# Google Cloud Fundamentals: Getting Started with Cloud Marketplace

25 minutesFree

Rate Lab

**Overview**

In this lab, you use Cloud Marketplace to quickly and easily deploy a LAMP stack on a Compute Engine instance. The Bitnami LAMP Stack provides a complete web development environment for Linux that can be launched in one click.

|  |  |
| --- | --- |
| **Component** | **Role** |
| Linux | Operating system |
| Apache HTTP Server | Web server |
| MySQL | Relational database |
| PHP | Web application framework |
| phpMyAdmin | PHP administration tool |

For more information on the Bitnami LAMP stack, see [Bitnami LAMP Stack Documentation](https://docs.bitnami.com/google/infrastructure/lamp" \t "_blank).

**Objectives**

In this lab, you learn how to launch a solution using Cloud Marketplace.

**Task 1: Sign in to the Google Cloud Platform (GCP) Console**

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, img/time.png and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click img/start_lab.png.
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. 
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

1. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

**Task 2: Use Cloud Marketplace to deploy a LAMP stack**

1. In the GCP Console, on the **Navigation menu** (), click **Marketplace**.
2. In the search bar, type LAMP
3. In the search results, click **LAMP Certified by Bitnami**.

If you choose another LAMP stack, such as the Google Click to Deploy offering, the lab instructions will not work as expected.

1. On the LAMP page, click **Launch**.

If this is your first time using Compute Engine, the Compute Engine API must be initialized before you can continue.

1. For **Zone**, select the deployment zone that Qwiklabs assigned to you.
2. Leave the remaining settings as their defaults.
3. If you are prompted to accept the GCP Marketplace Terms of Service, do so.
4. Click **Deploy**.
5. If a **Welcome to Deployment Manager** message appears, click **Close** to dismiss it.

The status of the deployment appears in the console window: **lampstack-1 is being deployed**. When the deployment of the infrastructure is complete, the status changes to **lampstack-1 has been deployed**.

After the software is installed, a summary of the details for the instance, including the site address, is displayed.

Click *Check my progress* to verify the objective.

Use Cloud Marketplace to deploy a LAMP stack

Check my progress

**Task 3: Verify your deployment**

1. When the deployment is complete, click the **Site address** link in the right pane.

Alternatively, you can click **Visit the site** in the **Get started with LAMP Certified by Bitnami** section of the page. A new browser tab displays a congratulations message. This page confirms that, as part of the LAMP stack, the Apache HTTP Server is running.

1. Close the congratulations browser tab.
2. On the GCP Console, under **Get started with LAMP Certified by Bitnami**, click **SSH**.

In a new window, a secure login shell session on your virtual machine appears.

1. In the just-created SSH window, to change the current working directory to /opt/bitnami, execute the following command:
2. cd /opt/bitnami
3. To copy the phpinfo.php script from the installation directory to a publicly accessible location under the web server document root, execute the following command:
4. sudo sh -c 'echo "<?php phpinfo(); ?>" > apache2/htdocs/phpinfo.php'

The phpinfo.php script displays your PHP configuration. It is often used to verify a new PHP installation.

1. To close the SSH window, execute the following command:
2. exit
3. Open a new browser tab.
4. Type the following URL, and replace SITE\_ADDRESS with the URL in the **Site address** field in the right pane of the **lampstack** page.
5. http://SITE\_ADDRESS/phpinfo.php

A summary of the PHP configuration of your server is displayed.

1. Close the **phpinfo** tab.

**Congratulations!**

In this lab, you deployed a LAMP stack to a Compute Engine instance.

**End your lab**

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2020 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.

**More resources**

Read the [Google Cloud Platform documentation on Cloud Marketplace](https://cloud.google.com/marketplace/docs/).

# Google Cloud Fundamentals: Getting Started with Compute Engine

25 minutesFree

Rate Lab

**Overview**

In this lab, you will create virtual machines (VMs) and connect to them. You will also create connections between the instances.

**Objectives**

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

* Create a Compute Engine virtual machine using the Google Cloud Platform (GCP) Console.
* Create a Compute Engine virtual machine using the gcloud command-line interface.
* Connect between the two instances.

**Task 1: Sign in to the Google Cloud Platform (GCP) Console**

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, img/time.png and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click img/start_lab.png.
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. 
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

1. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

**Task 2: Create a virtual machine using the GCP Console**

1. In the **Navigation menu** (), click **Compute Engine** > **VM instances**.
2. Click **Create**.
3. On the **Create an Instance** page, for **Name**, type my-vm-1
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. Leave all other defaults unmodified.
10. To create and launch the VM, click **Create**.

**Note**: The VM can take about two minutes to launch and be fully available for use.

Click *Check my progress* to verify the objective.

