Access Control for Projects using IAM

Google Cloud Platform offers Identity and Access Management (IAM), which lets you give more granular access to specific Google Cloud Platform resources and prevents unwanted access to other resources. IAM lets you adopt the security principle of least privilege (https://en.wikipedia.org/wiki/Principle_of_least_privilege), so you grant only the necessary access to your resources.

IAM lets you control **who (users)** has **what access (roles)** to **which resources** by setting IAM policies. IAM policies grant specific role(s) to a user giving the user certain permissions.

This page explains the Identity and Access Management (IAM) (https://cloud.google.com/iam/) roles that are available at the project level. For a detailed description of Cloud IAM, read the IAM documentation (https://cloud.google.com/iam/docs/overview). In particular, see Granting, Changing, and Revoking Access (https://cloud.google.com/iam/docs/granting-changing-revoking-access).

Permissions and roles

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)
`resourcemanager.projects.create()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/create)	`resourcemanager.projects.create`
`resourcemanager.projects.delete()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/delete)	`resourcemanager.projects.delete`

`resourcemanager.projects.get()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/get)	`resourcemanager.projects.get`
`resourcemanager.projects.getIamPolicy()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/getIamPolicy)	`resourcemanager.projects.getIamPolicy`
`resourcemanager.projects.list()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/list)	Does not require any permission. The method lists projects for which the caller has `resourcemanager.projects.get` permission. If you provide a filter while calling `list()`, for example, `byParent`, the method lists projects for which you have the `resourcemanager.projects.get` permission and which satisfies the filter condition.
`resourcemanager.projects.setIamPolicy()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/setIamPolicy)	`resourcemanager.projects.setIamPolicy`
`resourcemanager.projects.testIamPermissions()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/testIamPermissions)	`Does not require any permission.`
`resourcemanager.projects.undelete()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/undelete)	`resourcemanager.projects.undelete`
`resourcemanager.projects.update()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/update)	To update a project's metadata, requires `resourcemanager.projects.update` permission. To update a project's parent and move the project into an organization, requires `resourcemanager.projects.create`

	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 primitive roles

The following table lists the primitive roles that you can grant to access a project, the description of what the role does, and the permissions bundled within that role. Avoid using primitive roles except when absolutely necessary. These roles are very powerful, and include a large number of permissions across all Google Cloud services. For more details on when you should use primitive roles, see the Cloud Identity and Access Management FAQ (https://cloud.google.com/iam/docs/faq#when_would_i_use_primitive_roles).

Cloud IAM predefined roles are much more granular, and allow you to carefully manage the set of permissions that your users have access to. See Understanding Roles (https://cloud.google.com/iam/docs/understanding-roles) for a list of roles that can be granted at the project level. Creating custom roles (https://cloud.google.com/iam/docs/understanding-custom-roles) can further increase the control you have over user permissions.

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.	• `resourcemanager.organizations.get` • `resourcemanager.projects.get` • `resourcemanager.projects.getIamPolicy` • `resourcemanager.projects.list` • `resourcemanager.projectInvites.get`

Creating Custom Roles

In addition to the predefined roles described in this topic, you can also create Custom Roles (https://cloud.google.com/iam/docs/understanding-custom-roles) that are collections of permissions that you tailor to your needs. When creating a Custom Role for use with Resource Manager, be aware of the following points:

- List and get permissions, such as `resourcemanager.projects.get/list`, should always be granted as a pair.
- When your Custom Role includes the `folders.list` and `folders.get` permissions, it should also include `projects.list` and `projects.get`.
- Be aware that the `setIamPolicy` permission for organizations, folders, and projects allows the user to grant all other permissions, and so should be assigned with care.

Access control at the project level

You can grant roles to users at the project level using the Google Cloud Console (https://console.cloud.google.com/), the Resource Manager API, and the `gcloud` command-line tool. For instructions, see Granting, Changing, and Revoking Access to Project Members (https://cloud.google.com/iam/docs/granting-changing-revoking-access).

Default roles

When you create a project, you are granted the **roles/owner** role for the project to provide you full control as the creator. See above for the permissions that this role provides. This default role can be changed as normal in a Cloud IAM policy.

VPC Service Controls

VPC Service Controls can provide additional security when using the Resource Manager API. To learn more about VPC Service Controls, see the VPC Service Controls overview (https://cloud.google.com/vpc-service-controls/docs/overview).

To learn about the current limitations in using Resource Manager with VPC Service Controls, see the supported products and limitations

(https://cloud.google.com/vpc-service-controls/docs/supported-products) page.

Last updated January 13, 2020.