

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

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 2: Deploy a web server VM instance



1. In the GCP Console, on the **Navigation menu** (), click **Compute Engine > VM instances**.
2. Click **Create**.
3. On the **Create an Instance** page, for **Name**, type `bloghost`

4. For **Region** and **Zone**, select the region and zone assigned by Qwiklabs.
5. For **Machine type**, accept the default.
6. For **Boot disk**, if the **Image** shown is not **Debian GNU/Linux 9 (stretch)**, click **Change** and select **Debian GNU/Linux 9 (stretch)**.
7. Leave the defaults for **Identity and API access** unmodified.
8. For **Firewall**, click **Allow HTTP traffic**.
9. Click **Management, security, disks, networking, sole tenancy** to open that section of the dialog.
10. Enter the following script as the value for **Startup script**:

```
apt-get update
apt-get install apache2 php php-mysql -y
service apache2 restart
```

Be sure to supply that script as the value of the **Startup script** field. If you accidentally put it into another field, it won't be executed when the VM instance starts.

11. Leave the remaining settings as their defaults, and click **Create**. Instance can take about two minutes to launch and be fully available for use.
12. On the **VM instances** page, copy the **bloghost** VM instance's internal and external IP addresses to a text editor for use later in this lab. Click *Check my progress* to verify the objective.

Deploy a web server VM instance

Check my progress

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

All Cloud Storage bucket names must be globally unique. To ensure that your bucket name is unique, these instructions will guide you to give your bucket the same name as your Cloud Platform project ID, which is also globally unique.

Cloud Storage buckets can be associated with either a region or a multi-region location: **US**, **EU**, or **ASIA**. In this activity, you associate your bucket with the multi-region closest to the region and zone that Qwiklabs or your instructor assigned you to.

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
```

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
```


Click *Check my progress* to verify the objective.

Create a Cloud Storage bucket using the `gsutil` command line

Check my progress

Task 4: Create the Cloud SQL instance



1. In the GCP Console, on the **Navigation menu** (), click **Storage > SQL**.
2. Click **Create instance**.
3. For **Choose a database engine**, select **MySQL**.
4. For **Instance ID**, type **blog-db**, and for **Root password** type a password of your choice.

Choose a password that you remember. There's no need to obscure the password because you'll use mechanisms to connect that aren't open access to everyone.

5. Set the region and zone assigned by Qwiklabs.

This is the same region and zone into which you launched the **bloghost** instance. The best performance is achieved by placing the client and the database close to each other.

6. Click **Create**.

Wait for the instance to finish deploying. It will take a few minutes.

7. Click on the name of the instance, **blog-db**, to open its details page.
8. From the SQL instances details page, copy the **Public IP address** for your SQL instance to a text editor for use later in this lab.
9. Click the **Users** tab, and then click **Create user account**.
10. For **User name**, type `blogdbuser`
11. For **Password**, type a password of your choice. Make a note of it.
12. Click **Create** to create the user account in the database.

Wait for the user to be created.

13. Click the **Connections** tab, and then click **Add network**.

If you are offered the choice between a **Private IP** connection and a **Public IP** connection, choose **Public IP** for purposes of this lab. The **Private IP** feature is in beta at the time this lab was written.

The **Add network** button may be grayed out if the user account creation is not yet complete.

14. For **Name**, type `web front end`
15. For **Network**, type the external IP address of your **bloghost** VM instance, followed by `/32`

The result will look like this:

Be sure to use the external IP address of your VM instance followed by /32. Do not use the VM instance's internal IP address. Do not use the sample IP address shown here.

16. Click **Done** to finish defining the authorized network.

17. Click **Save** to save the configuration change.


Click *Check my progress* to verify the objective.

Create the Cloud SQL instance

Check my progress

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



1. On the **Navigation menu** (), click **Compute Engine > VM instances**.
2. In the VM instances list, click **SSH** in the row for your VM instance **bloghost**.
3. In your ssh session on **bloghost**, change your working directory to the document root of the web server:

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

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

```
sudo nano index.php
```

5. 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.

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

```
sudo service apache2 restart
```

9. 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
```

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.

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

```
sudo nano index.php
```

11. 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.
12. 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.
13. Press **Ctrl+O**, and then press **Enter** to save your edited file.
14. Press **Ctrl+X** to exit the nano text editor.
15. Restart the web server:

```
sudo service apache2 restart
```

16. 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. In the GCP Console, click **Storage > Browser**.
2. Click on the bucket that is named after your GCP project.
3. In this bucket, there is an object called **my-excellent-blog.png**. Copy the URL behind the link icon that appears in that object's **Public access** column, or behind the words "Public link" if shown.

If you see neither a link icon nor a "Public link", try refreshing the browser. If you still do not see a link icon, return to Cloud Shell and confirm that your attempt to change the object's Access Control list with the **gsutil acl ch** command was successful.

4. Return to your ssh session on your **bloghost** VM instance.
5. Enter this command to set your working directory to the document root of the web server:

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

6. Use the **nano** text editor to edit **index.php**:

```
sudo nano index.php
```

7. 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.
8. Paste this HTML markup immediately before the URL:


```
<img src='
```

9. 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-0005e186fa559a09/my-excellent-blog.png'>
```

The effect of these steps is to place the line containing `` immediately before the line containing `<h1>...</h1>`

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

10. Press **Ctrl+O**, and then press **Enter** to save your edited file.
11. Press **Ctrl+X** to exit the nano text editor.
12. Restart the web server:

```
sudo service apache2 restart
```

13. 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!