Identity and Access Management (IAM)

Welcome to module 3 of Managing Security in Google Cloud—**Identity and Access Management**.

Identity and Access Management (or IAM as it is known) lets administrators authorize who can take actions on specific resources, giving you full control and visibility to manage your cloud resources centrally.
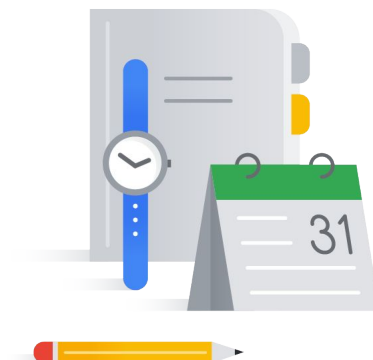
Module overview

Resource Manager

IAM roles

Service accounts

Workload identity federation

IAM & Organization policies

Policy Intelligence

IAM best practices

Google Cloud

More specifically, we will cover the Resource Manager which enables you to centrally manage projects, folders, and organizations.

We will then cover IAM roles and service accounts.

We will then cover Workload Identity Federation, which allows you to grant on-premises or multi-cloud workloads access to Google Cloud resources, without using a service account key.

We will continue with IAM and organization policies.

After this, we will cover Policy Intelligence, which helps you understand and manage your policies to proactively improve your security configuration.

We will end the module with best practices and a lab, where you will configure IAM to grant roles and create custom roles.

Let's get started!

Identity and Access Management (IAM)
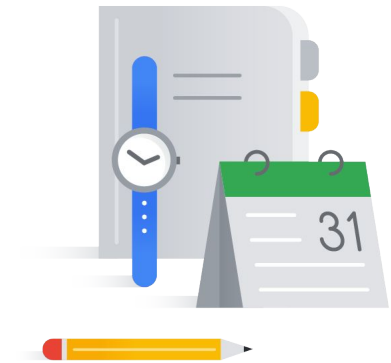
Resource Manager

IAM roles

Service accounts

Workload identity federation

IAM & Organization policies

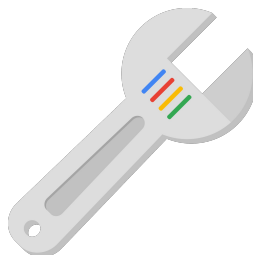Policy Intelligence

IAM best practices

OK, let's dive into IAM and how to centrally manage your resources with the Resource Manager.

Identity and Access Management

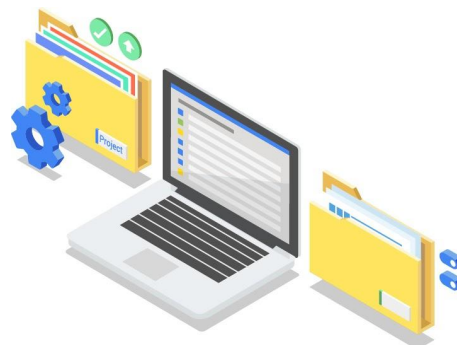| Who	can do what	on which resource
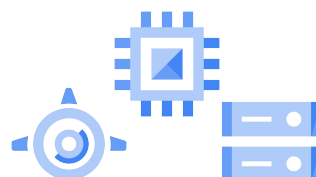
Google Cloud

IAM lets administrators authorize who can do what on which resources in Google Cloud. It provides full control and visibility to manage cloud resources centrally.

Resource Manager

- Resources in Google Cloud are hierarchically managed by organization, folders, and projects.

- Resources Manager enables you to programmatically manage these resource containers.

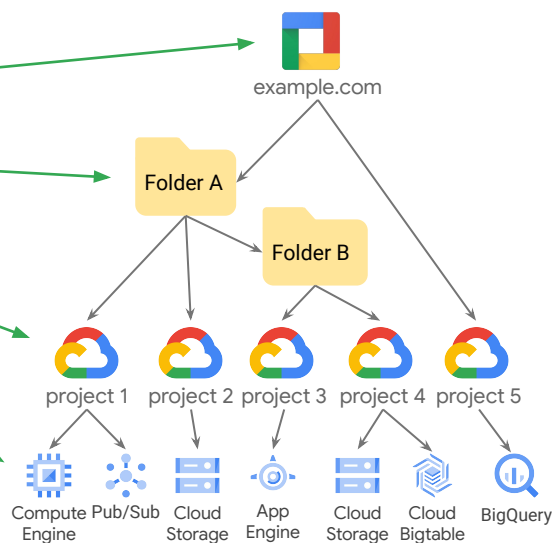Google Cloud

---

Google Cloud provides resource containers such as Organizations, Folders, and Projects, which allow you to group and hierarchically organize cloud resources. This hierarchical organization lets you easily manage common aspects of your resources, like access control and configuration settings.

The Resource Manager enables you to programmatically manage these resource containers.

There are several objects that are important when discussing IAM in Google Cloud.

These objects are:

- Organization
- Folders
- Projects
- Resources
- Members and
- Roles

These objects together form a resource hierarchy that can be managed using the Resource Manager.

This Google Cloud resource hierarchy allows you to map your organization onto appropriate Google Cloud objects and presents logical attach points for access management policies.

Organization node

The organization node:

- Is the root node for Google Cloud resources

- Contains all of your projects and resources

example.com

Folder A

Folder B

project 1   project 2   project 3   project 4   project 5

Compute Engine   Pub/Sub   Cloud Storage   App Engine   Cloud Storage   Cloud Bigtable   BigQuery
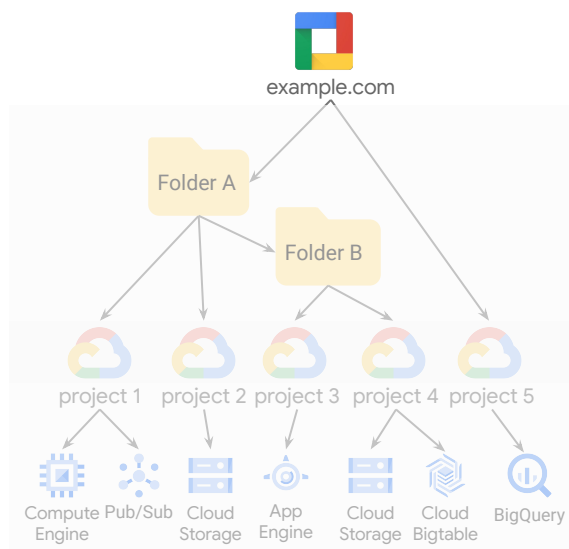
Google Cloud

The organization node is the root node for Google Cloud resource hierarchy. It is the "super node" for all of your projects and resources and represents your organization.

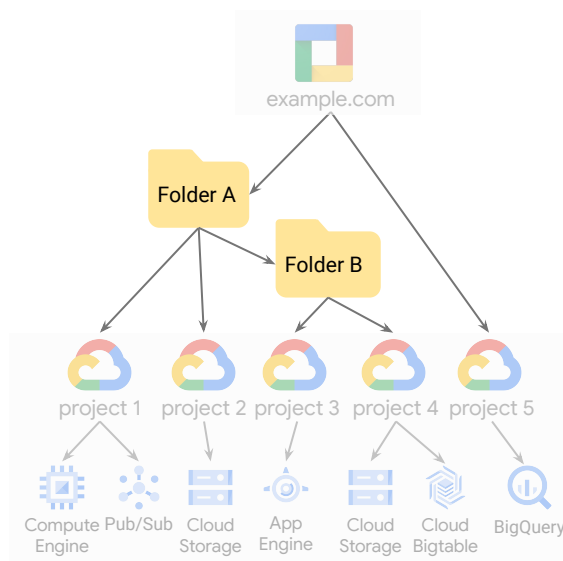# Folders offer flexible management

Folders:

- Optionally group projects under an Organization.

