**Compute and Storage Security Lab-1**

**Lab: Applying Signed URLs to Cloud Storage Objects**

**Introduction:** GCP can offer access to the individual at a specified time to access the URL. You can use the signed URL and access a secure bucket object for a predetermined period, irrespective of whether they are in your organization or even have a Google account.

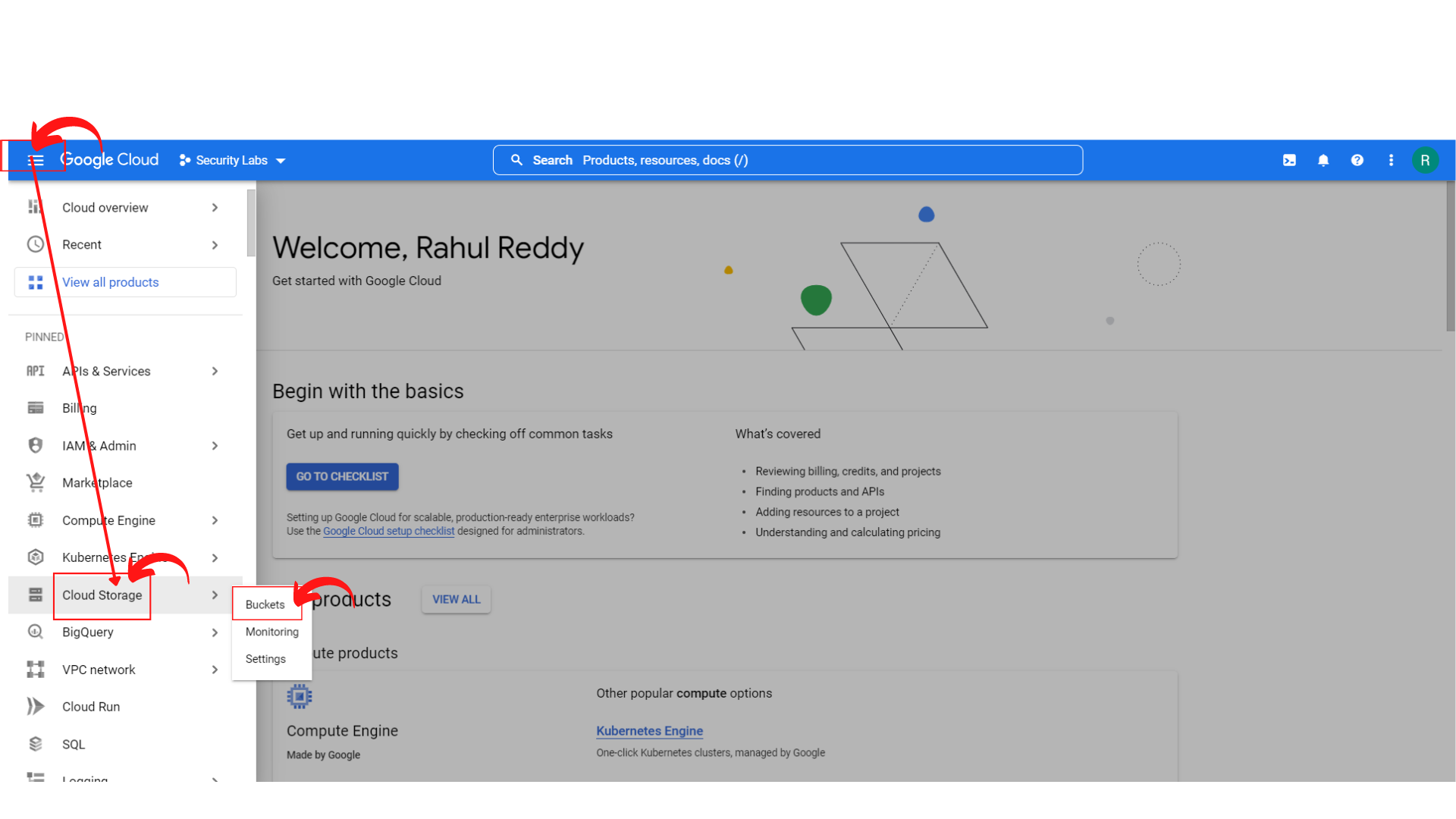
**Problem:** The contents of Cloud Storage buckets are not accessible to the public by default. You may, however, make an object or a bucket visible to everyone. However, neither of these options -- completely barring or completely permitting access -- is always the best option. **The signed URL is a Google Cloud feature that allows you to grant limited access to one or more people for a set period of time.** Regardless of whether they are in your business or even have a Google account, anyone who uses the signed URL can access a secure bucket object for the predetermined period of time.

Solution: In this hand-on lab, we will establish the appropriate structure -- a service account with a private key -- to generate a bucket containing an object that is not publicly exposed.

