In this module, we cover Cloud Identity and Access Management (or Cloud IAM).

Cloud IAM is a sophisticated system built on top of email-like address names, job-type roles, and granular permissions. If you're familiar with IAM from other implementations, look for the differences that Google has implemented to make IAM easier to administer and more secure.

Agenda

Cloud Identity and Access Management (IAM)

Organization

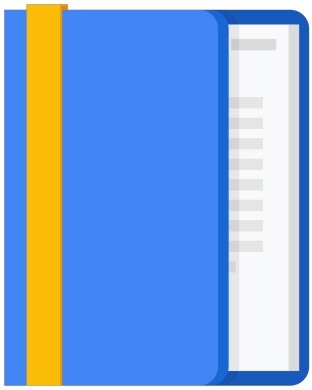
Roles

Members

Service accounts

Cloud IAM best practices

Lab

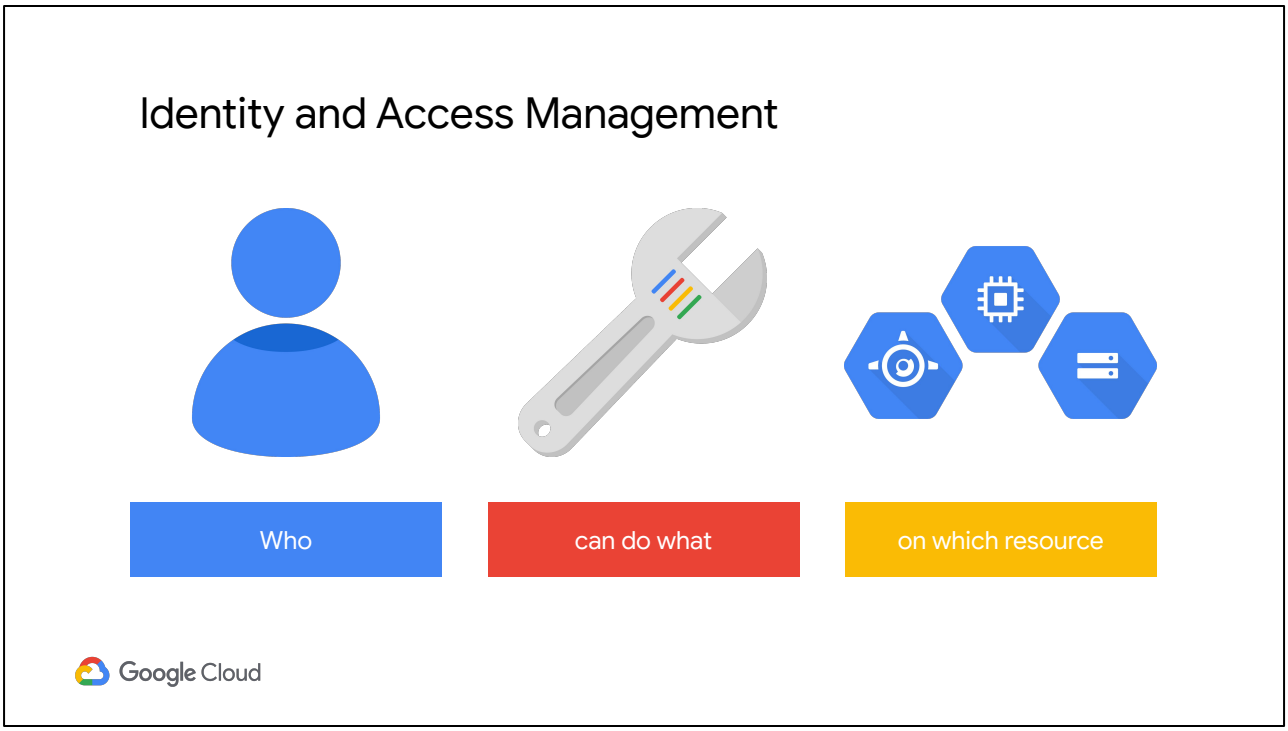


Google Cloud

We will start by introducing Cloud IAM from a high-level perspective. We will then dive into each of the components within Cloud IAM, which are organizations, roles, members, and service accounts. We will also introduce some best practices to help you apply these concepts in your day-to-day work.

Finally, you will gain first-hand experience with Cloud IAM through a lab.

Let’s get started with an overview of Cloud Identity and Access Management!



So what is identity access management? It is a way of identifying who can do what on which resource.

The who can be a person, group, or application. The what refers to specific privileges or actions, and the resource could be any GCP service.

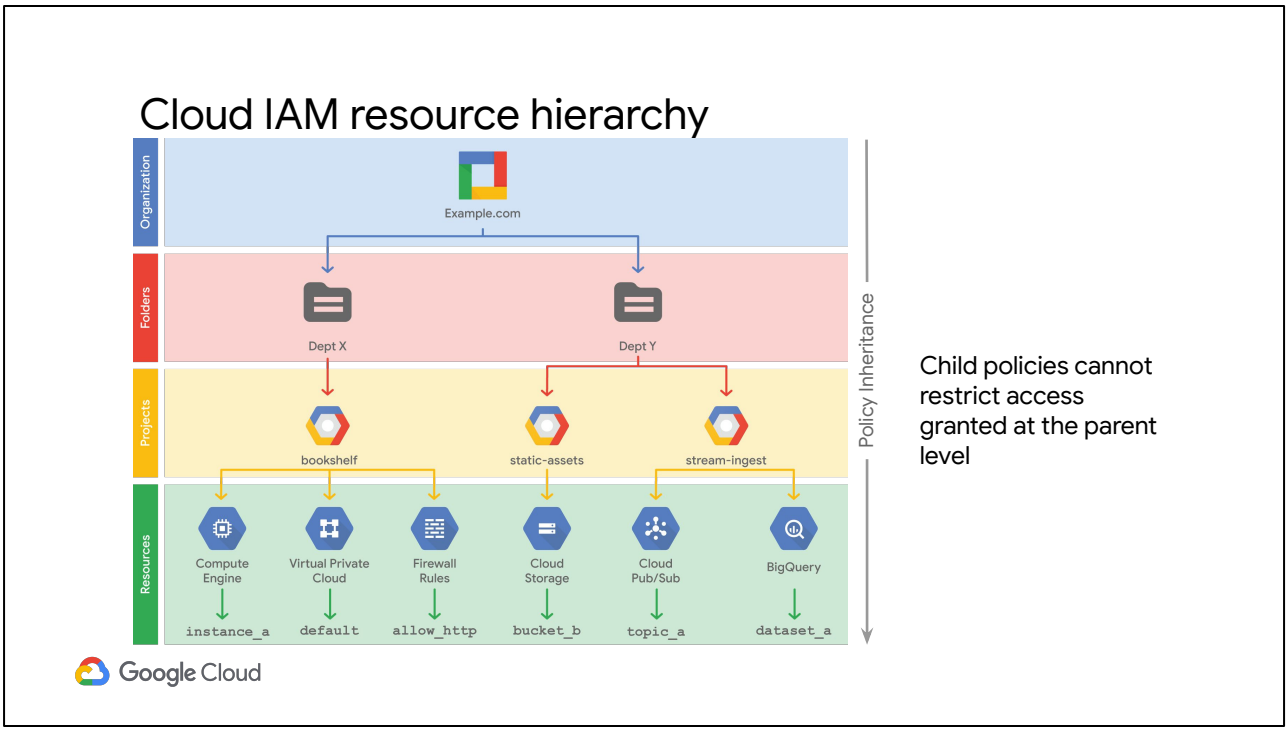
For example, we could give you the privilege or role of Compute Viewer. This provides you with read-only access to get and list Compute Engine resources, without being able to read the data stored on them.

Cloud IAM objects

Organization Folders Projects Resources Roles Members

Google Cloud

Cloud IAM is composed of different objects as shown on this slide. We are going to cover each of these in this module. To get a better understanding of where these fit in, let's look at the Cloud IAM resource hierarchy.



Google Cloud Platform resources are organized hierarchically, as shown in this tree structure. The Organization node is the root node in this hierarchy, folders are the children of the organization, projects are the children of the folders, and the individual resources are the children of projects. Each resource has exactly one parent.

Cloud IAM allows you to set policies at all of these levels, where a policy contains a set of roles and role members. Let me go through each of the levels from top to bottom, because resources inherit policies from their parent.

The organization resource represents your company. Cloud IAM roles granted at this level are inherited by all resources under the organization.

The folder resource could represent your department. Cloud IAM roles granted at this level are inherited by all resources that the folder contains.

Projects represent a trust boundary within your company. Services within the same project have a default level of trust.

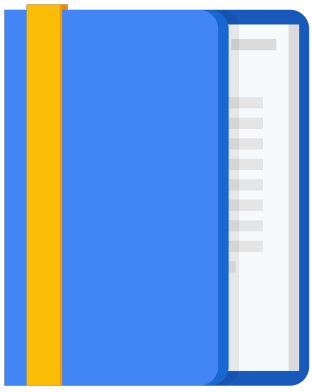
The Cloud IAM policy hierarchy always follows the same path as the GCP resource hierarchy, which means that if you change the resource hierarchy, the policy hierarchy also changes. For example, moving a project into a different organization will update the project's Cloud IAM policy to inherit from the new organization's Cloud IAM policy.

