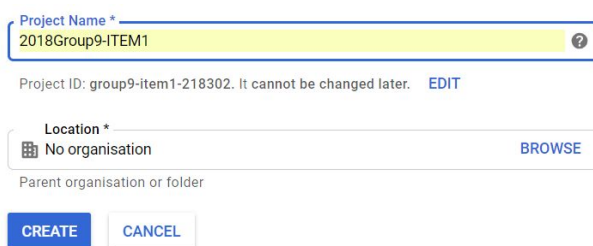


# 2018 Group 9 Item1

## Instructions

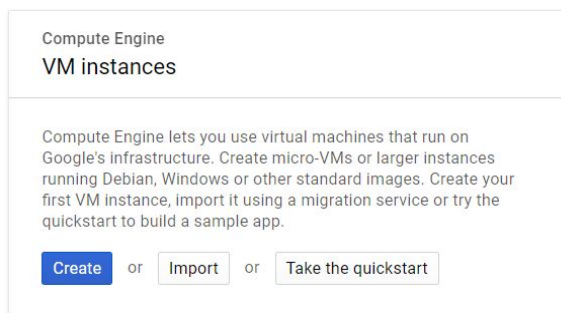
### Create an instance on Google Cloud Platform

1. Go to [Google Cloud Platform](#), create a new project called *2018Group9-ITEM1*;



The screenshot shows the 'Create Project' form in the Google Cloud Platform console. The 'Project Name' field is filled with '2018Group9-ITEM1'. Below it, the 'Project ID' is shown as 'group9-item1-218302'. The 'Location' dropdown is set to 'No organisation'. At the bottom, there are 'CREATE' and 'CANCEL' buttons.

2. Select this project and open it;
3. Click **Compute Engine** from navigation bar. When it is ready, click **Create** to launch a new instance, this will bring you to instance setting page;



The screenshot shows the 'Compute Engine VM instances' page. It includes a description of Compute Engine and three buttons: 'Create', 'Import', and 'Take the quickstart'.

4. In the **boot disk** section, change **OS images** to **Ubuntu 16.04 LTS** and click **Select**;
5. In the **identify and API access** section, select **Allow full access to all Cloud APIs**;
6. In the **firewall** section, select **Allow HTTP traffic** and **Allow HTTPS traffic**;

**Boot disk** ?

New 10 GB standard persistent disk

Image  
 Change

**Identity and API access** ?

Service account ?  
 Compute Engine default service account

Access scopes ?

- ☐ Allow default access
- ☒ Allow full access to all Cloud APIs
- ☐ Set access for each API

**Firewall** ?  
 Add tags and firewall rules to allow specific network traffic from the Internet.

- ☒ Allow HTTP traffic
- ☒ Allow HTTPS traffic

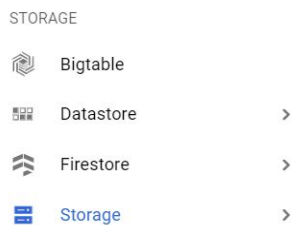
5. Make sure you have selected all the options right, click **Create**;
6. After the instance is ready, you can see it from your VM console. We will later refer to this instance as **local instance**.

## Transfer local folder to the instance

### Platform-independent file transfer method

We will use **Google Cloud Storage** to upload your local folder to a Storage Bucket and download it to your local instance;

1. Click **Storage** in the main navigation bar;



2. Create a new Bucket, find an available name by yourself, (let's assume the name is [your\\_bucket\\_name](#)) keep the other settings default and Click **Create**;

**Name** ?  
Must be unique across Cloud Storage. If you're [serving website content](#), enter the website domain as the name.

**Default storage class** ?  
[Compare storage classes](#)

☒ Multi-Regional  
☐ Regional  
☐ Nearline  
☐ Coldline

**Location**  
United States

<b>Storage cost</b> \$0.026 per GB-month	<b>Retrieval cost</b> Free	<b>Class A operations</b> ? \$0.005 per 1,000 ops	<b>Class B operations</b> ? \$0.0004 per 1,000 ops
---	-------------------------------	--	---

[Show advanced settings](#)

[Create](#) [Cancel](#)

- Click **Upload folder**, you can then upload our folder *2018Group9-ITEM1* to the bucket;

**Note:** What you have downloaded from Stream should be a zip file, please unzip it to a folder and upload this folder, do not change the folder name!

[Upload files](#)
[Upload folder](#)
[Create folder](#)
[Delete](#)

Filter by prefix...