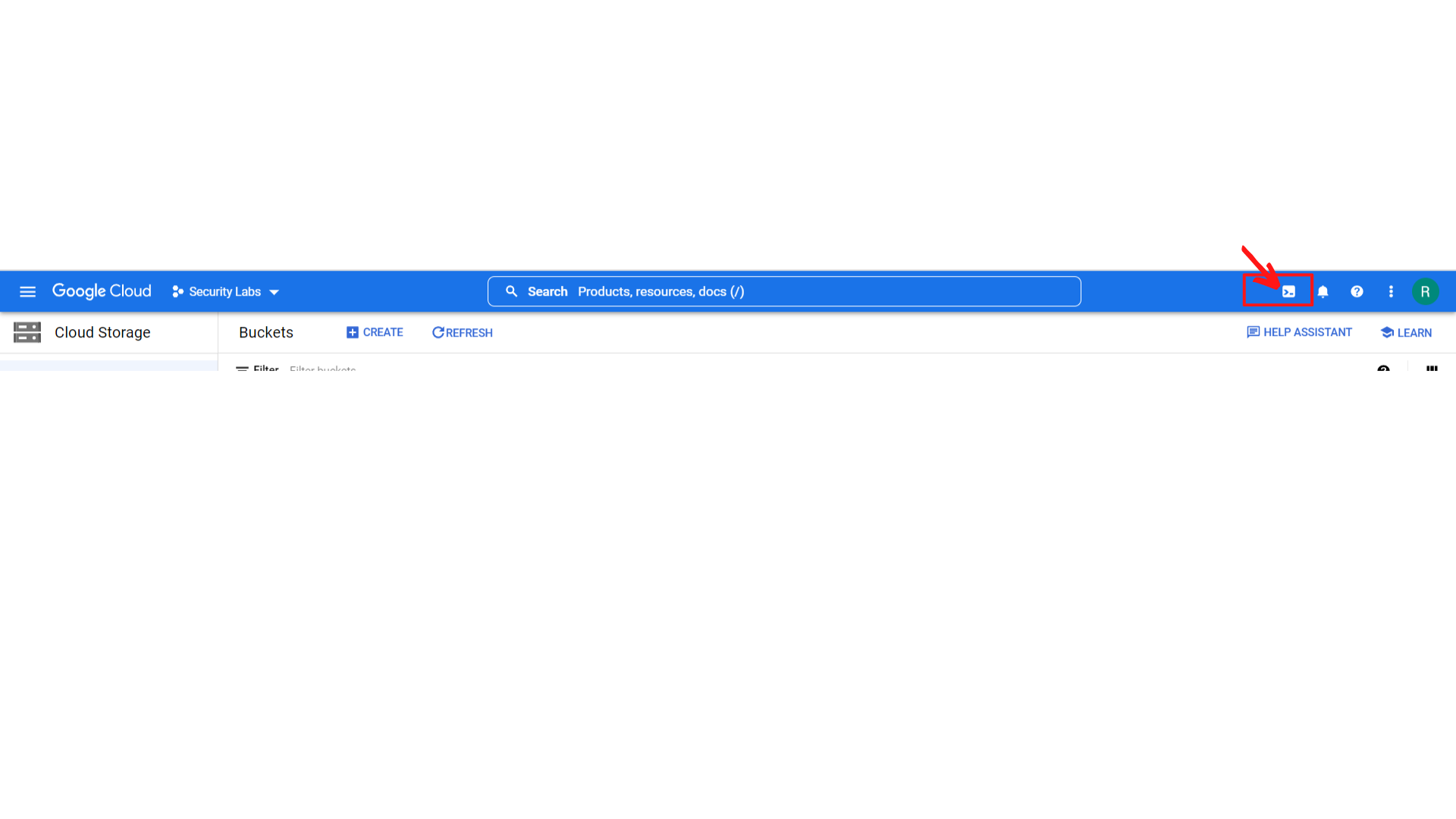
**Step1:** **Create a Cloud Storage Bucket.**

There is a document to create a storage bucket with the name **“Creating Storage Bucket for Signed URL Lab”** please follow the instructions from the document and create a storage bucket then follow the instructions below.

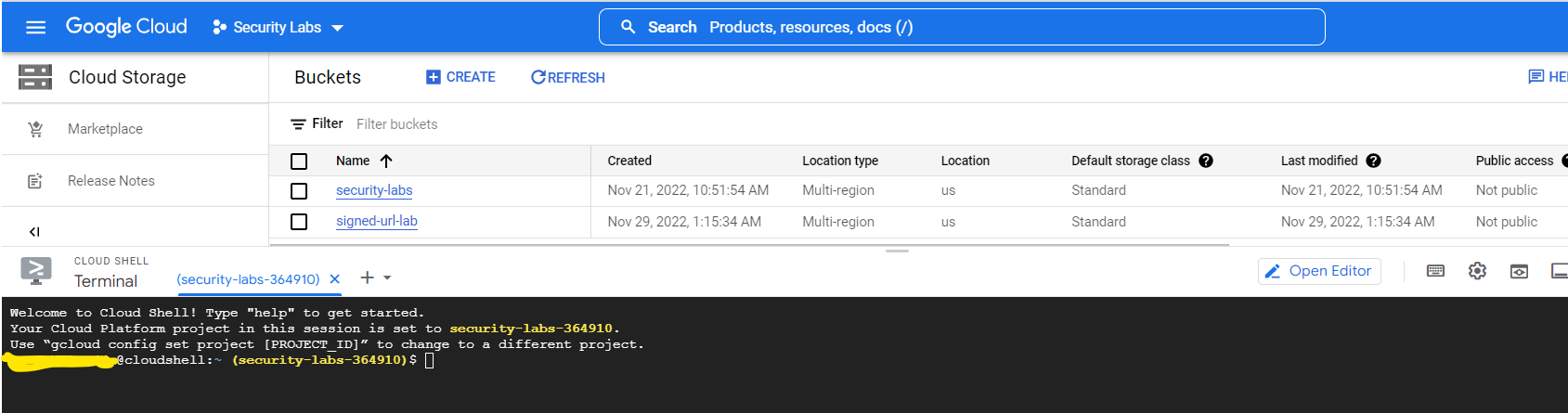
1. Click on *“Navigation menu”* on top left, then click on *“Cloud Storage” → “Buckets”*



1. Your screen should look similar to this and ***“click → Cloud shell”*** Icon.



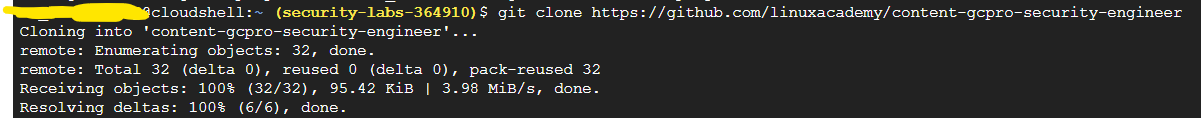
1. Your screen should look similar to this with a cloud shell terminal.



