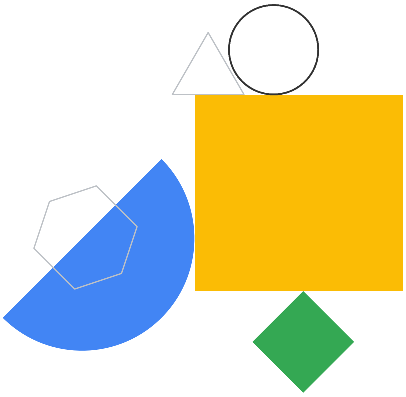
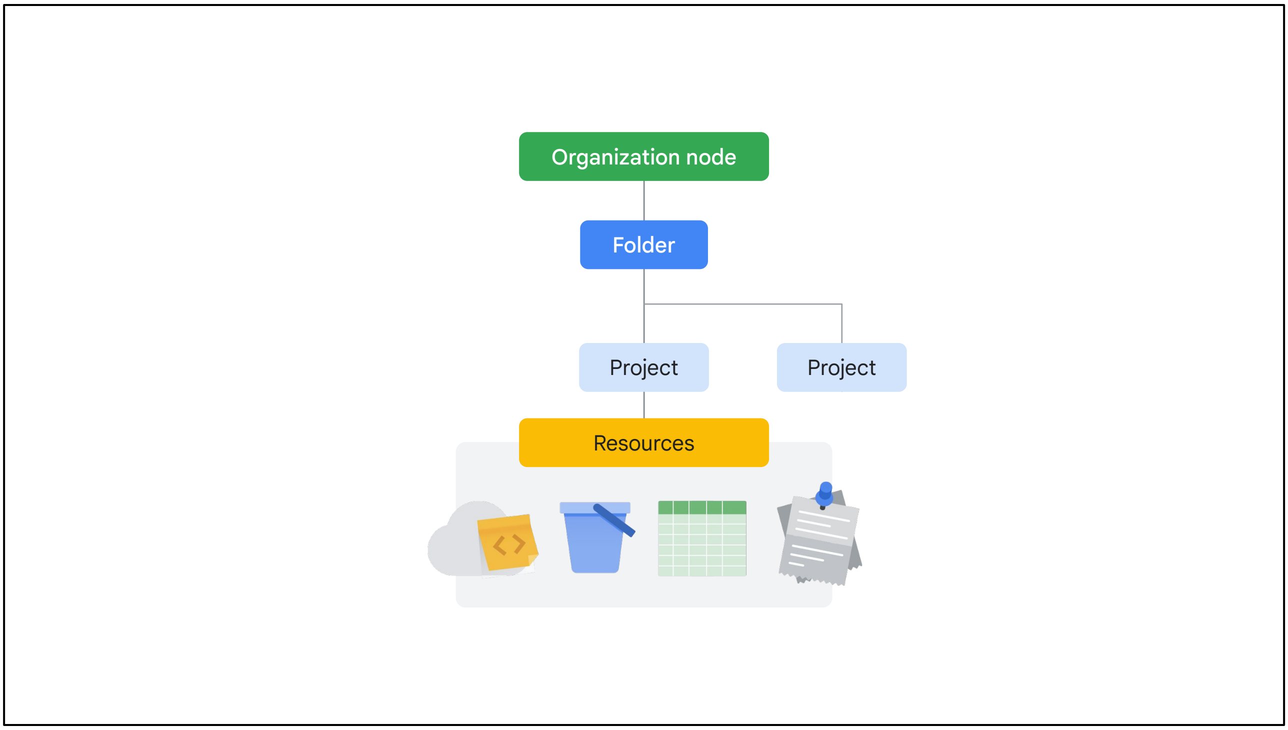
Google Cloud



