

GCP Certification Series, 1.1



Prashanta Paudel

Oct 12, 2018 · 13 min read

From this series, I will be trying to go through sections in the GCP Certification and document the steps to carry out the tasks mentioned under the subheading.

Google has mentioned Associate Cloud Engineer as the one who could perform the following tasks

Associate Cloud Engineer

An Associate Cloud Engineer deploys applications, monitors operations, and manages enterprise solutions. This individual is able to use the Google Cloud Console and the command-line interface to perform common platform-based tasks to maintain one or more deployed solutions that leverage Google-managed or self-managed services on Google Cloud.

The Associate Cloud Engineer exam assesses your ability to:

- Set up a cloud solution environment
- Plan and configure a cloud solution
- Deploy and implement a cloud solution
- Ensure the successful operation of a cloud solution
- Configure access and security

The whole Certification is divided into 5 sections and other subsections. My plan is to go through each subsection so that learning will be easy and piece by piece.

So, Today we will go through section 1.1 of section 1.

Before going directly into section 1.1 we should have information on GCP resource hierarchy. The purpose of this is to bind the resources

with the owner and maintain inheritance of ownership as well as provide access control and policy for the resources.

Generally, we can compare the resource hierarchy with the file system in the traditional OS. This hierarchical organization of resources enables you to set access control policies and configuration settings on a parent resource, and the policies and IAM settings are inherited by the child resources.

Google says :

At the lowest level, resources are the fundamental components that make up all GCP services. Examples of resources include Compute Engine Virtual Machines (VMs), Cloud Pub/Sub topics, Cloud Storage buckets, App Engine instances. All these lower level resources can only be parented by projects, which represent the first grouping mechanism of the GCP resource hierarchy.

G Suite and Cloud Identity customers have access to additional features of the GCP resource hierarchy that provide benefits such as centralized visibility and control, and further grouping mechanisms, such as folders. We have launched the Cloud Identity management tool. For details on how to use Cloud Identity, see [Migrating to Cloud Identity](#).

GCP resources are organized hierarchically. Starting from the bottom of the hierarchy, projects are the first level, and they contain other resources. All resources must belong to exactly one project.

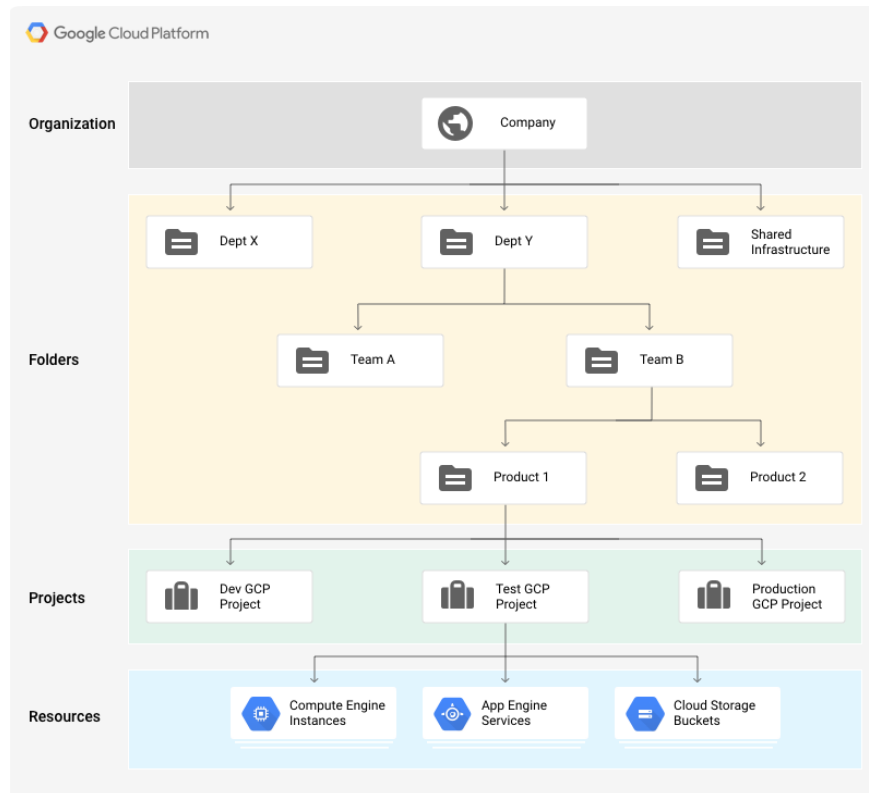
The Organization resource is the root node of the GCP resource hierarchy and all resources that belong to an organization are grouped under the organization node. This provides central visibility and control over every resource that belongs to an organization.

Folders are an additional grouping mechanism on top of projects. You are required to have an Organization resource as a prerequisite to using folders. Folders and projects are all mapped under the Organization resource.

The GCP resource hierarchy, especially in its most complete form which includes an Organization resource and folders, allows companies to map their organization onto GCP and provides logical attach points for access management policies (Cloud Identity and Access Management)

and Organization policies. Both Cloud IAM and Organization policies are inherited through the hierarchy, and the effective policy at each node of the hierarchy is the result of policies directly applied at the node and policies inherited from its ancestors.

The diagram below represents an example GCP resource hierarchy in complete form:



Section 1: Setting up a cloud solution environment

1.1 Setting up cloud projects and accounts. Activities include:

- Creating projects.
- Assigning users to pre-defined IAM roles within a project.
- Linking users to G Suite identities.
- Enabling APIs within projects.
- Provisioning one or more Stackdriver accounts.