1. Now **run the following command to clone the Github repository:**

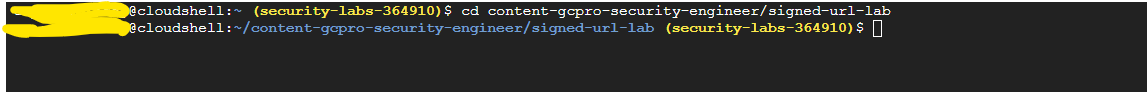
**git clone https://github.com/linuxacademy/content-gcpro-security-engineer**

Note: If you have already done the “Encryption on GCP Lab-1” lab, the git has been already cloned into your project, so you skip this step.



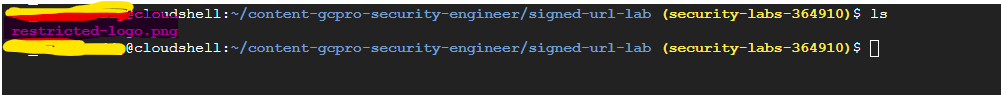
1. Now change to the content-gcpro-security-engineer/signed-url-lab directory using the following command:

**cd content-gcpro-security-engineer/signed-url-lab/**



1. To get the list of contents of the current directory, use the following command:

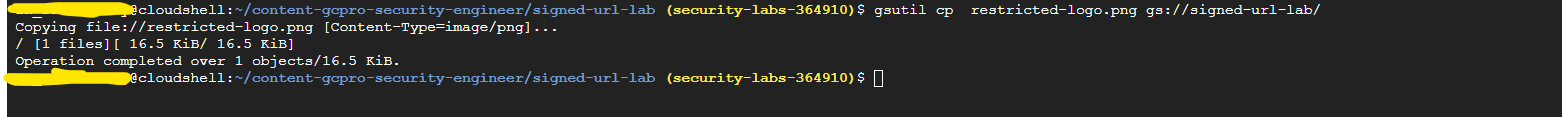
**ls**



You can see there is a file named **“restricted-logo.png”** in that directory.

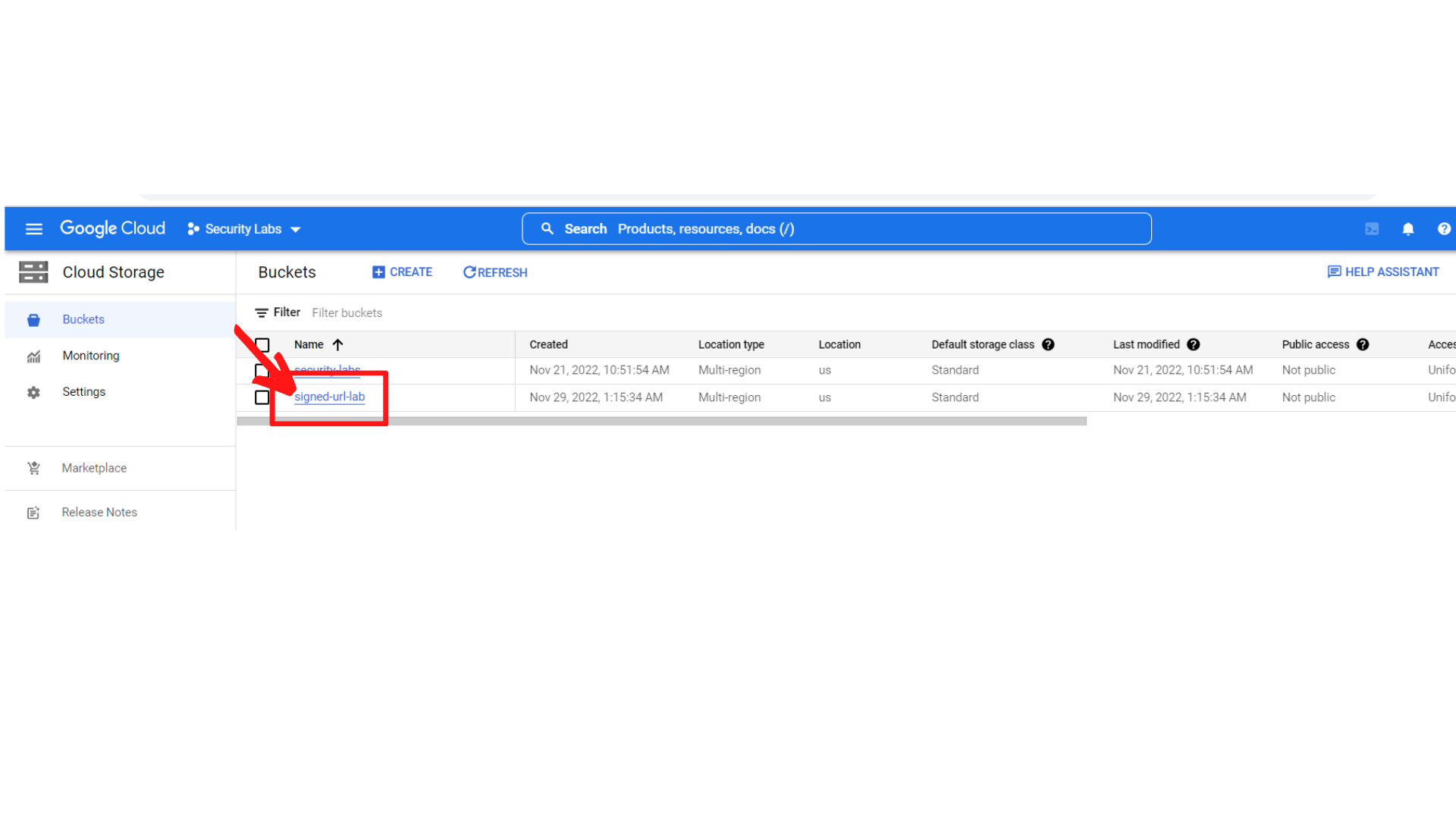
1. Now copy that file(restricted-logo.png) to the storage bucket you have created using the document.
2. Using the following command copy the file: Here my storage bucket name **“signed-url-lab”**