Google Cloud Core Infrastructure Module 2

On-demand course

March 2022



Organization node

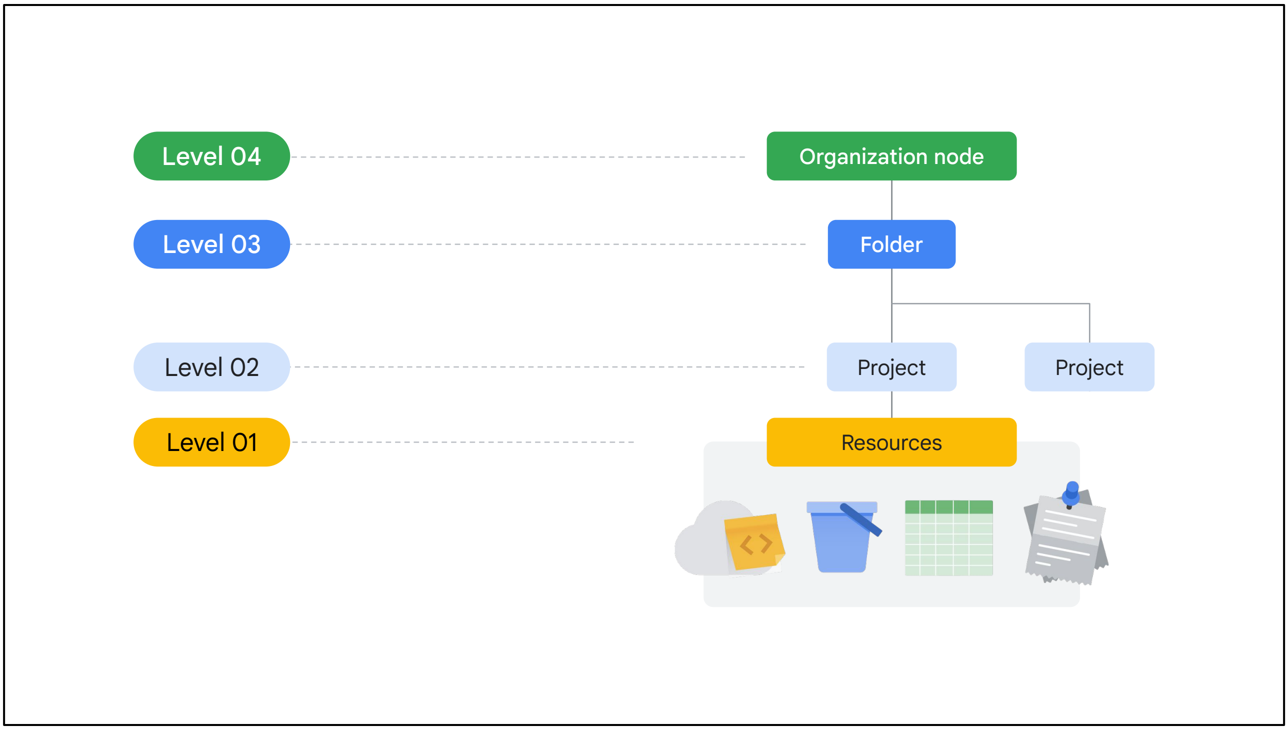
Folder

Project

Project

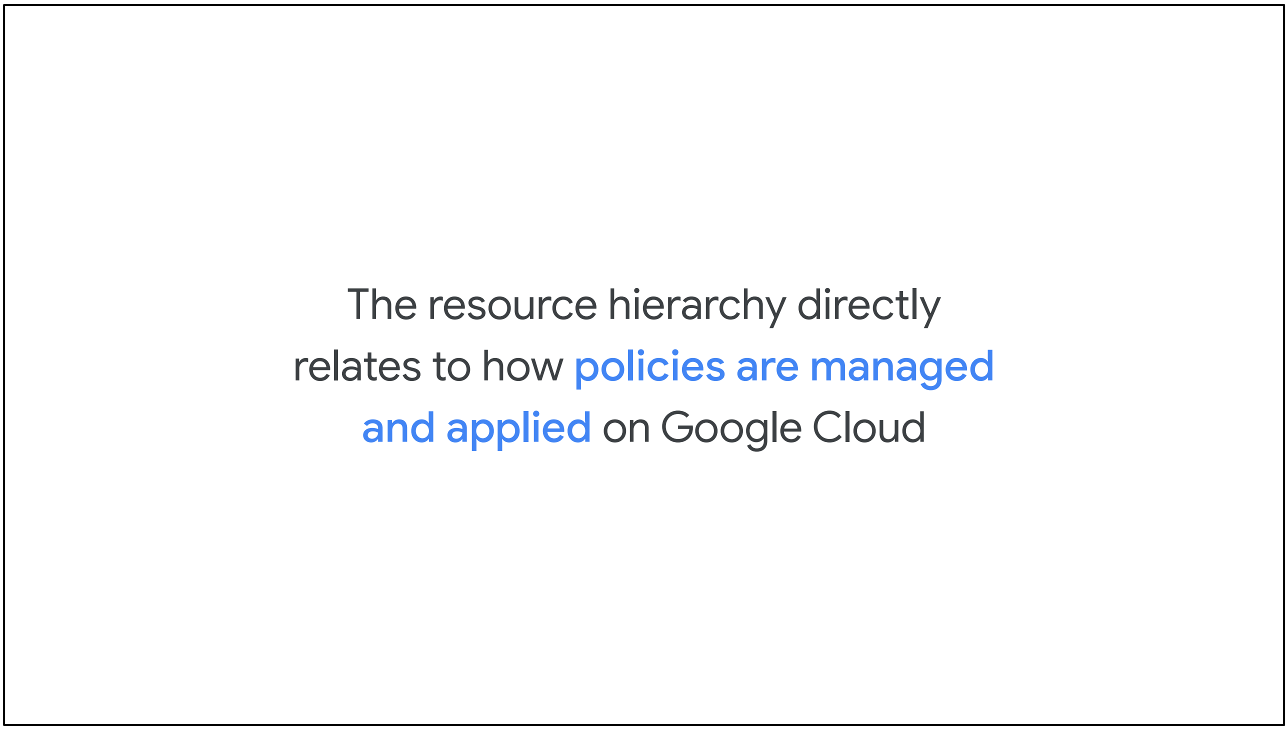
Resources

