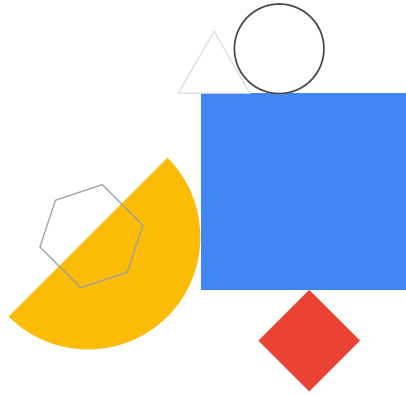Google Cloud

Identity and Access Management

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

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

01	Identity and Access Management (IAM)
02	Organization
03	Roles
04	Members
05	Service Accounts
06	Organization Restrictions
07	IAM Best Practices
	Lab: Exploring IAM

Google Cloud
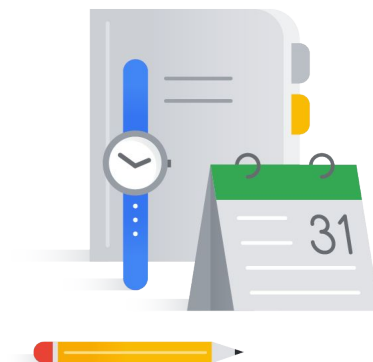
We start by introducing IAM from a high-level perspective. We will then dive into each of the components within IAM, which are organizations, roles, members, and service accounts. We talk about the Organization Restrictions feature and also introduce some best practices to help you apply these concepts in your day-to-day work.
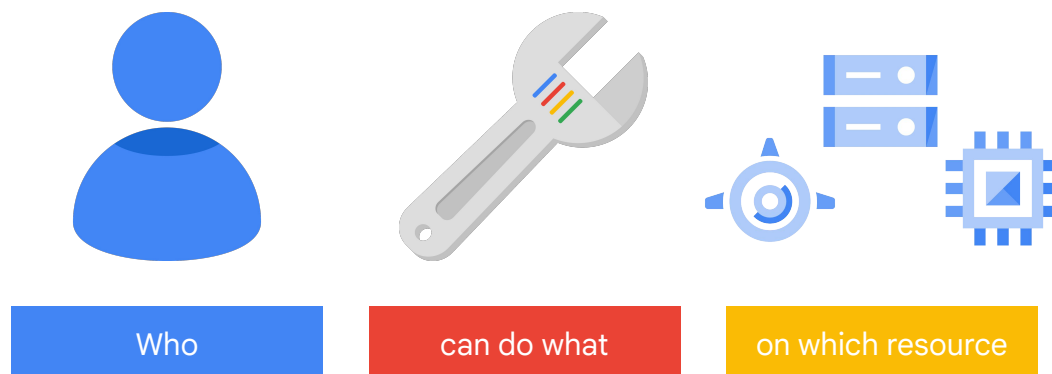
Finally, you will gain first-hand experience with IAM through a lab.

01

Identity and Access Management (IAM)

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

Identity and Access Management

| Who	can do what	on which resource

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 Google Cloud service.

For example, I 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.

IAM objects

| Organization | Folders | Projects | Resources | Roles | Members |

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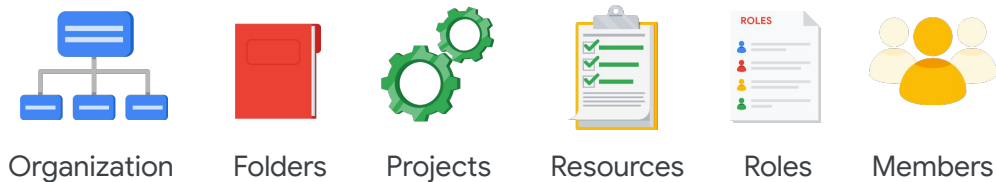
IAM is composed of different objects as shown on this slide. We are going to cover each of these in this module. To get a better understanding of where these fit in, let's look at IAM policies, and the IAM resource hierarchy.
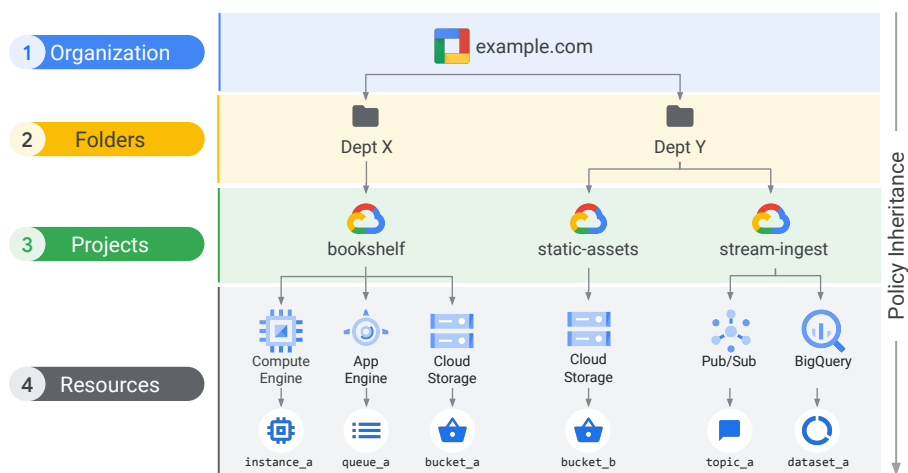
IAM resource hierarchy

Google Cloud 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.

