklabs.n

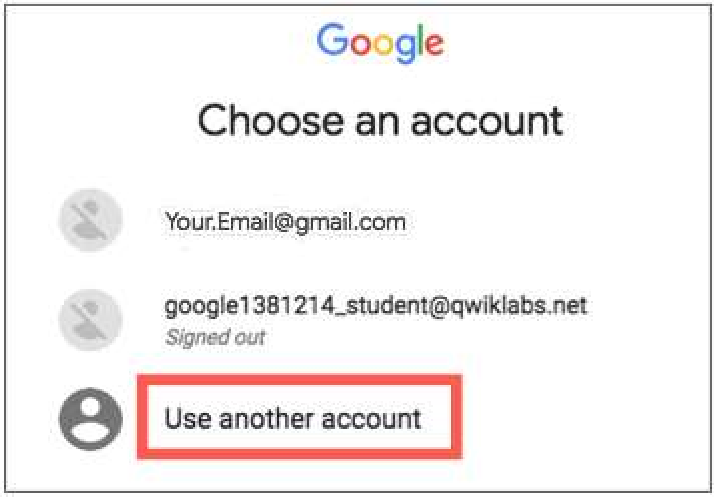
Password
k68CZXsxMZ

GCP Project ID
qwiklabs-gcp-4fbfecac8667e457

2. 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.

Note: Open the tabs in separate windows, side-by-side.

3. On the Choose an account page, click **Use Another Account**. The Sign in page opens.



Google

Choose an account

Your.Email@gmail.com

google1381214_student@qwiklabs.net
Signed out

Use another account

4. Paste the username that you copied from the Connection Details panel. Then copy and paste the password.

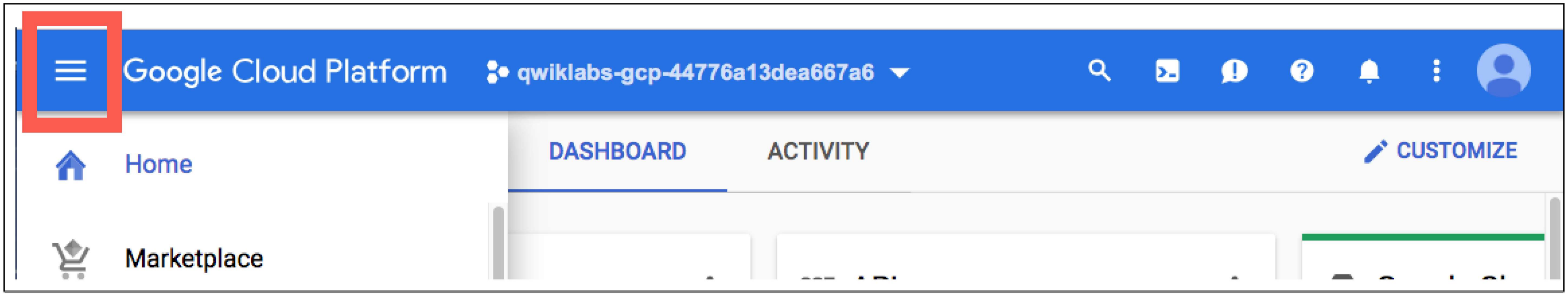
Note: You must use the credentials from the Connection Details panel. Do not use your Google Cloud Skills Boost credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

5. 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 Cloud console opens in this tab.

Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.

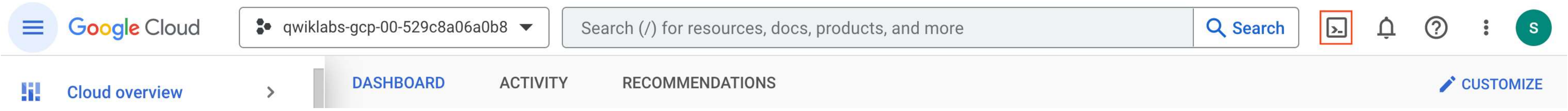


Activate Google Cloud Shell

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.