Also, child policies cannot restrict access granted at the parent level. For example, if

we grant you the Editor role for Department X, and we grant you the Viewer role at the bookshelf project level, you still have the Editor role for that project. Therefore, it is a best practice is to follow the principle of least privilege. The principle applies to identities, roles, and resources. Always select the smallest scope that’s necessary for the task in order to reduce your exposure to risk.

Agenda

- Cloud Identity and Access Management (IAM)
- **Organization**
- Roles
- Members
- Service accounts
- Cloud IAM best practices
- Lab

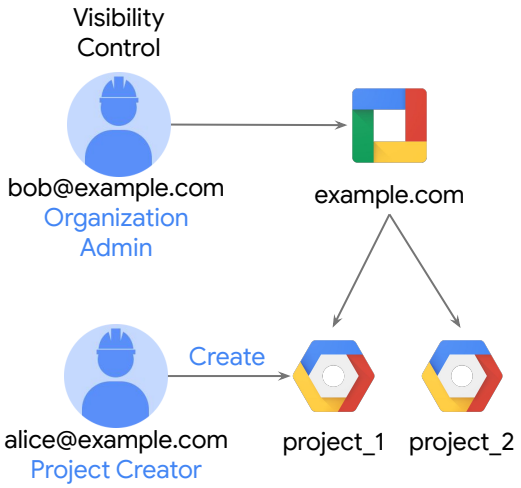


Google Cloud

Let’s learn more about the organization node.

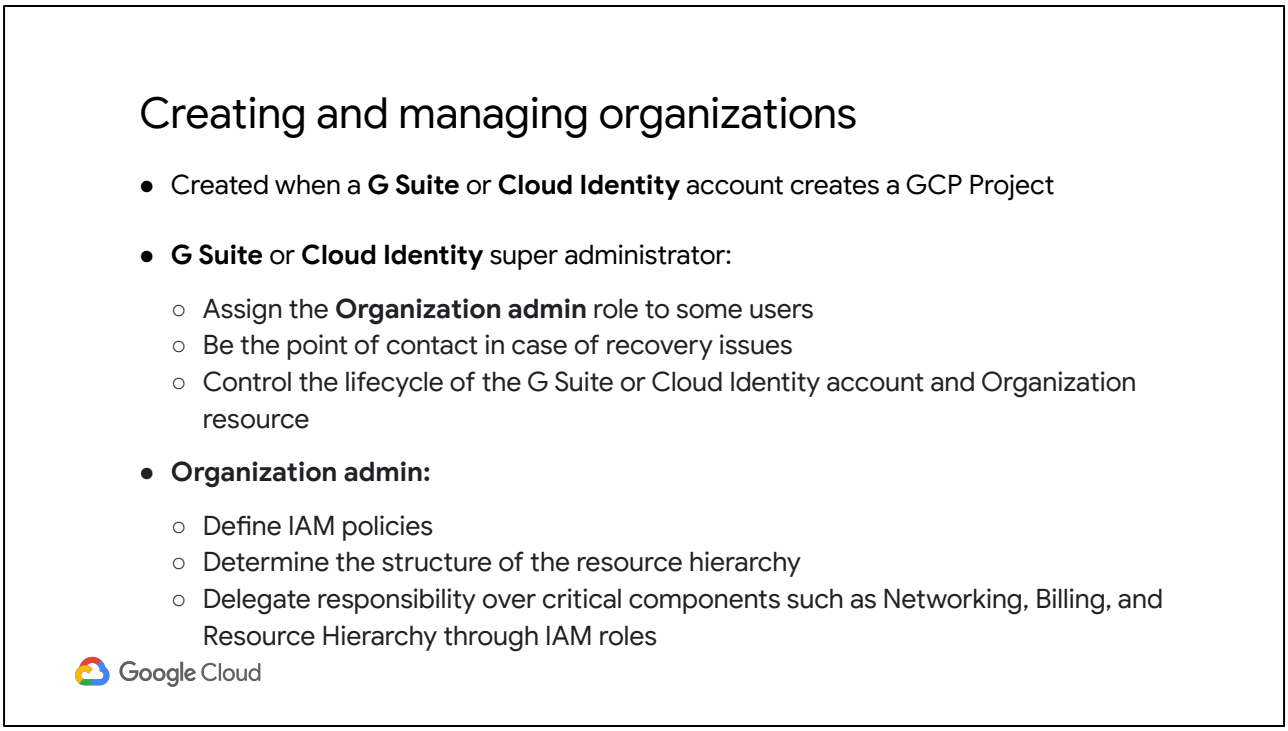
Organization node

- An organization node is a root node for Google Cloud resources
- Organization roles:
  - Organization Admin: Control over all cloud resources; useful for auditing
  - Project Creator: Controls project creation; control over who can create projects



Google Cloud

As we mentioned earlier, the organization resource is the root node in the GCP resource hierarchy. This node has many roles, like the Organization Admin.
The Organization Admin provides a user like Bob with access to administer all resources belonging to his organization, which is useful for auditing.
There is also a Project Creator role, which allows a user like Alice to create projects within her organization. We are showing the project creator role here because it can also be applied at the organization level, which would then be inherited by all the projects within the organization.



The Organization resource is closely associated with a G Suite or Cloud Identity account.
When a user with a G Suite or Cloud Identity account creates a GCP Project, an Organization resource is automatically provisioned for them. Then, Google Cloud communicates its availability to the G Suite or Cloud Identity super admins. These super admin accounts should be used carefully because they have a lot of control over your organization and all the resources underneath it.

The G Suite or Cloud Identity super administrators and the GCP Organization admin are key roles during the setup process and for lifecycle control for the Organization resource. The two roles are generally assigned to different users or groups, although this depends on the organization structure and needs.

In the context of GCP Organization setup, the G Suite or Cloud Identity super administrator responsibilities are:

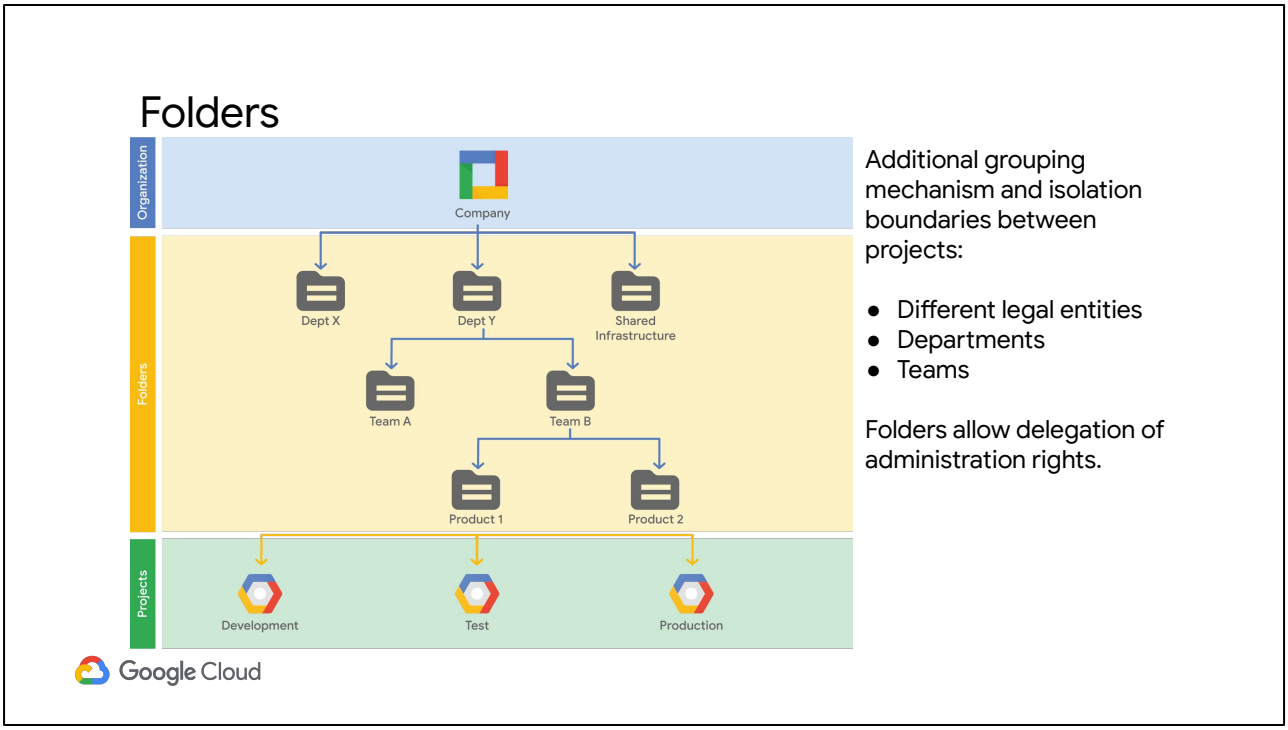
- Assign the Organization admin role to some users.
- Be a point of contact in case of recovery issues.
- Control the lifecycle of the G Suite or Cloud Identity account and Organization resource.

The responsibilities of the Organization admin role are:

- Define IAM policies.
- Determine the structure of the resource hierarchy.