**gsutil cp restricted-logo.png gs://signed-url-lab/**

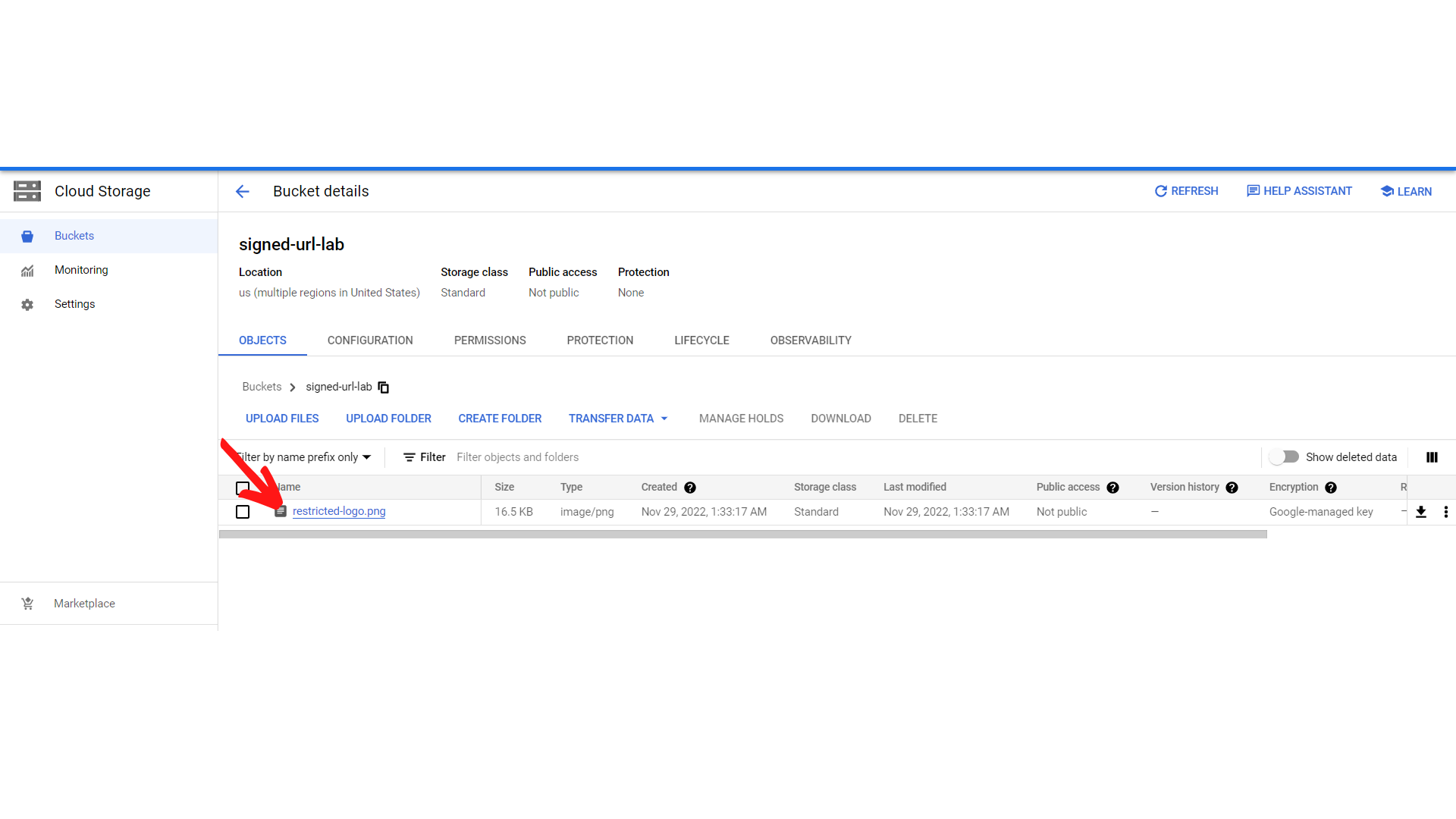


That’s it, you have copied the file to your storage bucket.

