

Assigning a VM for Our Website

In the last lesson, we set up the basic environment for deploying our web application. Now, we'll learn how to allocate a virtual server to the project.

WE'LL COVER THE FOLLOWING ^

- VM Instance
- VM Communication

Picking up where we left off in the last lesson, we can observe that there are no resources allocated to our project:



Project info



Project name

Post Web Example

Project ID

post-web-example

Project number

612065733566



Go to project settings



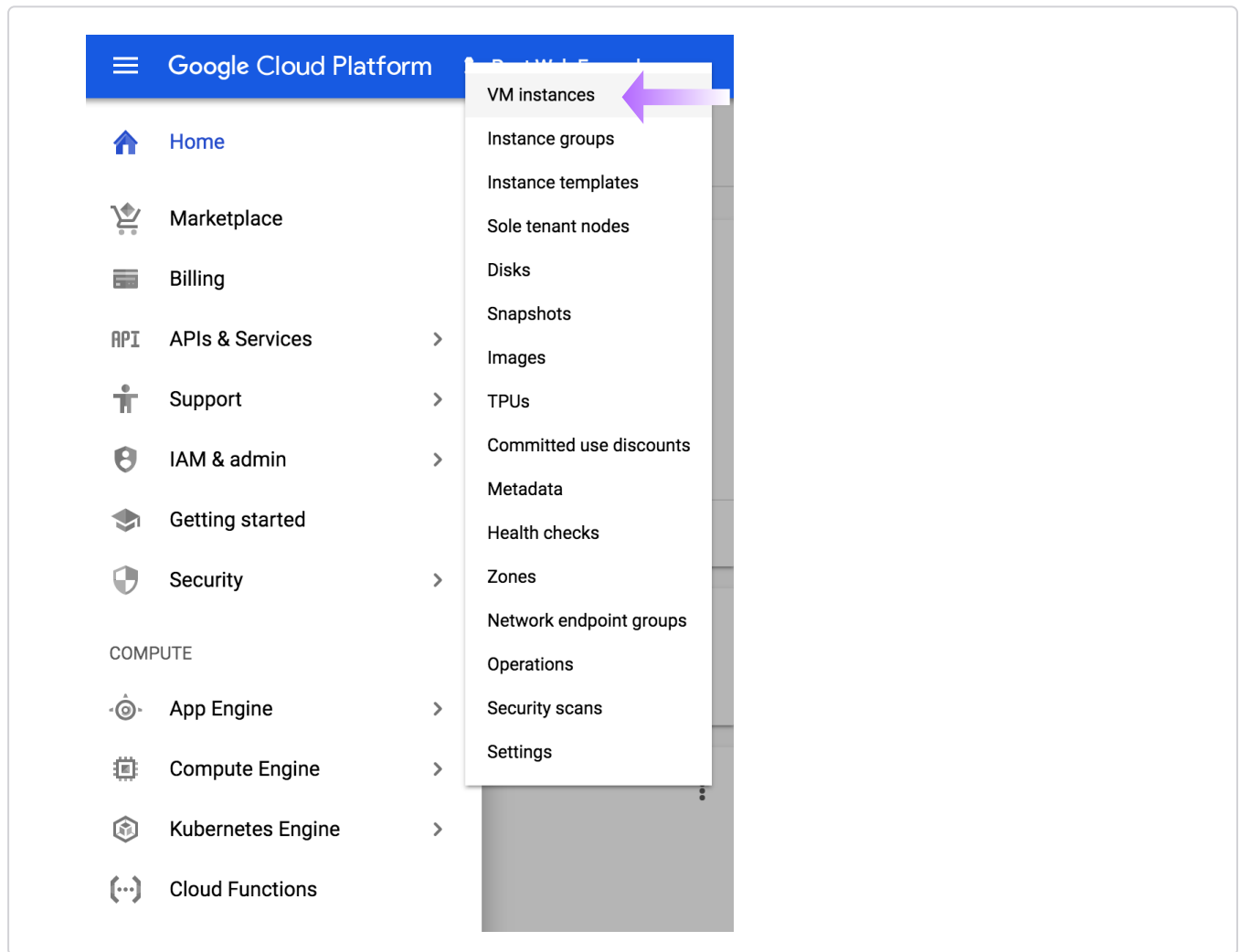
Resources



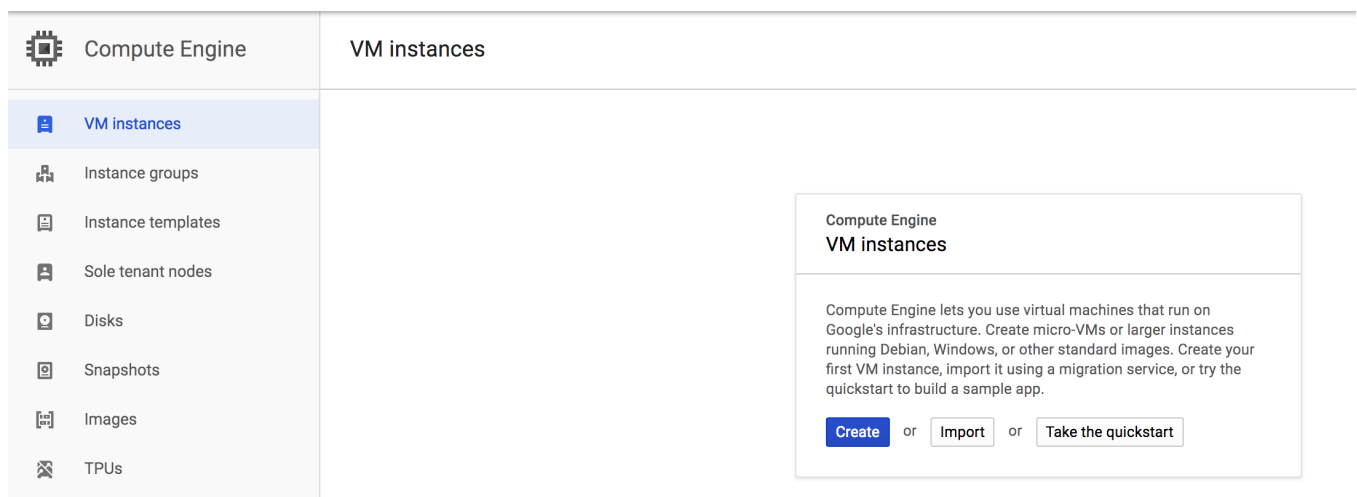
This project has no resources

VM Instance

To assign a server to our website, we need to create a **VM instance**. Go the navigation bar at the top left corner and scroll down to the **COMPUTE** section. Here, move on to **Compute Engine** and select **VM instances**:



After initialization, refreshing the page will direct us to this window:



The **Create** button will open up the **Create an instance** page where we can specify the properties of our virtual server:

The screenshot shows the Google Cloud VM creation interface. Key sections include:

- Name:** A text input field containing "instance-1".
- Region:** A dropdown menu set to "us-east1 (South Carolina)".
- Zone:** A dropdown menu set to "us-east1-b".
- Machine type:** A section with a dropdown set to "1 vCPU" and "3.75 GB memory". A "Customize" link is present. A callout bubble points to this section with the text "Specify the RAM and CPUs".
- Container:** A checkbox labeled "Deploy a container image to this VM instance." which is unchecked. A "Learn more" link is next to it.
- Boot disk:** A section showing a "New 10 GB standard persistent disk" with the image "Debian GNU/Linux 9 (stretch)". A "Change" button is at the bottom right. A callout bubble points to this section with the text "Specify the server OS and hard disk".
- Identity and API access:** A section with a "Service account" dropdown set to "Compute Engine default service account" and "Access scopes" radio buttons, with "Allow default access" selected.
- Firewall:** A section with checkboxes for "Allow HTTP traffic" and "Allow HTTPS traffic", both of which are unchecked.