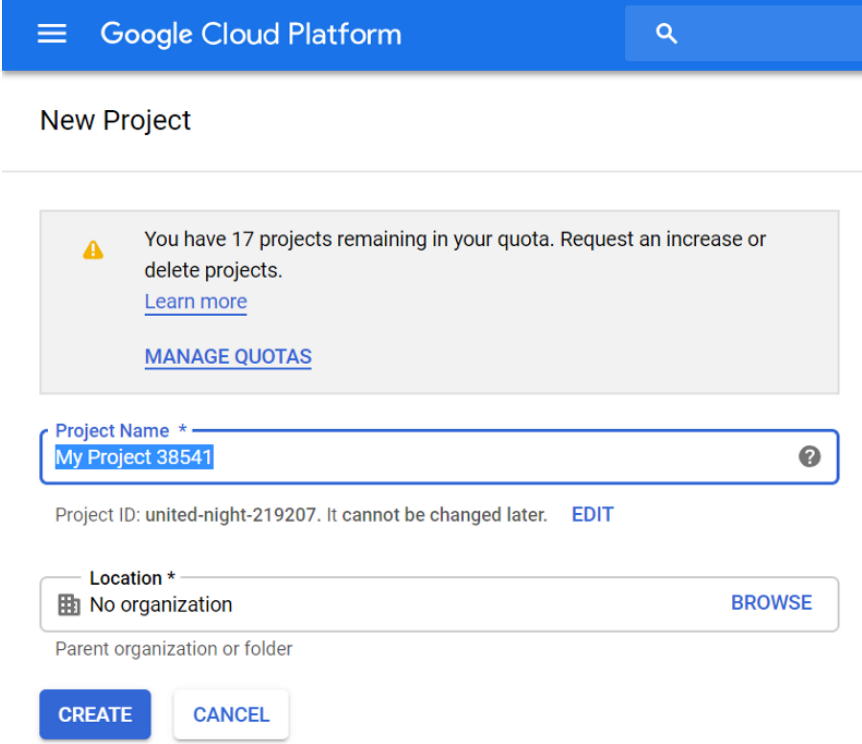
Creating the project

The most basic thing to do in GCP is creating the project. Creating the project in itself doesn't complete anything but it will start the process so that you can add entities in the project and build your own network, create a database, write code, build server etc.

The project number and project ID are unique across the Google Cloud Platform. If another user owns a project ID for their project, you won't be able to use the same project ID. Also, the name such as Google or SSL cannot be used for the project name.

All the instances are attached to the project. So, you got the idea everything is inside the project.

To create the project, first, you need to have access to the cloud console.



The screenshot shows the 'New Project' page in the Google Cloud Platform console. At the top is a blue header with the 'Google Cloud Platform' logo and a search icon. Below the header, the title 'New Project' is displayed. A warning box indicates that the user has 17 projects remaining in their quota and provides links to 'Learn more' and 'MANAGE QUOTAS'. The 'Project Name' field is required and contains the text 'My Project 38541'. Below this, the 'Project ID' is shown as 'united-night-219207', with a note that it cannot be changed later and an 'EDIT' link. The 'Location' field is also required and currently shows 'No organization' with a 'BROWSE' button. At the bottom, there are 'CREATE' and 'CANCEL' buttons, and a 'create project' link.

Google Cloud Platform

New Project

You have 17 projects remaining in your quota. Request an increase or delete projects.
[Learn more](#)
[MANAGE QUOTAS](#)

Project Name *
My Project 38541

Project ID: united-night-219207. It cannot be changed later. [EDIT](#)

Location *
No organization [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

[create project](#)

If you are running a free account you should select “individual” while creating the account itself. Free users get limited quota so if the number of projects that can be created is less than 30 you will get the message as shown above.

The project name is a human-readable name for simplicity while all the processing will be done using the project ID mentioned just below the project name.

You may be working in the console for more than one project so, it is always a good idea to check the project name while performing tasks.

To create a new project, use the `gcloud projects create` command:

```
gcloud projects create PROJECT_ID
```

Eg

```
prashantagcppaudel@cloudshell:~ (webproject-217416)$ gcloud
projects create testdemotrial123
Create in progress for
[https://cloudresourcemanager.googleapis.com/v1/projects/tes
tdemotrial123].
Waiting for [operations/cp.6134727994789518289] to
finish...done.
```

Where PROJECT_ID is the ID for the project you want to create. A project ID must start with a lowercase letter, and can contain only ASCII letters, digits, and hyphens, and must be between 6 and 30 characters.

To create a project with an organization(not for a free account)or a folder as a parent, use

```
--organization or --folder flags.
```

As a resource can only have one parent, only one of these flags can be used:

```
gcloud projects create PROJECT_ID --organization=ORGANIZATION_ID
```

```
gcloud projects create PROJECT_ID --folder=FOLDER_ID
```

To check the metadata of the project see the Dashboard of the project or use

```
# gcloud projects describe PROJECT_ID
```

```
root@cs-6000-devshell-vm-4663cdf8-e36b-46af-a81b-16d2c81e115c:/home/prashantagcppaudel# gcloud projects describe testdemotrial123
createTime: '2018-10-12T07:50:46.560Z'
lifecycleState: ACTIVE
name: testdemotrial123
projectId: testdemotrial123
projectNumber: '376521124726'
```

Assigning users to pre-defined IAM roles within a project

When a new project is created, the account used in GCP will automatically get the owner access. There are few standard user access types or say roles.