- Can contain both projects and other folders.

example.com

Folder A

Folder B

project 1 project 2 project 3 project 4 project 5

Compute   Pub/Sub Cloud   App   Cloud   Cloud   BigQuery
Engine   Storage   Engine   Storage Bigtable

Google Cloud

Folders can be used to implement organizational structure and/or group projects by department, team, application or environment.

A folder can contain projects, other folders, or a combination of both.

Organizations can use folders to group projects under the organization node in a hierarchy. For example, your organization might contain multiple departments, each with its own set of Google Cloud resources.

Folders allow you to group these resources on a per-department basis. While a folder can contain multiple child folders or other resources, each folder or resource can only have exactly one parent.

Folders offer flexible management

Use folders to assign policies; changes will apply across all the projects and resources.

example.com

Folder A

Folder B

project 1   project 2   project 3   project 4   project 5

Compute   Pub/Sub   Cloud   App   Cloud   Cloud   BigQuery
Engine   Storage   Engine   Storage   Bigtable
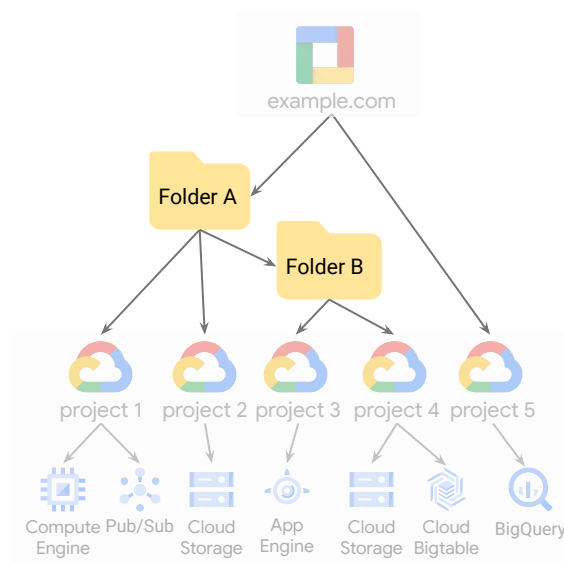
Google Cloud

Folders are used to group resources that share common IAM policies. You can use folder-level IAM policies to control access to the resources the folder contains.
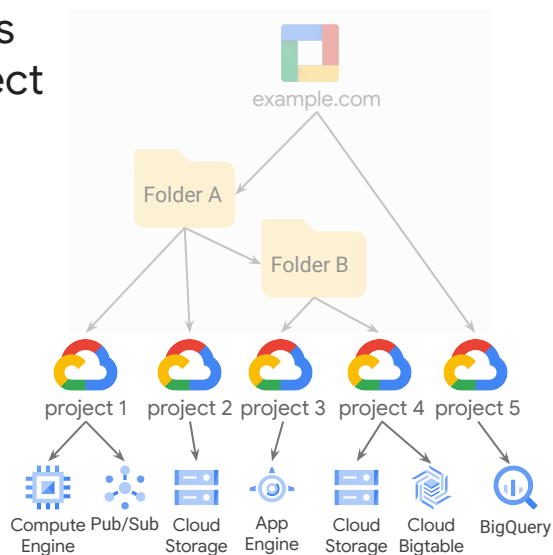
For example, if a user is granted the Compute Instance Admin role on a folder, that user has the Compute Instance Admin role for all of the projects in the folder.

You can also use deny policies in combination with roles, to restrict access to resources in a folder.

It is important to note that the use of folders to organize resources is **optional**.

Projects, however, are required in Google Cloud and any resource that is deployed must be associated with a project.

Projects provide many management-related features, such as the ability to:

- Track resource and quota usage.
- Assign projects to different billing accounts.
- Assign manager permissions and credentials and selectively enable specific services and APIs at the project level.

Sample hierarchy
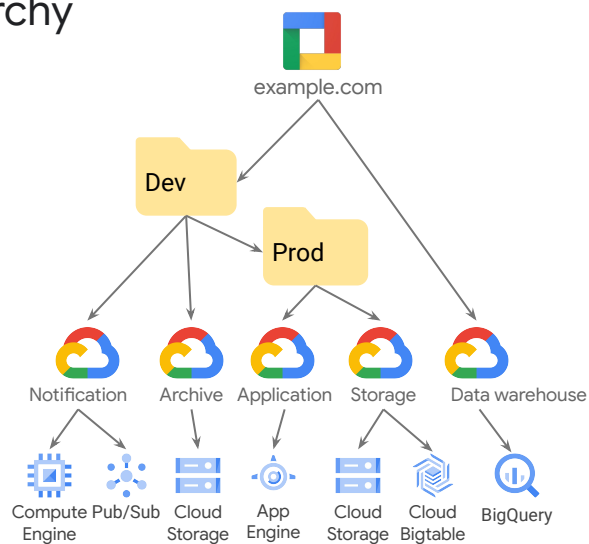
The following slide demonstrates a sample organization—example.com—that has two folders: one for development and one for production.

Each folder contains projects that contain Google Cloud resources.

Identity and Access Management (IAM)

Resource Manager

IAM roles

Service accounts

Workload identity federation

IAM & Organization policies

Policy Intelligence

IAM best practices
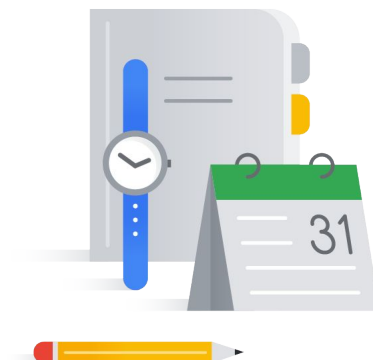
Google Cloud

Now that you've learned a little bit about resource manager, let's dive into Cloud IAM roles.

In Google Cloud, you can grant permissions by granting roles.

In this section we will first review, and then take a more in-depth look at, the different types of roles.

There are three kinds of IAM roles in Google Cloud

Basic **Predefined** **Custom**
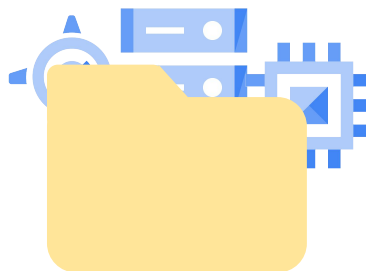
There are three kinds of roles in IAM:

- *Basic roles*: The roles that have been historically available in the Cloud Console. These roles existed prior to the introduction of IAM.
- *Predefined roles*: Also sometimes called "curated roles," are the IAM roles that give finer-grained access control than the basic roles. Each Google Cloud service offers a set of predefined roles.
- *Custom roles*: You can define roles consisting of permissions and resources of your choice.

IAM basic roles are applied at the project level

can do what	on all resources

Basic roles offer fixed, coarse-grained levels of access.

The IAM basic roles are applied at the project or service resource levels and control access to **all** resources in that project or resource.

The level of access these provide is very coarse-grained and that is why they are called basic roles.

They control what can be done on all resources in a project.