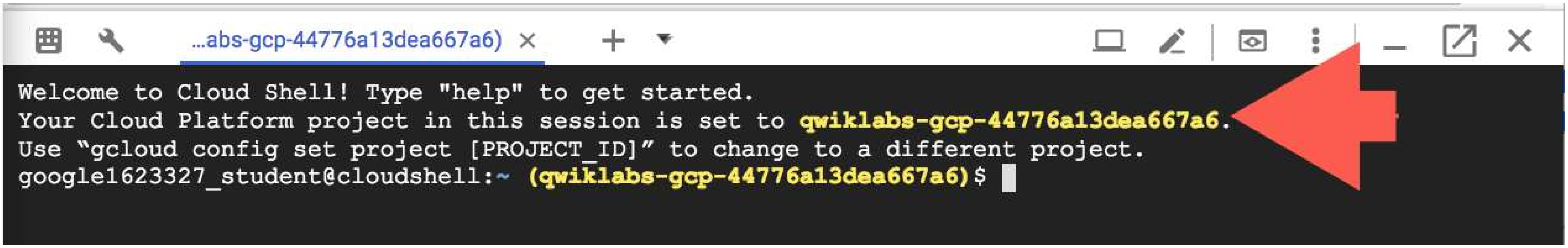
Google Cloud Shell provides command-line access to your Google Cloud resources.

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


Google Cloud
qwiklabs-gcp-00-529c8a06a0b8
Search (/) for resources, docs, products, and more
Search
S
Cloud overview
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE

2. 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:



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6)$
```

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

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

```
gcloud auth list
```

content_c

Output:

```
Credentialed accounts:
 - @.com (active)
```

Example output:

```
Credentialed accounts:
 - google1623327_student@qwiklabs.net
```

- You can list the project ID with this command:

content_c

```
gcloud config list project
```

Output:

```
[core]
project =
```

Example output:

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

Note: Full documentation of **gcloud** is available in the gcloud CLI overview guide .

Task 1. Create a Compute Engine template

In this task you create an instance template. This is a resource that you use to create virtual machine (VM) instances and managed instance groups (MIGs).

1. In the Google Cloud Console, in the **Navigation menu** (≡), click **Compute Engine > Instance templates**.

2. Click **Create Instance Template**.

3. On the **Create an instance template** page, specify the following, and leave the remaining settings as their defaults:

Property	**Value**
Series	**E2**
Machine type	**e2-micro (2 vCPU)**
Access scopes	**Set access for each API**
Access scopes > Compute Engine	**Read Only**
Firewall	**Allow HTTP traffic**

4. Click **Advanced options**.

5. Click **Management**.

6. In **Automation > Startup script**, copy and paste the following script:

content_c

```
# Copyright 2021 Google LLC
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.#
You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
```

