# GCP Worldwide Autoscaling and Load Balancing

GitHub Repo: programming mentor

## Part 1: Create GCP Account and Project

> **Note:** Skip this if you already have an account

### Objectives

- Register for the Google Cloud Platform free trial
- Create a project using the Google Developers Console

### Description

In this step, you register for the Google Cloud Platform free trial and you create a project. The free trial provides you:

- $300 Credit for Free
- Access to Google Cloud Platform Products
- You Won't be Billed (though you need to enter your credit card)
- Build with the Power, Speed, Security, Reliability, and Scalability of Google

### Account

To register for the free trial:

**Step 1**

Open the free trial registration page - https://console.developers.google.com/freetrial

**Step 2**

If you do not have a Gmail account, follow the steps to create one. Otherwise, login and proceed to the next step.

**Step 3**

Complete the registration form.

**Step 4**

Read and agree to the terms of service.

**Step 5**

Click Accept and start free trial.

### Project

You create your first project using the Google Cloud Platform Console. The project is used to complete the rest of the lab.

To create a project:

**Step 1**

In the Google Cloud Platform Console, click Select a project > Create a project.

**Step 2**

In the 'New Project' dialog:

For Project name, type `gcpssproject`. Make a note of the `Project ID` in the text below the project name box; you need it later. Click Create.

**Step 3 (*OPTIONAL*)**

> **Important:** In this step you can Upgrade your account to **full paid** account. When you upgrade your account, you immediately have access to standard service quotas, which are higher than those available on the free trial. But this is completely optional if you want to experiment with larger quantity and/or size of VMs. **You will be able to complete this lab without upgrading**.

In the upper-right corner of the console, a button will appear asking you to upgrade your account. Click Upgrade when you see it. If the Upgrade button does not appear, you may skip this step. If the button appears later, click it when it does.

When you upgrade your account, you immediately have access to standard service quotas, which are higher than those available on the free trial.

**Step 4**

On the GCP Console, use the left-hand side menu to navigate to Compute Engine and ensure that there are no errors.

> **Important:** By navigating to Compute Engine, a default network is set up and and some APIs are enabled. If you don't do this, then you won't be able to do the later exercises easily. You will have to enable everything manualy

At the end of this lab, you may delete this project and close your billing account if desired.

---

## Part 2: Images and Regions

### Objectives

- Configure a machine to run a web application
- Create a snapshot and custom image from your configured instance
- Create new machines using your custom image

### Step 1

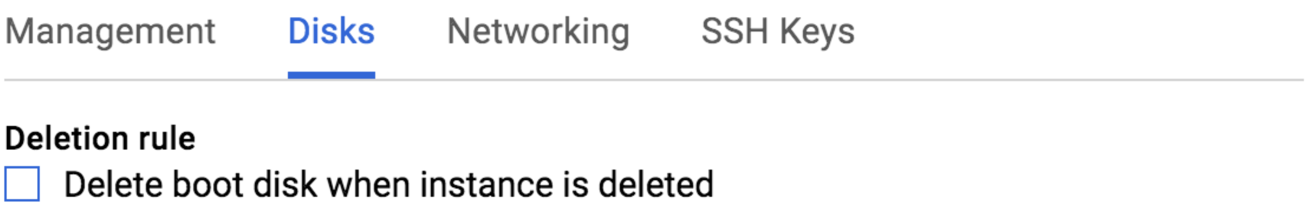
From the **Products and Services** menu, choose **Compute Engine**. Click the **Create** button to create a virtual machine.

Name your new machine `web-server`. Use any zone you wish.

Scroll to the **Firewalls** section and check the box to enable **HTTP** (*leave HTTPS unchecked*).

Click the **Management, disk, networking, SSH keys** link to expand the advanced options.

Then, click the **Disks** tab. Uncheck the **Delete boot disk when instance is deleted** checkbox as shown below.



Deletion rule

□ Delete boot disk when instance is deleted

Leave all the other settings at their defaults and then click **Create**.

### Step 2

When the machine is ready, SSH into it. Enter the following commands to install Apache and Git.

```
sudo apt-get update
sudo apt-get install -y git apache2
```

### Step 3

You will now remove Apache's default home page and replace it with a already written website.