Basic roles apply across all Google Cloud services in a project

Owner
- Invite members
- Remove members
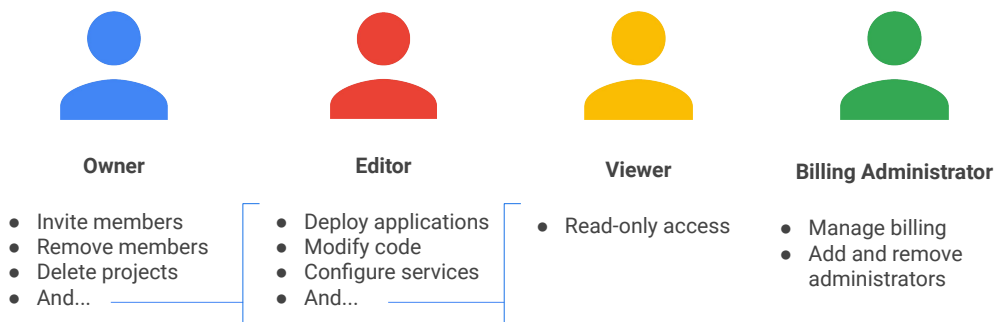- Delete projects
- And...

Editor
- Deploy applications
- Modify code
- Configure services
- And...

Viewer
- Read-only access

Billing Administrator
- Manage billing
- Add and remove administrators

A project can have multiple owners, editors, viewers, and billing administrators.

Google Cloud

There are three basic roles: Owner, Editor, and Viewer. These roles are concentric; that is, the Owner role includes the permissions in the Editor role, and the Editor role includes the permissions in the Viewer role.

The viewer role, as its name implies, provides view or read-only access to a project and all its resources.

The editor role provides the ability to modify or edit all resources in the project, as well as all the read-only access from the viewer role.

The owner role provides the ability to manage the project itself, such as deleting the project, and adding or removing other members to the project, as well as all the editor role permissions plus the read-only access from the viewer role.

The Billing Administrator is an owner role for a billing account. Use it to manage payment instruments, configure billing exports, view cost information, link and unlink projects and manage other user roles on the billing account.

___
IAM predefined roles

Predefined roles are designed to map to job functions:
Compute Network Admin, Security Reviewer, etc.

| can do what | on Compute Engine resources in this project, or folder, or org |

As you have seen, basic roles are coarse-grained and are applied at the project level. Often items, these roles provide unnecessary access to Google Cloud services and resources, which can pose a Security risk to your cloud environment.

Predefined roles on the other hand provide granular access for a specific service. They are designed to map to job functions, for example, Compute Network Admin, Security Reviewer, Storage Admin, etc.

Predefined roles are managed by Google Cloud. So if a new feature or service is added in the future, the appropriate permissions will be added to any predefined role that requires them.

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

InstanceAdmin Role →

- ✓ *compute.instances.delete*
- ✓ *compute.instances.get*
- ✓ *compute.instances.list*
- ✓ *compute.instances.setMachineType*
- ✓ *compute.instances.start*
- ✓ *compute.instances.stop*

. . .

Google Cloud

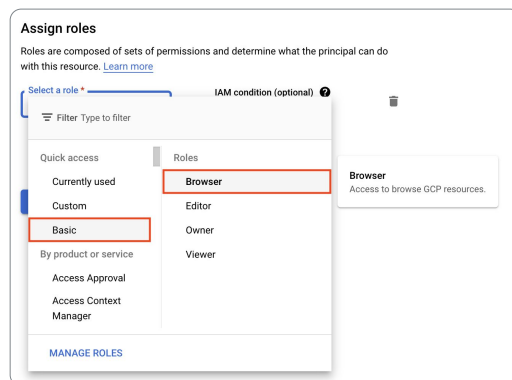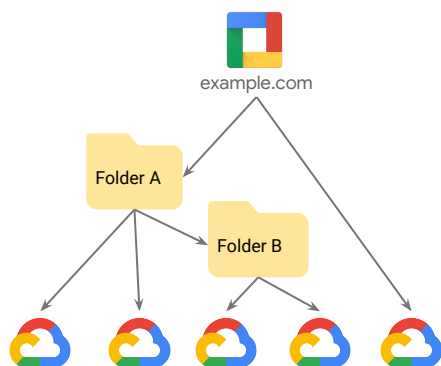A predefined role is simply a collection of permissions for a particular service. For example, the *InstanceAdmin* predefined role provides the permissions needed to manage Compute Engine instances.

An example of some of the permissions inherent in this role are shown here on the slide.

As you can see, predefined roles give granular access to specific Google Cloud resources and prevent unwanted access to other resources.

The predefined **Browser** role

This role provides read access to browse the hierarchy for a project, including the organization and folders.

example.com

Folder A

Folder B

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. Learn more

Select a role *   IAM condition (optional) ?

Filter Type to filter

Quick access	Roles
Currently used	Browser
Custom	Editor
Basic	Owner
By product or service	Viewer
Access Approval	
Access Context Manager	

Browser
Access to browse GCP resources.

MANAGE ROLES

The predefined Browser role provides read-access to browse the **hierarchy** for a project, including the folder, organization, and IAM policy.

The Browser role does not include permission to view **resources** in the project.

IAM custom roles let you define a precise set of permissions

Google Group

PrivacyReviewer Role

✓ *storage.buckets.list*
✓ *storage.objects.list*
✓ *storage.objects.get*
✓ *spanner.databases.get*
✓ *spanner.databases.list*
✓ *bigtable.tables.get*
✓ *bigtables.tables.list*
✓ *bigtable.tables.readRows*

. . .

Google Cloud
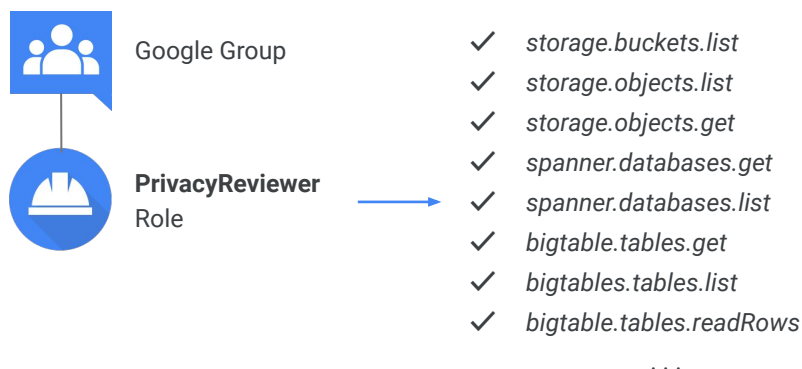
What if you need something even finer-grained?

This is when you might use a Custom role, which will allow you to map specific permissions to specific job roles. For example, maybe you need to define a "Privacy Reviewer" role, to allow some users the ability to audit data that is stored in Google Cloud Storage, Cloud Spanner, Cloud Bigtable, and other data repositories.

You can create a Custom role which contains **all** of the specific permissions needed to do that particular job—and **only** those permissions. Be aware that once Custom roles are created, you must manage the permissions granted for them.

If, for example, a new data storage service is created in the future that will need to be audited, permissions for that new service would need to be added to your Privacy Reviewer role.