1. Browser
2. Editor
3. Owner
4. Viewer

After the project has been created, click on the project and then IAM and Admin to get to the Identity and access management page.

Type	Member	Name	Role	Inheritance
<input type="checkbox"/>	757213253994:compute@developer.gserviceaccount.com	Compute Engine default service account	Editor	<input checked="" type="checkbox"/>
<input type="checkbox"/>	757213253994:cloudservices.gserviceaccount.com	Google APIs Service Agent	Editor	<input checked="" type="checkbox"/>
<input type="checkbox"/>	prashantagcppaudel@gmail.com	Prashanta Paudel	Owner	<input checked="" type="checkbox"/>
<input type="checkbox"/>	service-757213253994@compute-system.iam.gserviceaccount.com	Compute Engine Service Agent	Compute Engine Service Agent	<input checked="" type="checkbox"/>
<input type="checkbox"/>	service-757213253994@container-engine-robot.iam.gserviceaccount.com	Kubernetes Engine Service Agent	Kubernetes Engine Service Agent	<input checked="" type="checkbox"/>
<input type="checkbox"/>	service-757213253994@containerregistry.iam.gserviceaccount.com	Google Container Registry Service Agent	Editor	<input checked="" type="checkbox"/>

when you click ADD, you will get the option to add users to the project. Here you will see various options to include access for projects and billings. Select the project and then access type.

Add members, roles to "Webproject" project

Enter one or more members below. Then select a role for these members to grant them access to your resources. Multiple roles allowed. [Learn more](#)

New members

?

Must select at least one member to add

Type to filter

Project

Android Manageme...

App Engine

AutoML

BigQuery

Billing

Binary Authorization

Cloud Asset

Browser

Editor

Owner

Viewer

MANAGE ROLES

Google says:

With Cloud IAM, every Google Cloud Platform method requires that the account making the API request has appropriate permissions to access the resource. Permissions allow users to perform specific actions on Cloud resources. For example, the `resourcemanager.projects.list` permission allows a user to list the projects they own, while `resourcemanager.projects.delete` allows a user to delete a project.

The following table lists the permissions that the caller must have to call a projects API:

Method	Required Permission(s)
<code>resourceManager.projects.create()</code>	<code>resourceManager.projects.create</code>
<code>resourceManager.projects.delete()</code>	<code>resourceManager.projects.delete</code>
<code>resourceManager.projects.get()</code>	<code>resourceManager.projects.get</code>
<code>resourceManager.projects.getIamPolicy()</code>	<code>resourceManager.projects.getIamPolicy</code>
<code>resourceManager.projects.list()</code>	Does not require any permission. The method lists projects for which the caller has <code>resourceManager.projects.get</code> permission. If you provide a filter while calling <code>list()</code> , for example, <code>byParent</code> , the method lists projects for which you have the <code>resourceManager.projects.get</code> permission and which satisfies the filter condition.
<code>resourceManager.projects.setIamPolicy()</code>	<code>resourceManager.projects.setIamPolicy</code>
<code>resourceManager.projects.testIamPermissions()</code>	Does not require any permission.
<code>resourceManager.projects undelete()</code>	<code>resourceManager.projects.undelete</code>
<code>resourceManager.projects.update()</code>	To update a project's metadata, requires <code>resourceManager.projects.update</code> permission. To update a project's parent and move the project into an organization, requires <code>resourceManager.projects.create</code> permission on the organization.

You don't directly give users permissions; instead, you grant them *roles*, which have one or more permissions bundled within them.

You can grant one or more roles on the same project. When using the `resourceManager.projects.getIamPolicy()` method to view permissions, only the permissions assigned to the project itself will appear, not any inherited permissions.

Using Predefined Roles

The following table lists the roles that you can grant to access a project, the description of what the role does, and the permissions bundled within that role.

Role	Description	Permissions
roles/owner	Full access to all resources.	All permissions for all resources.
roles/editor	Edit access to all resources.	Create and update access for all resources.
roles/viewer	Read access to all resources.	Get and list access for all resources.
roles/browser ^{Beta}	Access to browse resources in the project.	<ul style="list-style-type: none"> resourcemanager.organizations.get resourcemanager.projects.get resourcemanager.projects.getIamPolicy resourcemanager.projects.list resourcemanager.projectInvites.get

Protection Against Accidental deletion

To protect any owners/administrator accidentally delete the project GCP has a feature called liens. You can use liens in the project to block projects deletion until it is revoked back. The easiest approach to using liens is the gcloud shell

Putting liens

To place a lien on a project, a user must have the

`resourcemanager.projects.updateLiens` permission which is granted by the `roles/owner` and `roles/resourcemanager.lienModifier` roles.

```
gcloud alpha resource-manager liens create \
  --restrictions=resourcemanager.projects.delete \
  --reason="Super important production system"
```

The available parameters to `liens create` are:

- `--project` - The project the lien applies to.
- `--restrictions` - A comma-separated list of IAM permissions to block.
- `--reason` - A human-readable description of why this lien exists.
- `--origin` - A short string denoting the user/system which originated the lien. Required, but the gcloud tool will automatically populate it with the user's email address if left out.