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

The folder resource could represent your department. 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 the same default level of trust.
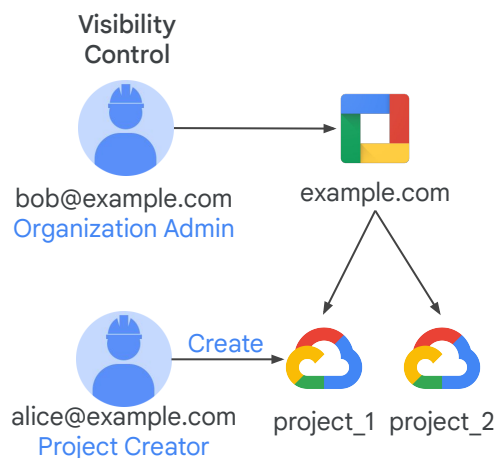
02

Organization

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

Visibility Control

bob@example.com
Organization Admin

example.com

alice@example.com
Project Creator

Create

project_1 project_2

As I 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. I am 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.

Creating and managing organizations

- Created when a **Google Workspace** or **Cloud Identity** account creates a Google Cloud Project

- **Workspace** or **Cloud Identity** super administrator:
  - Assign the **Organization admin** role to some users
  - Be the point of contact in case of recovery issues
  - Control the lifecycle of the Workspace or Cloud Identity account and Organization resource

- **Organization admin:**
  - 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

Google Cloud

The Organization resource is closely associated with a Google Workspace or Cloud Identity account.

When a user with a Workspace or Cloud Identity account creates a Google Cloud Project, an Organization resource is automatically provisioned for them. Then, Google Cloud communicates its availability to the Workspace 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 Workspace or Cloud Identity super administrators and the Google Cloud 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 Google Cloud Organization setup, the Workspace 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 Workspace 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 the how-to guide:
https://cloud.google.com/resource-manager/docs/creating-managing-organization#adding_an_organization_admin

Folders
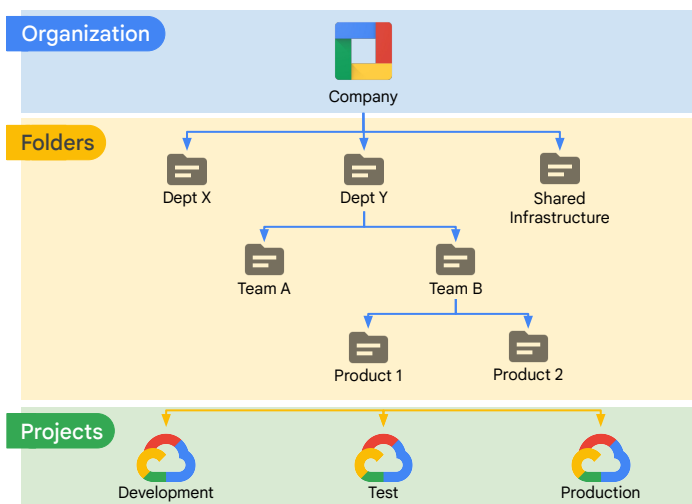
Additional grouping mechanism and isolation boundaries between projects:

- Different legal entities
- Departments
- Teams

Folders allow delegation of administration rights.

Google Cloud

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.

Resource manager roles

Organization

- **Admin:** Full control over all resources
- **Viewer:** View access to all resources

Folder

- **Admin:** Full control over folders
- **Creator:** Browse hierarchy and create folders
- **Viewer:** View folders and projects below a resource

Project

- **Creator:** Create new projects (automatic owner) and migrate new projects into organization
- **Deleter:** Delete projects

Policy Inheritance

Google Cloud

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.

03

Roles

Let's talk more about roles, which define the "can do what on which resource" part of IAM.

