

GCP Fundamentals: Getting Started with Cloud Storage and Cloud SQL

Overview

In this lab, you create a Cloud Storage bucket and place an image in it. You'll also configure an application running in Compute Engine to use a database managed by Cloud SQL. For this lab, you will configure a web server with PHP, a web development environment that is the basis for popular blogging software. Outside this lab, you will use analogous techniques to configure these packages. You also configure the web server to reference the image in the Cloud Storage bucket.

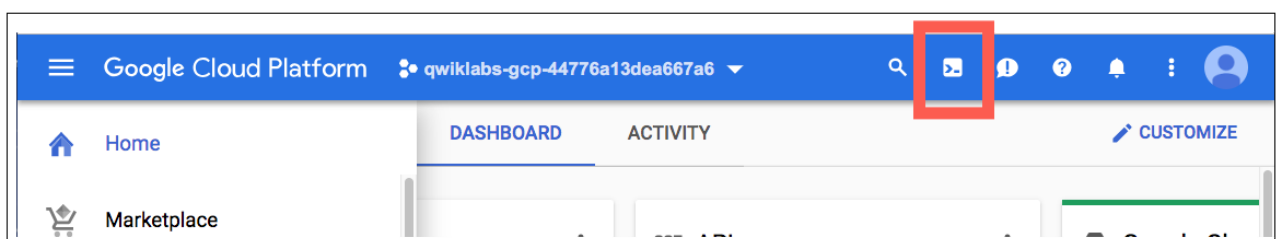
Objectives

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

- Create a Cloud Storage bucket and place an image into it.
- Create a Cloud SQL instance and configure it.
- Connect to the Cloud SQL instance from a web server.
- Use the image in the Cloud Storage bucket on a web page

Task 1: Sign in to the Google Cloud Platform (GCP) Console, with your credentials.

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



2. If a dialog box appears, click Start Cloud Shell.

Task 2: Deploy a web server VM instance.

This VM with startup-script will install a web server.

```
gcloud compute instances create "bloghost" \  
--image-project "debian-cloud" \  
--image "debian-9-stretch-v20190213" \  
--subnet "default" \  

```

```
--zone "us-central1a" \  
--tags http \  
--metadata  
startup-script=apt-get update  
apt-get install apache2 php php-mysql -y  
service apache2 restart
```

Add this line of code to configure your HTTP firewall -rule

```
gcloud compute firewall-rules create allow-http \  
--action=ALLOW \  
--destination=INGRESS \  
--rules=http:80 \  
--target-tags=http
```

NOTE: Copy the bloghost VM Instances internal and external IP addresses to a text editor, for use later.

Task 3: Create a Cloud Storage bucket using the gsutil command line

- All Cloud Storage bucket names must be globally unique
- Cloud Storage buckets can be associated with either a region or a multi-region location: US, EU, or ASIA

1. On the Google Cloud Platform menu, click Activate Cloud Shell. If a dialog box appears, click Start Cloud Shell.
2. For convenience, enter your chosen location into an environment variable called LOCATION. Enter one of these commands: [export LOCATION=US or export LOCATION=EU or export LOCATION=ASIA]

```
export LOCATION=US
```

3. In Cloud Shell, the DEVSHHELL_PROJECT_ID environment variable contains your project ID. Enter this command to make a bucket named after your project ID:

```
gsutil mb -l $LOCATION gs://$DEVSHHELL_PROJECT_ID
```

4. Retrieve a banner image from a publicly accessible Cloud Storage location:

```
gsutil cp gs://cloud-training/gcpfci/my-excellent-blog.png my-excellent-blog.png
```

5. Copy the banner image to your newly created Cloud Storage bucket:

```
gsutil cp my-excellent-blog.png gs://$DEVSHHELL_PROJECT_ID/my-excellent-blog.png
```

6. Modify the Access Control List of the object you just created so that it is readable by everyone:

```
gsutil acl ch -u allUsers:R gs://$DEVSHHELL_PROJECT_ID/my-excellent-blog.png
```

Task 4: Create the Cloud SQL instance

1. Show the list of potential machine types:

```
gcloud sql tiers list
```

NOTE: Note the values that begin with db-. You must choose one of these values to create a Second Generation instance

2. create the instance:

```
gcloud sql instances create [INSTANCE_NAME] \  
--tier=[MACHINE_TYPE] \  
--zone=[ZONE]
```

For example, the following command creates a Second Generation instance called blog-db. Make your machine-type and zone are the same into which you launched the bloghost instance. The best performance is achieved by placing the client and the database close to each other.

```
gcloud sql instances create blog-db --tier=db-n1-standard-2 --zone=us-central1a
```

NOTE: Do not include sensitive or personally identifiable information in your instance name; it is externally visible. You do not need to include the project ID in the instance name. This is done automatically where appropriate (for example, in the log files).

3. Let's set Root password with this line of code below:

```
gcloud sql users set-password root --host=% \  
--instance [INSTANCE_NAME] \  
--password [PASSWORD]
```

NOTE: Replace INSTANCE_NAME with your sql instance name: "blog-db" and choose a password that you remember and replace PASSWORD.

Your code should look like this:

```
gcloud sql users set-password root --host=% \  
--instance blog-db \  
--password password
```

4. copy the Public IP address for your blog-db SQL instance to a text editor for use later in this lab.

