Caching Content with Cloud CDN

1 hourFree

Rate Lab

**Overview**

In this lab, you configure Cloud Content Delivery Network (Cloud CDN) for a backend bucket and verify caching of an image. Cloud CDN uses Google's globally distributed edge points of presence to cache HTTP(S) load-balanced content close to your users. Caching content at the edges of Google's network provides faster delivery of content to your users while reducing serving costs.

For an up-to-date list of Google's Cloud CDN cache sites, refer to this documentation: <https://cloud.google.com/cdn/docs/locations>

Objectives

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

* Create and populate a Cloud Storage bucket
* Create an HTTP load balancer with Cloud CDN
* Verify the caching of your bucket's content

**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 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 the Google Cloud Platform 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 GCP account or project, do not use it for this lab.

**Task 1. Create and populate a Cloud Storage bucket**

Cloud CDN content can originate from two types of backends:

* Google Compute Engine virtual machine (VM) instance groups
* Google Cloud Storage buckets

In this lab, you configure a Cloud Storage bucket as the backend.

**Create a unique Cloud Storage bucket**

1. In the GCP Console, on the **Navigation menu** (Navigation menu), click **Storage** > **Browser**.
2. Click **Create bucket**.
3. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | *Enter a globally unique name* |
| Default storage class | Regional |
| Location | *Choose a location that is very far from you* |
| Access control model | Set object-level and bucket-level permissions |

Try to choose a location that is either halfway around the world from you or at least in a different continent. This provides a greater difference between accessing the image with and without Cloud CDN enabled.

1. Click **Create**.
2. Note the name of your storage bucket for the next subtask. It will be referred to as [your-storage-bucket].

**Copy an image file into your bucket**

Copy an image from a public Cloud Storage bucket to your own bucket.

1. In the GCP Console, click **Activate Cloud Shell** (Cloud Shell).
2. If prompted, click **Continue**.
3. Run the following command in Cloud Shell, replacing [your-storage-bucket] with your bucket's name:

gsutil cp gs://cloud-training/gcpnet/cdn/cdn.png gs://[your-storage-bucket]

1. In the GCP Console, click **Refresh Bucket** to verify that the image was copied.
2. Publish Cloud Storage image file to the web with following command:

gsutil acl ch -u AllUsers:R gs://[your-storage-bucket]/cdn.png

1. Click on the **Public Link** under **Public access** to verify that you can access the image.
2. Exit **Cloud Shell**:

exit

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

Create and populate a Cloud Storage bucket

Check my progress

**Task 2. Create the HTTP load balancer with Cloud CDN**

HTTP(S) load balancing provides global load balancing for HTTP(S) requests of static content to a Cloud Storage bucket (backend). When you enable Cloud CDN on your backend, your content is cached at a [location at the edge of Google's network](https://cloud.google.com/cdn/docs/locations), which is usually far closer to the user than your backend is.

**Start the HTTP load balancer Configuration**

1. In the GCP Console, on the **Navigation menu** (Navigation menu), click **Network Services** > **Load balancing**.
2. Click **Create load balancer**.
3. Under **HTTP(S) Load Balancing**, click **Start configuration**.
4. For **Name**, type **cdn-lb**.

**Configure the backend**

1. Click **Backend configuration**.
2. For **Backend services & backend buckets**, click **Create or select backend services & backend buckets** > **Backend buckets** > **Create a backend bucket**.
3. For **Name**, type **cdn-bucket**.
4. Click **Browse** under **Cloud Storage bucket**.
5. Select your bucket, and click **Select**.
6. Select **Enable Cloud CDN**.
7. Click **Create**.

Yes, enabling Cloud CDN is as simple as selecting **Enable Cloud CDN**!

**Configure the frontend**

The host and path rules determine how your traffic will be directed. For example, you could direct video traffic to one backend and image traffic to another backend. However, you are not configuring the Host and path rules in this lab.