There are three types of IAM roles

Basic
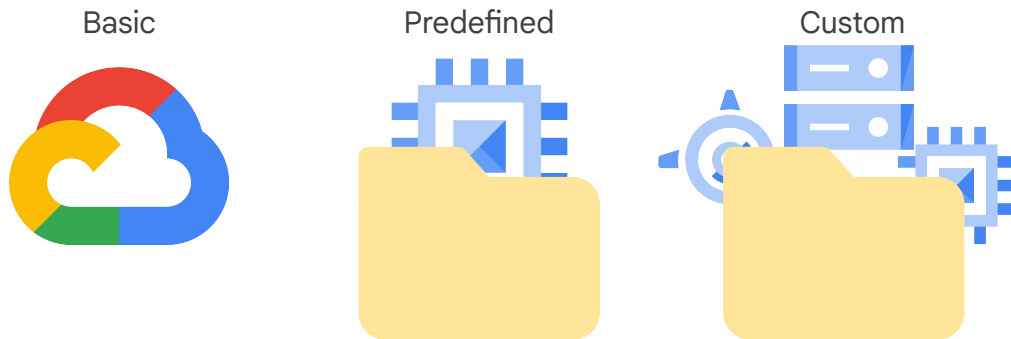
Predefined

Custom

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

IAM basic roles apply across all Google Cloud services in a project
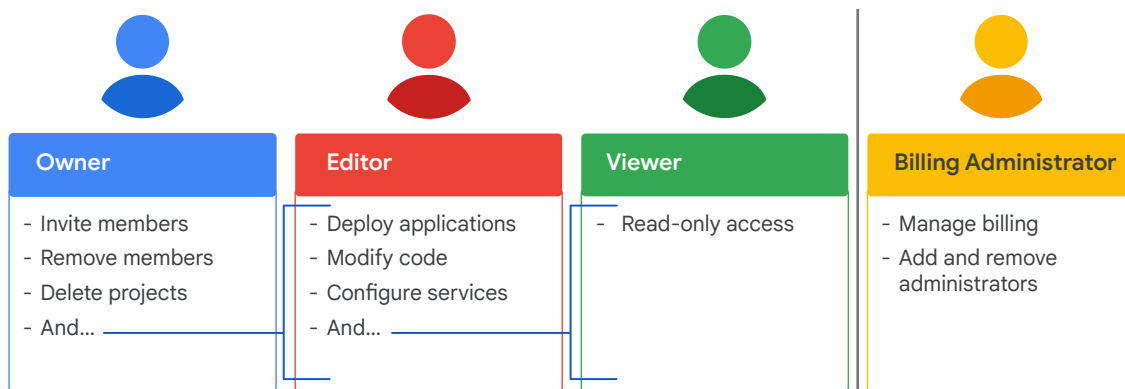

can do what

on all resources

Basic roles are the original roles that were available in the Cloud Console, but they are broad.

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

IAM **basic** roles offer fixed, coarse-grained levels of access

Owner	Editor	Viewer	Billing Administrator
- Invite members - Remove members - Delete projects - And…	- Deploy applications - Modify code - Configure services - And…	- Read-only access	- Manage billing - Add and remove administrators

Google Cloud

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

The basic 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.

IAM **predefined** roles apply to a particular Google Cloud service in a project
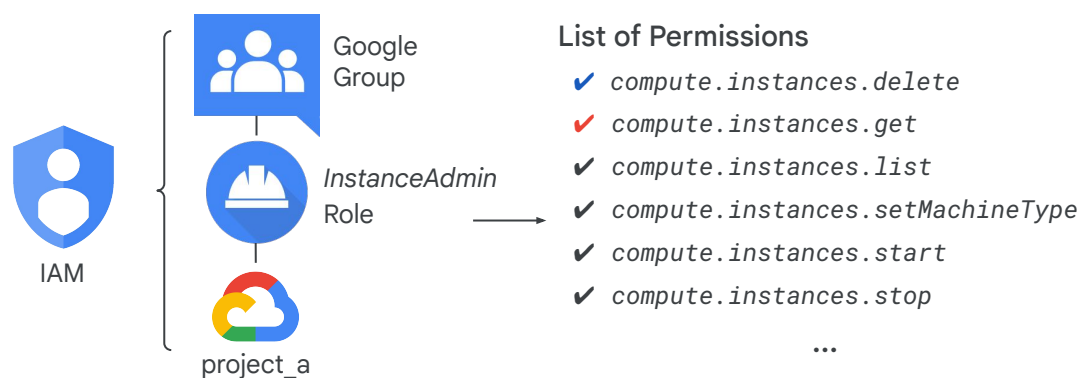
can do what

on Compute Engine resources in this project, or folder, or org

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.

IAM **predefined** roles offer more fine-grained permissions on particular services

Google Group

InstanceAdmin Role

IAM

project_a

List of Permissions

✔ `compute.instances.delete`
✔ `compute.instances.get`
✔ `compute.instances.list`
✔ `compute.instances.setMachineType`
✔ `compute.instances.start`
✔ `compute.instances.stop`

...

Google Cloud

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

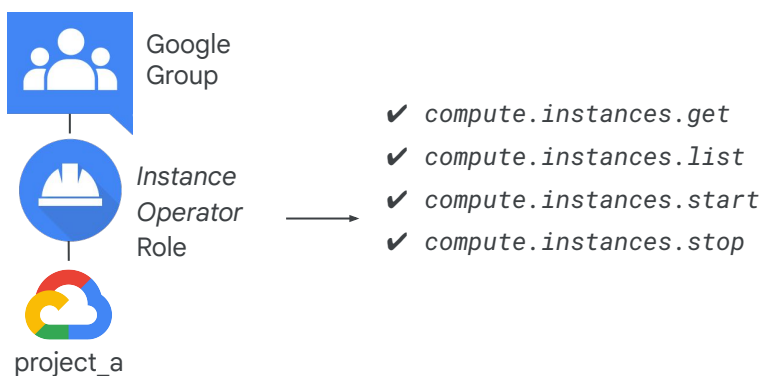Compute Engine has several predefined IAM roles. Let's look at three of those:

- 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, refer to the documentation: [https://cloud.google.com/compute/docs/access/iam#iam_roles]

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?

IAM **custom** roles let you define a precise set of permissions

Google Group

Instance Operator Role

⟶

✔ `compute.instances.get`
✔ `compute.instances.list`
✔ `compute.instances.start`
✔ `compute.instances.stop`