```
# See the License for the specific language governing permissions and
# limitations under the License.
sudo apt-get -y update
sudo apt-get -y install git
sudo apt-get -y install virtualenv
git clone https://github.com/GoogleCloudPlatform/python-docs-samples
cd python-docs-samples/iap
virtualenv venv -p python3
source venv/bin/activate
pip install -r requirements.txt
cat example_gce_backend.py |
sed -e "s/YOUR_BACKEND_SERVICE_ID/$(gcloud compute backend-services
describe my-backend-service --global --format="value(id)")/g" |
    sed -e "s/YOUR_PROJECT_ID/$(gcloud config get-value account | tr -
cd "[0-9]")/g" > real_backend.py
gunicorn real_backend:app -b 0.0.0.0:80
```

7. Click **Create**.

It takes a few moments to create your instance template.

Task 2. Create a managed instance group

In this task you create a managed instance group (MIG), a collection of virtual machine (VM) instances that you manage as a single entity.

1. In the Google Cloud Console, in the **Navigation menu** (≡), click **Compute Engine > Instance groups**.

2. Click **Create Instance Group**.

3. Click **New managed instance group (stateless)** from the left-hand menu.

4. On the **New managed instance group (stateless)** page, specify the following, and leave the remaining settings as their defaults:

Property	**Value**
Name	**my-managed-instance-group**
Instance template	Select the instance template you created in Task 1.
Location	**Multiple zones**
Autoscaling > Autoscaling mode	**Off: do not autoscale**
Number of instances	To change the number of instances, you must first turn off autoscaling; see below. When this is done, set the maximum number value to **3**.

Note: Do not forget to set the number of instances after you change the autoscaling mode.

5. Click **Create**.

It will take a few minutes to create the MIG.

Task 3. Create a Google Cloud self-managed SSL certificate resource

In this task you create a private key, a certificate, and then a self-managed SSL certificate resource. Before you can create a Google Cloud SSL certificate resource, you must have a private key and certificate.

A Google Cloud SSL certificate includes both a private key and the certificate itself, both in PEM format.

Self-managed SSL certificates are certificates that you obtain, provision, and renew yourself. You use this resource to secure communication between clients and your load balancer, which you create in the next task.

Create a private key and certificate

1. On the Google Cloud Console title bar, click **Activate Cloud Shell** (>_). If prompted, click **Continue**.

2. To create a new private key with RSA-2048 encryption in the PEM format OpenSSL, run the following command:

```
openssl genrsa -out PRIVATE_KEY_FILE 2048
```

content_c

Create a CSR

Generate a certificate signing request (CSR) in the PEM format using OpenSSL.

1. Click **Open Editor**. If prompted, click **Open in a new window**.

2. Click **File > New File**.

3. Copy and paste the following configuration into the **Cloud Editor** window:

```
[req]
default_bits = 2048
req_extensions = extension_requirements
distinguished_name = dn_requirements
prompt = no
[extension_requirements]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
[dn_requirements]
countryName = US
stateOrProvinceName = CA
localityName = Mountain View
0.organizationName = Cloud
organizationalUnitName = Example
commonName = Test
```

content_c

4. Click **File** > **Save As**.

5. In the location drop-down at the top of the panel, select **/home** > **student-XX-XXXXXXXXXX@qwiklabs.net**.

6. In **Name**, type **ssl_config**, and then click **Save**.

7. Return to the previous tab. You may have to click **Open Terminal** to resume your **Cloud Shell** session.

8. To create a certificate signing request (CSR) file, run the following OpenSSL command:

```
openssl req -new -key PRIVATE_KEY_FILE \
 -out CSR_FILE \
 -config ssl_config
```

content_c

Sign the CSR

When a Certificate Authority (CA) signs your CSR, it uses its own private key to create a certificate.

- To create a self-signed certificate for testing, run the following OpenSSL command:

```
openssl x509 -req \
 -signkey PRIVATE_KEY_FILE \
 -in CSR_FILE \
 -out CERTIFICATE_FILE.pem \
 -extfile ssl_config \
 -extensions extension_requirements \
 -days 365
```

content_c

Create a self-managed SSL certificate resource

Before you can create a Google Cloud SSL certificate resource, you must have a private key and certificate.

1. To create a global SSL certificate, use the `gcloud compute ssl-certificates create` command with the `--global` flag:

```
gcloud compute ssl-certificates create my-cert \
 --certificate=CERTIFICATE_FILE.pem \
 --private-key=PRIVATE_KEY_FILE \
 --global
```

content_c

2. In the **Authorize Cloud Shell** prompt, click **Authorize**.

Task 4. Create a load balancer

In this task you create a load balancer. HTTP(S) Load Balancing is implemented on Google Front End (GFE). GFEs are distributed globally and operate together using Google's global network and control plane.

To set up a load balancer, your VMs must be in an instance group, which you created in the previous tasks.

1. In the Google Cloud Console, in the **Navigation menu** (☰), click **Network services > Load balancing**.

2. Click **Create load balancer**.

3. Under **Application Load Balancer (HTTP/S)**, click **Start Configuration**.

4. For **Internet facing or internal only**, select **From Internet to my VMs or serverless services**.

5. For **Global or Regional**, select **Global external Application Load Balancer**, and then click **Continue**.

6. On the **New Classic HTTP(S) load balancer** page, for **Load Balancer name**, type **my-load-balancer**.

7. Click **Backend Configuration** > **Backend services & backend buckets > Create a Backend Service**.

8. On the **Create backend service** panel, for **Name**, type **my-backend-service**.
   You must use the exact name *my-backend-service*. If you use a different name, the startup script on your VMs won't be able to find the correct Backend Service ID to authenticate requests.

9. In **New backend**, specify the following, and leave the remaining settings as their defaults:

Property	**Value**

Instance group	**my-managed-instance-group**
Port Numbers	**80**

10. Uncheck **Enable Cloud CDN**.

11. In **Health check**, click **Create a Health Check**.

12. For **Name**, type **my-health-check**.

13. For **Protocol**, select **HTTP**.

14. Click **Save**.

15. Click **Create**.

The New Classic HTTP(S) load balancer pane reappears.

16. Click **Host and path rules** to load the default values. You don't need to add any rules.

17. Click **Frontend configuration**. Specify the following, and leave the remaining settings as their defaults:

Property	**Value**
Protocol	**Https (includes Http/2)**
IP address	Click **Ephemeral**, and then select **Create IP address**
Name	Type **static-ip**, and then click **Reserve**.
Certificate	**my-cert**

18. Click **Done**.

The New Classic HTTP(S) load balancer pane reappears.

19. In **New Classic HTTP(S) load balancer**, click **Create**.