In this section of the course we'll look at the functional structure of Google Cloud.



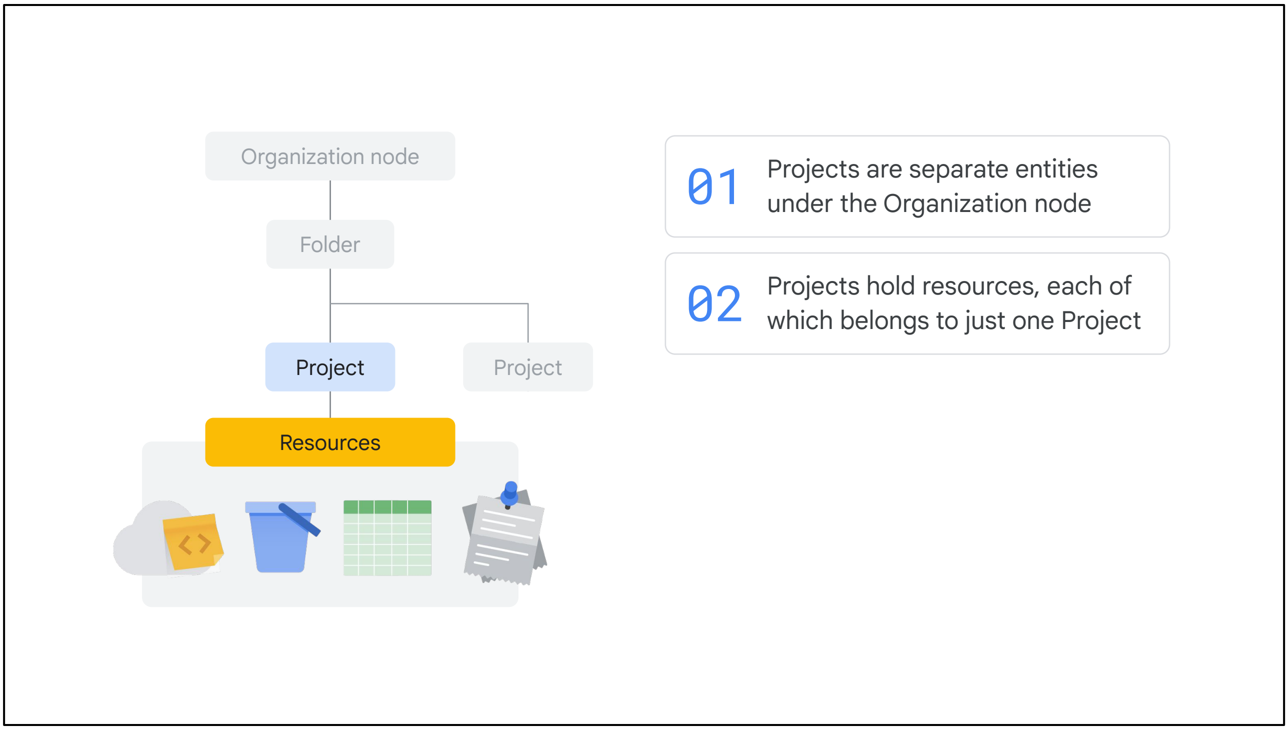
Google Cloud’s resource hierarchy contains four levels, and starting from the bottom up they are:

- Resources
- Projects
- Folders
- And an organization node.



The resource hierarchy directly relates to how **policies are managed and applied** on Google Cloud

It’s important to understand this resource hierarchy because it directly relates to how policies are managed and applied when you use Google Cloud.



Organization node

Folder

Project

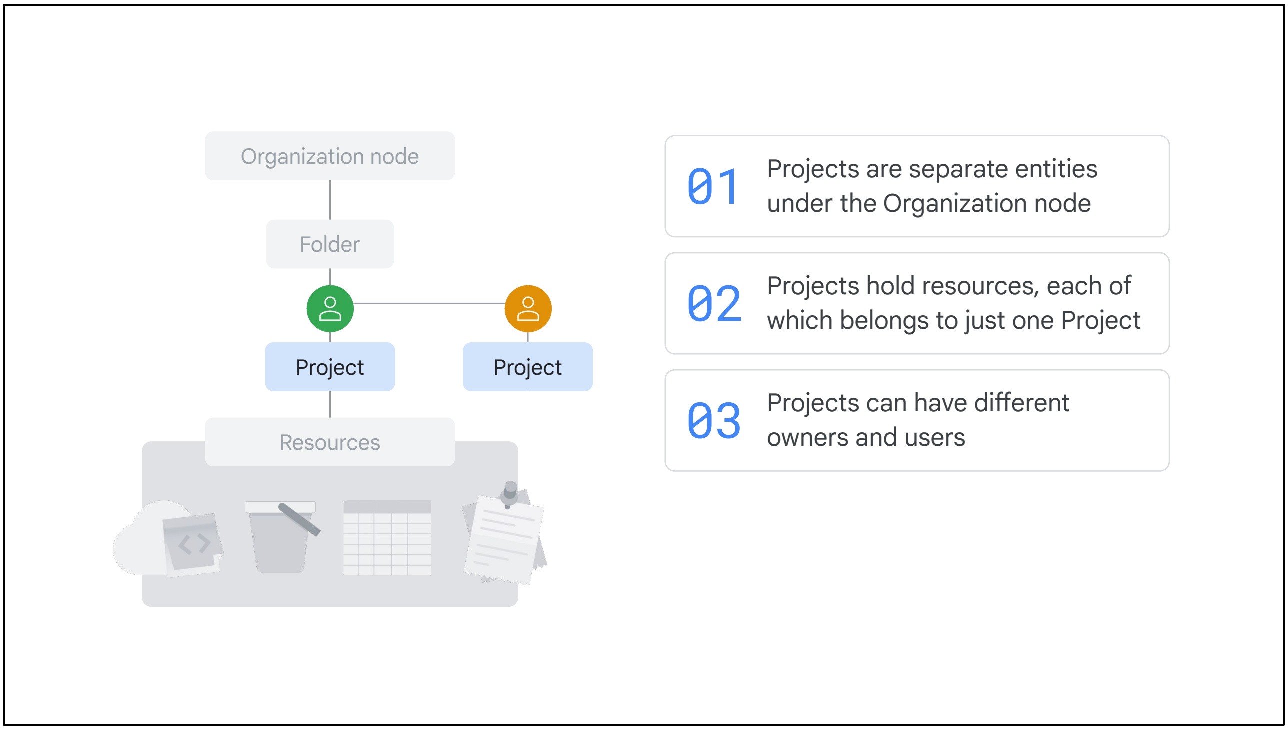
Project

Resources

01 Projects are separate entities under the Organization node