project_a

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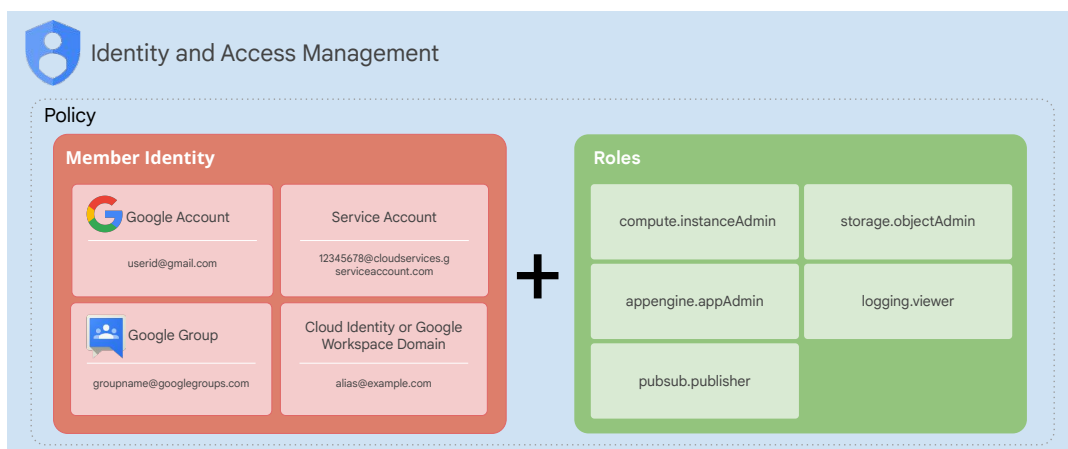
Let's say you want to define an "Instance Operator" role to allow some users to start and stop Compute Engine virtual machines, but not reconfigure them. Custom roles allow you to do that.

04

Members

Let's talk more about members, which define the "who" part of "who can do what on which resource."

Members

Identity and Access Management

Policy

Member Identity

G Google Account	Service Account
userid@gmail.com	12345678@cloudservices.g serviceaccount.com
Google Group	Cloud Identity or Google Workspace Domain
groupname@googlegroups.com	alias@example.com

Roles

compute.instanceAdmin	storage.objectAdmin
appengine.appAdmin	logging.viewer
pubsub.publisher	

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

Google Cloud

There are five different types of members: Google Accounts, Service Accounts, Google Groups, Google Workspace domains, and Cloud Identity domains.

A Google account represents a developer, an administrator, or any other person who interacts with Google Cloud. 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 Google Cloud, 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 Workspace domain represents a virtual group of all the Google accounts that have been created in an organization's Workspace account. Workspace domains represent your organization's internet domain name, such as example.com, and when you add a user to your Workspace domain, a new Google account is created for the user

inside this virtual group, such as username@example.com.

Google Cloud customers who are not Workspace 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 Workspace collaboration products such as Gmail, Docs, Drive, and Calendar.

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

[Cloud Identity: https://support.google.com/cloudidentity/answer/7319251?hl=en]

IAM policies

- A policy consists of a list of bindings.
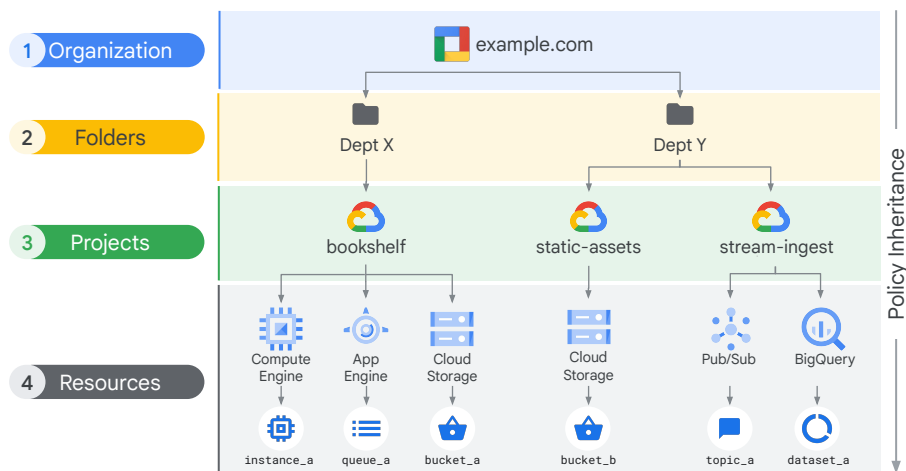
- A binding binds a list of members to a role.
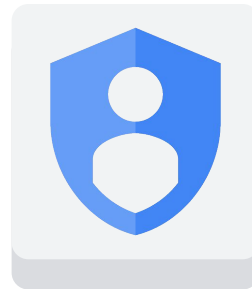
A policy consists of a list of bindings.

A binding binds a list of members to a role, where the members can be user accounts, Google groups, Google domains, and service accounts. A role is a named list of permissions defined by IAM. Let's revisit the IAM resource hierarchy.

IAM resource hierarchy

A policy is a collection of access statements attached to a resource.

Each policy contains a set of roles and role members, with resources inheriting policies from their parent. Think of it like this: resource policies are *a union of parent and resource*, where a *less* restrictive parent policy will always override a *more* restrictive resource policy.

The IAM policy hierarchy always follows the same path as the Google Cloud 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 IAM policy to inherit from the new organization's 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.

You can also use a recommender for role recommendations to identify and remove excess permissions from your principals, improving your resources' security configurations. Each role recommendation suggests that you remove or replace a role that gives your principals excess permissions. At scale, these recommendations help you enforce the principle of least privilege by ensuring that principals have only the

permissions that they actually need. Recommender identifies excess permissions using policy insights. Policy insights are ML-based findings about permission usage in your project, folder, or organization.