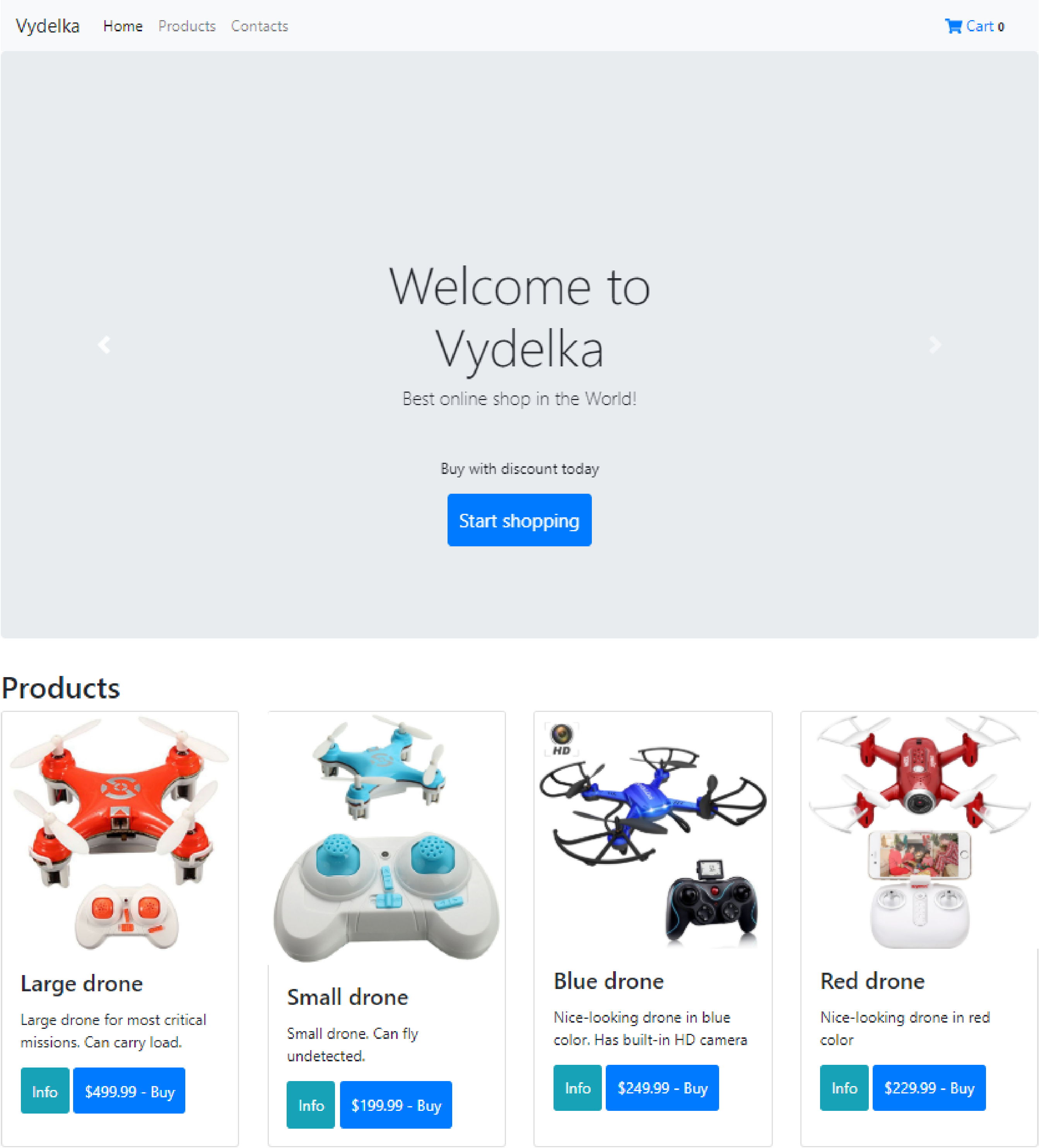
Enter the following commands to change into Apache's home folder and delete the default home page.

```
cd /var/www/html
sudo rm index.html -f
```

Enter the command below to install a website from GitHub.

```
sudo git init
sudo git pull https://github.com/programmingmentor/gcp-lb-lab.git
service apache2 restart
```

Go back to the web console and click on the external IP address of the web server and see if the website works. It should look as shown below.