- Delegate responsibility over critical components such as Networking, Billing, and Resource Hierarchy through IAM roles.

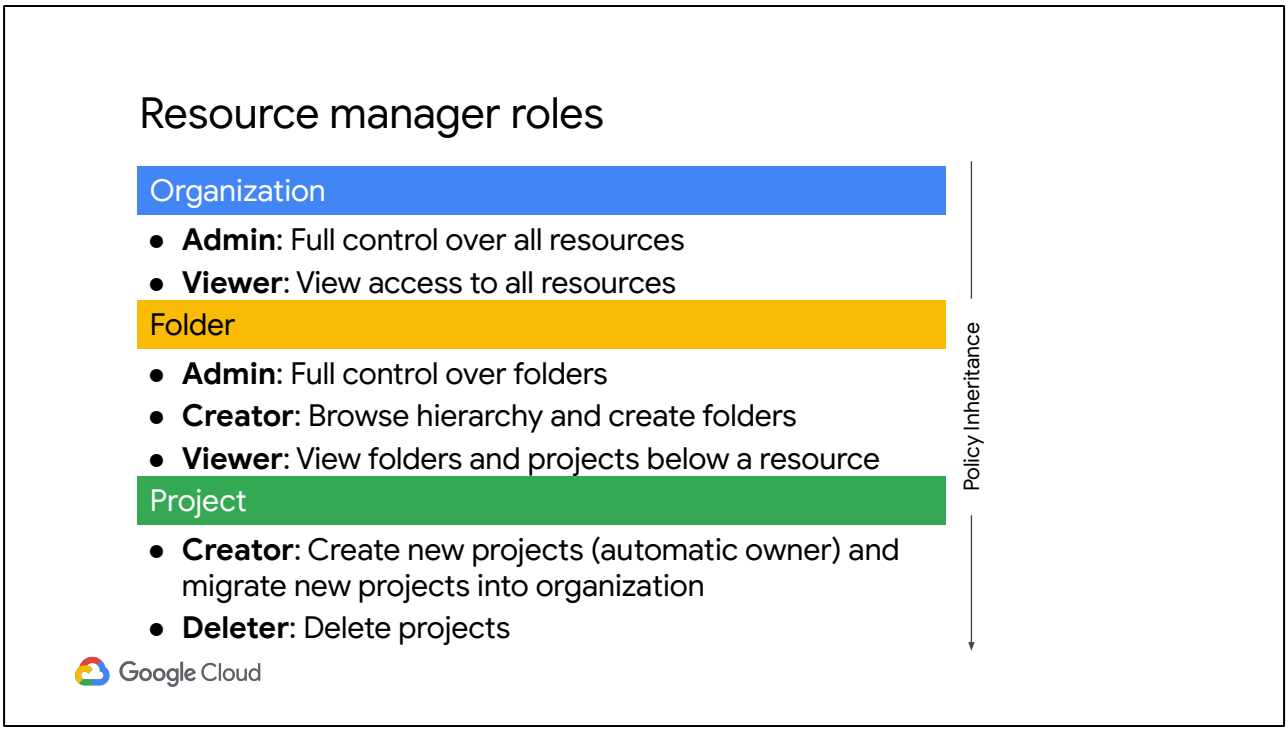
Following the principle of least privilege, this role does not include the permission to perform other actions, such as creating folders. To get these permissions, an Organization admin must assign additional roles to their account. For more information about creating and managing organizations, see this how-to guide.



Let's talk more about folders, because they can be viewed as sub-organizations within the organization.

Folders provide an additional grouping mechanism and isolation boundary between projects. Folders can be used to model different legal entities, departments, and teams within a company. For example, a first level of folders could be used to represent the main departments in your organization, like departments X and Y. Because folders can contain projects and other folders, each folder could then include other sub-folders to represent different teams, like teams A and B.
Each team folder could contain additional sub-folders to represent different applications, like Products 1 and 2.

Folders allow delegation of administration rights, so for example, each head of a department can be granted full ownership of all GCP resources that belong to their department. Similarly, access to resources can be limited by folder, so users in one department can only access and create GCP resources within that folder.



Let's look at some other resource manager roles, while remembering that policies are inherited from top to bottom.

The organization node also has a Viewer role that grants view access to all resources within an organization.

The folder node has multiple roles that mimic the organizational roles but are applied to resources within a folder. There is an admin role that provides full control over folders; a creator role to browse the hierarchy and create folders; and a viewer role to view folders and projects below a resource.

Similarly, for projects, there is a creator role that allows a user to create new projects, making that user automatically the owner. There is also a project deleter role that grants deletion privileges for projects.

Agenda

Cloud Identity and Access Management (IAM)

Organization

Roles

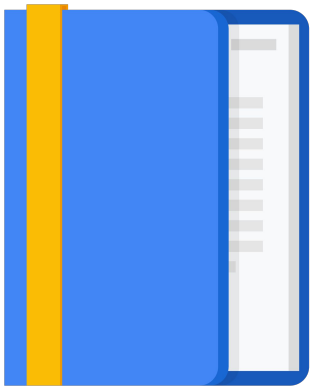
Members

Service accounts

Cloud IAM best practices

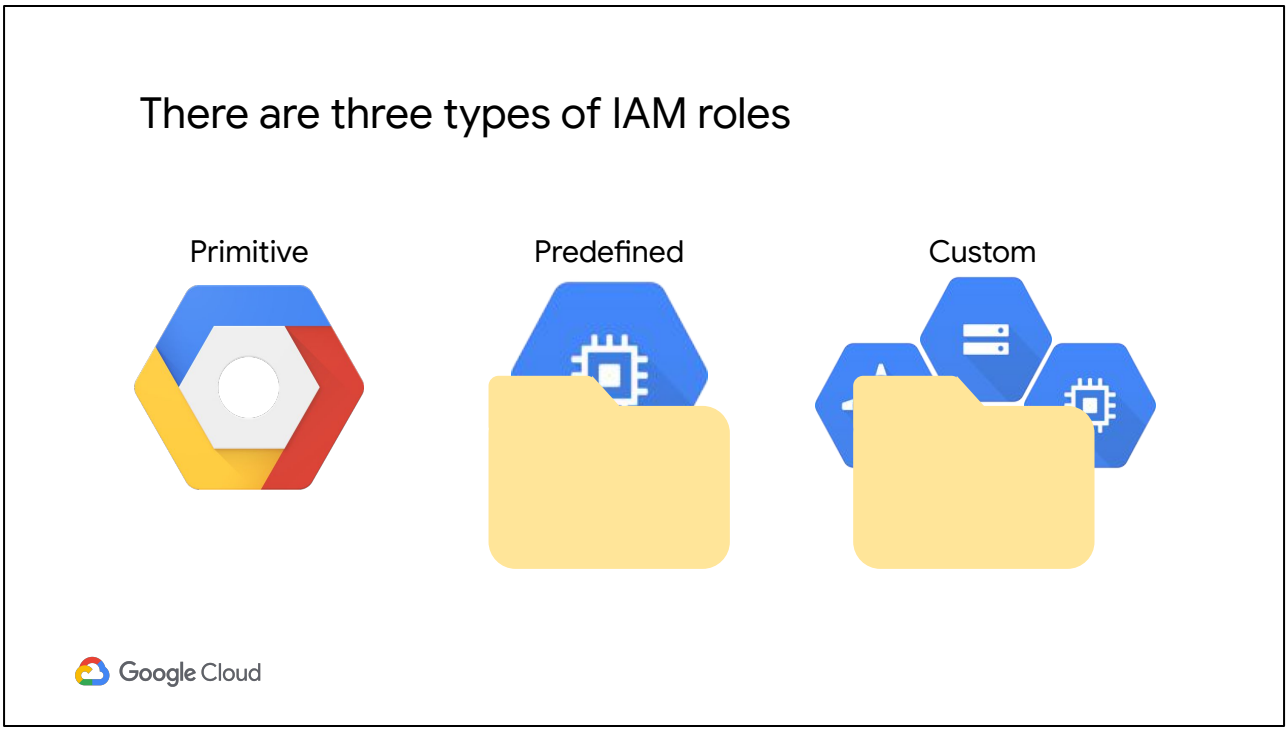
Lab

Quiz



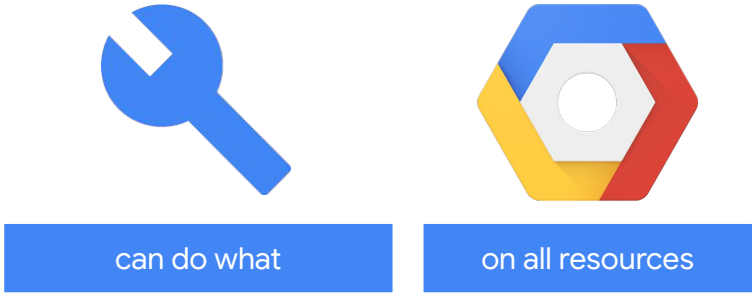
Google Cloud

Let’s talk more about roles, which define the “can do what on which resource” part of Cloud IAM.



There are three types of roles in Cloud IAM: primitive roles, predefined roles, and custom roles.