1. Click **Frontend configuration**.
2. Specify the following, leaving all other values with their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Protocol | HTTP |
| IP version | IPv4 |
| IP address | Ephemeral |
| Port | 80 |

1. Click **Done**.

**Review and create the HTTP load balancer**

1. Click **Review and finalize**.
2. Review the **Backend Buckets** and **Frontend**.
3. Click **Create**. Wait for the load balancer to be created.
4. Click on the name of the load balancer (**cdn-lb**).
5. Note the IP address of the load balancer for the next task. It will be referred to as [LB\_IP\_ADDRESS].

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

Create the HTTP Load Balancer with Cloud CDN

Check my progress

**Task 3. Verify the caching of your bucket's content**

Now that you have created the HTTP load balancer for your bucket and enabled Cloud CDN, it is time to verify that the image is cached on the edge of Google's network.

**Time the HTTP request for the image**

One way to verify that the image is cached is to time the HTTP request for the image. The first request should take significantly longer, because content is only cached at an edge location after being accessed through that location.

1. In the GCP Console, click **Activate Cloud Shell** (Cloud Shell).
2. If prompted, click **Continue**.
3. Store the IP address of the load balancer in an environment variable:

export LB\_IP\_ADDRESS=<Enter the IP address of the load balancer>

1. Run the following command 3 times for consecutive HTTP requests:

for i in {1..3};do curl -s -w "%{time\_total}\n" -o /dev/null http://$LB\_IP\_ADDRESS/cdn.png; done

The output should look like this (**do not copy; this is example output**):

1.234357

0.009600

0.006652

In this example output, the second and third request take less than 1% of the time of the first request. This demonstrates that the image was cached during the first request and accessed from an edge location on further requests. Depending on how far you placed your storage bucket and where your closest edge location is, you will see different results.

**Explore the Cloud CDN logs**

Another way to verify that the image got cached in the previous step, is to explore the Cloud CDN logs. These logs will contain information on when content was cached and when the cache was accessed.

While [Cloud CDN logging](https://cloud.google.com/cdn/docs/logging) is in Alpha, the following steps might not work as expected.

It might take a couple of minutes to select the Cloud CDN logs.

1. In the GCP Console, on the **Navigation menu** (Navigation menu), click **Logging** > **Logs Viewer**.
2. Under the **Resources** filter, select **Cloud HTTP Load Balancer > cdn-lb-forwarding-rule > cdn-lb**.

It might take a couple of minutes to select the Cloud CDN logs.

1. Expand the first log entry (on top).
2. Within the entry, expand the **httpRequest** and notice that the **cacheLookup** is *true* but there is no **cacheHit** field.

This illustrates that the cache did not contain the image on this first request.

1. Expand the **jsonPayload** and notice that the **statusDetails** field contains *response\_sent\_by\_backend*.

This also illustrates that the image came from the backend bucket on this first request.

1. Close the current log entry and expand a different log entry.
2. Within the entry, expand the **httpRequest** and notice that the **cacheHit** is *true*.

This illustrates that the cache contained the image on this request.

1. Expand the **jsonPayload** and notice that the **statusDetails** field contains *response\_from\_cache*.

This also illustrates that the cache, instead of the backend, provided the image on this request.

The Cloud CDN logs clearly demonstrate that the image was provided from the backend on the first request. This request filled the cache on the edge location, and all future requests received the image from that cache.

**Task 4. Review**

In this lab, you configured Cloud CDN for a backend bucket by configuring an HTTP load balancer and enabling Cloud CDN with a simple checkbox. You verified the caching of the bucket's content by accessing an image multiple times and exploring the Cloud CDN logs. The first time you accessed the image, it took longer because the cache of the edge location did not contain the image yet. All other requests were quicker because the image was provided from the cache of the edge location closest to your Cloud Shell instance.

For an up-to-date list of Google's Cloud CDN cache sites, refer to this documentation: <https://cloud.google.com/cdn/docs/locations>