Identity and Access Management (IAM)
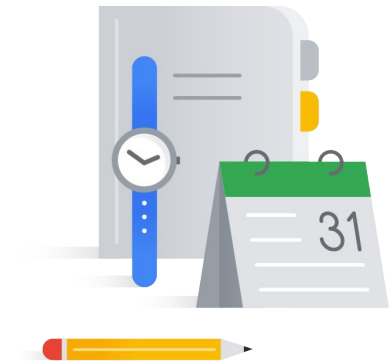
Resource Manager

IAM roles

Service accounts

Workload identity federation

IAM & Organization policies

Policy Intelligence

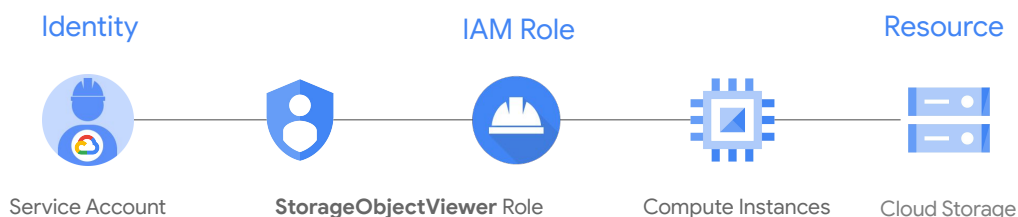IAM best practices

Google Cloud

Now let's talk about service accounts.

Service accounts

Service accounts control server-to-server interactions:

- Used to authenticate from one service to another
- Used to control privileges used by resources

Identity	IAM Role		Resource
Service Account	**StorageObjectViewer** Role	Compute Instances	Cloud Storage

Google Cloud

In addition to the members already mentioned, you can also grant roles to service accounts.

Service Accounts control server-to-server interactions and are used to authenticate from one service to another and control what actions applications running on a service can perform.

For example, if an application running on a Compute Engine instance needs to read a file from Cloud Storage, a service account with Cloud Storage Object Viewer role can be assigned to the Compute Engine instance.

An application running on that instance would then be permitted to read a file from Cloud Storage.

Service accounts are identified with a Google-managed email address in the `gserviceaccount.com` domain.

There are two types of Google service accounts

Google-managed service accounts

- Google creates and manages service accounts for many Google Cloud services.

- Google managed service accounts are not listed in the Service accounts page.

- Accounts are not listed in your project and you cannot access them directly.

User-managed service accounts

- Service accounts that you create in your projects.

- You can update, disable,enable, and delete these service accounts.

- Can be administered via the IAM API, Google Cloud CLI, or the Google Cloud console.

- Can create up to 100 user-managed services accounts in a project.

Google Cloud

There are two types of Google Service Accounts: Service accounts that Google manages, and service accounts that you manage.

There are two types of Google service accounts

Google-managed service accounts

- Google creates and manages service accounts for many Google Cloud services.

- Google managed service accounts are not listed in the Service accounts page.

- Accounts are not listed in your project and you cannot access them directly.

User-managed service accounts

- Service accounts that you create in your projects.

- You can update, disable,enable, and delete these service accounts.

- Can be administered via the IAM API, Google Cloud CLI, or the Google Cloud console.

- Can create up to 100 user-managed services accounts in a project.

Google Cloud

Some Google Cloud services need access to your resources so that they can act on your behalf. For example, when you use Cloud Run to run a container, the service needs access to any Pub/Sub topics that can trigger the container.

To meet this need, Google creates and manages service accounts for many Google Cloud services. These service accounts are known as Google-managed service accounts. You might see Google-managed service accounts in your project's allow policy, in audit logs, or on the IAM page in the Google Cloud console.

Google-managed service accounts aren't created in your projects, so you won't see them when viewing your projects' service accounts.

And because accounts are not listed in your project, you won't be able to access them directly.

There are two types of Google service accounts

Google-managed service accounts

- Google creates and manages service accounts for many Google Cloud services.

- Google managed service accounts are not listed in the Service accounts page.

- Accounts are not listed in your project and you cannot access them directly.

User-managed service accounts

- Service accounts that you create in your projects.

- You can update, disable,enable, and delete these service accounts.

- Can be administered via the IAM API, Google Cloud CLI, or the Google Cloud console.

- Can create up to 100 user-managed services accounts in a project.

Google Cloud

User-managed service accounts are service accounts that you create in your projects.

You can update, disable, enable, and delete these service accounts at your discretion. You can also manage other principals' access to these service accounts.

You can create user-managed service accounts in your project using the IAM API, the Google Cloud console, or the Google Cloud CLI.

By default, you can create up to 100 user-managed service accounts in a project.

Two types of service account keys

Google-managed account keys

- 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 account keys

- Google only stores the public portion of a user-managed key and does not save your user-managed private keys.

- Users are responsible for private key security. If you lose them, Google cannot help you recover them.

- Can create up to 10 user-managed service account keys per service.

- Can be administered via the IAM API, gcloud, or the Google Cloud console.

Google Cloud

Unlike normal users, service accounts do not have passwords. Instead, service accounts use RSA key pairs for authentication: If you know the private key of a service account's key pair, you can use the private key to create a JWT bearer token and use the bearer token to request an access token. The resulting access token reflects the service account's identity and you can use it to interact with Google Cloud APIs on the service account's behalf.

Because the private key lets you authenticate as the service account, having access to the private key is similar to knowing a user's password.

The private key is known as a service account key. The key pairs used by service accounts fall into two categories, Google-managed and user-managed.

Two types of service account keys

Google-managed account keys

- 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 account keys

- Google only stores the public portion of a user-managed key and does not save your user-managed private keys.

- Users are responsible for private key security. If you lose them, Google cannot help you recover them.

- Can create up to 10 user-managed service account keys per service.

- Can be administered via the IAM API, gcloud, or the Google Cloud 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 account keys

- 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 account keys

- Google only stores the public portion of a user-managed key and does not save your user-managed private keys.

- Users are responsible for private key security. If you lose them, Google cannot help you recover them.

- Can create up to 10 user-managed service account keys per service.

- Can be administered via the IAM API, gcloud, or the Google Cloud 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. If you lose them, Google cannot help you recover them.

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.