[Role recommendations: https://cloud.google.com/iam/docs/recommender-overview]

IAM allow policies

- Grant access to Google Cloud resources
- Controls access to the resource itself, as well as any descendants of that resource
- Associates, or binds, one or more principals (also known as a member or identity) with a single IAM role

```json
{
  "bindings": [
    {
      "members": [
        "user:jie@example.com"
      ],
      "role": "roles/resourcemanager.organizationAdmin"
    },
    {
      "members": [
        "user:raha@example.com",
        "user:jie@example.com"
      ],
      "role": "roles/resourcemanager.projectCreator"
    }
  ],
  "etag": "BwUjMhCsNvY=",
  "version": 1
}
```

Google Cloud

You can grant access to Google Cloud resources by using allow policies, also known as Identity and Access Management (IAM) policies, which are attached to resources.

The allow policy controls access to the resource itself and any descendants of that resource that inherit the allow policy.

An allow policy associates, or binds, one or more principals (also known as a member or identity) with a single IAM role and any context-specific conditions that change how and when the role is granted.

In the example on this slide, Jie (jie@example.com) is granted the Organization Admin predefined role (roles/resourcemanager.organizationAdmin) in the first role binding. This role contains permissions for organizations, folders, and limited projects operations.

In the second role binding, both Jie and Raha (raha@example.com) are granted the ability to create projects via the Project Creator role (roles/resourcemanager.projectCreator). Together, these role bindings grant fine-grained access to both Jie and Raha, and Jie is granted more access than Raha.

IAM deny policies

Deny rules prevent certain principals from using certain permissions, regardless of the roles they're granted.

Deny policies are made up of deny rules. Each deny rule specifies:

- A set of principals that are denied permissions
- The permissions that the principals are denied, or unable to use
- Optional: The condition that must be true for the permission to be denied

When a principal is denied a permission, they can't do anything that requires that permission.

Google Cloud

IAM deny policies let you set guardrails on access to Google Cloud resources.

With deny policies, you can define *deny rules* that prevent certain principals from using certain permissions, regardless of the roles they're granted.
Deny policies are made up of deny rules. Each deny rule specifies:

- A set of principals that are denied permissions
- The permissions that the principals are *denied*, or unable to use
- Optional: The condition that must be true for the permission to be denied

When a principal is denied a permission, they can't do anything that requires that permission, regardless of the IAM roles they've been granted. This is because IAM always checks relevant deny policies before checking relevant allow policies.

IAM Conditions

Enforce conditional, attribute-based access control for Google Cloud resources.

- Grant resource access to identities (members) only if configured conditions are met.

- Specified in the role bindings of a resource's IAM policy.

Google Cloud

IAM Conditions allow you to define and enforce conditional, attribute-based access control for Google Cloud resources.

With IAM Conditions, you can choose to grant resource access to identities (members) only if configured conditions are met. For example, this could be done to configure temporary access for users in the event of a production issue or to limit access to resources only for employees making requests from your corporate office.

Conditions are specified in the role bindings of a resource's IAM policy. When a condition exists, the access request is only granted if the condition expression evaluates to `true`. Each condition expression is defined as a set of logic statements allowing you to specify one or more attributes to check.

Organization policies
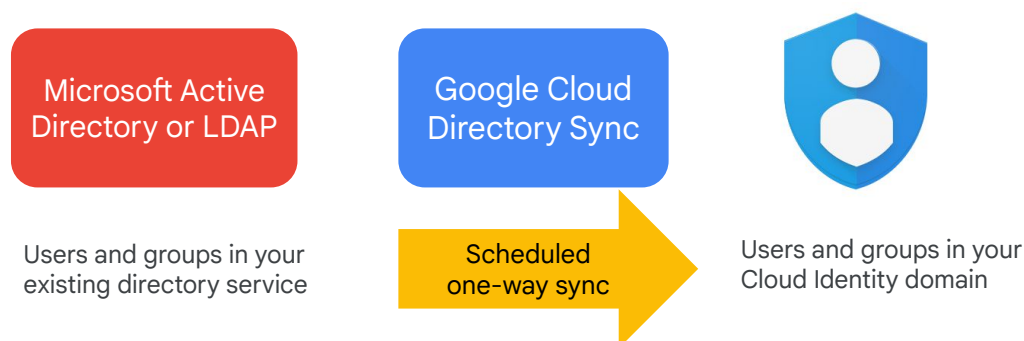
An organization policy is:

- A configuration of restrictions

- Defined by configuring a constraint with desired restrictions.

- Applied to the organization node, folders or projects.

An organization policy is a configuration of restrictions, defined by configuring a constraint with the desired restrictions for that organization.

An organization policy can be applied to the organization node, and all of its folders or projects within that node. Descendants of the targeted resource hierarchy node inherit the organization policy that has been applied to their parents.

Exceptions to these policies can be made, but only by a user who has the organization policy admin role.

What if I already have a different corporate directory?

Microsoft Active Directory or LDAP	Google Cloud Directory Sync	
Users and groups in your existing directory service	Scheduled one-way sync →	Users and groups in your Cloud Identity domain

Google Cloud

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

Using Google Cloud Directory Sync, your administrators can log in and manage Google Cloud 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).