1. To verify that, **“click → signed-url-lab”**

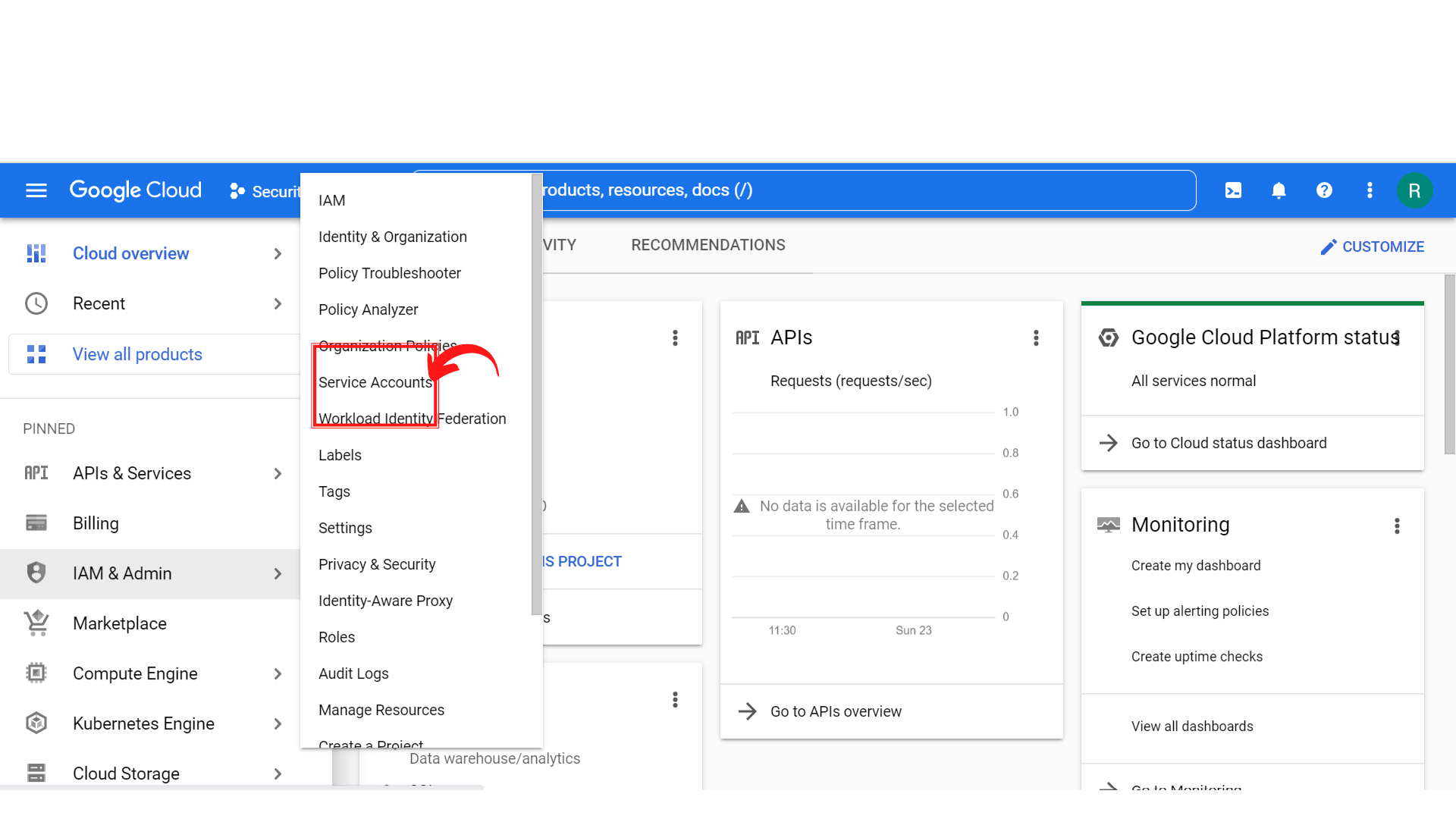


1. Now your screen should look similar to this and scroll down to see the file **“restricted-logo.png”**

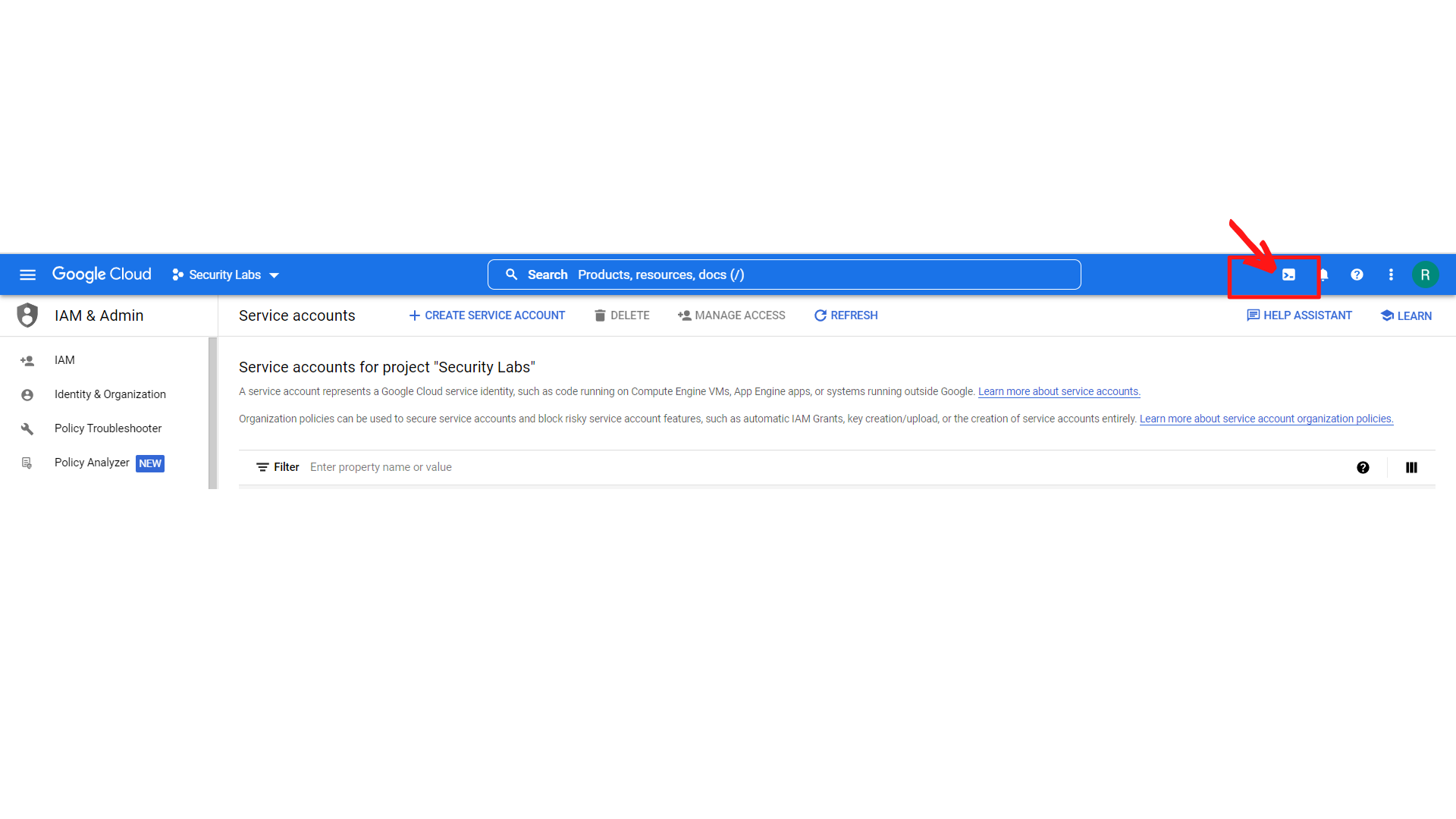