At present, the only valid restriction for a project is

```
resourcemanager.projects.delete .
```

Listing liens on a project

To list liens applied to a project, a user must have the

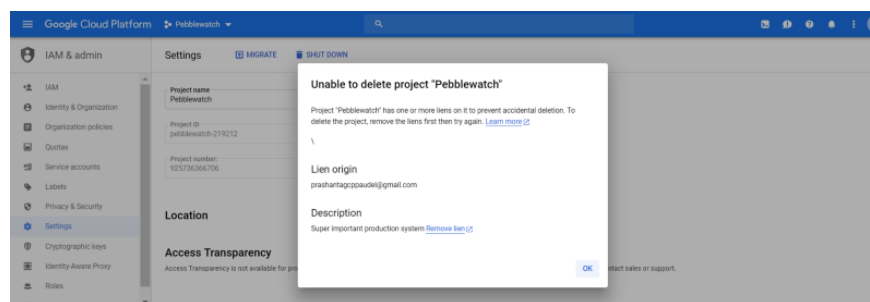
`resourcemanager.projects.get` permission. Use the `liens list` `gcloud` command.

```
gcloud alpha resource-manager liens list
```

Here is some example output for this command:

```
gcloud alpha resource-manager liens list
NAME                                     ORIGIN
REASON
p1061081023732-l3d8032b3-ea2c-4683-ad48-5ca23ddd00e7
user@example.com testing
```

If you try to delete the project then you will be presented with an error message



Removing liens from a project

To remove a lien from a project, a user must have the

`resourcemanager.projects.updateLiens` permission which is granted by `roles/owner` and `roles/resourcemanager.lienModifier` .

```
gcloud alpha resource-manager liens delete [LIEN_NAME]

gcloud alpha resource-manager liens delete p925736366706-
lb2d80913-a41b-47a4-b4af-799489f09f96
Deleted [ liens/p925736366706-lb2d80913-a41b-47a4-b4af-
799489f09f96].
```

where:

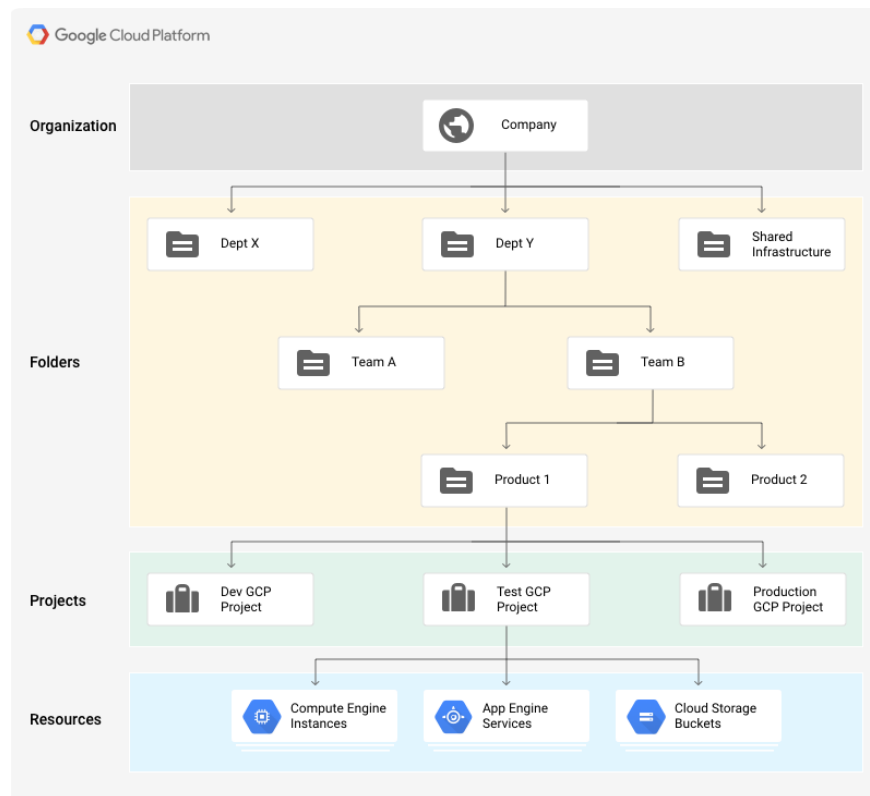
- [LIEN_NAME] is the name of the lien to be deleted.

IAM policy inheritance

Google Cloud Platform offers Cloud Identity and Access Management (IAM), which lets you assign granular access to specific Google Cloud Platform resources and prevents unwanted access to other resources. IAM lets you control who (**users**) has what access (**roles**) to which **resources** by setting IAM policies on the resources.

You can set an IAM policy at the organization level, the folder level, the project level, or (in some cases) the resource level. Resources inherit the policies of the parent node. If you set a policy at the Organization level, it is inherited by all its child folders and projects, and if you set a policy at the project level, it is inherited by all its child resources.

The effective policy for a resource is the union of the policy set on the resource and the policy inherited from its ancestors. This inheritance is transitive. In other words, resources inherit policies from the project, which inherit policies from the organization. Therefore, the organization-level policies also apply at the resource level.



For example, in the resource hierarchy diagram above, if you set a policy on folder “Dept Y” that grants Project Editor role to bob@example.com, then Bob will have editor role on projects “Dev GCP,” “Test GCP,” and “Production.” Conversely, if you assign alice@example.com the Instance Admin role on project “Test GCP”, she will only be able to manage Compute Engine instances in that project.