☐ Setup SSO with third party identity provider

To setup third party as your identity provider, please provide the information below. ❓

Sign-in page URL

URL for signing in to your system and G Suite

Sign-out page URL

URL for redirecting users to when they sign out

Change password URL

URL to let users change their password in your system; when defined here, this is shown even when Single Sign-on is not enabled

Verification certificate

CHOOSE FILE No file chosen UPLOAD

The certificate file must contain the public key for Google to verify sign-in requests. ❓

☐ Use a domain specific issuer ❓

Google Cloud

Google Cloud 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 is given; otherwise, the user is prompted to sign in.
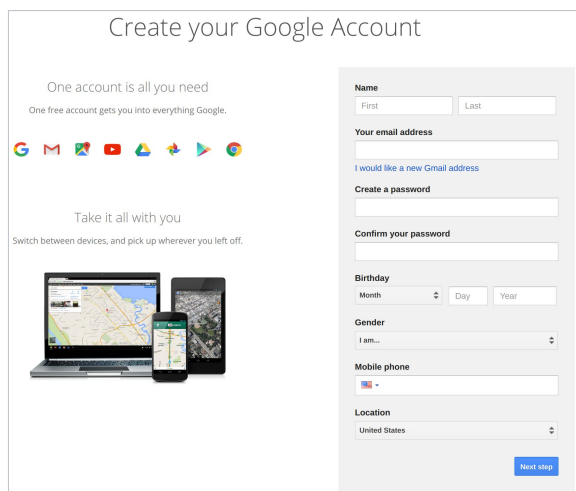
This allows you to also revoke access to Google Cloud.

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.

Additional reading: Using your existing identity management system with Google Cloud Platform

Google Cloud access without Gmail

Create your Google Account

One account is all you need

One free account gets you into everything Google.

Take it all with you

Switch between devices, and pick up wherever you left off.

Name
First | Last

Your email address

I would like a new Gmail address

Create a password

Confirm your password

Birthday
Month | Day | Year

Gender
I am...

Mobile phone

Location
United States

Next step

- 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.

Additional reading: https://accounts.google.com/SignUpWithoutGmail?hl=en

05

Service Accounts

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

Service accounts provide an identity for carrying out service-to-service 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 service-to-service 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 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 IAM roles to them, and assign the service accounts to any virtual machine instance.
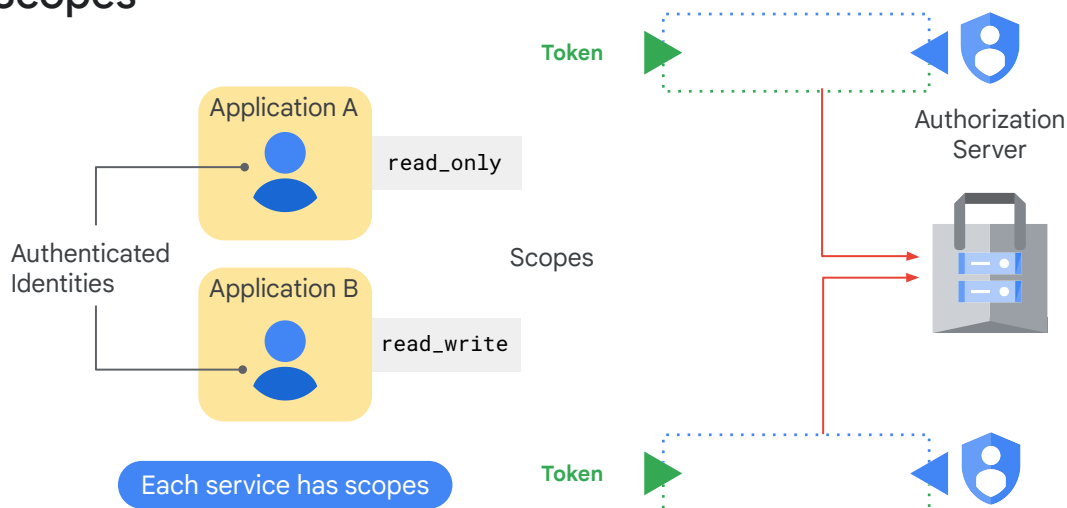
Default Compute Engine service account

- 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 the Google Cloud console

Google Cloud

Let's talk more about the default Compute Engine service account. As I 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.

Scopes

Application A

read_only

Authenticated Identities

Scopes

Application B

read_write

Each service has scopes

Token

Authorization Server

Token

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.

Proprietary + Confidential

Customizing scopes for a VM

Identity and API access

Service account
Compute Engine default service account ▾

Access scopes
○ Allow default access
○ Allow full access to all Cloud APIs
● Set access for each API

BigQuery
None

Bigtable Admin
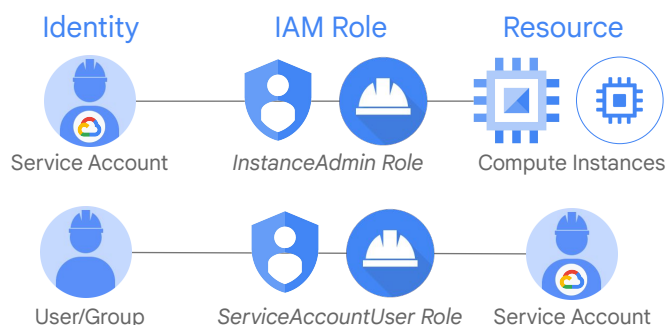None

Bigtable Data
None

Cloud Datastore
None

- Scopes can be changed after an instance is created.