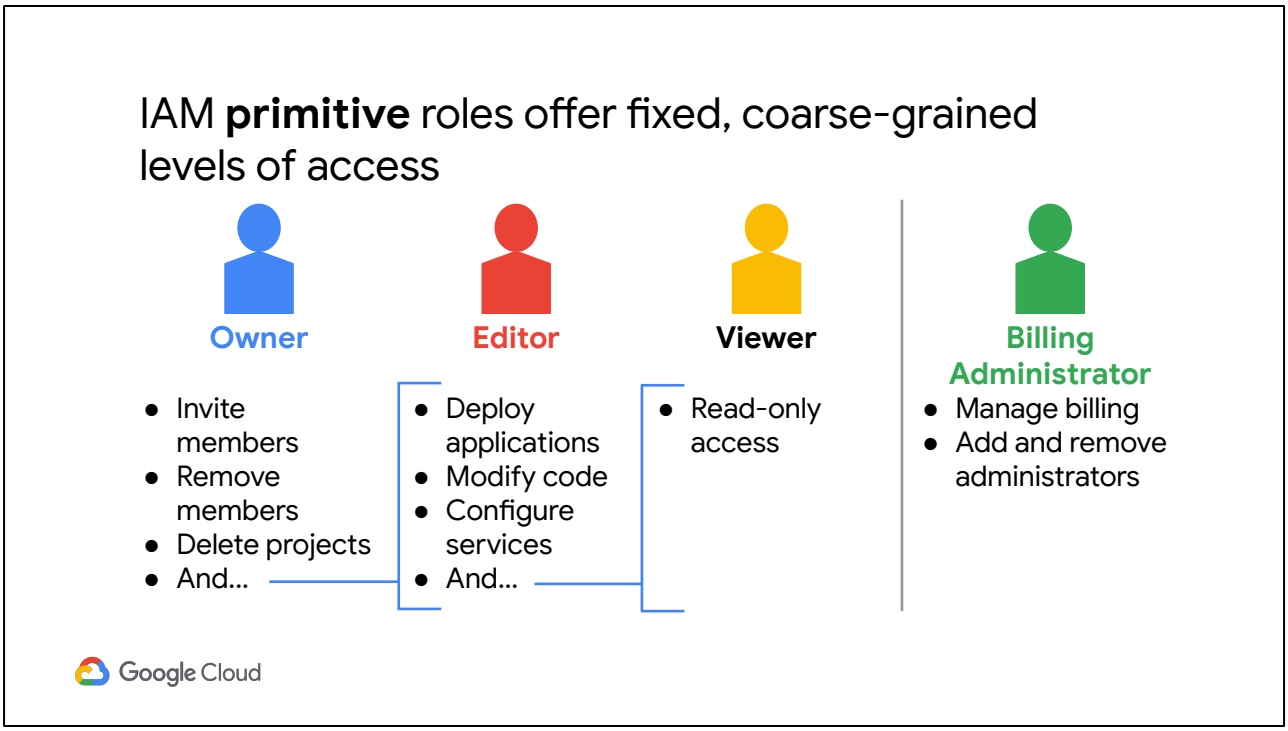
IAM **primitive** roles apply across all GCP services in a project



Google Cloud

Primitive roles are the original roles that were available in the GCP Console, but they are broad.

You apply them to a GCP project, and they affect all resources in that project.

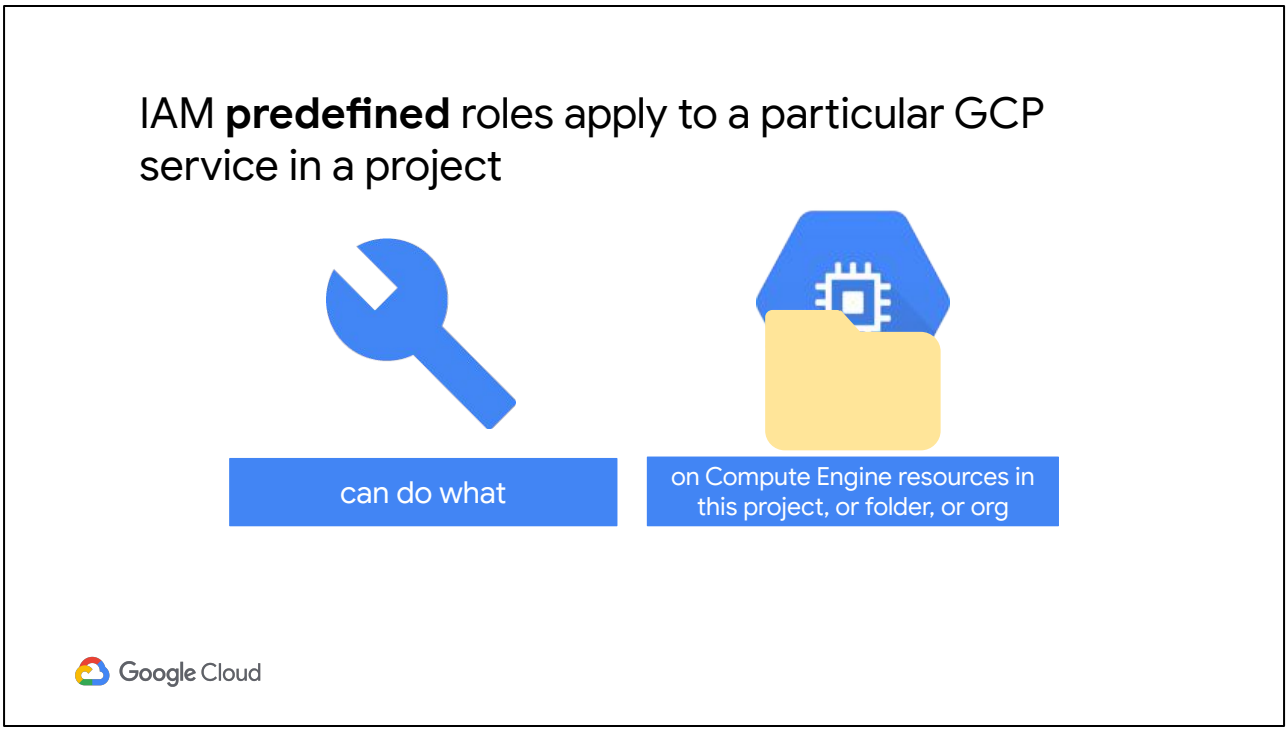


In other words, IAM primitive roles offer fixed, coarse-grained levels of access.

The primitive roles are the Owner, Editor, and Viewer roles.

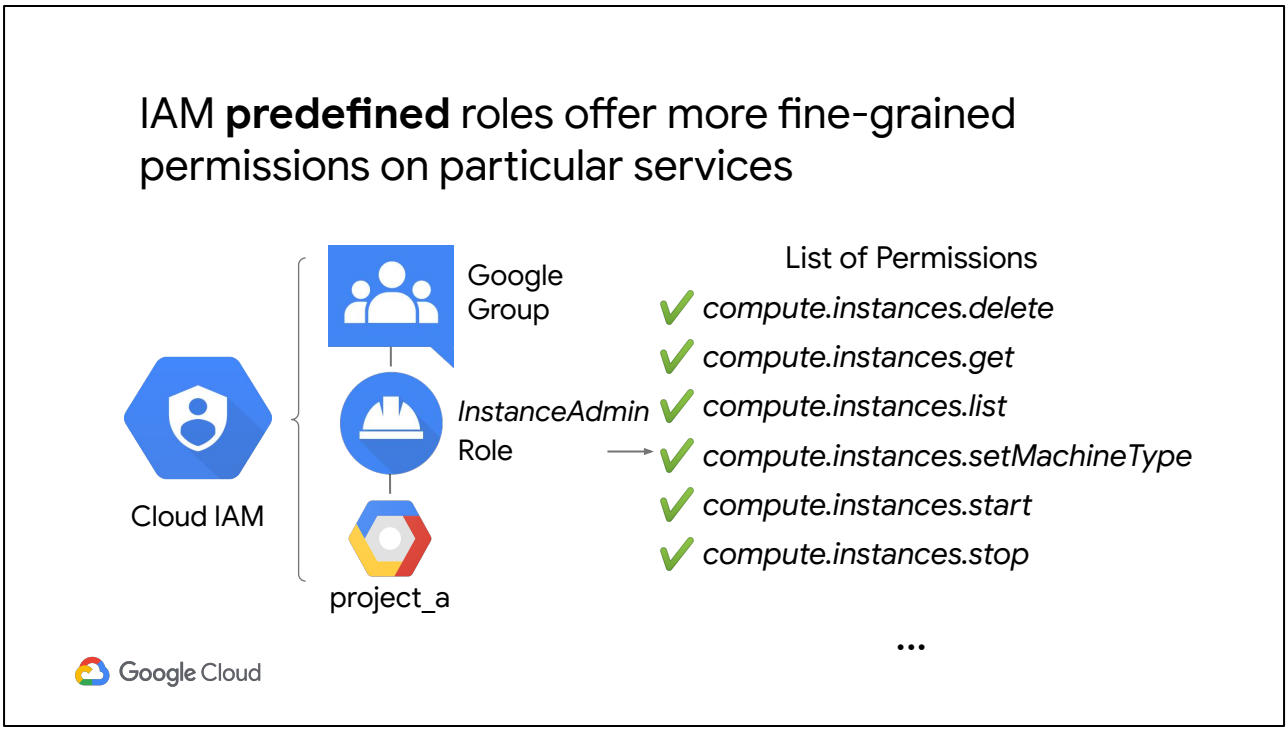
- The owner has full administrative access. This includes the ability to add and remove members and delete projects.
- The editor role has modify and delete access. This allows a developer to deploy applications and modify or configure its resources.
- The viewer role has read-only access.