Welcome to Vydelka

Best online shop in the World!

Buy with discount today

Products

Large drone

Large drone for most critical missions. Can carry load.

Small drone

Small drone. Can fly undetected.

Blue drone

Nice-looking drone in blue color. Has built-in HD camera

Red drone

Nice-looking drone in red color

**Step 4**

In the web console, select your `web-server` VM and delete it.

> **Note:** Recall that when you created this VM, you unchecked the box that said to delete the disk when the instance was deleted. So, after deleting this VM, the boot disk should remain.

**Step 5**

Click **Disks** in the navigation pane on the left. Select your `web-server` disk and click the button all the way to the right in that row. Then, select **Create** snapshot.

Name the snapshot `web-server-snapshot` and click **Create**.

**Step 6**

Click **Images** in the navigation pane on the left. Then, click the **Create image** button.

Name the image `web-server-image`. Set the family to `web-game`.

Then, select `web-server` from the **Source disk** dropdown.

Finally, click the Create button.

**Step 7**

Go back to the VM instances page and click the **Create** button. Name your instance `web-server-2`.

Click the **Change** button in the **Boot disk** section. Then, click the **Custom images** tab and select your `web-server` image. Click the Select button.

Scroll to the **Firewall** section and check **HTTP** traffic (*leave HTTPS unchecked*).

Click the **Create** button.

When the machine is ready, click the external IP address to verify that everything worked.

**Step 8**

You could also create an instance from the snapshot you created. Try creating another virtual machine, but use the snapshot as the boot disk.

Try to figure out how to create a virtual machine using your custom image, but use the CLI instead.

> **Hint:** Use the web console to fill out the form for creating a VM, but click the command line link at the bottom of the form instead of click the create button.

Then, run the command from Google Cloud Shell.

**Step 9**

You can delete any virtual machines you have running.

For the time being, leave your custom image, disk, and snapshot. You will use those later.

---

# Part 3: Global Load Balancing and Autoscaling

## Objectives

- Create an Instance Template and Health Check
- Create Instance Groups in the United States and Europe

- Set up a load balancer to balance traffic across your instance groups
- Set up autoscaling
- Send simulated load to the load balancer using ApacheBench
- Monitor traffic to your load balancer

**Step 1**

From the **Products and Services** menu, select **Compute Engine**. Then, click **Instance templates** from the navigation pane.

**Step 2**

Click the **Create instance template** button. Name your instance template `web-game-instance-template`.

From the machine type dropdown, select micro (1 shared vCPU).

Click the **Change** button in the **Boot disk** section. Click the **Custom images** tab and select your `web-server-image`. Then, click the **Select** button.

Check the Allow HTTP traffic checkbox (*leave HTTPS unchecked*).

**Step 3**

Click the **Management, disk, networking, SSH keys** link to expand the advanced options. Add the following code to the **Startup script** text box.

```
#! /bin/bash

ZONE=$(curl "http://metadata.google.internal/computeMetadata/v1/instance/zone" -H
"Metadata-Flavor: Google")

sed -i "s|zone-here|$ZONE|" /var/www/html/index.html
```

> **Note:** The startup script runs when the server is booted. For testing, we want to easily see what zones our instances are running in. In the script above, the command that begins with "ZONE=..." retrieves the zone a machine is in and stores it in the variable ZONE. The home page of our app has a hard-coded string on it, "zone-here". The sed command in the above script replaces that hard-coded string with the value of the ZONE variable.

Click the **Create** button to create the instance template.

**Step 4**

From the navigation pane, select **Instance groups**.

We will start simple and just create an instance group with three identical machines.

Click the **Create instance group** button.

Name your instance group `web-game-us`. Select the multi-zone radio button, and set the region to `us-central1`.

Select the `web-game-instance-template` you just created from the **Instance template** dropdown.