Create a virtual machine using the GCP Console

Check my progress

**Task 3: Create a virtual machine using the gcloud command line**

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



1. Click **Continue**. 
2. To display a list of all the zones in the region to which Qwiklabs assigned you, enter this partial command gcloud compute zones list | grep followed by the region that Qwiklabs or your instructor assigned you to.

Your completed command will look like this:

gcloud compute zones list | grep us-central1

1. Choose a zone from that list other than the zone to which Qwiklabs assigned you. For example, if Qwiklabs assigned you to region us-central1 and zone us-central1-a you might choose zone us-central1-b.
2. To set your default zone to the one you just chose, enter this partial command gcloud config set compute/zone followed by the zone you chose.

Your completed command will look like this:

gcloud config set compute/zone us-central1-b

1. To create a VM instance called **my-vm-2** in that zone, execute this command:
2. gcloud compute instances create "my-vm-2" \
3. --machine-type "n1-standard-1" \
4. --image-project "debian-cloud" \
5. --image "debian-9-stretch-v20190213" \
6. --subnet "default"

**Note**: The VM can take about two minutes to launch and be fully available for use.

1. To close the Cloud Shell, execute the following command:
2. exit

Click *Check my progress* to verify the objective.

Create a virtual machine using the gcloud command line

Check my progress

**Task 4: Connect between VM instances**

1. In the **Navigation menu** (), click **Compute Engine > VM instances**.

You will see the two VM instances you created, each in a different zone.

Notice that the Internal IP addresses of these two instances share the first three bytes in common. They reside on the same subnet in their Google Cloud VPC even though they are in different zones.

1. To open a command prompt on the **my-vm-2** instance, click **SSH** in its row in the **VM instances** list.
2. Use the ping command to confirm that **my-vm-2** can reach **my-vm-1** over the network:
3. ping my-vm-1

Notice that the output of the ping command reveals that the complete hostname of **my-vm-1** is **my-vm-1.c.PROJECT\_ID.internal**, where PROJECT\_ID is the name of your Google Cloud Platform project. GCP automatically supplies Domain Name Service (DNS) resolution for the internal IP addresses of VM instances.

1. Press **Ctrl+C** to abort the ping command.
2. Use the **ssh** command to open a command prompt on **my-vm-1**:
3. ssh my-vm-1

If you are prompted about whether you want to continue connecting to a host with unknown authenticity, enter **yes** to confirm that you do.

1. At the command prompt on **my-vm-1**, install the Nginx web server:
2. sudo apt-get install nginx-light -y
3. Use the **nano** text editor to add a custom message to the home page of the web server:
4. sudo nano /var/www/html/index.nginx-debian.html
5. Use the arrow keys to move the cursor to the line just below the h1 header. Add text like this, and replace YOUR\_NAME with your name:
6. Hi from YOUR\_NAME
7. Press **Ctrl+O** and then press **Enter** to save your edited file, and then press **Ctrl+X** to exit the nano text editor.
8. Confirm that the web server is serving your new page. At the command prompt on **my-vm-1**, execute this command:
9. curl http://localhost/

The response will be the HTML source of the web server's home page, including your line of custom text.

1. To exit the command prompt on **my-vm-1**, execute this command:
2. exit

You will return to the command prompt on **my-vm-2**

1. To confirm that **my-vm-2** can reach the web server on **my-vm-1**, at the command prompt on **my-vm-2**, execute this command:
2. curl http://my-vm-1/

The response will again be the HTML source of the web server's home page, including your line of custom text.

1. In the **Navigation menu** (), click **Compute Engine > VM instances**.
2. Copy the External IP address for **my-vm-1** and paste it into the address bar of a new browser tab. You will see your web server's home page, including your custom text.

If you forgot to click **Allow HTTP traffic** when you created the **my-vm-1** VM instance, your attempt to reach your web server's home page will fail. You can add a [firewall rule](https://cloud.google.com/vpc/docs/firewalls) to allow inbound traffic to your instances, although this topic is out of scope for this course.

**Congratulations!**

In this lab, you created virtual machine (VM) instances in two different zones and connected to them using ping, ssh, and HTTP.

**End your lab**

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2020 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.

**More Resources**

Read the [Google Cloud Platform documentation on Google Compute Engine](https://cloud.google.com/compute/docs/).

Read about [Google Cloud Platform Virtual Private Cloud (VPC)](https://cloud.google.com/compute/docs/vpc/).

# Google Cloud Fundamentals: Getting Started with Cloud Storage and Cloud SQL

50 minutesFree

Rate Lab

**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**

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, img/time.png and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click img/start_lab.png.
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. 
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

1. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

**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.

1. Leave the remaining settings as their defaults, and click **Create**.

Instance can take about two minutes to launch and be fully available for use.