**Step2:** **Create a Service Account and Key.**

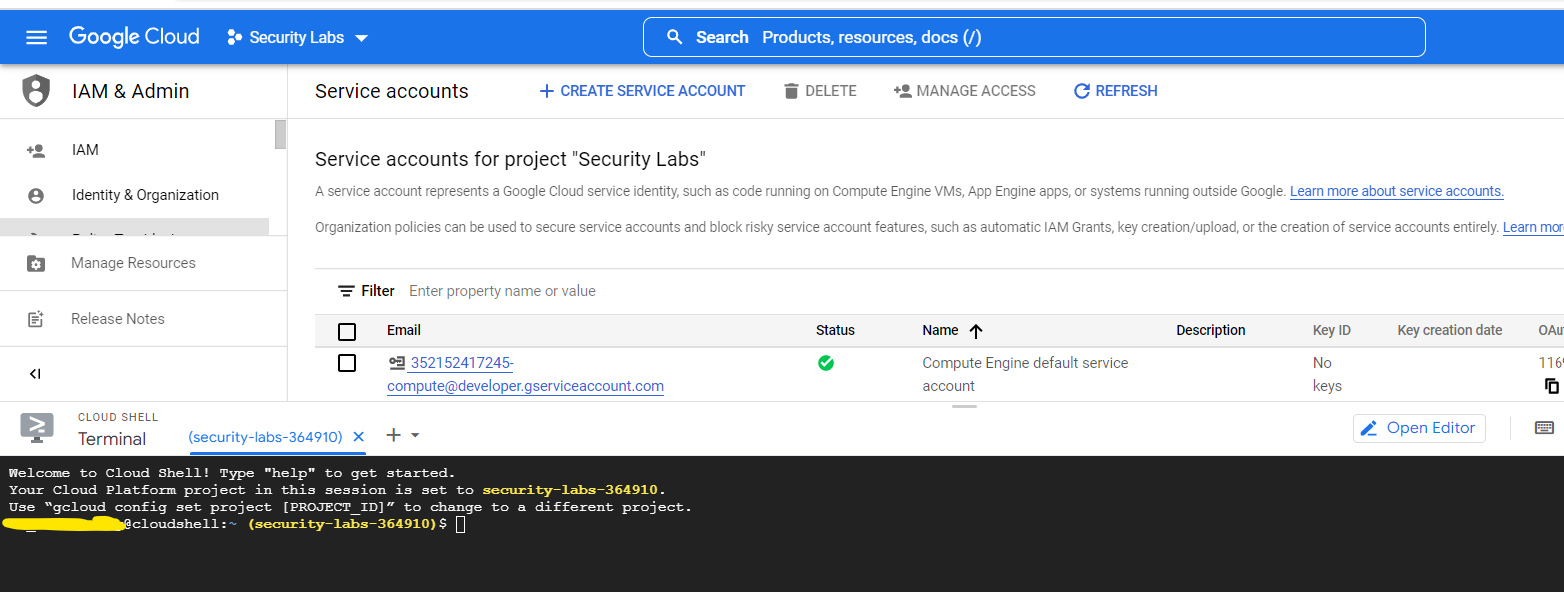
1. To create a new service account, click on “*IAM & Admin → Service Accounts”.*



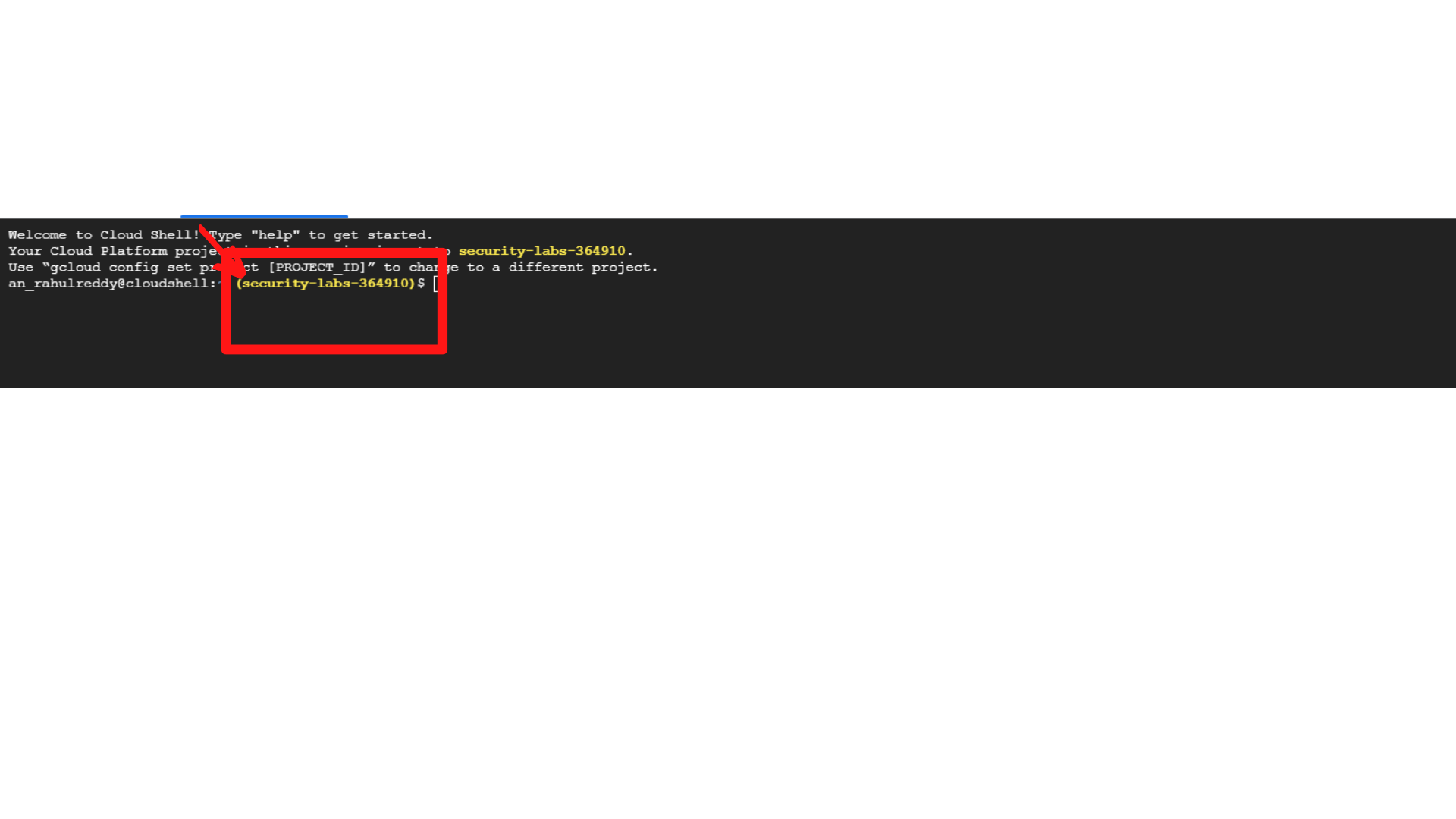
1. Your screen should look similar to this and ***“click → Cloud shell”*** Icon.