The IAM policy hierarchy follows the same path as the GCP resource hierarchy. If you change the resource hierarchy, the policy hierarchy changes as well. For example, moving a project into an organization will update the project’s IAM policy to inherit from the organization’s IAM policy. Similarly, moving a project from one folder to another will change the inherited permissions. Permissions that were inherited by the project from the original parent will be lost when the project is moved to a new folder. Permissions set at the destination folder will be inherited by the project as it is moved.

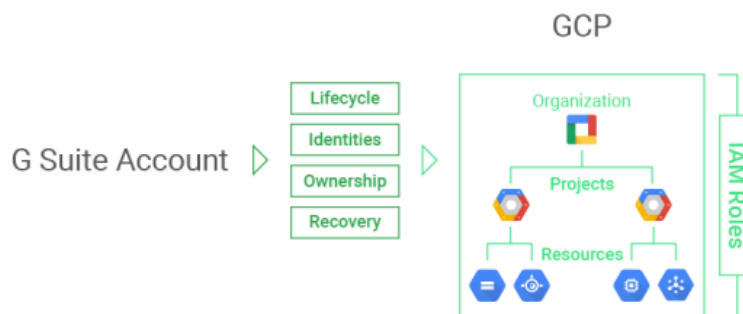
Linking users to G Suite identities

If a new user wants to have access to the resources he should already possess G-Suite or Cloud identity account in order to access those

resources. G-Suite generally represents a company and is prerequisite to access organization resources.

Google says:

The G Suite or Cloud Identity account represents a company and is a prerequisite to have access to the Organization resource. In the GCP context, it provides identity management, recovery mechanism, ownership and lifecycle management. The picture below shows the link between the G Suite account, Cloud Identity, and the GCP resource hierarchy.

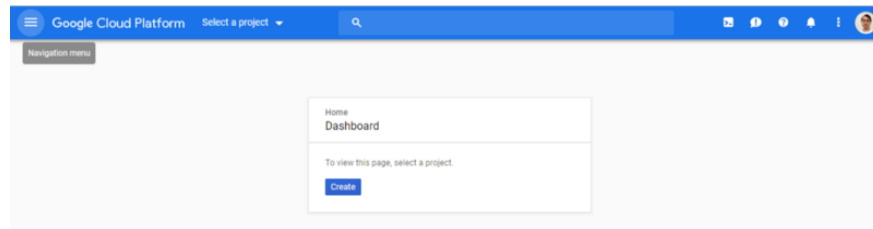


The G Suite super admin is the individual responsible for domain ownership verification and the contact in cases of recovery. For this reason, the G Suite super admin is granted the ability to assign Cloud IAM roles by default. The G Suite super admin's main duty with respect to GCP is to assign the Organization Administrator IAM role to appropriate users in their domain. This will create the separation between G Suite and GCP administration responsibilities that users typically seek.

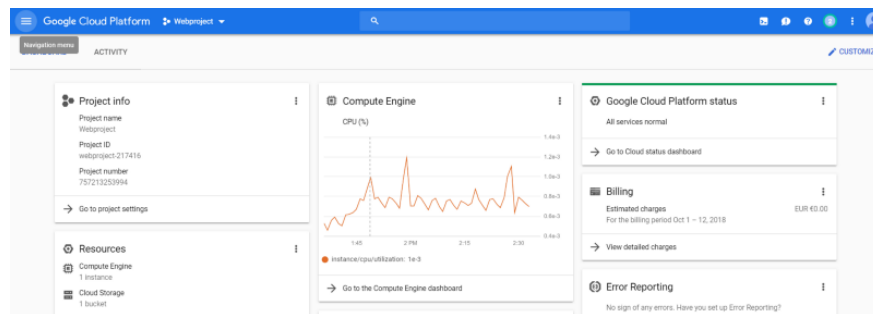
GCP users are not required to have an Organization resource. A user acquires an Organization resource only if they are also G Suite or Cloud Identity customers. The Organization resource is closely associated with a G Suite or Cloud Identity account. Each G Suite or Cloud Identity account may have exactly one Organization provisioned with it. Once an Organization resource is created for a domain, all GCP projects created by members of the account domain will by default belong to the Organization resource.

Eg

This user has not been associated with any project or G-Suit.



Now I am going to give access to the project from another user as shown below. This project has one compute engine and 1 cloud storage.



Add user as shown previously

Add members to "Webproject"

Add members, roles to "Webproject" project

Enter one or more members below. Then select a role for these members to grant them access to your resources. Multiple roles allowed. [Learn more](#)

New members

@gmail.com

Role

Owner

Full access to all resources.