1. 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

1. In Cloud Shell, the DEVSHELL\_PROJECT\_ID environment variable contains your project ID. Enter this command to make a bucket named after your project ID:

gsutil mb -l $LOCATION gs://$DEVSHELL\_PROJECT\_ID

1. 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

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

gsutil cp my-excellent-blog.png gs://$DEVSHELL\_PROJECT\_ID/my-excellent-blog.png

1. 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://$DEVSHELL\_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 **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.

1. 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.

1. Click **Create**.

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

1. Click on the name of the instance, **blog-db**, to open its details page.
2. 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.
3. Click on **Users** menu on the left-hand side, and then click **ADD USER ACCOUNT**.
4. For **User name**, type blogdbuser
5. For **Password**, type a password of your choice. Make a note of it.
6. Click **Create** to create the user account in the database.

Wait for the user to be created.

1. 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.

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

The result will look like this:

35.192.208.2/32

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.

1. Click **Done** to finish defining the authorized network.
2. 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

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

sudo nano index.php

1. 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.

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

sudo service apache2 restart

1. 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.

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

sudo nano index.php

1. 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.
2. 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.
3. Press **Ctrl+O**, and then press **Enter** to save your edited file.
4. Press **Ctrl+X** to exit the nano text editor.
5. Restart the web server:

sudo service apache2 restart

1. 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.

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

cd /var/www/html

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

sudo nano index.php

1. 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.
2. Paste this HTML markup immediately before the URL:

<img src='

1. 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 <img src='...'> 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.

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

sudo service apache2 restart

1. 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.

**End your lab**

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2020 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.

**More resources**

Read the [Google Cloud Platform documentation on Cloud SQL](https://cloud.google.com/sql/docs/).

Read the [Google Cloud Platform documentation on Cloud Storage](https://cloud.google.com/storage/docs/).

# Google Cloud Fundamentals: Getting Started with GKE

35 minutesFree

Rate Lab

**Overview**

In this lab, you create a Google Kubernetes Engine cluster containing several containers, each containing a web server. You place a load balancer in front of the cluster and view its contents.

**Objectives**

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

* Provision a [Kubernetes](http://kubernetes.io/) cluster using [Kubernetes Engine.](https://cloud.google.com/container-engine)
* Deploy and manage Docker containers using kubectl.

**Task 1: Sign in to the Google Cloud Platform (GCP) Console**

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, img/time.png and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click img/start_lab.png.
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. 
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

1. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

**Task 2: Confirm that needed APIs are enabled**

1. Make a note of the name of your GCP project. This value is shown in the top bar of the Google Cloud Platform Console. It will be of the form qwiklabs-gcp- followed by hexadecimal numbers.
2. In the GCP Console, on the **Navigation menu** (), click **APIs & Services**.
3. Scroll down in the list of enabled APIs, and confirm that both of these APIs are enabled:

* Kubernetes Engine API
* Container Registry API

If either API is missing, click **Enable APIs and Services** at the top. Search for the above APIs by name and enable each for your current project. (You noted the name of your GCP project above.)

**Task 3: Start a Kubernetes Engine cluster**

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



1. Click **Continue**. 
2. For convenience, place the zone that Qwiklabs assigned you to into an environment variable called MY\_ZONE. At the Cloud Shell prompt, type this partial command:
3. export MY\_ZONE=

followed by the zone that Qwiklabs assigned to you. Your complete command will look similar to this:

export MY\_ZONE=us-central1-a

1. Start a Kubernetes cluster managed by Kubernetes Engine. Name the cluster **webfrontend** and configure it to run 2 nodes:
2. gcloud container clusters create webfrontend --zone $MY\_ZONE --num-nodes 2

It takes several minutes to create a cluster as Kubernetes Engine provisions virtual machines for you.

1. After the cluster is created, check your installed version of Kubernetes using the kubectl version command:
2. kubectl version

The gcloud container clusters create command automatically authenticated kubectl for you.

1. View your running nodes in the GCP Console. On the **Navigation menu** (), click **Compute Engine > VM Instances**.

Your Kubernetes cluster is now ready for use.

Click *Check my progress* to verify the objective.

Start a Kubernetes Engine cluster

Check my progress

**Task 4: Run and deploy a container**

1. From your Cloud Shell prompt, launch a single instance of the nginx container. (Nginx is a popular web server.)
2. kubectl create deploy nginx --image=nginx:1.17.10

In Kubernetes, all containers run in pods. This use of the kubectl create command caused Kubernetes to create a deployment consisting of a single pod containing the nginx container. A Kubernetes deployment keeps a given number of pods up and running even in the event of failures among the nodes on which they run. In this command, you launched the default number of pods, which is 1.