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 service-account-email-id
```

Tip: 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.

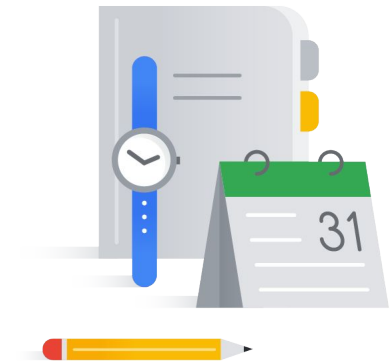# Identity and Access Management (IAM)

Resource Manager

IAM roles

Service accounts

Workload identity federation

IAM & Organization policies

Policy Intelligence

IAM best practices

Google Cloud

Let's discuss workload identity federation.

Workload identity federation - overview

- Grant apps running outside of Google Cloud access to data without service account keys

- Problem: service account keys are powerful
  - Security risk if not managed correctly

- Solution: grant external identities IAM roles
  - Impersonate service accounts and access resources
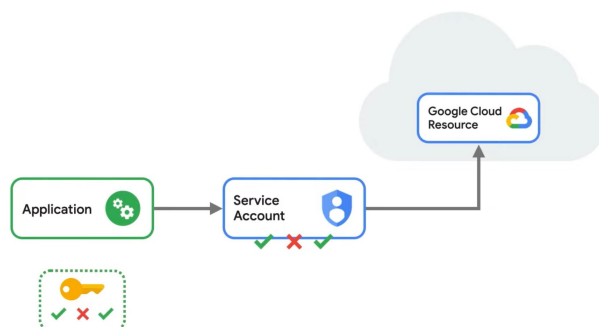
Google Cloud

Using identity federation, you can grant on-premises or multi-cloud workloads to access data stored in Google Cloud services such as GCS without service account keys.

Traditionally, applications running outside Google Cloud have used service account keys to access Google Cloud resources. Service account keys are powerful credentials, and can represent a security risk if they are not managed correctly.

Service account keys are akin to user passwords, allowing the holder to act as the service account and gain access to any resource that service account has access to.

Unfortunately, there's no way to verify that the application holding the key has permission to use it. It's a key without an expiration date and with no guarantee around where it's stored or who has access to it.

Because of this risk, managing the storage, distribution, and rotation of service account keys becomes a top priority, effectively turning an identity management problem into a secrets management problem.

So what's the solution?

Ditch the keys. With workload identity federation, you can use Identity and Access Management (IAM) to grant external identities IAM roles, including the ability to impersonate service accounts.

This lets you access resources directly, using temporary credentials—or "tokens"—and eliminates the maintenance and security burden associated with service account keys.

Workload identity federation - how it works

1. Create a workload identity pool in your Google Cloud project

2. App authenticates with identity provider → receives account credentials

3. App calls security token service → get short-lived Google Cloud access token

4. Use token to impersonate service account → access Google Cloud resources
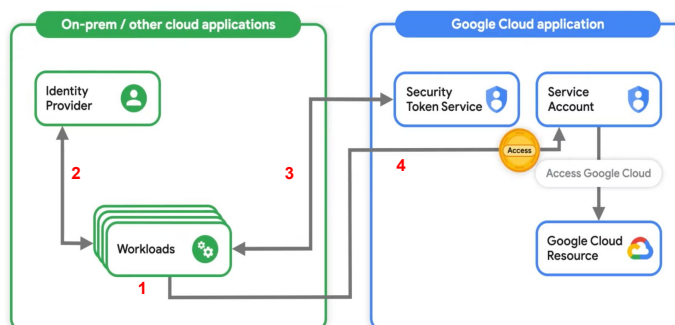
Google Cloud

To set up workload identity federation, you'll first need to create a workload identity pool in your Google Cloud project. (1)

Fortunately, you don't need to be a super admin to do this. You only need permission to manage workload identity pools, service accounts, and IAM policies at the project level.

A workload identity pool allows you to organize and manage external identities. A project can have multiple pools with each one allowing access from a different external identity provider.

You'll need to create an IAM policy that allows identities in the workload identity pool to impersonate the service account. This allows you to create collections of identities and easily control the permissions granted to identities from each identity provider.

To learn more about identity pools, check out the link in the speaker notes:
- **Link:** cloud.google.com/iam/docs/workload-identity-federation#pools

After you have your identity pool set up, your application authenticates to your identity provider and receive account credentials. (2)

The application can then call our security token service to exchange the account credentials issued by your identity provider for a short-lived Google Cloud access token. (3)

This access token can then be used to impersonate a service account, inheriting the permissions of that service account to access Google Cloud resources.

Then, configure a one-way trust between your identity provider and the workload identity pool by providing relevant metadata about your provider. (4)

Now, when an application attempts to exchange their IDP credential, the security token service will be able to validate that the credential is from a trusted provider before issuing an access token to the application.

We've mitigated a security risk while maintaining the ability to reliably make Google Cloud API requests.

To learn more about how workload identity federation can help you better protect your Google Cloud access, check out the video link in the speaker notes:
- **Link:** www.youtube.com/watch?v=4vajaXzHN08&t=1s

Identity and Access Management (IAM)
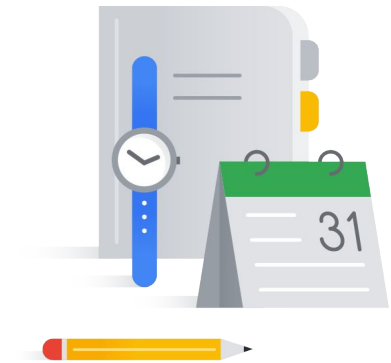
Resource Manager

IAM roles

Service accounts

Workload identity federation
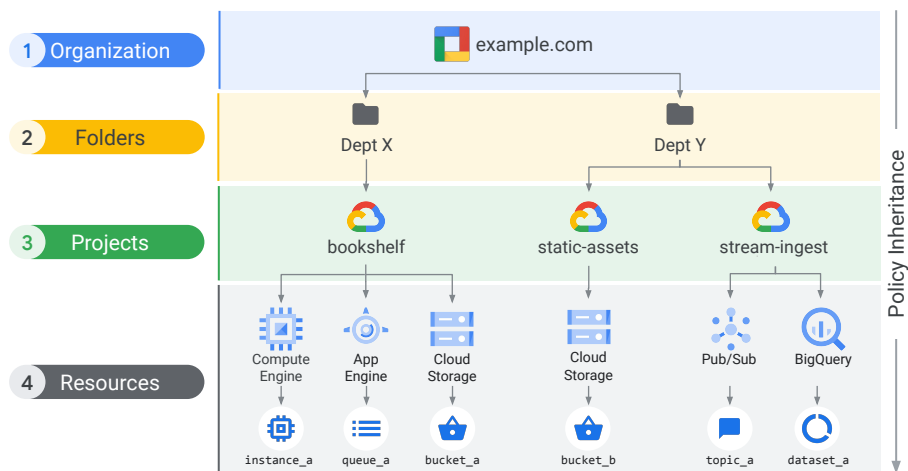
IAM & Organization policies

Policy Intelligence

IAM best practices

Google Cloud

Let's now move on and discuss IAM policies.

IAM policy 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.

In addition to policies which grant access, there are also deny policies that can be used to restrict access. We will talk about deny policies in a moment.

Another way to express the hierarchy of policies is: "allowed permissions are always inherited down the resource hierarchy unless overridden by a deny policy."

IAM 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

```
{
  "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, as well as 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

Let you 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.

Google Cloud

Now let's talk about IAM deny policies. IAM deny policies let you define deny rules that prevent certain principals from using certain permissions, regardless of the roles they're granted.

Each deny rule specifies:
- A set of principles that are denied permissions,
- The permissions that the principals are denied, or unable to use,
- And optionally, 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 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.

IAM Conditions

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

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

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

- Condition attributes
  - Resource attributes
  - Request attributes

Allow access only to Cloud Storage buckets whose names start with a specified prefix

```
resource.type ==
"storage.googleapis.com/Bucket" &&
resource.name.startsWith("projects/_
/buckets/exampleco-site-assets-")
```

Google Cloud

Conditions are specified in the role bindings of a resource's IAM policy.

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.

Condition attributes are either based on the requested resource—for example, its type or name—or on details about the request—for example, its timestamp or destination IP address.

There are two subtypes of condition attributes:

- Resource attributes, which allow you to write conditions that evaluate the resource in the access request.
- Request attributes, which allow you to write conditions that evaluate details about the request.