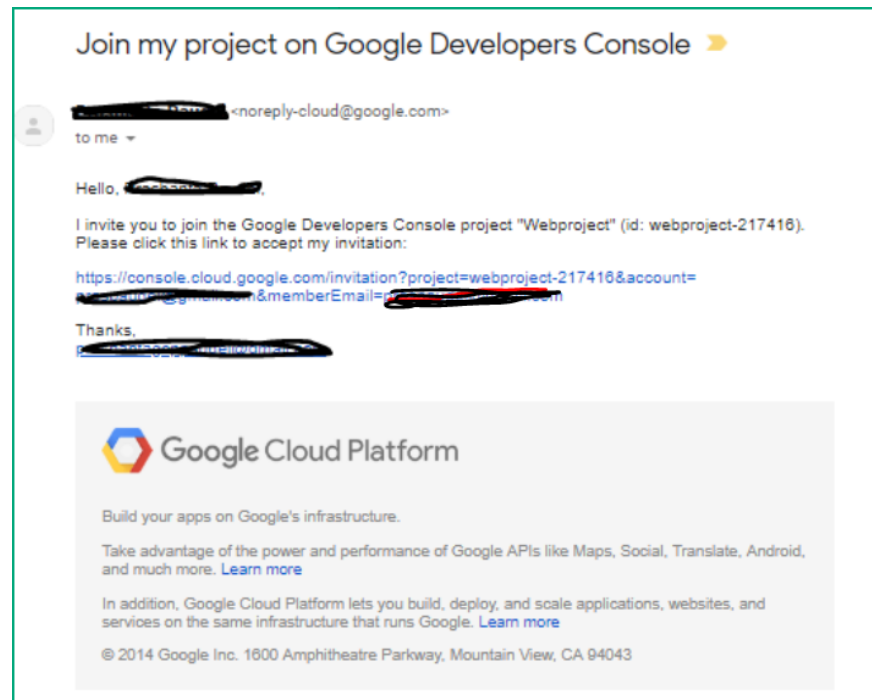
+ ADD ANOTHER ROLE

SAVE

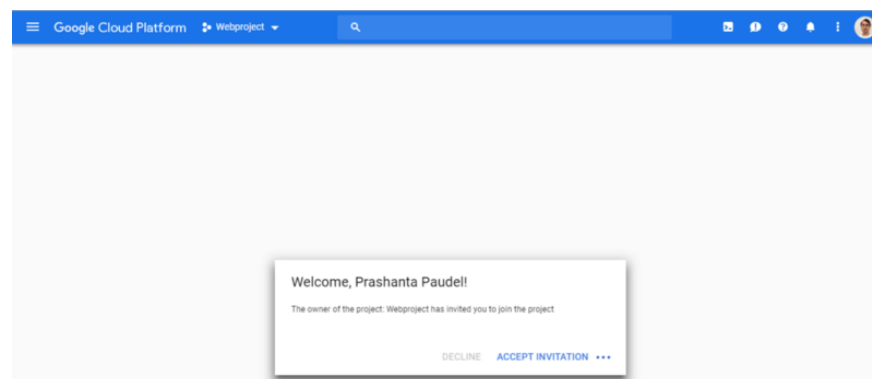
CANCEL

Here I have given access to the project as a whole so it should be accessible from the first user.

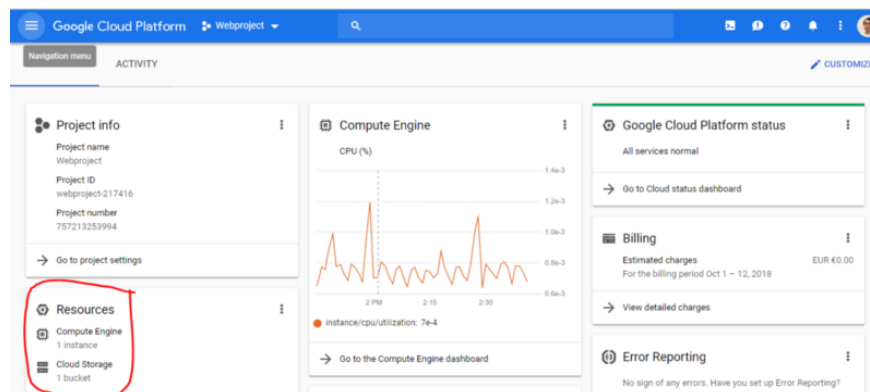
The invited user will get an email asking for accepting invite as shown below



when the link is clicked you will be redirected to the console and the project will be displayed as shown below.



Also, notice that the instances under the project are also listed in the resources available to the user.