**Note**: If you see any deprecation warning about future version you can simply ignore it for now and can proceed further.

1. View the pod running the nginx container:
2. kubectl get pods
3. Expose the nginx container to the Internet:
4. kubectl expose deployment nginx --port 80 --type LoadBalancer

Kubernetes created a service and an external load balancer with a public IP address attached to it. The IP address remains the same for the life of the service. Any network traffic to that public IP address is routed to pods behind the service: in this case, the nginx pod.

1. View the new service:
2. kubectl get services

You can use the displayed external IP address to test and contact the nginx container remotely.

It may take a few seconds before the **External-IP** field is populated for your service. This is normal. Just re-run the kubectl get services command every few seconds until the field is populated.

1. Open a new web browser tab and paste your cluster's external IP address into the address bar. The default home page of the Nginx browser is displayed.
2. Scale up the number of pods running on your service:
3. kubectl scale deployment nginx --replicas 3

Scaling up a deployment is useful when you want to increase available resources for an application that is becoming more popular.

1. Confirm that Kubernetes has updated the number of pods:
2. kubectl get pods
3. Confirm that your external IP address has not changed:
4. kubectl get services
5. Return to the web browser tab in which you viewed your cluster's external IP address. Refresh the page to confirm that the nginx web server is still responding.

Click *Check my progress* to verify the objective.

Run and deploy a container

Check my progress

**Congratulations!**

In this lab, you configured a Kubernetes cluster in Kubernetes Engine. You populated the cluster with several pods containing an application, exposed the application, and scaled the application.

**End your lab**

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2020 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.

# Google Cloud Fundamentals: Getting Started with App Engine

20 minutesFree

Rate Lab

## Overview

In this lab, you create and deploy a simple App Engine application using a virtual environment in the Google Cloud Shell.

## Objectives

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

* Initialize App Engine.
* Preview an App Engine application running locally in Cloud Shell.
* Deploy an App Engine application, so that others can reach it.
* Disable an App Engine application, when you no longer want it to be visible.

## Set up your lab environment

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, img/time.png and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click img/start_lab.png.
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. 
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

1. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

### 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 GCP resources.

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



1. 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:



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

You can list the active account name with this command:

gcloud auth list

Output:

Credentialed accounts:

- <myaccount>@<mydomain>.com (active)

Example output:

Credentialed accounts:

- google1623327\_student@qwiklabs.net

You can list the project ID with this command:

gcloud config list project

Output:

[core]

project = <project\_ID>

Example output:

[core]

project = qwiklabs-gcp-44776a13dea667a6

Full documentation of **gcloud** is available on [Google Cloud gcloud Overview](https://cloud.google.com/sdk/gcloud).

## Task 1: Initialize App Engine

1. Initialize your App Engine app with your project and choose its region:
2. gcloud app create --project=$DEVSHELL\_PROJECT\_ID

When prompted, select the [region](https://cloud.google.com/appengine/docs/locations)where you want your App Engine application located.

1. Clone the source code repository for a sample application in the **hello\_world** directory:
2. git clone https://github.com/GoogleCloudPlatform/python-docs-samples
3. Navigate to the source directory:
4. cd python-docs-samples/appengine/standard\_python3/hello\_world

## Task 2: Run Hello World application locally

In this task, you run the Hello World application in a local, virtual environment in Cloud Shell.

Ensure that you are at the Cloud Shell command prompt.

1. Execute the following command to download and update the packages list.
2. sudo apt-get update
3. Set up a virtual environment in which you will run your application. Python virtual environments are used to isolate package installations from the system.
4. sudo apt-get install virtualenv

If prompted [Y/n], press Y and then Enter.

virtualenv -p python3 venv

1. Activate the virtual environment.
2. source venv/bin/activate
3. Navigate to your project directory and install dependencies.
4. pip install -r requirements.txt
5. Run the application:
6. python main.py

Please ignore the warning if any.

1. In **Cloud Shell**, click **Web preview** (Web Preview) > **Preview on port 8080** to preview the application.

To access the **Web preview** icon, you may need to collapse the **Navigation menu**.

Result:



1. To end the test, return to Cloud Shell and press **Ctrl+C** to abort the deployed service.
2. Using the Cloud Console, verify that the app is not deployed. In the Cloud Console, on the **Navigation menu** (), click **App Engine** > **Dashboard**.

Notice that no resources are deployed.