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

Service account permissions

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

| Identity	IAM Role	Resource
Service Account	*InstanceAdmin Role*	Compute Instances
User/Group	*ServiceAccountUser Role*	Service Account

Google Cloud

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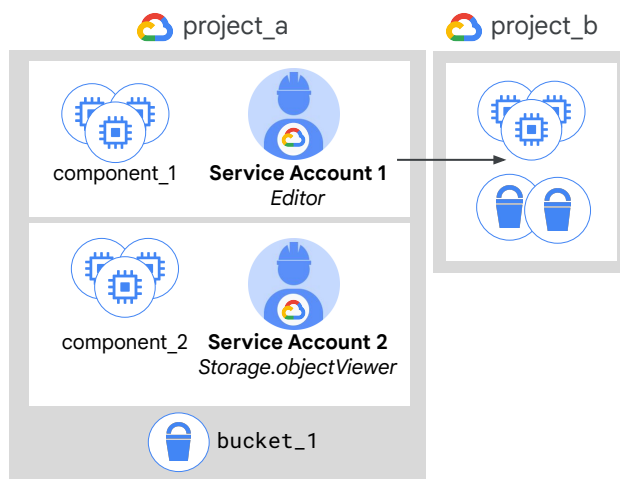
Here is another example. The VMs running component_1 are granted Editor access to project_b using Service Account 1. VMs running component_2 are granted objectViewer access to bucket_1 using an isolated Service Account 2. This way you can scope permissions for VMs without re-creating VMs.

Essentially, 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 Google Cloud manages security for you.

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

Two types of service account keys

Google-managed service accounts

- All service accounts have Google-managed keys.
- Google stores both the public and private portion of the key.
- Each public key can be used for signing for a maximum of two weeks.
- Private keys are never directly accessible.

User-managed service accounts

- Google only stores the public portion of a user-managed key.
- Users are responsible for private key security.
- Can create up to 10 user-managed service account keys per service.
- Can be administered via the IAM API, gcloud, or the console.

Google Cloud

There are two types of service account keys.

By default, when using service accounts within Google Cloud (for example, from Compute Engine or App Engine) Google automatically manages the keys for service accounts. However, if you want to be able to use service accounts outside of Google Cloud, or want a different rotation period, it is possible to also manually create and manage your own service account keys.

Two types of service account keys

Google-managed service accounts

- All service accounts have Google-managed keys.
- Google stores both the public and private portion of the key.
- Each public key can be used for signing for a maximum of two weeks.
- Private keys are never directly accessible.

User-managed service accounts

- Google only stores the public portion of a user-managed key.
- Users are responsible for private key security.
- Can create up to 10 user-managed service account keys per service.
- Can be administered via the IAM API, gcloud, or the console.

Google Cloud

All service accounts have Google-managed key-pairs.

With Google-managed service account keys, Google stores both the public and private portion of the key, and rotates them regularly.

Each public key can be used for signing for a maximum of two weeks.

Your private key is always held securely in escrow and is never directly accessible.

Two types of service account keys

Google-managed service accounts	User-managed service accounts
• All service accounts have Google-managed keys.	• Google only stores the public portion of a user-managed key.
• Google stores both the public and private portion of the key.	• Users are responsible for private key security.
• Each public key can be used for signing for a maximum of two weeks.	• Can create up to 10 user-managed service account keys per service.
• Private keys are never directly accessible.	• Can be administered via the IAM API, gcloud, or the console.

Google Cloud

You may optionally create one or more user-managed key pairs (also known as "external" keys) that can be used from outside of Google Cloud. Google only stores the public portion of a user-managed key.

The User is responsible for security of the private key and performing other management operations such as key rotation, whether manually or programmatically.

Users can create up to 10 service account keys per service account to facilitate key rotation.

User-managed keys can be managed by using the IAM API, the gcloud command-line tool, or the Service Accounts page in the Google Cloud console.

Keeping your user-managed keys safe is vital - and is the creator's responsibility

! **Remember:** Google does not save your user-managed private keys - if you lose them, Google cannot help you recover them.

Google does not save your user-managed private keys, so if you lose them, Google cannot help you recover them.

You are responsible for keeping these keys safe and also responsible for performing key rotation.

User-managed keys should be used as a last resort. Consider the other alternatives, such as short-lived service account credentials (tokens), or service account impersonation.

Use the `gcloud` command-line tool to quickly list all of the keys associated with a Service Account

```
gcloud iam service-accounts keys list --iam-account user@email.com
```

The `gcloud` command line shown on this slide is a fast and easy way to list all of the keys associated with a particular service account.

06

Organization Restrictions

Let's talk now about Organization Restrictions.

Organization Restrictions let you prevent data exfiltration through phishing or insider attacks.

The **Organization Restrictions** feature lets you prevent data exfiltration through phishing or insider attacks.

For managed devices in an organization, the Organization Restrictions feature restricts access only to resources in authorized Google Cloud organizations.

Organization Restrictions
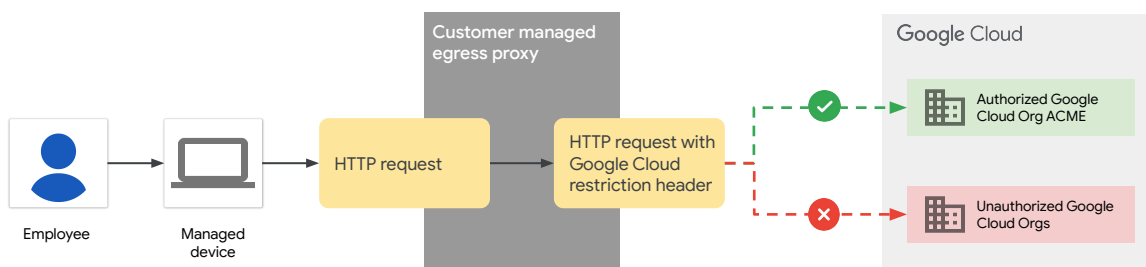
There is a need in organizations to restrict access of their employees only to resources in authorized Google Cloud organizations. Google Cloud administrators who administer Google Cloud, and egress proxy administrators, who configure the egress proxy, engage together to set up organization restrictions.