1. Your screen should look similar to this with a cloud shell terminal.



1. **Now you have to know your project ID**, here is the screenshot where you can know the project ID.



1. Now in the Cloud shell, run the following command to establish a variable:

**export PROJECT\_ID=[YOUR\_PROJECT\_ID]**

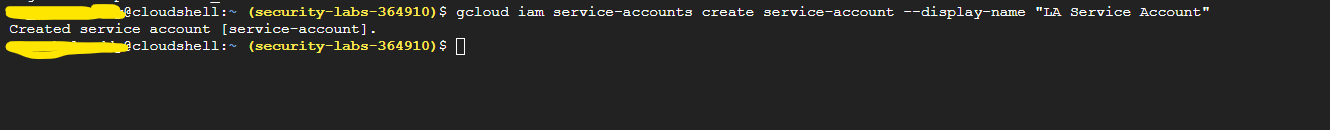
Here my project id is “security-labs-364910”

**export PROJECT\_ID=security-labs-364910**



1. Create a new service account using the following command:

**gcloud iam service-accounts create service-account --display-name “LA Service Account”**



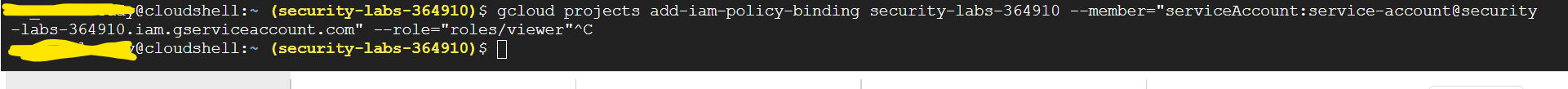
This will create a new service account with name **LA Service Account**

1. Now to grant your service account an IAM role on your project, run the following command:

**gcloud projects add-iam-policy-binding PROJECT\_ID --member="serviceAccount:service-account@PROJECT\_ID.iam.gserviceaccount.com" --role="roles/viewer"**

**Note: Please replace project id with your project id**

**gcloud projects add-iam-policy-binding security-labs-364910 --member="serviceAccount:service-account@security-labs-364910.iam.gserviceaccount.com" --role="roles/viewer"**

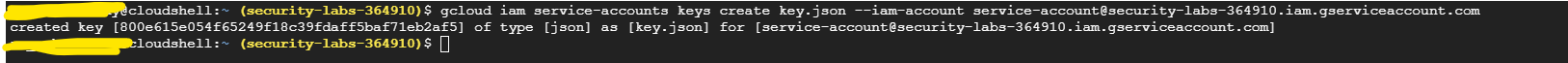


1. Now create a JSON key for authentication using the following command:

gcloud iam service-accounts keys create key.json --iam-account service-account@PROJECT\_ID.iam.gserviceaccount.com

**Note: Please replace project id with your project id**

**gcloud iam service-accounts keys create key.json --iam-account service-account@security-labs-364910.iam.gserviceaccount.com**



That’s it, you successfully created service account with key(json)

**Step3: Generate a Signed URL**

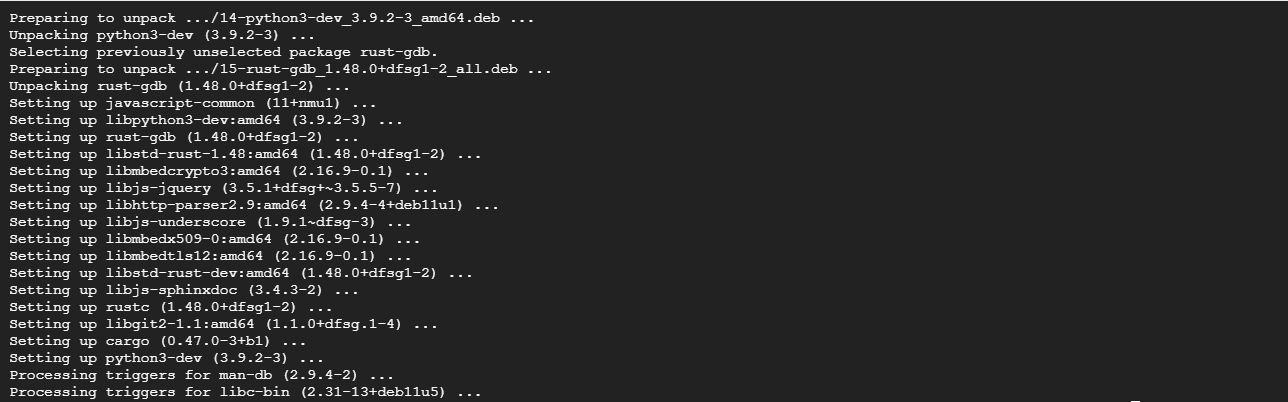
**Please be on cloud shell(terminal)**

1. In the cloud shell, install the **openssl headers for debian GNU/Linux** using the following commands:

**sudo apt-get install build-essential libssl-dev libffi-dev \python3-dev cargo**

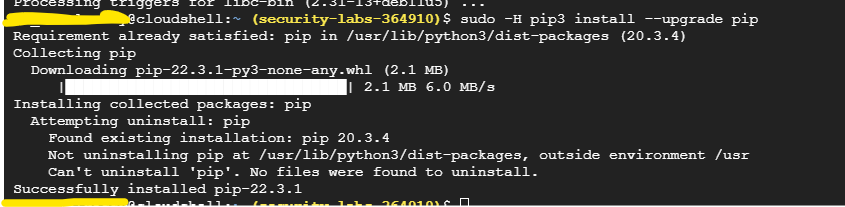
**Type “Y” and press “Enter” to continue if asked on cloud shell.**