## Task 3: Deploy and run Hello World on App Engine

To deploy your application to the App Engine Standard environment:

1. Navigate to the source directory:
2. cd ~/python-docs-samples/appengine/standard\_python3/hello\_world
3. Deploy your Hello World application.
4. gcloud app deploy

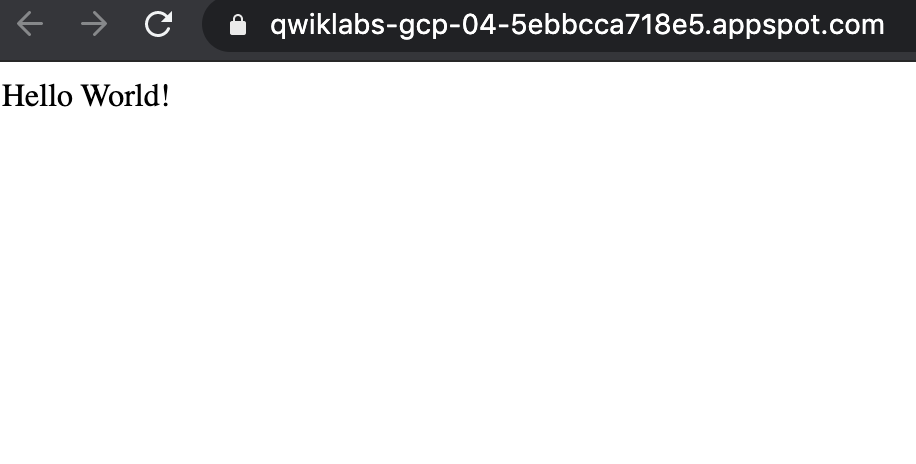
If prompted "Do you want to continue (Y/n)?", press Y and then Enter.

This **app deploy** command uses the app.yaml file to identify project configuration.

1. Launch your browser to view the app at http://YOUR\_PROJECT\_ID.appspot.com
2. gcloud app browse

Copy and paste the URL into a new browser window.

Result:



Congratulations! You created your first application using App Engine.

Click Check my progress to verify the objective.

Deploy the Hello World application to App Engine

Check my progress

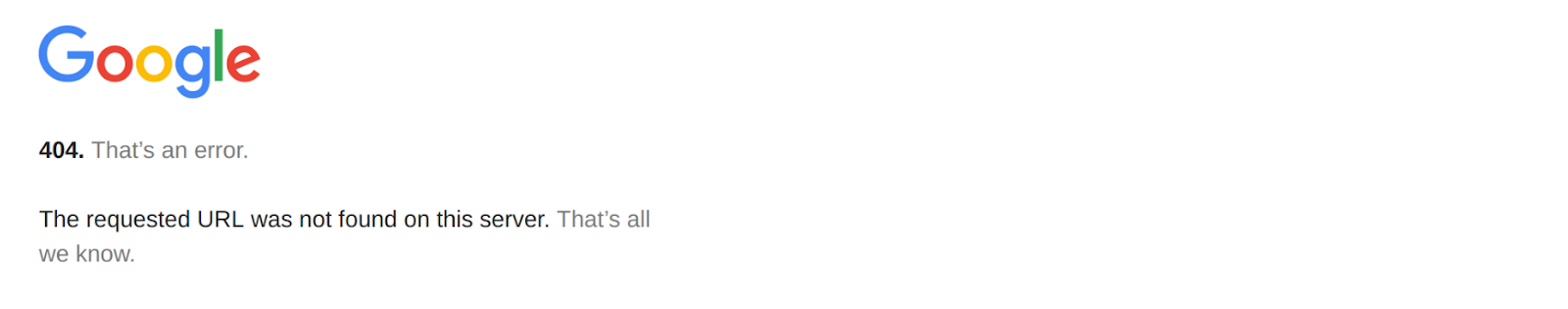
## Task 4: Disable the application

App Engine offers no option to **Undeploy** an application. After an application is deployed, it remains deployed, although you could instead replace the application with a simple page that says something like "not in service."

However, you can disable the application, which causes it to no longer be accessible to users.

1. In the Cloud Console, on the **Navigation menu** (), click **App Engine** > **Settings**.
2. Click **Disable application**.
3. Read the dialog message. Enter the **App ID** and click **DISABLE**.

If you refresh the browser window you used to view to the application site, you'll get a 404 error.



## Congratulations!

You created your first application using App Engine!

## End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2020 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.

# Google Cloud Fundamentals: Getting Started with Deployment Manager and Cloud Monitoring

45 minutesFree

Rate Lab

## Overview

In this lab, you create a deployment using Deployment Manager and use it to maintain a consistent state of your deployment. You will also view resource usage in a VM instance using Cloud Monitoring.

## Objectives

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

* Create a Deployment Manager deployment.
* Update a Deployment Manager deployment.
* View the load on a VM instance using Cloud Monitoring.