The managed device is governed by the organizational policies of a company. Employees of an organization use a managed device to access the organization resources. An egress proxy administrator configures the proxy to add organization restrictions headers to any requests originating from a managed device. This proxy configuration prevents users from accessing any Google Cloud resources in non-authorized Google Cloud organization. The Organization Restrictions feature in Google Cloud inspects all requests for organization restrictions header, and allows or denies the requests based on the organization being accessed.

Organization Restrictions can be used to restrict access to employees in your organization so that employees can access resources only in your Google Cloud organization and not other organizations. They can also be used to allow your employees to read from Cloud Storage resources but restrict employee access only to resources in your Google Cloud organization. Or, allow your employees to access a vendor Google Cloud organization in addition to your Google Cloud organization.

07

IAM Best Practices

Let's talk about some 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 IAM policy.
- Audit membership of groups used in policies.
- Control the ownership of the Google group used in IAM policies.

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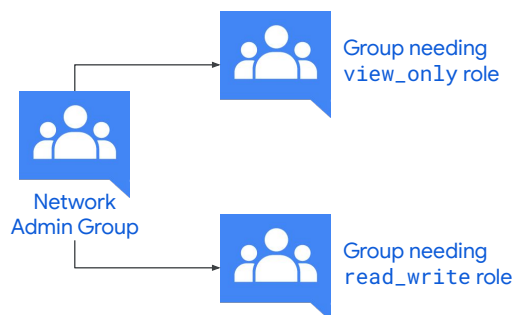
Next, I recommend granting roles to groups instead of individuals. This allows you to update group membership, instead of changing an IAM policy. If you do this, make sure to audit membership of groups used in policies and control the ownership of the Google group used in IAM policies.

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.

Identity-Aware Proxy (IAP)

Enforce access control policies for applications and resources:

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

IAM policy is applied after authentication.


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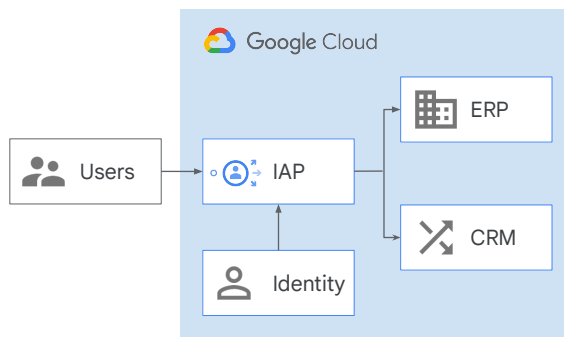
Finally, I recommend using Identity-Aware proxy, or IAP. IAP lets you establish a central authorization layer for applications accessed by HTTPS, so you can use an application-level access control model instead of relying on network-level firewalls.

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

Additional reading: [Identity-Aware Proxy overview | Google Cloud](#)

Lab Intro

Exploring IAM
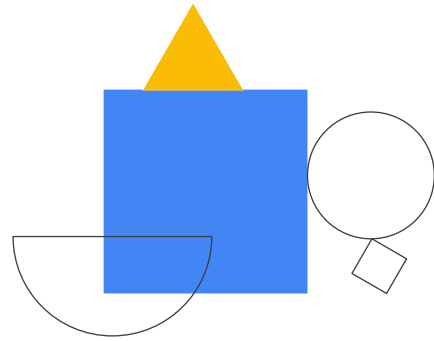
It's time to apply what you learned.

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.

Lab Review

Exploring IAM
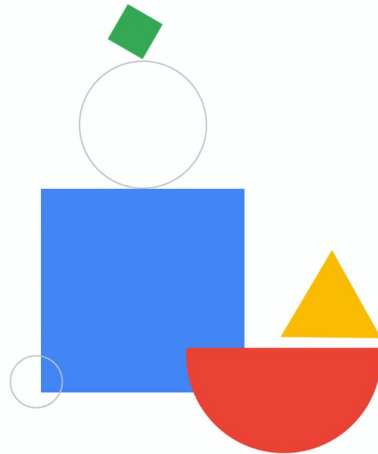
In this lab, you granted and revoked IAM roles, first to a user, Username 2, and then to a Service Account User. Having access to both users allowed you to see the results of the changes you made.

You can stay for a lab walkthrough, but remember that GCP's user interface can change, so your environment might look slightly different.

Review: Identity and Access Management
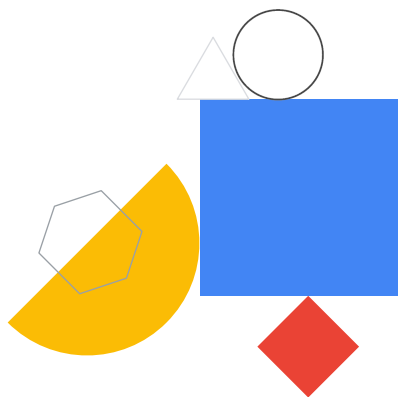
In this module, we covered Identity and Access Management along with its components and best practices. IAM builds on top of other Google Cloud identity services.

The creation and administration of corporate identities occurs through the Workspace admin or Cloud Identity interface, and is commonly handled by a person separate from the Google Cloud 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 Workspace 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.