All of these roles are concentric; that is, the Owner role includes the permissions of the Editor role, and the Editor role includes the permissions of the Viewer role.
There is also a billing administrator role to manage billing and add or remove administrators without the right to change the resources in the project.
Each project can have multiple owners, editors, viewers, and billing administrators.



GCP services offers their own sets of predefined roles, and they define where those roles can be applied. This provides members with granular access to specific GCP resources and prevents unwanted access to other resources.

These roles are *collections* of permissions because, to do any meaningful operations, you usually need more than one permission.



For example, as shown here, a group of users is granted the InstanceAdmin role on project_a. This provides the users of that group with all the Compute Engine permissions listed on the right and more. Grouping these permissions into a role makes them easier to manage. The permissions themselves are classes and methods in the APIs.

For example, compute.instances.start can be broken down into the service, resource, and verb that mean that this permission is used to start a stopped Compute Engine instance.

These permissions usually align with the action's corresponding REST API.

Compute Engine IAM roles

Role Title	Description
Compute Admin	Full control of all Compute Engine resources (compute.*)
Network Admin	Permissions to create, modify, and delete networking resources, except for firewall rules and SSL certificates
Storage Admin	Permissions to create, modify, and delete disks, images, and snapshots

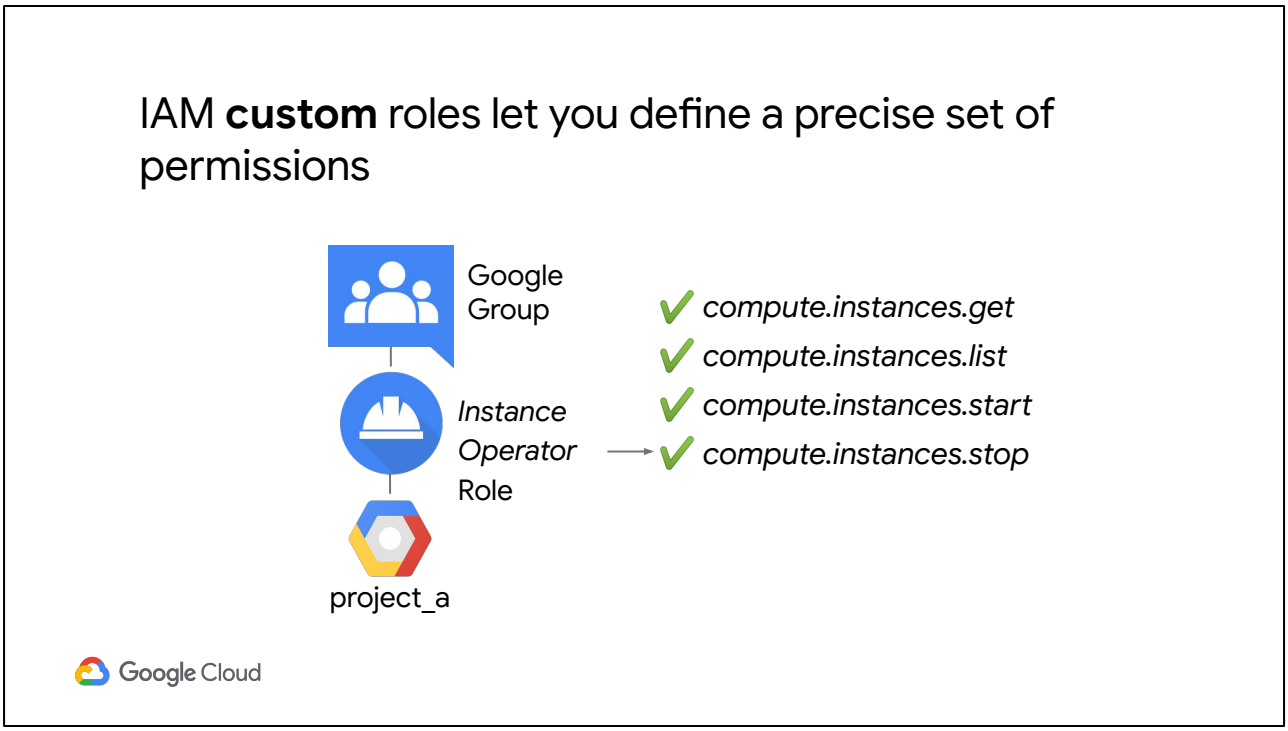
Google Cloud

Compute Engine has several predefined IAM roles. Let's look at three of those:

- The Compute Admin role provides full control of all Compute Engine resources. This includes all permissions that start with *compute*, which means that every action for any type of Compute Engine resource is permitted.
- The Network Admin role contains permissions to create, modify, and delete networking resources, *except* for firewall rules and SSL certificates. In other words, the network admin role allows read-only access to firewall rules, SSL certificates, and instances to view their ephemeral IP addresses.
- The Storage Admin role contains permissions to create, modify, and delete disks, images, and snapshots.

For example, if your company has someone who manages project images and you don't want them to have the editor role on the project, grant their account the Storage Admin role on the project.
For the full list of predefined roles for Compute Engine, please refer to this documentation page.

Now, roles are meant to represent abstract functions and are customized to align with real jobs. But what if one of these roles does not have enough permissions, or you need something even finer-grained?



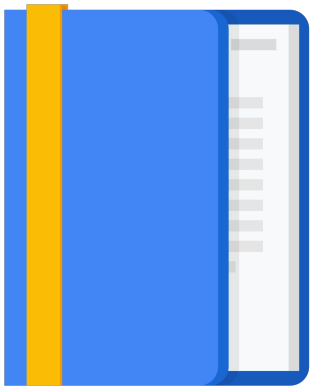
That's what custom roles permit. A lot of companies use the "least-privilege" model, in which each person in your organization is given the minimal amount of privilege needed to do their job.

Let's say you want to define an "Instance Operator" role to allow some users to start and stop Compute Engine virtual machines, but not reconfigure them. Custom roles allow you to do that.

For a demo on how to create a custom role in GCP, refer to this video.

Agenda

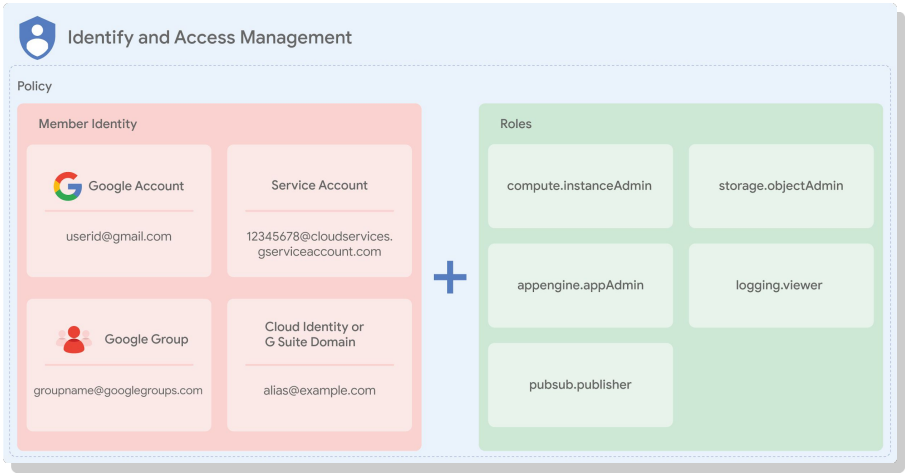
- Cloud Identity and Access Management (IAM)
- Organization
- Roles
- **Members**
- Service accounts
- Cloud IAM best practices
- Lab
- Quiz



Google Cloud

Let’s talk more about members, which define the “who” part of “who can do what on which resource.”

Members



Note: You *cannot* use Cloud IAM to create or manage your users or groups.

Google Cloud

There are five different types of members: Google Accounts, Service Accounts, Google groups, G Suite domains, and Cloud Identity domains.

A Google account represents a developer, an administrator, or any other person who interacts with GCP. Any email address that is associated with a Google account can be an identity, including gmail.com or other domains. New users can sign up for a Google account by going to the Google account signup page, without receiving mail through Gmail.

A service account is an account that belongs to your application instead of to an individual end user. When you run code that is hosted on GCP, you specify the account that the code should run as. You can create as many service accounts as needed to represent the different logical components of your application.