## Task 1: Sign in to the Google Cloud Platform (GCP) Console

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, img/time.png and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click img/start_lab.png.
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. 
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

1. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

## Task 2: Confirm that needed APIs are enabled

1. Make a note of the name of your GCP project. This value is shown in the top bar of the Google Cloud Platform Console. It will be of the form qwiklabs-gcp- followed by hexadecimal numbers.
2. In the GCP Console, on the **Navigation menu** (), click **APIs & services**.
3. Scroll down in the list of enabled APIs, and confirm that these APIs are enabled:

* Cloud Deployment Manager v2 API
* Cloud Runtime Configuration API
* Cloud Monitoring API

1. If one or more APIs is missing, click the **Enable APIs and Services** button at top. Search for the above APIs by name and enable each for your current project. (You noted the name of your GCP project above.)

## Task 3: Create a Deployment Manager deployment

1. In GCP console, on the top right toolbar, click the Open Cloud Shell button (). Click **Continue**.
2. For your convenience, place the zone that Qwiklabs assigned you to into an environment variable called MY\_ZONE. At the Cloud Shell prompt, type this partial command:

export MY\_ZONE=

followed by the zone that Qwiklabs assigned you to. Your complete command will look similar to this:

export MY\_ZONE=us-central1-a

1. At the Cloud Shell prompt, download an editable Deployment Manager template:

gsutil cp gs://cloud-training/gcpfcoreinfra/mydeploy.yaml mydeploy.yaml

1. In the Cloud Shell, use the sed command to replace the PROJECT\_ID placeholder string with your Google Cloud Platform project ID using this command:

sed -i -e "s/PROJECT\_ID/$DEVSHELL\_PROJECT\_ID/" mydeploy.yaml

1. In the Cloud Shell, use the sed command to replace the ZONE placeholder string with your Google Cloud Platform zone using this command:

sed -i -e "s/ZONE/$MY\_ZONE/" mydeploy.yaml

1. View the mydeploy.yaml file, with your modifications, with this command:

cat mydeploy.yaml

The file will look something like this:

resources:

- name: my-vm

type: compute.v1.instance

properties:

zone: us-central1-a

machineType: zones/us-central1-a/machineTypes/n1-standard-1

metadata:

items:

- key: startup-script

value: "apt-get update"

disks:

- deviceName: boot

type: PERSISTENT

boot: true

autoDelete: true

initializeParams:

sourceImage: https://www.googleapis.com/compute/v1/projects/debian-cloud/global/images/debian-9-stretch-v20180806

networkInterfaces:

- network: https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-dcdf854d278b50cd/global/networks/default

accessConfigs:

- name: External NAT

type: ONE\_TO\_ONE\_NAT

Do not use the above text literally in your own **mydeploy.yaml** file. Be sure that the zone that is named on the **zone:** and **machineType:** lines in your file matches the zone to which Qwiklabs assigned you. Be sure that the GCP project ID on the **network:** line in your file matches the project ID to which Qwiklabs assigned you, not the one in this example.

1. Build a deployment from the template:

gcloud deployment-manager deployments create my-first-depl --config mydeploy.yaml

When the deployment operation is complete, the **gcloud** command displays a list of the resources named in the template and their current state.

1. Confirm that the deployment was successful. In the GCP Console, on the **Navigation menu** (), click **Compute Engine > VM instances**. You will see that a VM instance called **my-vm** has been created, as specified by the template.
2. Click on the VM instance's name to open its VM instance details screen.
3. Scroll down to the **Custom metadata** section. Confirm that the startup script you specified in your Deployment Manager template has been installed.

Click Check my progress to verify the objective.

Create a Deployment Manager deployment

Check my progress

## Task 4: Update a Deployment Manager deployment

1. Return to your Cloud Shell prompt. Launch the nano text editor to edit the **mydeploy.yaml** file:

nano mydeploy.yaml

1. Find the line that sets the value of the startup script, value: "apt-get update", and edit it so that it looks like this:

value: "apt-get update; apt-get install nginx-light -y"

Do not disturb the spaces at the beginning of the line. The YAML templating language relies on indented lines as part of its syntax. As you edit the file, be sure that the v in the word value in this new line is immediately below the k in the word key on the line above it.

1. Press **Ctrl+O** and then press **Enter** to save your edited file.
2. Press **Ctrl+X** to exit the **nano** text editor.
3. Return to your Cloud Shell prompt. Enter this command to cause Deployment Manager to update your deployment to install the new startup script:

gcloud deployment-manager deployments update my-first-depl --config mydeploy.yaml

Wait for the **gcloud** command to display a message confirming that the update operation was completed successfully.

1. In the GCP console, on the **Navigation menu** (), click **Compute Engine > VM instances**.
2. Click on the **my-vm** VM instance's name to open its **VM instance details** pane.
3. Scroll down to the **Custom metadata** section. Confirm that the startup script has been updated to the value you declared in your Deployment Manager template.

Click Check my progress to verify the objective.