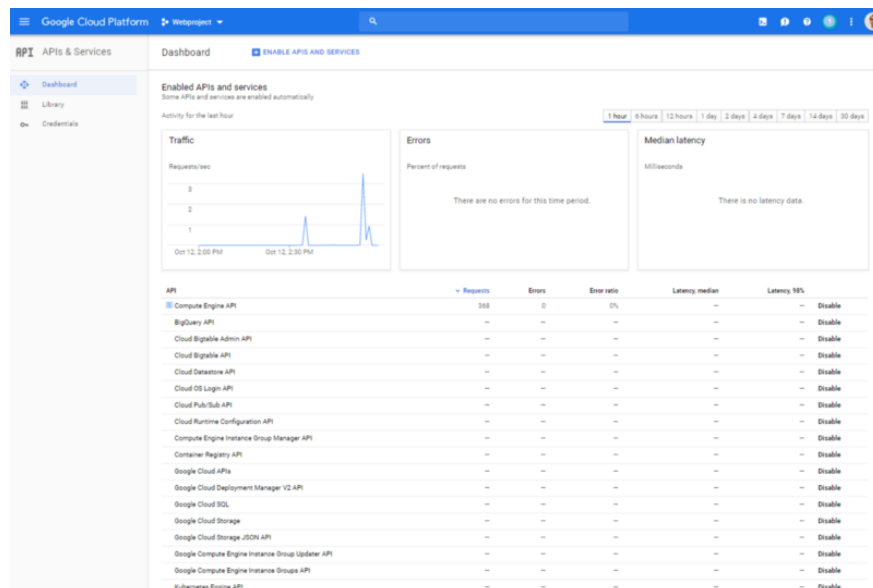
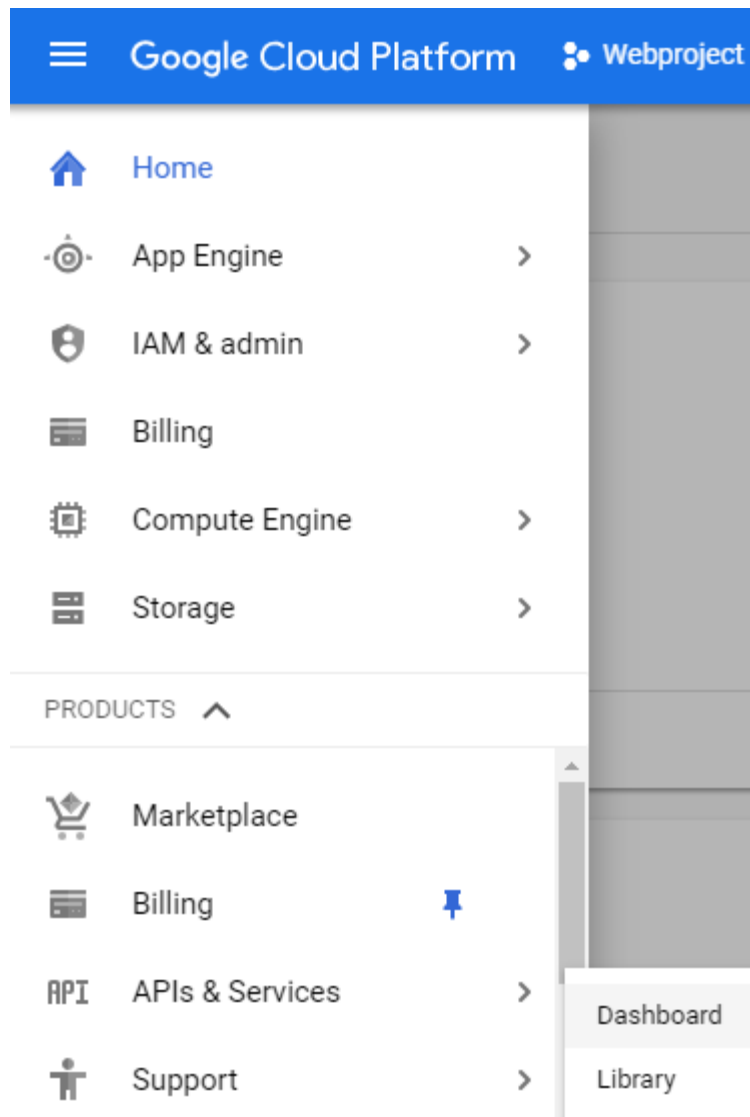


The second user can create, delete and modify instances if he has proper access.

If accidentally you remove yourself from the project and there are no other users in the project, then you are locked out of the project.

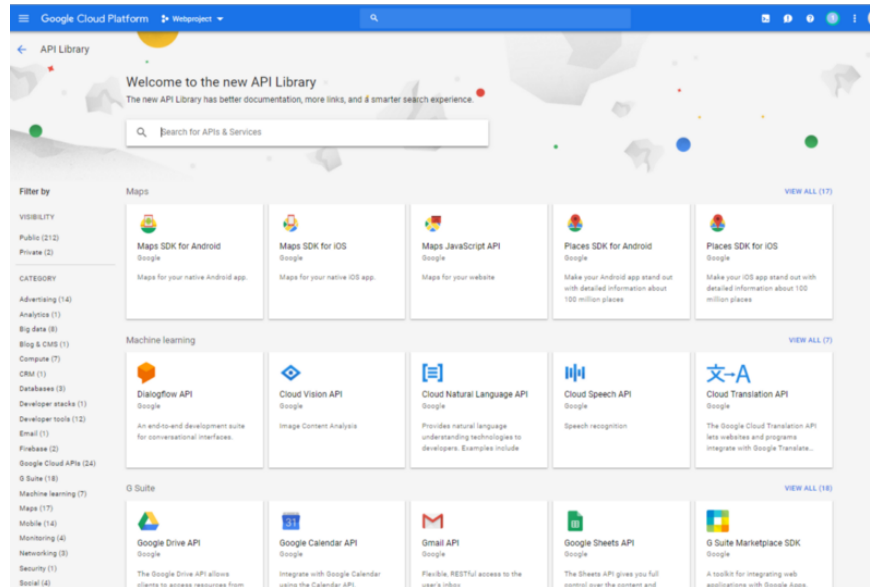
Enabling APIs within projects.

All the API's are accessible from console page > API & Services page. after clicking API & Services you will be presented with the API Dashboard.

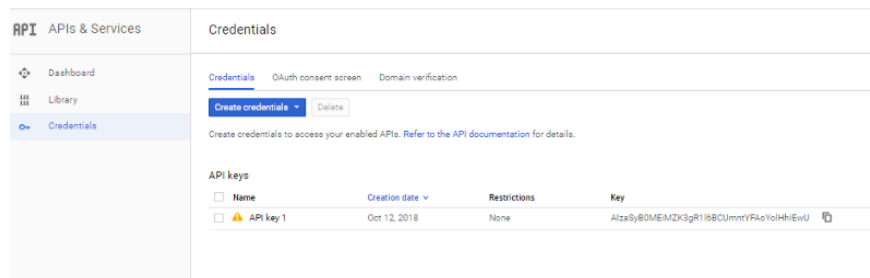


To Enable new API click on Disable button at rightmost of each row.