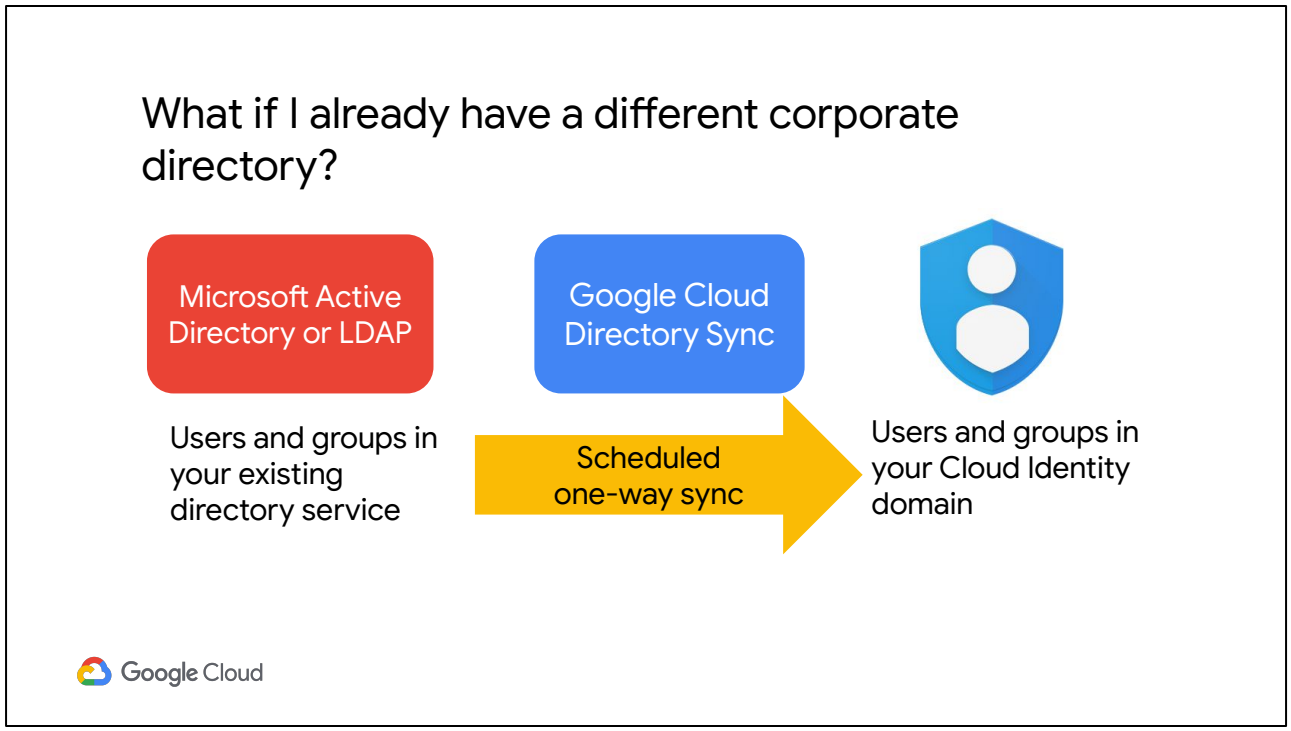
A Google group is a named collection of Google accounts and service accounts. Every group has a unique email address that is associated with the group. Google groups are a convenient way to apply an access policy to a collection of users. You can grant and change access controls for a whole group at once instead of granting or changing access controls one-at-a-time for individual users or service accounts.

A G Suite domain represents a virtual group of all the Google accounts that have

been created in an organization's G Suite account. G Suite domains represent your organization's internet domain name, such as example.com, and when you add a user to your G Suite domain, a new Google account is created for the user inside this virtual group, such as username@example.com.

GCP customers who are not G Suite customers can get these same capabilities through Cloud Identity. Cloud Identity lets you manage users and groups using the Google Admin Console, but you do not pay for or receive G Suite's collaboration products such as Gmail, Docs, Drive, and Calendar. Cloud Identity is available in free and premium editions. The premium edition adds capabilities for mobile device management.

Now it's important to note that you cannot use Cloud IAM to create or manage your users or groups. Instead, you can use Cloud Identity or G Suite to create and manage users.



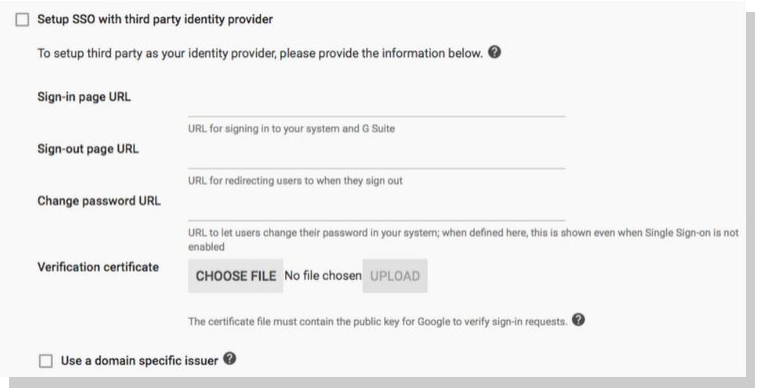
What if you already have a different corporate directory? How can you get your users and groups into GCP?

Using Google Cloud Directory Sync, your administrators can log in and manage GCP resources using the same usernames and passwords they already use. This tool synchronizes users and groups from your existing Active Directory or LDAP system with the users and groups in your Cloud Identity domain.

The synchronization is one-way only; which means that no information in your Active Directory or LDAP map is modified. Google Cloud Directory Sync is designed to run scheduled synchronizations without supervision, after its synchronization rules are set up.

Single sign-on (SSO)

- Use Cloud Identity to configure SAML SSO,
- If SAML2 isn't supported, use a third-party solution (ADFS, Ping, or Okta).



Google Cloud

GCP also provides single sign-on authentication.

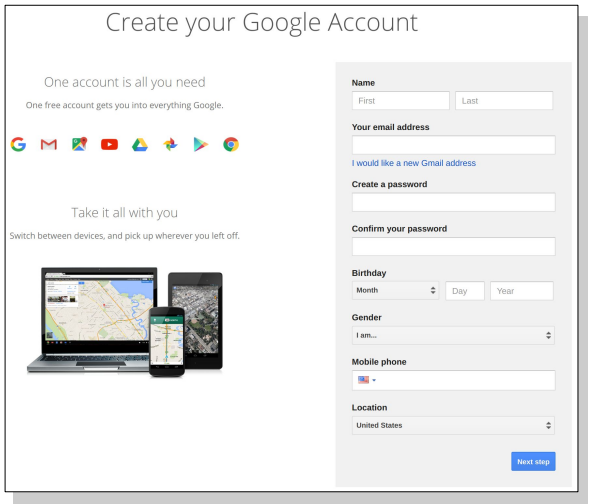
If you have your identity system, you can continue using your own system and processes with SSO configured. When user authentication is required, Google will redirect to your system. If the user is authenticated in your system, access to Google Cloud Platform is given; otherwise, the user is prompted to sign in.

This allows you to also revoke access to GCP.

If your existing authentication system supports SAML2, SSO configuration is as simple as 3 links and a certificate, as shown on this slide. Otherwise, you can use a third-party solution, like ADFS, Ping, or Okta.

For more information about using your existing identity management system, please refer to this documentation page.

GCP access without Gmail



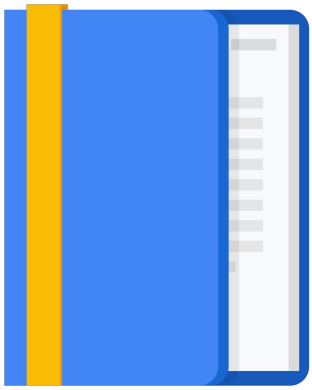
- You can get a Google password without Gmail.
- There are benefits to having a domain, including group permissions.

Google Cloud

Also, if you want to use a Google account but are not interested in receiving mail through Gmail, you can still create an account without Gmail. For more information about this, please refer to this page.

Agenda

- Cloud Identity and Access Management (IAM)
- Organization
- Roles
- Members
- **Service accounts**
- Cloud IAM best practices
- Lab
- Quiz



Google Cloud

As we mentioned earlier, another type of member is a service account.

Service accounts provide an identity for carrying out server-to-server interactions

- Programs running within Compute Engine instances can automatically acquire access tokens with credentials.
- Tokens are used to access any service API in your project and any other services that granted access to that service account.
- Service accounts are convenient when you're not accessing user data.

Google Cloud

A service account is an account that belongs to your application instead of to an individual end user. This provides an identity for carrying out server-to-server interactions in a project without supplying user credentials.

For example, if you write an application that interacts with Google Cloud Storage, it must first authenticate to either the Google Cloud Storage XML API or JSON API.

You can enable service accounts and grant read-write access to the account on the instance where you plan to run your application.

Then, program the application to obtain credentials from the service account. Your application authenticates seamlessly to the API without embedding any secret keys or credentials in your instance, image, or application code.

Service accounts are identified by an email address

- `123845678986-compute@project.gserviceaccount.com`
- Three types of service accounts:
  - User-created (custom)
  - Built-in
    - Compute Engine and App Engine default service accounts
  - Google APIs service account
    - Runs internal Google processes on your behalf.

Google Cloud

Service accounts are identified by an email address, like the example shown here.

There are three types of service accounts: user-created or custom, built-in, and Google APIs service accounts.

By default, all projects come with the built-in Compute Engine default service account.

Apart from the default service account, all projects come with a Google Cloud Platform APIs service account, identifiable by the email: project-number@cloudservices.gserviceaccount.com. This is a service account designed specifically to run internal Google processes on your behalf, and it is automatically granted the Editor role on the project.