Turn **Autoscaling** off and set the number of instances to **3**.

#### Step 5

From the **Health check** dropdown, select **Create a health check**. Name the health check `web-game-health-check`. Set the Check interval to **10** and set the Unhealthy threshold to **3**. Save the health check and continue configuring the instance group.

Set the initial delay in the instance group to **120**. Then, click the **Create** button.

Wait a few seconds and then click the **VM instances** link in the navigation pane. You should see three instances get created. You might have to hit the Refresh button a couple times.

> **Note:** The three instances are all in the us-central1 region, but they should all be in different zones. Placing them in different zones, gives us extremely high availability.

#### Step 6

Click on each of the machine's' external IP addresses and verifying the website is working on each one. Notice at the bottom of the page is the zone each machine is running in. This is a result of the startup script in the instance template.

#### Step 7

Go back the **Instance groups** page. Select your instance group by checking the checkbox, and then click the **Edit** button at the top. Turn **Autoscaling On**. Set the minimum number of instances to **1** and set the maximum number of instances to **5**.

Leave everything else at its default value and click the **Save** button.

Since there is no load on the web servers, after a few minutes the instance group will scale to **1** and turn the unneeded machines off.

#### Step 8

Create another instance group. This time name it `web-game-eu`.

As before, make it a Multi-zone instance group. This time select `europe-west1` as the region though.

Choose your instance template.

Turn autoscaling On and scale between **1** and **5** instances. Select the health check you created earlier, and set the initial delay to **120**.

Finally, click **Create** to create the instance group.

If enough time has gone by, your first instance group will have scaled down.

Go to the **VM Instances** page. You should have a machine starting in Europe and one or more machines in the US. Wait for the machine in Europe to be ready and then click its external IP address to verify website is working.

**Step 9**

From the **Products and Services** menu, choose **Network Services** and then click the **Load balancing** section.

Click the **Create load balancer** button, then click the Start configuration button in the **HTTP(S) Load Balancing** section.

Name your load balancer `web-game-lb`.

Click **Backend configuration**, then from the **Create or select backend services & backend buckets** dropdown, select **Backend services | Create a backend service**.

In the **Create backend service** dialog, name the Backend service `web-game-bes`. In the **Backends** section, select one of your instance groups from the Instance group dropdown. Leave other values as the default and click **Done**. Then, click the **Add backend** button and select the other instance group and click **Done** again.

Scroll down a little and select your health check for webservers. Leave everything else for the backend service at their default values.

**Step 10**

Click Create.

You should see a checkmark next to Backend configuration.

Now, click **Host and path rules**. The defaults for this should be fine.

Click **Frontend configuration**. The defaults should be fine for this as well.

Click **Review and finalize**. At this point, you should see **3** checkmarks on the left side of the screen and your instance groups in the right part.

If you have all check marks, click the Create button to create your load balancer.

**Step 11**

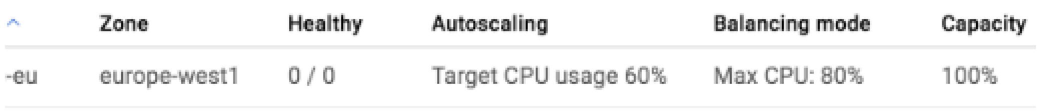
It takes a few minutes for the load balancer to be ready. It's a good time for a break. Take about **5 minutes to get a coffee or check your email.

**Step 12**

After a few minutes, your screen will show an active load balancer.

Click the load balancer to see its details.

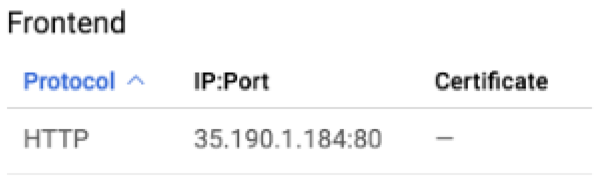
In the details section, look at the Backend services section. If there are zero healthy instances as shown in the screenshot below, then the load balancer isn't ready.



| | Zone | Healthy | Autoscaling | Balancing mode | Capacity |
|---|---|---|---|---|---|
| -eu | europe-west1 | 0 / 0 | Target CPU usage 60% | Max CPU: 80% | 100% |

When you see that there are healthy instances, then you can make requests to the load balancer.