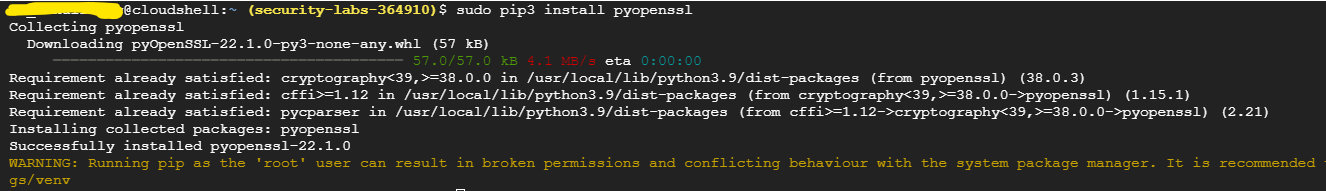


1. Now then **install the pyopenssl library** using the following command:

**sudo -H pip3 install --upgrade pip**

****

**sudo pip3 install pyopenssl**



That's it, openssl has been installed.

1. Now generate a signed URL using the following command:

**gsutil signurl -d 10m key.json gs://[BUCKET\_NAME]/restricted-logo.png**

**Note: Please replace BUCKET\_NAME with your Bucket\_Name.**

**gsutil signurl -d 10m key.json gs://signed-url-lab/restricted-logo.png**



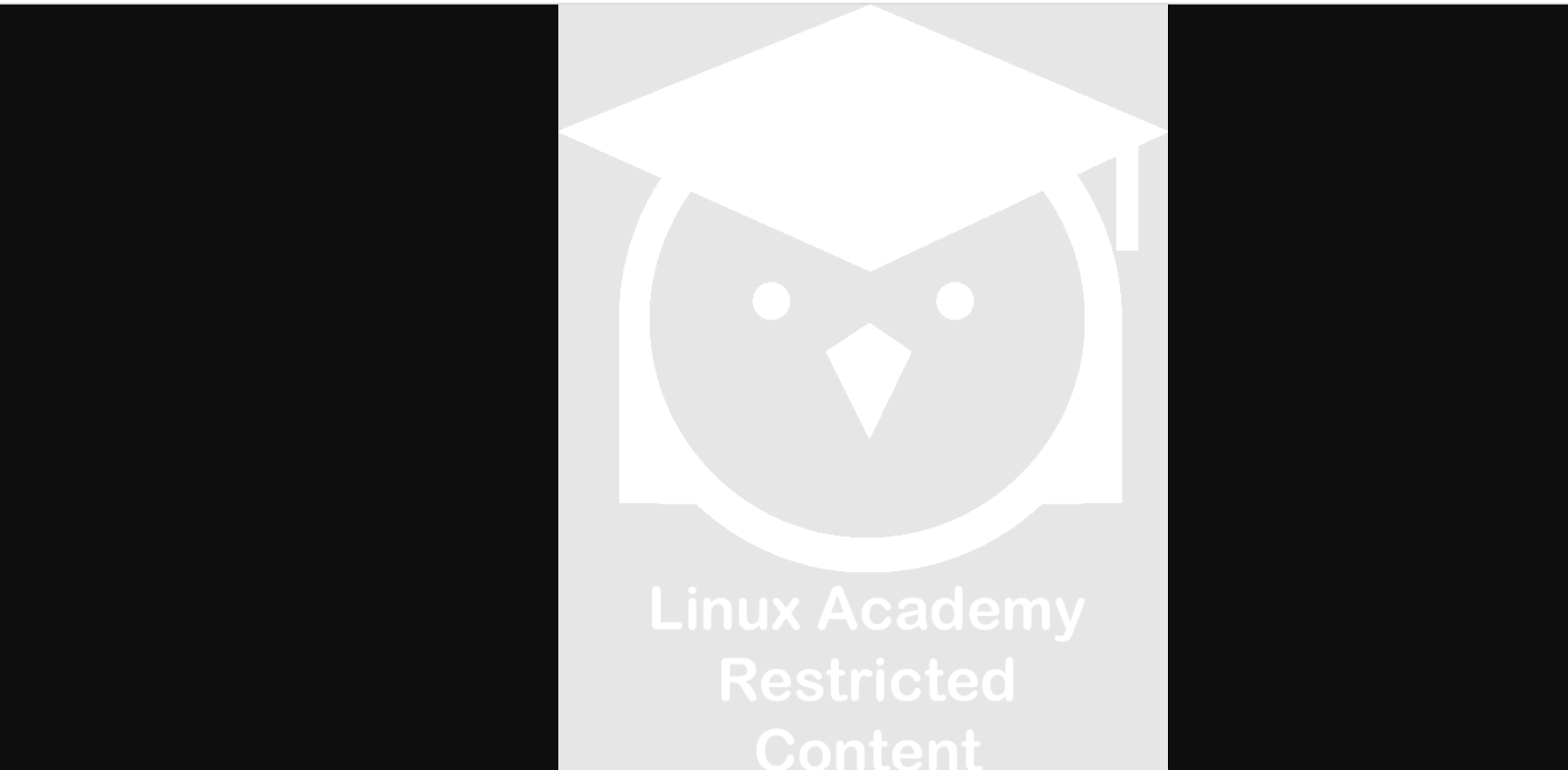
**After entering the command, the signed url will be generated.**



1. Test the URL by clicking on the generated link.



1. Once you click on the URL, A new window will open and display the image.



**That’s it, you successfully applied a signed url to cloud storage objects.**