Alternatively, you can also start an instance with a custom service account. Custom service accounts provide more flexibility than the default service account, but they require more management from you. You can create as many custom service accounts as you need, assign any arbitrary access scopes or Cloud IAM roles to them, and assign the service accounts to any virtual machine instance.

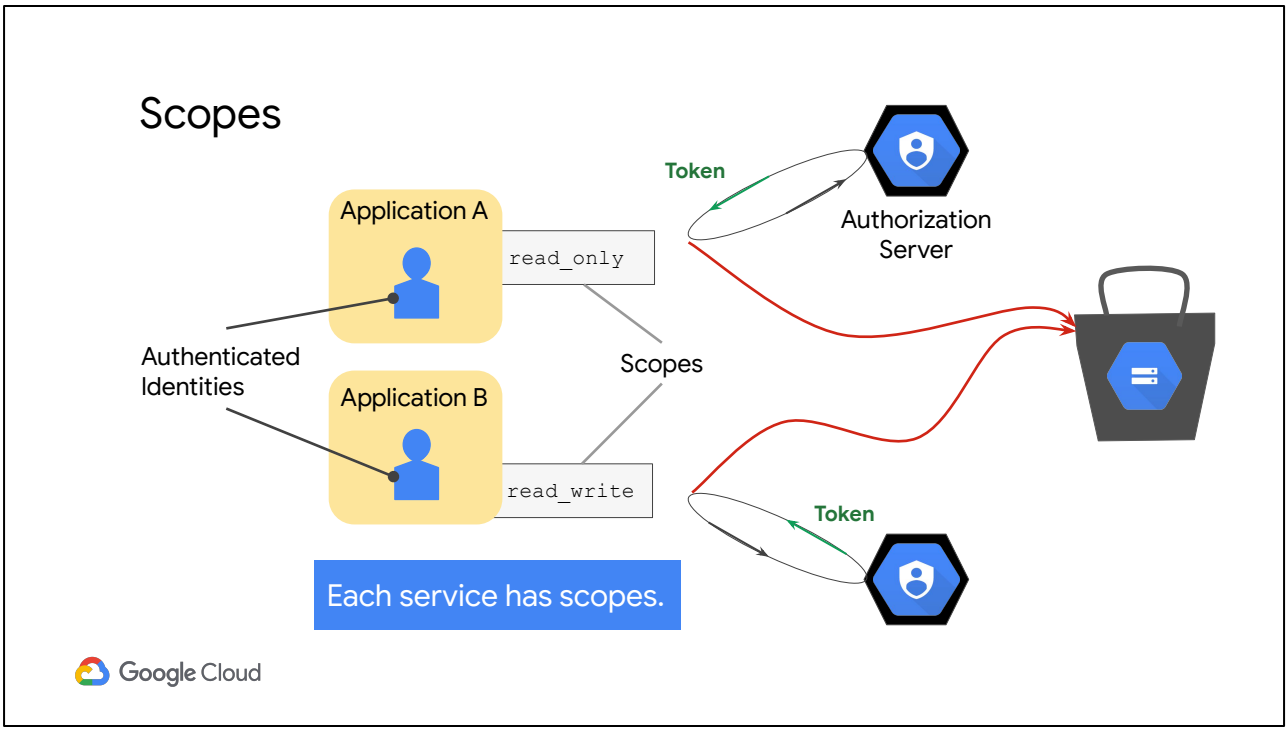
Default Compute Engine service account

- Automatically created per project with auto-generated name and email address:
  - Name has -compute suffix
    `39xxxx0965-compute@developer.gserviceaccount.com`
- Automatically added as a project Editor
- By default, enabled on all instances created using gcloud or GCP Console

Google Cloud

Let's talk more about the default Compute Engine service account. As we mentioned, this account is automatically created per project. This account is identifiable by the email project-number-compute@developer.gserviceaccount.com, and it is automatically granted the Editor role on the project.

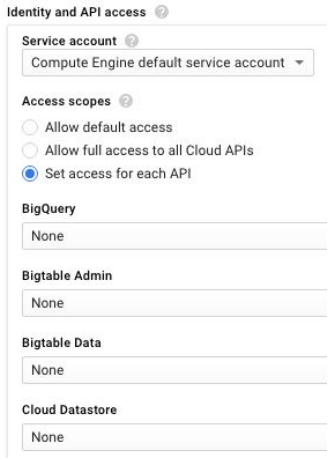
When you start a new instance using gcloud, the default service account is enabled on that instance. You can override this behavior by specifying another service account or by disabling service accounts for the instance.



Now, authorization is the process of determining what permissions an authenticated identity has on a set of specified resources. Scopes are used to determine whether an authenticated identity is authorized.

In the example shown here, Applications A and B contain Authenticated Identities (or service accounts). Let's assume that both applications want to use a Cloud Storage bucket. They each request access from the Google Authorization server, and in return they receive an access token. Application A receives an access token with read-only scope, so it can only read from the Cloud Storage bucket. Application B, in contrast, receives an access token with read-write scope, so it can read and modify data in the Cloud Storage bucket.

Customizing scopes for a VM



- Scopes can be changed after an instance is created.
- For user-created service accounts, use Cloud IAM roles instead.

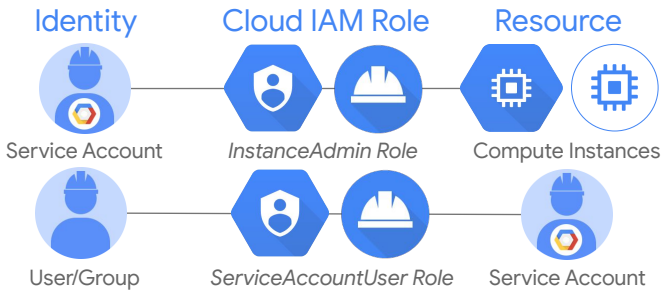
Google Cloud

Scopes can be customized when you create an instance using the default service account, as shown in this screenshot. These scopes can be changed after an instance is created by stopping it. Access scopes are actually the legacy method of specifying permissions for your VM. Before the existence of IAM roles, access scopes were the only mechanism for granting permissions to service accounts.

For user-created service accounts, use Cloud IAM roles instead to specify permissions.

Service account permissions

- Default service accounts: primitive and predefined roles
- User-created service accounts: predefined roles
- Roles for service accounts can be assigned to groups or users



Google Cloud

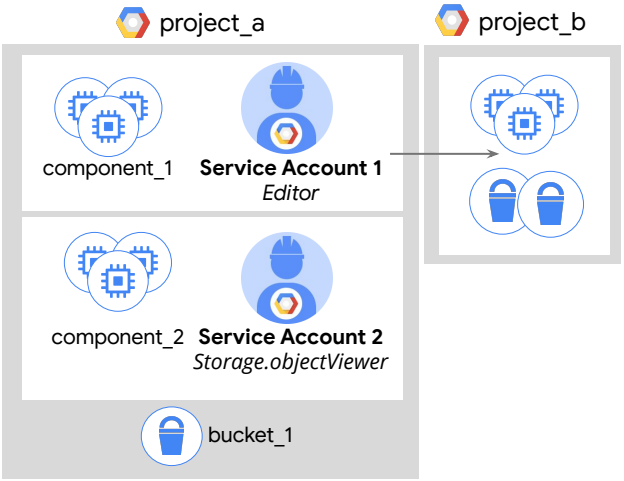
Another distinction between service accounts is that default service accounts support both primitive and predefined IAM roles, but user-created service accounts only use predefined IAM roles.

Now, roles for service accounts can also be assigned to groups or users. Let's look at the example shown on this slide. First, you create a service account that has the InstanceAdmin role, which has permissions to create, modify, and delete virtual machine instances and disks. Then you treat this service account as the resource, and decide who can use it by providing users or a group with the Service Account User role. This allows those users to act as that service account to create, modify, and delete virtual machine instances and disks.

Users who are Service Account Users for a service account can access all the resources that the service account has access to. Therefore, be cautious when granting the Service Account User role to a user or group.

Example: Service accounts and Cloud IAM

- VMs running component_1 are granted Editor access to project_b using *Service Account 1*.
- VMs running component_2 are granted objectViewer access to bucket_1 using *Service Account 2*.
- Service account permissions can be changed without re-created VMs.



Google Cloud

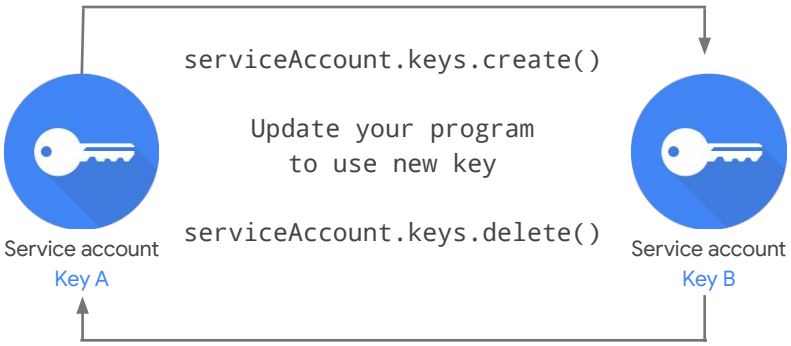
Here is another example. The VMs running component_1 are granted Editor access to project_b using Service Account 1. VMs running component_2 are granted objectViewer access to bucket_1 using an isolated Service Account 2. This way you can scope permissions for VMs without re-creating VMs.