You can also select from the library of API's listed under the Library section.



Another important part is credentials, to use any API you need to have valid credentials. To find the keys for authenticating uses of API you can add and remove credentials from the list.



Google says:

Restoring a project

Project owners can restore a deleted project within the 30-day recovery period that starts when the project is shut down. Restoring a project returns it to the state it was in before it was shut down. Cloud Storage resources are deleted before the 30-day period ends, and may not be fully recoverable.

Some services might need to be restarted manually. For more information, see [Restarting Google Cloud Platform Services](#).

To restore a project:

1. Go to the **Manage Resources** page in the Google Cloud Platform Console.
2. [GO TO THE MANAGE RESOURCES PAGE](#)
3. In the **Organization** drop-down in the upper left, select your organization.
4. Below the list of projects, click **Resources pending deletion**.
5. Check the box for the project you want to restore, then click **Restore**. In the dialog that appears, confirm that you want to restore the project.

Provisioning one or more Stackdriver accounts

First, let's check what is stack driver

This embedded content is from a site that does not comply with the Do Not Track (DNT) setting now enabled on your browser.

Please note, if you click through and view it anyway, you may be tracked by the website hosting the embed.

[Learn More about Medium's DNT policy](#)

This embedded content is from a site that does not comply with the Do Not Track (DNT) setting now enabled on your browser.

Please note, if you click through and view it anyway, you may be tracked by the website hosting the embed.

[Learn More about Medium's DNT policy](#)

Install the Stackdriver Agents recommended

Get the most out of your free Workspace by installing the Stackdriver Monitoring and Logging agents on each of your VM instances. Agents collect more information from your VM instances, including metrics and logs from third-party applications:

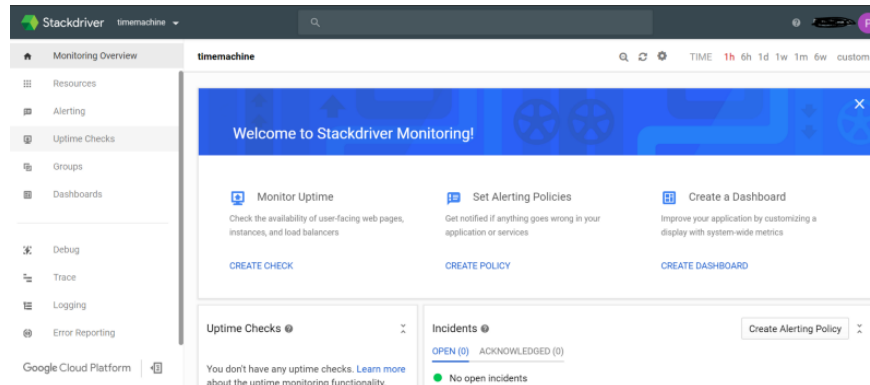
1. Switch to the terminal connected to your VM instance, or create a new one.
2. Install the Stackdriver agents by running the following commands on your instance:

```
# To install the Stackdriver monitoring agent:
$ curl -sSO https://dl.google.com/cloudagents/install-monitoring-agent.sh
$ sudo bash install-monitoring-agent.sh

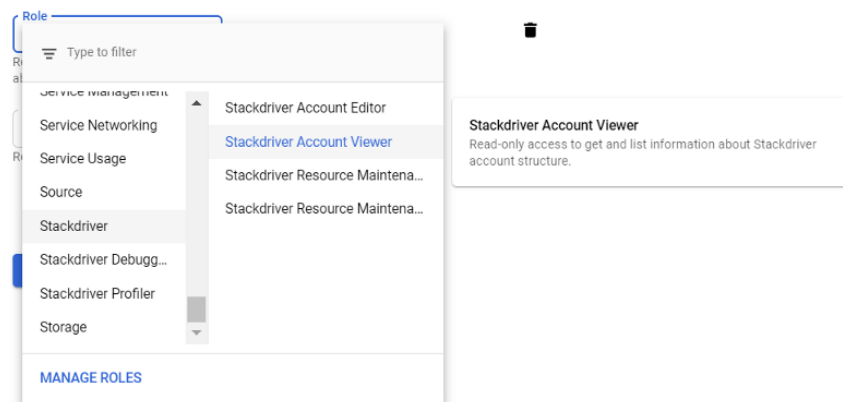
# To install the Stackdriver logging agent:
$ curl -sSO https://dl.google.com/cloudagents/install-logging-agent.sh
$ sudo bash install-logging-agent.sh
```

I install one VM and installed stack driver monitoring and agent in it as shown in the command above.

Now go to GCP console and then to stackdriver>monitoring. You will be presented with welcome page



Now we need to add users for monitoring. Adding the users with various roles is pretty forward for stack driver. We don't have to validate separately different users but add in IAM & services as adding the new user to the project but have to select Stackdriver and Stackdriver roles and save the configuration.



After adding the user, the access given will allow the user to view/edit/delete/ monitoring new services/apps.

So, we finished 1.1 of the GCP Certification.

NOTE: The projects shown in the article are used only for a demonstration purpose. All the projects and users are canceled and non-existing.