5. User: Set the password for the "root@%" blog-db MySQL user:

```
gcloud sql users set-password root -host=% --instance [INSTANCE_NAME] -password  
[PASSWORD]
```

Replace your INSTANCE_NAME with blogdbuser and PASSWORD with password of your choice, make a note of it. Your result should look like this:

```
gcloud sql users set-password root -host=% --instance blogdbuser -password  
password
```

6. Let us add Network control, so that our MySQL blog-db database can only be contacted from our bloghost VM instance, and not having broad internet access.

Now is time to use the bloghost External IP address that was copied earlier into notepad, name your network web front.

Replace [instance_name] with web front end, and with the public IP address of your bloghost VM instance [35.224.196.147], follow by /32

- Add the External ip address to the instance:

```
gcloud sql instances path [instance_name]  
--assign-ip
```

- Authorize the public IP addresses you want to keep.

```
gcloud sql instances path [instance_name]  
--authorized-networks=<IP address>
```

your result should look like this:

```
gcloud sql instances path web front end
--authorized-networks=35.224.196.147/32
```

Task 5: Configure an application in a Compute Engine instance to use Cloud SQL

1. Connect to SSH bloghost:

```
gcloud compute ssh bloghost
```

2. In your ssh session on bloghost, change your working directory to the document root of the web server:

```
cd /var/www/html
```

3. Use the nano text editor to edit a file called index.php:

```
sudo nano index.php
```

4. Paste the content below into the file:

```
<html>
<head><title>Welcome to my excellent blog</title></head>
<body>
<h1>Welcome to my excellent blog</h1>
<?php
    $dbserver = "CLOUDSQLIP";
    $dbuser = "blogdbuser";
    $dbpassword = "DBPASSWORD";
    // In a production blog, we would not store the MySQL
    // password in the document root. Instead, we would store it in a
    // configuration file elsewhere on the web server VM instance.

    $conn = new mysqli($dbserver, $dbuser, $dbpassword);

    if (mysqli_connect_error()) {
        echo ("Database connection failed: " . mysqli_connect_error());
    } else {
        echo ("Database connection succeeded.");
    }
?>
</body></html>
```

In a later step, you will insert your Cloud SQL instance's IP address and your database password into this file. For now, leave the file unmodified.

5. Press Ctrl+O, and then press Enter to save your edited file.
6. Press Ctrl+X to exit the nano text editor.
7. Restart the web server:

```
sudo service apache2 restart
```

8. Open a new web browser tab and paste into the address bar your bloghost VM instance's external IP address followed by /index.php. The URL will look like this

```
35.192.208.2/index.php
```

NOTE: Be sure to use the external IP address of your VM instance followed by /index.php. Do not use the VM instance's internal IP address. Do not use the sample IP address shown here.

When you load the page, you will see that its content includes an error message beginning with the words:

```
Database connection failed: ...
```

This message occurs because you have not yet configured PHP's connection to your Cloud SQL instance.

9. Return to your ssh session on bloghost. Use the nano text editor to edit index.php again.

```
sudo nano index.php
```

10. In the nano text editor, replace CLOUDSQLIP with the Cloud SQL instance Public IP address that you noted above. Leave the quotation marks around the value in place.
11. In the nano text editor, replace DBPASSWORD with the Cloud SQL database password that you defined above. Leave the quotation marks around the value in place.
12. Press Ctrl+O, and then press Enter to save your edited file.
13. Press Ctrl+X to exit the nano text editor.
14. Restart the web server:

```
sudo service apache2 restart
```

15. Return to the web browser tab in which you opened your bloghost VM instance's external IP address. When you load the page, the following message appears:

```
Database connection succeeded.
```

In an actual blog, the database connection status would not be visible to blog visitors. Instead, the database connection would be managed solely by the administrator.

Task 6: Configure an application in a Compute Engine instance to use a Cloud Storage object

1. Let us work with our subdirectory graphical image [my-excellent-blog.png] saved previously in [\$DEVSHHELL_PROJECT_ID] cloud storage:

Replace <BUCKET_NAME> with \$DEVSHHELL_PROJECT_ID and subdir with my-excellent-blog.png

```
gsutil ls -d gs://<BUCKET_NAME>/subdir
```

2. use -r to copy the entire directory tree of my-excellent-blog.png url

```
gsutil cp -r dir gs://<BUCKET_NAME>
```

3. SSH into bloghost bloghost VM instance

```
gcloud compute ssh bloghost  
or  
ssh bloghost
```

4. Enter this command to set your working directory to the document root of the web server:

```
cd /var/www/html
```

5. Use the nano text editor to edit index.php:

```
sudo nano index.php
```

6. Use the arrow keys to move the cursor to the line that contains the h1 element. Press Enter to open up a new, blank screen line, and then paste the URL you copied earlier into the line.
7. Paste this HTML markup immediately before the URL:

```
<img src='
```

8. Place a closing single quotation mark and a closing angle bracket at the end of the URL:

```
'>
```

The resulting line will look like this:

```
<img src='https://storage.googleapis.com/qwiklabs-gcp-02-17b34294521a/my-  
excellent-blog.png'>
```

The effect of these steps is to place the line containing



My Excellent Blog

immediately before the line containing

...

NOTE: Do not copy the URL shown here. Instead, copy the URL shown by the Storage browser in your own Cloud Platform project.

9. Press Ctrl+O, and then press Enter to save your edited file.

10. Press Ctrl+X to exit the nano text editor.

11. Restart the web server:

```
sudo service apache2 restart
```

12. Return to the web browser tab in which you opened your bloghost VM instance's external IP address. When you load the page, its content now includes a banner image.

Congratulations!

In this lab, you configured a Cloud SQL instance and connected an application in a Compute Engine instance to it. You also worked with a Cloud Storage bucket.

Compiled by:

JoyceMimi