Scroll up to the **Frontend** section of the load balancer details and you will find an IP address similar to what is shown below.



Frontend

| Protocol | IP:Port | Certificate |
|---|---|---|
| HTTP | 35.190.1.184:80 | — |

Copy and paste the IP address into a text file so you have it later on.

Open a new browser tab and make a request to that IP address. You don't need to include the port since 80 is the default when making an HTTP request.

> **Note:** When making a request to the load balancer, you should see our web game. You should also see the site come from the zone closer to you. So, if you are in or closer to the United States, you should see a `us-central1` zone and if you are closer to Europe you should see a `europe-west1` zone.

**Step 13**

Go to the Compute Engine VM instances page. Create two instances. Name the first `tester-us` and choose a zone in `us-central1`. Name the second machine `tester-eu` and choose a zone in `europe-west1`.

**Step 14**

SSH into each machine and run the following commands.

```
sudo apt-get update
sudo apt-get install -y apache2
```

> **Note:** Apache includes a testing tool called ApacheBench. It allows us to simulate load.

We will use it from each machine to make requests to the load balancer. The load balancer will send requests to the server closest to the test machine until that machine is saturated. Then, it will send requests to the other.

If we send enough requests, we should get the instance groups to autoscale and add more machines.

**Step 15**

Once the commands from the previous step complete on each machine, ApacheBench should be installed.

Let's start small. Enter the following command from each machine. It will make 50 requests, one at a time. You will need to enter your load balancer's IP address where indicated.

```
ab -n 50 -c 1 http://your-ip-address-here/
```

In the command above, **leave the slash at the end**.

**Step 16**

In the web console, go back to the **Networking services** and **Load balancing**. Click on your load balancer to see its details. Then click the **Monitoring** tab. In the Backend dropdown, select your `web-game-bes`.

The monitoring console should show some requests. The charts aren't updated immediately so be patient.

**Step 17**

Go back to the SSH window of each machine. Press the up arrow on your keyboard to repeat the previous ApacheBench command on each machine. Press Enter to repeat the command and let the command complete.

Press the up arrow again, but change the number of requests to **1000** and the number of concurrent requests to **10**. Do this on each machine.

Go back to the browser tab monitoring you load balancer. You should be seeing requests come in from both North America and Europe.

**Step 18**

Open another browser tab and go to the web console and the **Compute Engine** service and click on the **Instance groups** section.

At this point, the number of instances in both of your instance groups should be listed as one.

Run your ApacheBench command again. This time specify **100,000** requests, **1000** at a time. Do this from both machines.

Look at the load balancer's monitoring page. Be patient, the page won't update immediately. You will likely see some errors show up because we don't have enough machines to support the number of requests.

Go back to the Instance groups page and refresh it. You should see one or more of the groups autoscale.

Experiment by making some more requests. Monitor the load balancer and keep refreshing the Instance groups page. As you keep making requests, the instance groups will turn additional machines on to handle the load.

Try making a **million** requests, **5000** at a time.

As new machines are created, they show up in the monitoring page.

As load increases, the instance groups will add machines as needed.

If you look at VM instances, you will see new virtual machines appear or disappear based on how many machines the instance groups think they need.

**Step 19**

Take a 15 minute break. When you come back, refresh the instance groups page. The instance groups should eventually scale back to one when there is no more traffic.

**Step 20**

In the web console, go to the **Networking services**. Then click **Load balancing**. Click the trash can icon next to your load balancer to delete it. When prompted, **check** the box indicating you want to also delete your backend service. Confirm your request.

Go to the **Compute Engine** service and click **Instance groups**. Select all your instance groups and click the **Delete** button. **Confirm** that you want to do this.

Wait a few seconds and then go to the VM instances page. The machines created by the instance groups should be deleted automatically. Select your tester machines and delete those.

You can also delete the instance template you created in this exercise. You can also delete any **disks, snapshots, and images** you have created.

---