Essentially, Cloud IAM lets you slice a project into different microservices, each with access to different resources, by creating service accounts to represent each one. You assign the service accounts to the VMs when they are created, and you don't have to ensure that credentials are being managed correctly because GCP manages security for you.

Now, you might ask, how are service accounts authenticated?

Service accounts authenticate with keys

- GCP-managed: Cannot be downloaded, and are automatically rotated
- User-managed: Create, manage, and rotate yourself



Google Cloud

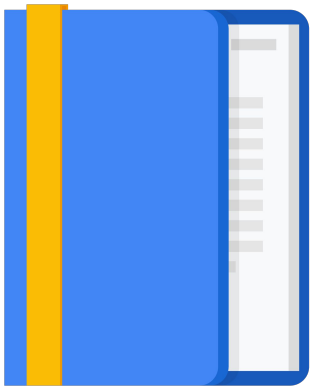
Although users require a username and password to authenticate, service accounts use keys. There are two types of service account keys: GCP-managed keys and user-managed keys.

GCP-managed keys are used by GCP services such as App Engine and Compute Engine. These keys cannot be downloaded and are automatically rotated and used for a maximum of two weeks.

User-managed keys are created, downloadable, and managed by users. When you create a new key pair, you download the private key, which is not retained by Google. With user-managed keys, you are responsible for security of the private key and other management operations such as key rotation, which is illustrated on this slide.

Agenda

- Cloud Identity and Access Management (IAM)
- Organization
- Roles
- Members
- Service accounts
- **Cloud IAM best practices**
- Lab
- Quiz



Google Cloud

Let’s talk about some Cloud IAM best practices to help you apply the concepts you just learned in your day-to-day work.

Leverage and understand the resource hierarchy

- Use projects to group resources that share the same trust boundary.
- Check the policy granted on each resource and make sure you understand the inheritance.
- Use “principles of least privilege” when granting roles.
- Audit policies in Cloud audit logs: `setiampolicy`.
- Audit membership of groups used in policies.

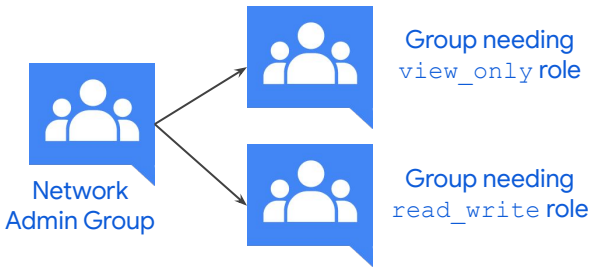
Google Cloud

First, leverage and understand the resource hierarchy.

- Specifically, use projects to group resources that share the same trust boundary.
- Check the policy granted on each resource and make sure you recognize the inheritance.
- Because of inheritance, use the "principle of least privilege" when granting roles.
- Finally, audit policies using Cloud audit logs and audit memberships of groups used in policies.

Grant roles to Google groups instead of individuals

- Update group membership instead of changing Cloud IAM policy.
- Audit membership of groups used in policies.
- Control the ownership of the Google group used in Cloud IAM policies.



Google Cloud

Next, we recommend granting roles to groups instead of individuals. This allows you to update group membership, instead of changing a Cloud IAM policy. If you do this, make sure to audit membership of groups used in policies and control the ownership of the Google group used in Cloud IAM policies.

You can also use multiple groups to get better control. In the example on this slide, there is a network admin group. Some of those members also need a read_write role to a Cloud Storage bucket, but others need the read_only role. Adding and removing individuals from all three groups controls their total access. Therefore, groups are not only associated with job roles but can exist for the purpose of role assignment.

Service accounts

- Be very careful granting `serviceAccountUser` role.
- When you create a service account, give it a display name that clearly identifies its purpose.
- Establish a naming convention for service accounts.
- Establish key rotation policies and methods.
- Audit with `serviceAccount.keys.list()` method.

Google Cloud

Here are some best practices for using service accounts:

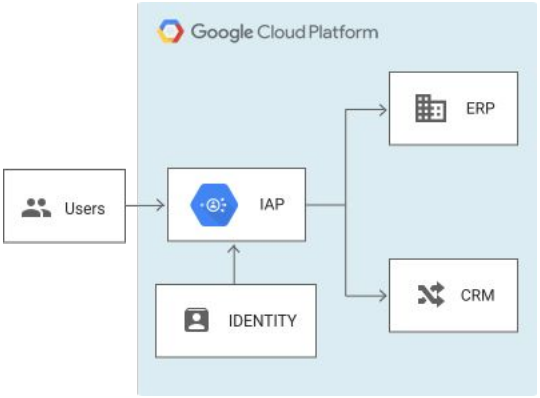
- As mentioned before, be very careful when granting the Service Account Users role, because it provides access to all the resources that the service account has access to.
- Also, when you create a service account, give it a display name that clearly identifies its purpose, ideally using an established naming convention.
- As for keys, establish key rotation policies and methods and audit keys with the serviceAccount.keys.list() method.

Cloud Identity-Aware Proxy (Cloud IAP)

Enforce access control policies for applications and resources:

- Identity-based access control
- Central authorization layer for applications accessed by HTTPS

Cloud IAM policy is applied after authentication.



Google Cloud

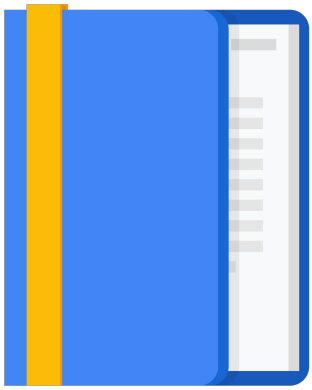
Finally, we recommend using Cloud Identity-Aware proxy, or Cloud IAP. Cloud IAP lets you establish a central authorization layer for applications accessed by HTTPS, so you can use an application-level access control model instead of relying on network-level firewalls.

Applications and resources protected by Cloud IAP can only be accessed through the proxy by users and groups with the correct Cloud IAM role. When you grant a user access to an application or resource by Cloud IAP, they're subject to the fine-grained access controls implemented by the product in use without requiring a VPN. Cloud IAP performs authentication and authorization checks when a user tries to access a Cloud IAP-secured resource, as shown on the right.

For more information about Cloud IAP, please refer to this documentation page.

Agenda

- Cloud Identity and Access Management (IAM)
- Organization
- Roles
- Members
- Service accounts
- Cloud IAM best practices
- Lab



Google Cloud

Let’s take some of the concepts that you just learned about and apply them in a lab.

30 minutes

Lab #1 of 1

Cloud IAM

Cloud IAM

Objectives

- Use Cloud IAM to implement access control
- Restrict access to specific features or resources
- Use the Service Account User role

Google Cloud

In this lab, you'll grant and revoke roles to change access. Specifically, you will use Cloud IAM to implement access control, restrict access to specific features and resources, and use the Service Account User role.

Now, anytime you make changes to IAM roles, the GCP Console refreshes faster than the actual system. Therefore, you should expect some short delays when making changes to a member's role.

Quiz

What abstraction is primarily used to administer user access in Cloud IAM?

A. Leases, an abstraction of periodic entitlements

B. Roles, an abstraction of job roles

C. Credentials, an abstraction of an authorization token

D. Privileges, an abstraction of access rights

Quiz

What abstraction is primarily used to administer user access in Cloud IAM?

A. Leases, an abstraction of periodic entitlements

B. Roles, an abstraction of job roles

C. Credentials, an abstraction of an authorization token

D. Privileges, an abstraction of access rights

Google Cloud

Explanation:
Cloud IAM administration uses pre-defined roles for administration of user access. The roles are defined by more granular permissions. But permissions are not applied to users directly; only through the roles that are assigned to them.

Quiz

Which of the following is not a type of IAM role?

A. Primitive

B. Predefined

C. Custom

D. Advanced

Quiz

Which of the following is not a type of IAM role?

A. Primitive

B. Predefined

C. Custom

D. Advanced

Explanation:
There are three types of roles in Cloud IAM: primitive roles, predefined roles, and custom roles.

Quiz

Which of the following is not a type of IAM member?

A. Google Account
B. Service Account
C. Google Group
D. Organization Account
E. Cloud Identity domain
F. G Suite domain

Quiz

Which of the following is not a type of IAM member?

A. Google Account
B. Service Account
C. Google Group
D. Organization Account
E. Cloud Identity domain
F. G Suite domain

Google Cloud

Explanation:
There are five different types of members: Google Accounts, Service Accounts, Google groups, G Suite domains, and Cloud Identity domains.

Review

Cloud IAM

Google Cloud

In this module, we covered Cloud IAM along with its components and best practices.

Cloud IAM builds on top of other GCP identity services. The creation and administration of corporate identities occurs through the G Suite admin or Cloud Identity interface, and is commonly handled by a person separate from the GCP administrator.

Google groups are a great way for these two business functions to collaborate. You establish the roles and assign them to the group, and then the G Suite admin administers membership in the group.

Finally, remember that service accounts are very flexible, and they can enable you to build an infrastructure-based level of control into your application.


Google Cloud