As an example, think about if you wanted to only allow access to Cloud Storage buckets whose names start with a specified prefix. Since this is a resource your condition is focused on, your condition attribute would look like what's on the slide.
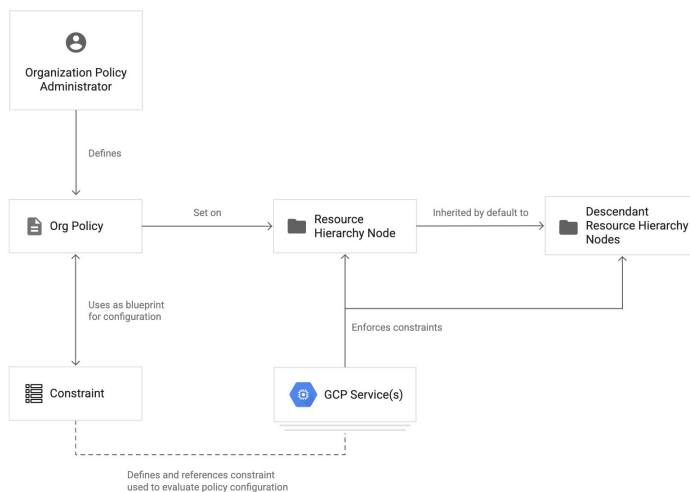
Check out the link in the speaker notes for more information on how to use IAM

Conditions:
- **Link:** cloud.google.com/iam/docs/conditions-overview

An organization policy gives you centralized and programmatic control over your organization's cloud resources.

Organization policies are set on organizations, folders, and projects in order to enforce the restrictions on that resource and its descendants.

Organization policy constraints

- Configured with constraints
  - Particular type of restriction against either a Google Cloud service or a group of Google Cloud services

- Descendants of the targeted resource hierarchy node inherit the organization policy.

```
resource: "organizations/ORGANIZATION_ID"
policy: {
  constraint: "constraints/iam.disableServiceAccountCreation"
  booleanPolicy: {
    enforced: true
  }
}
```

Google Cloud

In order to define an organization policy, you choose a constraint, which is a particular type of restriction against either a Google Cloud service or a group of Google Cloud services. You configure that constraint with your desired restrictions.

Descendants of the targeted resource hierarchy node inherit the organization policy. By applying an organization policy to the root organization node, you are able to effectively drive enforcement of that organization policy and configuration of restrictions across your organization.

The example on the slide shows how an organization policy that's used to disable service account creation.

Organization Policy constraint types

- List constraint type allow or disallow from a list of values.
  - Example: `compute.vmExternalIpAccess`
- Boolean constraint type turn on or turn off policies.
  - Example: `compute.disableSerialPortAccess`

You can think of a constraint as a blueprint that defines what behaviors are controlled. The enforcing service will evaluate the constraint type and value to determine the restriction.

There are 2 main constraint types: list and boolean.

The list constraint type allows or disallows values within a list. An example is the "compute.vmExternalIpAccess" list constraint.

This constraint defines the set of Compute Engine VM instances that are allowed to use external IP addresses. Remember that by default, all Compute Engine instances are allowed to use external IP addresses.

Boolean constraint types turn policies on or off. An example of this constraint type would be the "compute.disableSerialPortAccess" constraint.

Organization Policy constraints examples

Service	Constraint
Compute	constraints/compute.disableNestedVirtualization
	constraints/compute.disableSerialPortAccess
	constraints/compute.trustedImageProjects
	constraints/compute.vmExternalIpAccess
IAM	constraints/iam.disableServiceAccountCreation
	constraints/iam.disableServiceAccountKeyCreation
Google Cloud	constraints/serviceuser.services

Google Cloud

Of course, there are many different constraints for different Google Cloud services. This slide shows a few more constraints that are available for some other services.

Organization policies hierarchy

- All descendants of resource hierarchy node inherit the organization policy by default.

- If organization policy set at the root organization node, then those restrictions are inherited by all child folders, projects, and resources.
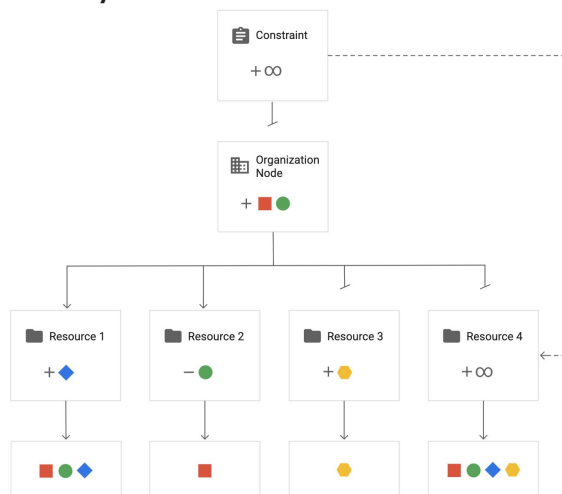
Google Cloud

When you set an organization policy on a resource hierarchy node, all descendants of that resource hierarchy node inherit the organization policy by default.

If you set an organization policy at the root organization node, then those restrictions are inherited by all child folders, projects, and resources.

In the resource hierarchy diagram in this slide, each node sets a custom organization policy and defines whether it inherits its parent node's policy.

You can learn more about organization policy inheritance from the link in the speaker notes.

- **Link:**
  cloud.google.com/resource-manager/docs/organization-policy/understanding-hierarchy

Organization policies versus IAM policies

Organization policies

- Focuses on **what**

- Lets the administrator set restrictions on specific resources to determine how they can be configured.

IAM policies

- Focuses on **who**

- Lets the administrator authorize who can take action on specific resources based on permissions.

Organization policies and their constraints are **not** the same things as IAM policies and bindings. Organization policies and IAM policies compliment one another.

Organization Policy focuses on **what**, and lets the administrator set restrictions on specific resources, services, or groups of services to determine how they can be configured and used.

Identity and Access Management focuses on **who**, and lets the administrator authorize who can take action on specific resources or services based on permissions.

Identity and Access Management (IAM)
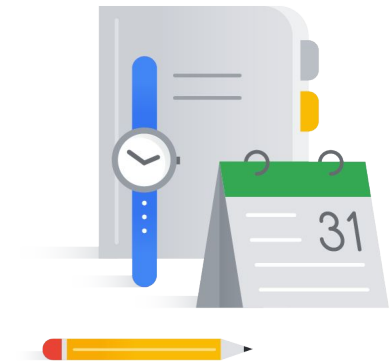
Resource Manager

IAM roles

Service accounts

Workload identity federation
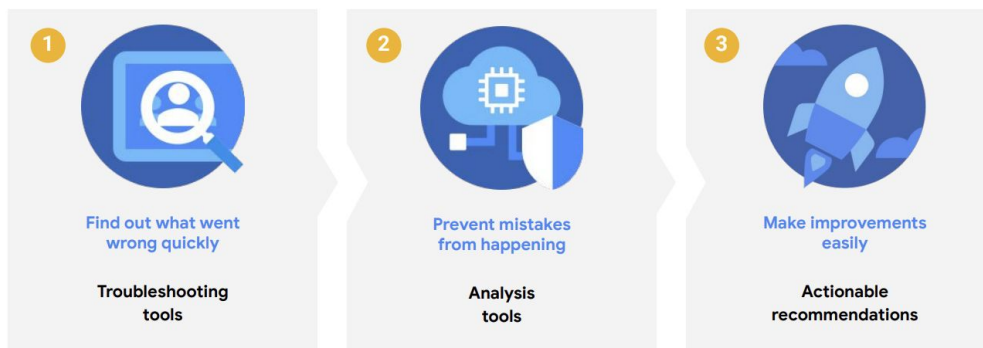
IAM & Organization policies

Policy Intelligence

IAM best practices

Google Cloud

Let's now discuss the Policy Intelligence suite.

Policy Intelligence

Policy Intelligence tools help you understand and manage your policies to proactively improve your security configuration.