[Buckets](#) / group-item1

<input type="checkbox"/>	Name	Size	Type	Storage class
<input type="checkbox"/>	2018Group9-ITEM1/	—	Folder	—

- After the entire folder has been uploaded, go back to your VM console, find you instance, click **SSH** to connect to it. Then a terminal window will open;

<input type="checkbox"/>	Name ^	Zone	Recommendation	Internal IP	External IP	Connect
<input type="checkbox"/>	instance-1	us-east1-b		10.142.0.2 (nic0)	35.196.61.169	SSH ▾ ⋮

- After the terminal is ready, switch to root user. Then download the folder from buckets you created before, replace the *your\_bucket\_name* with your own bucket name:

```
>>> sudo -i
```

```
>>> gsutil cp -r gs://your_bucket_name/2018Group9-ITEM1/ .
```

## Execute script and playbook

### Install Ansible

You need to install Ansible, we provide a shell script to help you do this. Besides installation, it will also change some directories inside the folder:

```
>>> sh 2018Group9-ITEM1/installAnsible.sh
```

### Run the playbook

Before you run the playbook, you need to pass an extra variable which is your project id, because this will make sure you are creating instances under your credential;

To find your project id, find and click your project name on the top side of the page, it will then prompt out a window, find the project id of *2018Group9-ITEM1* and copy it;



Replace the *your\_project\_id* with your own project id:

```
>>> ansible-playbook task3.yml --extra-vars "project_id=your_project_id"
```

**Note:** *Do not leave any extra blank space*

Then the playbook will start running, it firstly creates multiple instances, and configure them with the roles they supposed to be. Please be patient, this will take you a few minutes;

After all the tasks have been finished, refresh your VM console, you can now see 5 additional instances created : *managementserver*, *webserver1*, *webserver2*, *webserver3*, *databaseserver*

## Verification from test result

### Test Reverse Proxy

Copy the management server's **external ip address** and paste it into your web browser.

**Note:** Please don't add https in front of it, this is not allowed.

<input type="checkbox"/>	✔ databaseserver	australia-southeast1-b	10.152.0.3 (nic0)	35.189.15.190 ↗	SSH ▾	⋮
<input type="checkbox"/>	✔ instance-2	us-east1-b	10.142.0.2 (nic0)	35.231.67.73	SSH ▾	⋮
<input type="checkbox"/>	✔ managementserver	australia-southeast1-b	10.152.0.2 (nic0)	35.197.172.181 ↗	SSH ▾	⋮
<input type="checkbox"/>	✔ webserver1	australia-southeast1-b	10.152.0.4 (nic0)	35.189.54.96 ↗	SSH ▾	⋮
<input type="checkbox"/>	✔ webserver2	australia-southeast1-b	10.152.0.5 (nic0)	35.197.181.92 ↗	SSH ▾	⋮
<input type="checkbox"/>	✔ webserver3	australia-southeast1-b	10.152.0.6 (nic0)	35.197.180.160 ↗	SSH ▾	⋮

Then you can see Etherpad interface on the web page, this means reverse proxy has successfully built since you were visiting the management server instead of web server.

**Note:** Unfortunately, you cannot create a notebook of the etherpad by using management server ip address, an error will occurs. Etherpad hasn't fix this bug yet, please refer to this [bug report](#).

#### An error occurred

The error was reported with the following id: 'xwze61HP0F36MPIPOjsw'

**Please press and hold Ctrl and press F5 to reload this page, if the problem persists please send this error message to your webmaster:**

'Errorid: xwze61HP0F36MPIPOjsw  
URL: http://35.197.191.80/p/demo  
UserAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.92 Safari/537.36  
Uncaught TypeError: Cannot read property 'setStateIdle' of null in http://35.197.191.80/javascripts/lib/ep\_etherpad-lite/static/js/pad.js?callback=require.define at line 3'

In order to make our test more convincing, we provide a log file to prove that your request has been passed from management server to a web server behind it. Visit port 443 of management server (*your\_managementserver\_ip:443*), click **log**, this will download the log file of Nginx.

Inside the log file, it recorded when the management server pass its proxy to which ip address.

For example, the image below indicates that at 03/Oct/2018 8:51:22, it passed proxy to 35.189.15.190:80, which is one of these web servers' ip address.

```
"- 03/Oct/2018:08:51:22 +0000 -35.189.15.190:80 "
```

## Test Load Balancing

Firstly, you need to run a playbook, this will configure your local instance as a 'Load Tester', we use ApacheBench to simulate a large number of concurrent requests (total requests: 5000, concurrent number:10),

```
>>> ansible-playbook task3_test.yml
```

This will take you a few minutes, please be patient. After that, visit port 443 of management server (*your\_managementserver\_ip:443*). Click test\_result, this will bring you to the load testing result which is generated by ApacheBench.

## Test remote database access

You can pick up any web server's ip address, paste it into your web browser, connect to it.

Inside the web page, you should see etherpad is working fine. You can create a pad with name *demo*;

A screenshot of a web form. It consists of a text input field with the word "demo" typed inside, and an "OK" button to its right. The form has a simple, clean design with a light gray border.

**Note:** Please don't use create newpad button, because this will create a pad with a random name, instead, type the pad name into the text box.

Inside the pad, edit it whatever you want.

Then, connect to another web server's ip address, type the same pad name *demo*, you can see your previous pad, which demonstrates these web servers are sharing the same database.