Update the Deployment Manager deployment

Check my progress

## Task 5: View the Load on a VM using Cloud Monitoring

1. In the GCP Console, on the **Navigation menu** (), click **Compute Engine** > **VM instances**.
2. To open a command prompt on the **my-vm** instance, click **SSH** in its row in the **VM instances** list.
3. In the ssh session on **my-vm**, execute this command to create a CPU load:

dd if=/dev/urandom | gzip -9 >> /dev/null &

This Linux pipeline forces the CPU to work on compressing a continuous stream of random data.

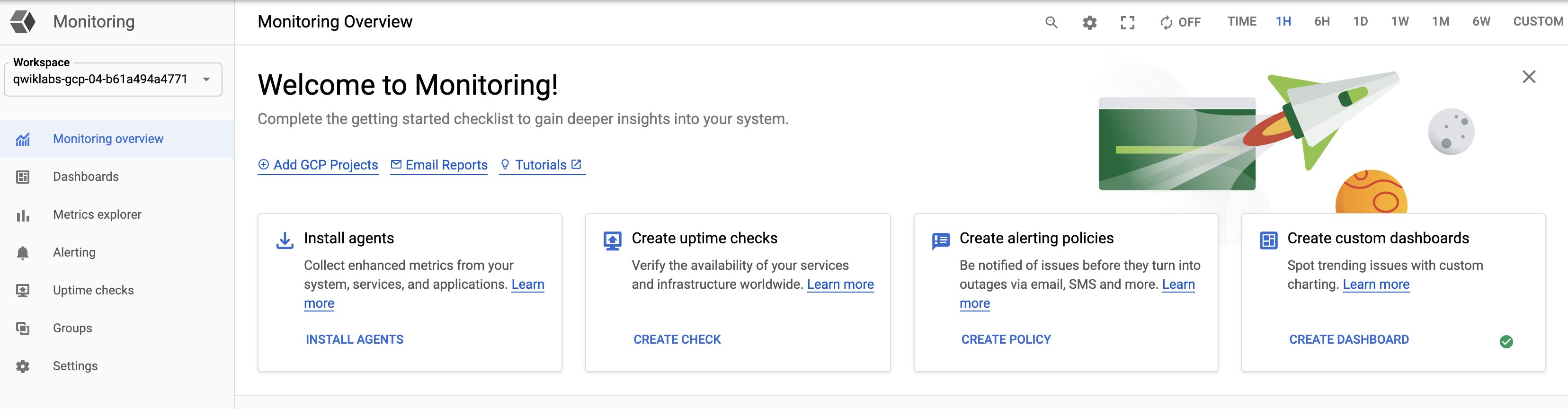
Leave the window containing your SSH session open while you proceed with the lab.

### Create a Monitoring workspace

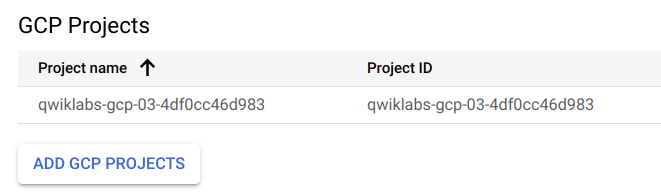
You will now setup a Monitoring workspace that's tied to your Qwiklabs GCP Project. The following steps create a new account that has a free trial of Monitoring.

1. In the Google Cloud Platform Console, click on **Navigation menu** > **Monitoring**.
2. Wait for your workspace to be provisioned.

When the Monitoring dashboard opens, your workspace is ready.



1. Click on **Settings** option from the left panel and confirm that the GCP project which Qwiklabs created for you is shown under the **GCP Projects** section.



1. Under the **Settings** tab menu, click **Agent**. Using your VM's open SSH window and the code shown on the Agents page, install both the Monitoring and Logging agents on your project's VM.
2. Once both of the agents have been installed on your project's VM, click **Metrics Explorer** under the main Cloud Monitoring menu on the far left.
3. In the **Metric** pane of **Metrics Explorer**, select the resource type **GCE VM instance** and the metric **CPU usage**.

In the resulting graph, notice that CPU usage increased sharply a few minutes ago.

1. Terminate your workload generator. Return to your ssh session on **my-vm** and enter this command:

kill %1

## End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2020 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.

# Google Cloud Fundamentals: Getting Started with BigQuery

30 minutesFree

Rate Lab

## Overview

In this lab, you load a web server log into a BigQuery table. After loading the data, you query it using the BigQuery web user interface and the BigQuery CLI.

BigQuery helps you perform interactive analysis of petabyte-scale databases, and it enables near-real time analysis of massive datasets. It offers a familiar SQL 2011 query language and functions.