On the right side of the interface, there is a summary of costs:

- Total cost for our machine:** A purple callout bubble points to this text.
- You have:** \$300.00 free trial credits remaining
- \$24.67 monthly estimate**
- That's about \$0.034 hourly**
- Pay for what you use:** No upfront costs and per second billing
- Details:** A link to expand more information.

The **Region** and **Zone** refer to the location of our VM. It is a wise practice to select a region which is close to our primary customer base so that they can access our website in the fastest way possible.

To avoid server failure, we should acquire machines in multiple zones. This will allow us to shift our application to another server in case the original one fails.

Machine Type and **Boot disk** can be used to specify the power of our server. Higher specs would lead to higher costs.

Container lets us set up an OS container on our VM.

Identify and API access allows us to specify the degree of freedom we want while working with other Google Cloud APIs. We will need a Service Account for this task. Visit [this documentation](#) for more details.

For now, we won't concern ourselves with the rest of the features. We can explore them later.

Let's hit the **Create!** Congratulations, a VM has been allocated to our project:

Name	Zone	Recommendation	Internal IP	External IP	Connect
instance-1	us-east1-b		10.142.0.2 (nic0)	35.196.64.74	SSH

VM Communication

The server is now running and we can transfer our website files to it. Click the **SSH** button under the **Connect** section in the image above. This will open a new tab where an SSH connection is set up on our virtual machine.

Install Git on your VM using:

```
sudo apt install git
```

Now, we can clone our Git repository to the current directory. Sometimes, we won't have default permission to run certain apps such as a React app (Node installation requires root access). For this, we can gain root access through,

```
sudo su
```

and simply copy our files there.

We'll cover repository cloning in the next lesson so don't worry about it for now. After that, all we have to do is deploy our website to the internet so that everyone can access it. Go to the next lesson for more details.