The **Load balancing** page appears and your new load balancer will be created in the list of load balancers.

20. When the Cloud Console finishes creating the new load balancer, click the name of the load balancer and note the external IP address under **Details > Frontend**. You will need it in task 7.

Task 5. Restart your VMs

In this task you restart the VMs in your MIG so that they can correctly authenticate requests from IAP.

1. In the Google Cloud Console, in the **Navigation menu** (≡), click **Compute Engine > Instance groups**.

2. Click **my-managed-instance-group**.

3. Click **Restart/Replace VMs**.

4. For **Operation**, click **Restart**.

5. For **Instances**, type **3**

6. For **Minimum wait time**, type **0**

7. Click **Restart VMs**.

Task 6. Set up IAP

Configure your firewall

In this task you configure your firewall to block access to the underlying VMs and only allow access through IAP.

1. In the Google Cloud Console, in the **Navigation menu** (≡), click **VPC network** > **Firewall**.

2. Select the **default-allow-internal** checkbox.

3. Click **Delete**, and then select **Delete** to confirm it.

4. Click **Create Firewall Rule**. Specify the following, and leave the remaining settings as their defaults:

Property	**Value**
Name	**allow-iap-traffic**
Targets	**All instances in the network**

Source IPv4 ranges	**130.211.0.0/22, 35.191.0.0/16** (Press **Enter** after you paste each value in the box)
Protocols and ports	**Specified protocols and ports**
TCP	**80, 78**

5. Click **Create**.

Set up IAP

In this step you set up IAP for your project.

1. In the Google Cloud Console, in the **Navigation menu** (≡), click **Security > Identity-Aware Proxy**.

2. Click **Enable API**.

3. Click **Go to Identity Aware Proxy**.

4. Click **Configure Consent Screen**.

Caution: Don't enter any confidential information on the OAuth consent screen. Any information you save to the OAuth consent screen may be publicly visible for anyone who accesses your URL. Email and product details are displayed on the login screen and when someone tries to access a resource for which they don't have permission.

5. In **User Type**, select **External** , and then click **Create**.

6. In **App name** , type **IAP**.

7. In **User support email**, select the student account. This has the value of **student-00-*********@qwiklabs.net**.

8. For **Developer contact information**, copy and paste the student account **Username** from the lab window. This matches the value in the previous step.