Data stored in BigQuery is highly durable. Google stores your data in a replicated manner by default and at no additional charge for replicas. With BigQuery, you pay only for the resources you use. Data storage in BigQuery is inexpensive. Queries incur charges based on the amount of data they process: when you submit a query, you pay for the compute nodes only for the duration of that query. You don't have to pay to keep a compute cluster up and running.

Using BigQuery involves interacting with a number of Google Cloud Platform resources, including projects (covered elsewhere in this course), datasets, tables, and jobs. This lab introduces you to some of these resources, and this brief introduction summarizes their role in interacting with BigQuery.

**Datasets:** A dataset is a grouping mechanism that holds zero or more tables. A dataset is the lowest level unit of access control. Datasets are owned by GCP projects. Each dataset can be shared with individual users.

**Tables:** A table is a row-column structure that contains actual data. Each table has a schema that describes strongly typed columns of values. Each table belongs to a dataset.

## Objectives

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

* Load data from Cloud Storage into BigQuery.
* Perform a query on the data in BigQuery.

## Task 1: Sign in to the Google Cloud Platform (GCP) Console

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, img/time.png and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click img/start_lab.png.
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. 
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

1. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

Make a note of whether your assigned region is closer to the United States or to Europe.

## Task 2: Load data from Cloud Storage into BigQuery

1. In the Console, on the **Navigation menu** () click **BigQuery** then click **Done**.
2. Create a new dataset within your project by selecting your project in the Resources section, then clicking on **CREATE DATASET** on the right.
3. In the **Create Dataset** dialog, for **Dataset ID**, type **logdata**.
4. For **Data location**, select the continent closest to the region your project was created in. click **Create dataset**.
5. Create a new table in the **logdata** to store the data from the CSV file.
6. Click on **Create Table**. On the **Create Table** page, in the **Source** section:

* For **Create table from**, choose select **Google Cloud Storage**, and in the field, type gs://cloud-training/gcpfci/access\_log.csv.
* Verify **File format** is set to **CSV**.

**Note:** When you have created a table previously, the Create from Previous Job option allows you to quickly use your settings to create similar tables.

1. In the **Destination** section:

* For **Dataset name**, leave **logdata** selected.
* For **Table name**, type **accesslog**.
* For **Table type**, **Native table** should be selected.

1. Under **Schema** section, for **Auto detect** check the **Schema and input Parameters**.
2. Accept the remaining default values and click **Create Table**.

BigQuery creates a load job to create the table and upload data into the table (this may take a few seconds).

1. (Optional) To track job progress, click **Job History**.
2. When the load job is complete, click **logdata** > **accesslog**.
3. On the table details page, click **Details** to view the table properties, and then click **Preview** to view the table data.

Each row in this table logs a hit on a web server. The first field, **string\_field\_0**, is the IP address of the client. The fourth through ninth fields log the day, month, year, hour, minute, and second at which the hit occurred. In this activity, you will learn about the daily pattern of load on this web server.

Click Check my progress to verify the objective.

Load data from Cloud Storage into BigQuery

Check my progress

## Task 3: Perform a query on the data using the BigQuery web UI

In this section of the lab, you use the BigQuery web UI to query the **accesslog** table you created previously.

1. In the **Query editor** window, type (or copy-and-paste) the following query:
2. Because you told BigQuery to automatically discover the schema when you load the data, the hour of the day during which each web hit arrived is in a field called **int\_field\_6**.
3. select int64\_field\_6 as hour, count(\*) as hitcount from logdata.accesslog
4. group by hour

order by hour

Notice that the Query Validator tells you that the query syntax is valid (indicated by the green check mark) and indicates how much data the query will process. The amount of data processed allows you to determine the price of the query using the [Cloud Platform Pricing Calculator](https://cloud.google.com/products/calculator/).

1. Click **Run** and examine the results. At what time of day is the website busiest? When is it least busy?

## Task 4: Perform a query on the data using the bq command

In this section of the lab, you use the bq command in Cloud Shell to query the **accesslog** table you created previously.

1. On the **Google Cloud Platform** Console, click **Activate Cloud Shell**  then click **Continue**.
2. At the Cloud Shell prompt, enter this command:
3. bq query "select string\_field\_10 as request, count(\*) as requestcount from logdata.accesslog group by request order by requestcount desc"

The first time you use the bq command, it caches your Google Cloud Platform credentials, and then asks you to choose your default project. Choose the project that Qwiklabs assigned you to. Its name will look like qwiklabs-gcp- followed by a hexadecimal number.

The bq command then performs the action requested on its command line. What URL offered by this web server was most popular? Which was least popular?

## Congratulations!

In this lab, you loaded data stored in Cloud Storage into a table hosted by Google BigQuery. You then queried the data to discover patterns.

## End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2020 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.