Policy Intelligence will assist you through the lifecycle of policy management to manage policies securely and with confidence. Policy Intelligence gives you a suite of tools for troubleshooting, analysis, and recommendations.

These allow you to:

1. Find out what went wrong quickly so you can take action right away with Troubleshooting tools.
2. Prevent mistakes from happening all together with Analytical Tools.
3. Improve your security posture with Actionable Recommendations.

Policy Troubleshooter exposes access policies that apply to a particular resource

Policy Troubleshooter:

- Requires a member email, a resource name, and a permission to check.

- Examines all IAM policies that apply to that resource.

- Reports on whether that member's roles include that permission to that resource.

- Reports on which policies bind that member to those roles.
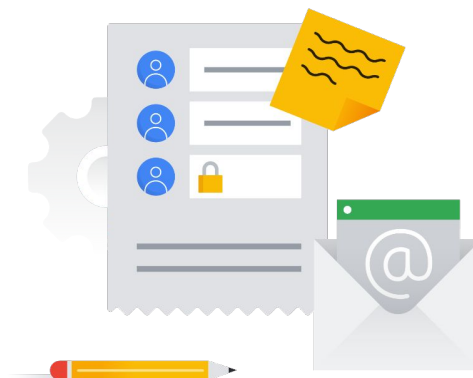
IAM Policy Troubleshooter helps you more closely examine policies that govern user access to a particular resource.

This tool makes it easier to understand why a user has access to a resource or doesn't have permission to call an API.

In order to generate a Policy Troubleshooter report, you will need the email of the user who needs access, the full name of the resource they need access to, and a permission that you want to check for.

Troubleshooter will take this information and examine all the IAM policies that apply to that particular resource and then report on whether it found that permission for that user in the resource's list of permissions. It will also report on the policies that bind that user to those roles.

Policy Troubleshooter will only access policies that the user has permissions to view

- Policy Troubleshooter may not always fully explain resource access.

- If you do not have access to a resource policy, it will not be analyzed.

- Maximum effectiveness requires the Security Reviewer (roles/iam.securityReviewer) role

For security reasons, Policy Troubleshooter can only examine policies that the person using it has permissions to access. Because Troubleshooter cannot analyze permissions it does not have access to, it may not always be able to fully explain a resource's access policies.

If maximum effectiveness is the overriding concern, the member using the Policy Troubleshooter must be granted the Security Reviewer (roles/iam.securityReviewer) role.

—

Policy Troubleshooter is accessible through the console, the Google Cloud CLI, or the REST API

- Console
  - Simple queries
- Google Cloud CLI
  - More complex scenarios
- REST API
  - More complex scenarios

Enter the following fields to check if the API call will grant the principal access to a resource.

If you have access logs turned on, you can view them in the Logs Explorer .

Principal (email) *
example@google.com

Enter an email address such as user@company.com

Resource permission pairs

Resource 1 *
🔍 //compute.googleapis.com/projects/looker-private-d

Permission 1 *
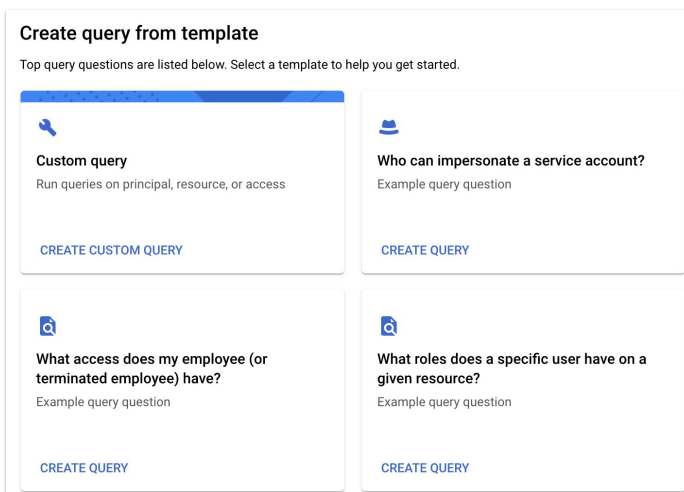compute.disks.deleteTagBinding ❓

➕ ADD ANOTHER PAIR

CHECK API CALL   CLEAR

Google Cloud

You can access Policy Troubleshooter using the console, the Google Cloud CLI, or the REST API. For simple queries, using the console is typically fastest. For more complex scenarios, consider the gcloud CLI or the REST API.

Policy Analyzer

- Which principles have what access to which Google Cloud resources?

- Examples:
  - Who can access IAM service account?
  - Who can read data in BigQuery dataset?
  - Who has access to resources in a project?

Create query from template

Top query questions are listed below. Select a template to help you get started.

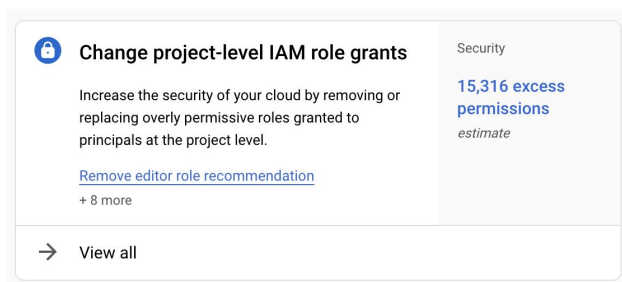|	
🔧 **Custom query** Run queries on principal, resource, or access CREATE CUSTOM QUERY	👤 **Who can impersonate a service account?** Example query question CREATE QUERY
🔍 **What access does my employee (or terminated employee) have?** Example query question CREATE QUERY	🔍 **What roles does a specific user have on a given resource?** Example query question CREATE QUERY

Google Cloud

Policy Analyzer lets you find out which principals (for example, users, service accounts, groups, and domains) have what access to which Google Cloud resources based on your IAM allow policies.

Policy Analyzer can help you answer questions like these:

- Who can access this IAM service account?
- Who can read data in this BigQuery dataset that contains personally identifiable information (PII)?
- What roles and permissions does the dev-testers group have on any resource in this project?

Role recommendations (Recommender)

- Identify and remove excess permissions from your principals.

- Use ML-based policy insights to analyze principal's permission usage.

- Help enforce principle of least privilege.

🔒 **Change project-level IAM role grants**

Increase the security of your cloud by removing or replacing overly permissive roles granted to principals at the project level.

Remove editor role recommendation
+ 8 more

Security

15,316 excess permissions
estimate

→ View all

Google Cloud

Role recommendations are one of the types of recommendations that Recommender generates.

Role recommendations help you identify and remove excess permissions from your principals, improving your resources' security configurations.

Recommender identifies excess permissions using policy insights. Policy insights are ML-based findings about a principal's permission usage.

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 evaluations

- Recommender compares project-level role grants with permissions used within the last 90 days.

- If a permission has not been used within that time, recommender will suggest revoking it.

- You have to review and apply recommendations; they will not be applied automatically.

Google Cloud
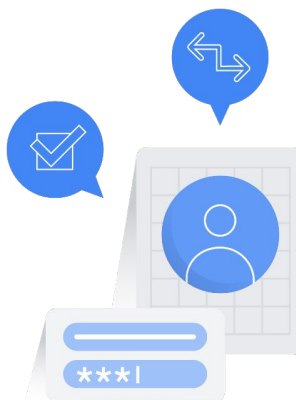
Recommender evaluates only role grants that were made at the project level and that have existed for at least 90 days. It does not evaluate any of the following items:

- Conditional role grants
- Role grants for Google-managed service accounts, and
- Access controls that are separate from IAM

Recommender gives you three types of recommendations

- Revoke an existing role.