9. Click **Save and Continue** three times, and then select **Back to Dashboard**.

To change information on the OAuth consent screen later, such as the product name or email address, repeat the preceding steps to configure the consent screen.

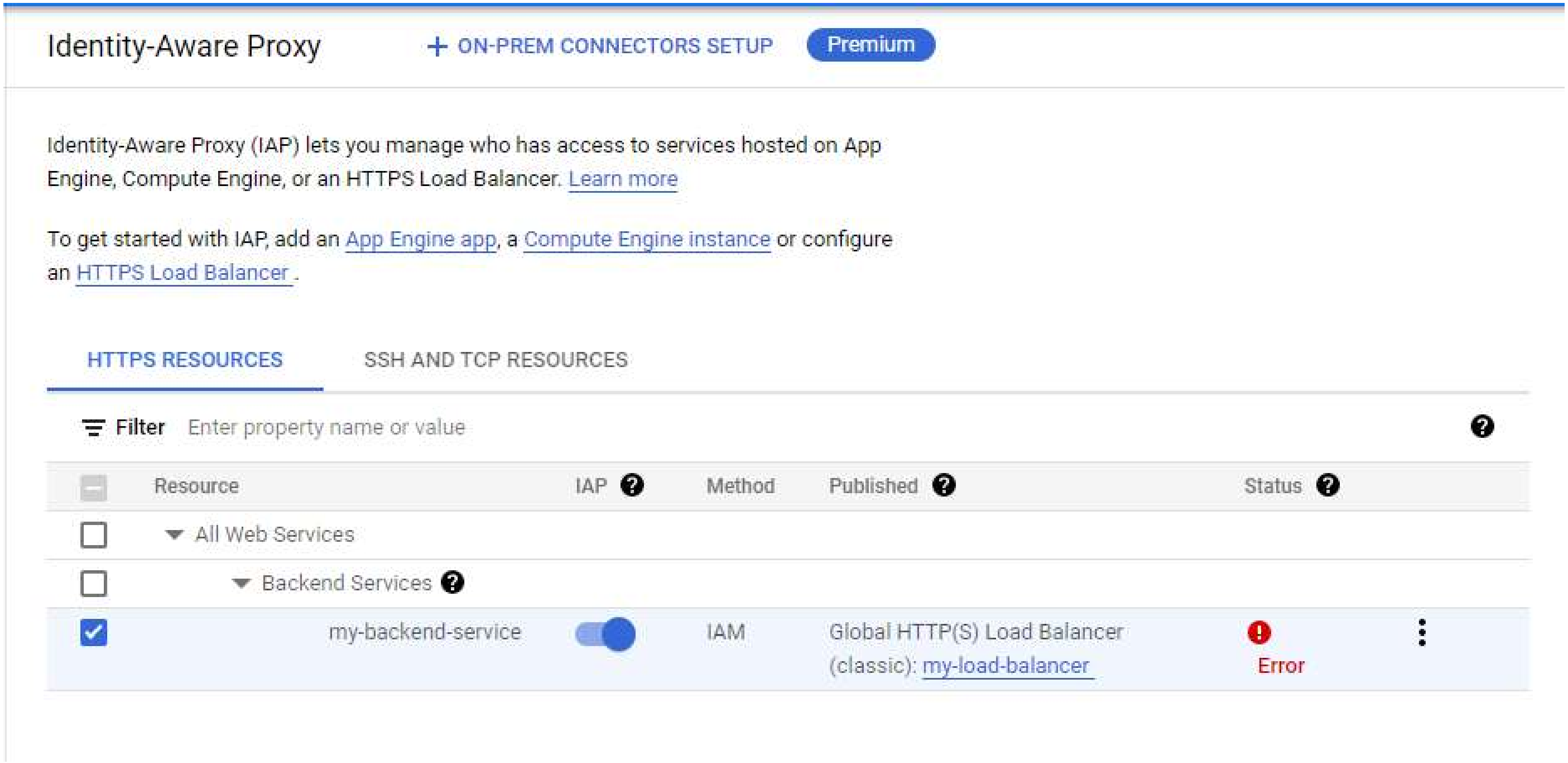
10. In the Google Cloud Console, in the **Navigation menu** (☰), click **Security** > **Identity-Aware Proxy**.

11. Next to **my-backend-service**, toggle the **on/off** switch in the **IAP** column.

12. In the **Turn on IAP** dialog, select the checkbox to confirm you have read the configuration requirements.

13. Click **Turn On**.

Note: If you see an error, click on the error. If you are then prompted to add a firewall rule, edit the rule you created previously to include the port number mentioned in the error.



Identity-Aware Proxy

Identity-Aware Proxy (IAP) lets you manage who has access to services hosted on App Engine, Compute Engine, or an HTTPS Load Balancer. Learn more

To get started with IAP, add an App Engine app, a Compute Engine instance or configure an HTTPS Load Balancer .

	Resource	IAP	Method	Published	Status
□	All Web Services				
□	Backend Services				
☑	my-backend-service	●	IAM	Global HTTP(S) Load Balancer (classic): my-load-balancer	Error

Confirm that OAuth Consent has been set up.

○ Check my progress

Assessment Completed!

Add principals to the access list

In this step you add principals to the IAP access list for your project.

1. In **Identity-Aware Proxy**, select the **my backend-service** checkbox.

2. Click **Add Principal**.

3. To grant access to yourself, in **New Principals**, copy and paste your qwiklabs **Username** from the lab credentials pane.

4. Select the Role of **Cloud IAP > IAP-secured Web App User**.

5. Click **Save**.

Confirm principal to access the application by configuring IAM.

○ Check my progress

Assessment Completed!

Task 7. Test IAP

In the task, you run a curl command to test access your external load balancer, and then verify that it is protected by IAP.

1. In the Google Cloud Console, in the **Navigation menu** (☰), click **Network services > Load balancing**.

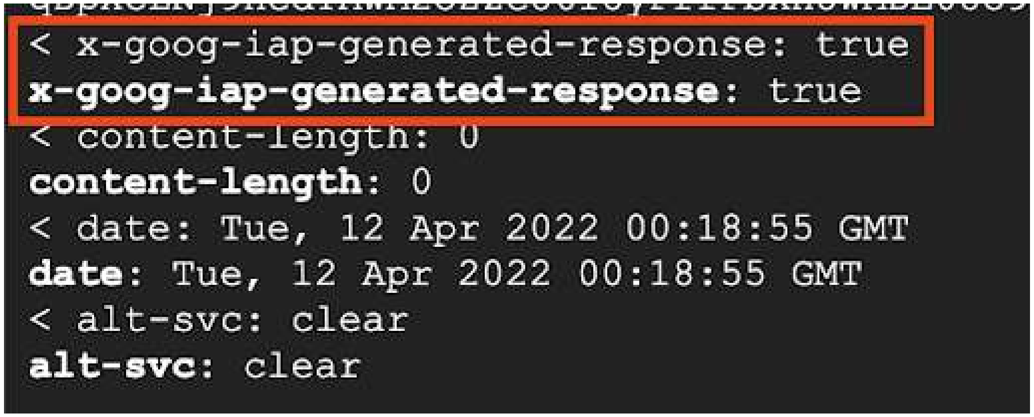
2. Click **Frontends**.

3. On the Google Cloud Console title bar, click **Activate Cloud Shell** (>_). If prompted, click **Continue**.

4. Run the following curl command, replacing **Load Balancer External IP address** with the External IP address of your load balancer:

```
curl -kvi https://<Load Balancer External IP address>                                    content_c
```

Note: If an IAP generated response that is true is displayed, you have successfully configured IAP for your Compute Engine instance.

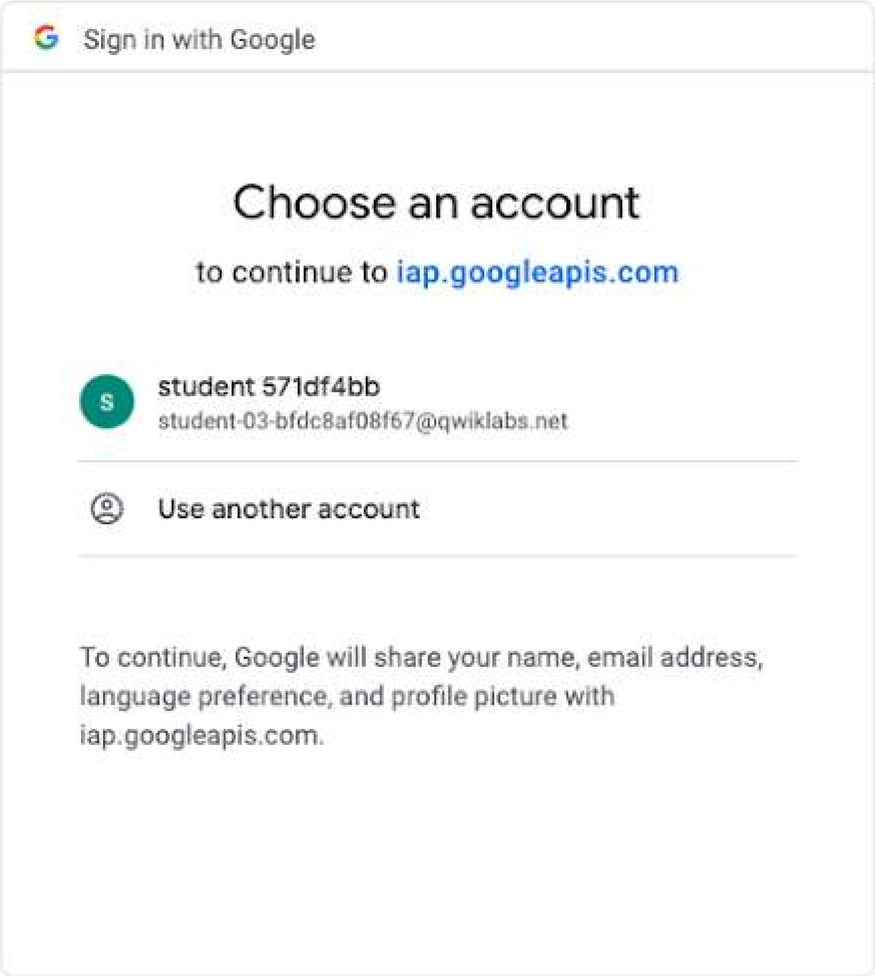


```
< x-goog-iap-generated-response: true
x-goog-iap-generated-response: true
< content-length: 0
content-length: 0
< date: Tue, 12 Apr 2022 00:18:55 GMT
date: Tue, 12 Apr 2022 00:18:55 GMT
< alt-svc: clear
alt-svc: clear
```

5. Scroll up the console page and locate the **Http/2 302** redirection to accounts.google.com.

6. Click on the **location** link.

The link opens to the **Google account integrated authentication** page.



Sign in with Google

Choose an account

to continue to iap.googleapis.com

student 571df4bb
student-03-bfdc8af08f67@qwiklabs.net

Use another account

To continue, Google will share your name, email address, language preference, and profile picture with iap.googleapis.com.

Note: Because you used a self-signed cert, you cannot access the application itself. However, this confirms that IAP is configured and is protecting traffic.

Confirm restrict access with IAP.

○ Check my progress

Assessment Completed!

Congratulations!

You have successfully used IAP to secure a web application running on a Compute Engine instance.

In this lab, you learned how to:

- Create an instance template.
- Create an instance group.
- Create a self-signed certificate.
- Create a load balancer.
- Configure an Oauth consent screen.
- Grant access to the application using IAP.

End your lab