- Replace an existing role.

- Add permissions to an existing role.

Recommender creates daily policy recommendations
and serves them to you automatically.

Recommender will suggest that you revoke an existing role when it has been in effect for 90 days or more and when it has not been used within the past 90 days.

The theory with this type of recommendation is that if the policy has not been used within the past 90 days, it may have been unnecessary originally, or it may have outlived its usefulness.

Removing such permissions keeps your roles pruned down to only those permissions that are actually required, which is a foundational security concept. Recommender may also suggest that you replace a particular role with another role or set of roles.

For example, if a service account has an assigned role with permissions that are not used, it would be more secure if you revise it to use a combination of less-permissive roles that have only the necessary permissions.

And, finally, Recommender may suggest that you add permissions to a role, even if those permissions are not currently being used. Recommender uses machine learning to predict which permissions may be needed by a particular role in the future. If those permissions are not currently enabled, Recommender will suggest adding them.

Recommender creates daily IAM policy recommendations and serves them to you automatically.

The easiest way to review and apply recommendations is to use Cloud Console

- View existing roles by visiting the IAM page.
- Look for the "over-granted permissions" column.
- If there are recommendations, you will see a Recommendation available 💡 icon.
- Click the Recommendation available icon for details.
- Choose to "apply" or to "dismiss" a recommendation.
- You can revert your choice within 90 days.
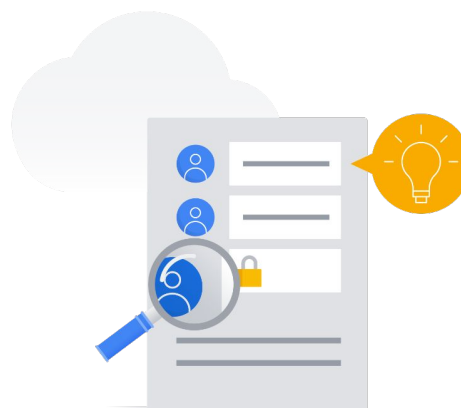
Google Cloud

Your recommendations can be found on the IAM page in the list of current roles for your account. Next to each role, in the "over granted permissions" column, you will see one of two icons: a lightbulb that is either greyed out or one that is golden-orange and "lit," indicating that there are recommendations available for that role.

If a role has recommendations, clicking on the Recommendation available icon will show you more details about the recommendation, and you can then choose to accept and apply a recommendation or dismiss it.

If you change your mind within 90 days about accepting or dismissing a recommendation, you can use the IAM Recommender logs to revert that decision.

While using the Cloud Console is the easiest way to manage your recommendations, you can also review and apply recommendations using the gcloud command-line tool and the Recommender API.

Identity and Access Management (IAM)
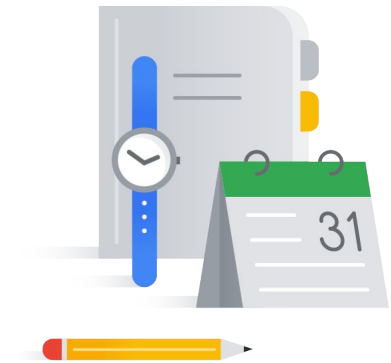
Resource Manager

IAM roles

Service accounts

Workload identity federation

IAM & Organization policies

Policy Intelligence

IAM best practices

Google Cloud

Now, let's discuss IAM best practices...

IAM best practices

- Adhere to the Principle of Least Privilege

- Manage IAM meticulously

- Cloud permissions more powerful than traditional on-premises environments
  - No physical boundaries between systems

Google Cloud

The first is to always use the principle of least privilege - which just means always apply the minimal access level required to get the job done. If a particular role has too many permissions for that job, create a Custom role so you can whittle permissions down to only what is needed. Not only is this practice more secure, it can also help prevent incidents from occurring - such as the accidental editing or removal of a required resource.

When creating policies, remember that a less restrictive parent policy will always override a more restrictive resource policy, so check when implementing parent policies to make sure you do not inadvertently grant more access to a child resource than you intended. For example, if someone in your organization is a project editor, you cannot restrict their access to a specific resource within that project. However, you can create a deny policy that will prevent their access. Remember that deny policies are always applied before access policies.

Disciplined and meticulous management of IAM is especially important in Google Cloud. Since access management is centralized, users can be given powerful permissions that allow them to control and manage Google Cloud resources.

In on-premises environments, the target systems were physically discrete, which made it a challenge to have centralised control. In turn, the blast radius of misconfigurations were managed to an extent due to these physical boundaries between systems.

IAM best practices

- Use groups when configuring Google Cloud access.

- Assign roles to the groups instead of individual users.

Google Cloud

It is best to use groups when configuring Google Cloud access - assign roles to the groups instead of individual users.

Groups are defined and maintained in the Cloud Console for Google Workspace or Cloud Identity domains, they are not configured in Google Cloud, so using groups will drastically reduce the administration needed by Google Cloud admins. Only minimal changes will be needed within Google Cloud once groups and roles are defined. Then users can simply be added or removed from groups by your Google Workspace or Cloud Identity admin.

IAM best practices

- Use predefined roles. Basic roles can be overly permissive.

- Utilizing predefined roles offers less administrative overhead.

- Predefined roles are managed by Google.

- Custom roles are not maintained by Google.

- Use the Policy Intelligence suite to manage policies securely and with confidence.

Another best practice is to use predefined roles as basic roles can be overly permissive. For example, providing a user with a Compute Engine admin role is more fine-grained than give a user an editor role, where they have access to any and all cloud resources.

Try to utilize predefined roles if they meet your requirements as predefined roles offer less administration. Predefined roles are managed by Google and their permissions are automatically updated as necessary.

For example: when new features or services are added to Google Cloud, all related predefined roles will be updated as needed.

Custom roles on the other hand are not maintained by Google. When new permissions, features, or services are added to Google Cloud, your custom roles will not be updated automatically.

And lastly, don't forget to use the Policy Intelligence suite to manage policies securely and with confidence.

Lab Intro

Configuring IAM

The objectives for this lab are for you to:

- Use IAM to implement access control,
- Restrict access to specific features or resources,
- Use predefined roles to provide Google Cloud access,
- Create custom IAM roles to provide permissions based on your own job roles, and
- Modify custom roles.

To wrap things up, here's an overview of some topics we covered in this module:

- IAM lets administrators authorize who can take action on specific resources.
- Resources in Google Cloud are hierarchically managed by organization, folders, and projects.
- Permissions are given to members by granting roles.
    - Google Cloud provides predefined roles, and the ability to create custom roles.
- Service Accounts control server-to-server interactions and are used to authenticate from one service to another.

Module review

- Workload identity federation allows you to grant on-premises or multi-cloud workloads access to Google Cloud resources without using a service account key.
- Recommender creates daily IAM policy recommendations and serves them to you automatically.
- IAM Policy Troubleshooter makes it easier to understand why a user has access to a resource or doesn't have permission to call an API.

Google Cloud

- Workload identity federation allows you to grant on-premises or multi-cloud workloads access to Google Cloud resources without using a service account key.
- Recommender creates daily IAM policy recommendations and serves them to you automatically.
- IAM Policy Troubleshooter makes it easier to understand why a user has access to a resource or doesn't have permission to call an API.