02 Projects hold resources, each of which belongs to just one Project

and each resource belongs to exactly one project.

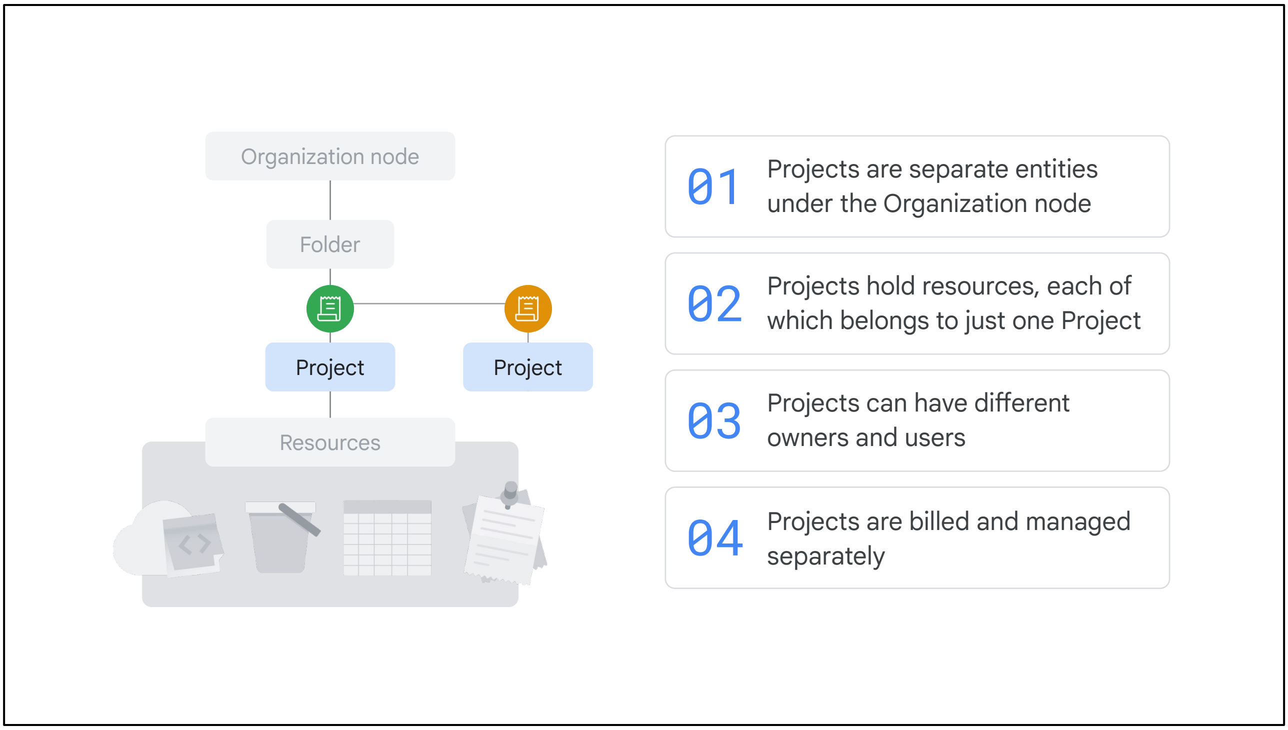


01 Projects are separate entities under the Organization node

02 Projects hold resources, each of which belongs to just one Project

03 Projects can have different owners and users

Projects can have different owners and users



Organization node

Folder

Project

Project

Resources

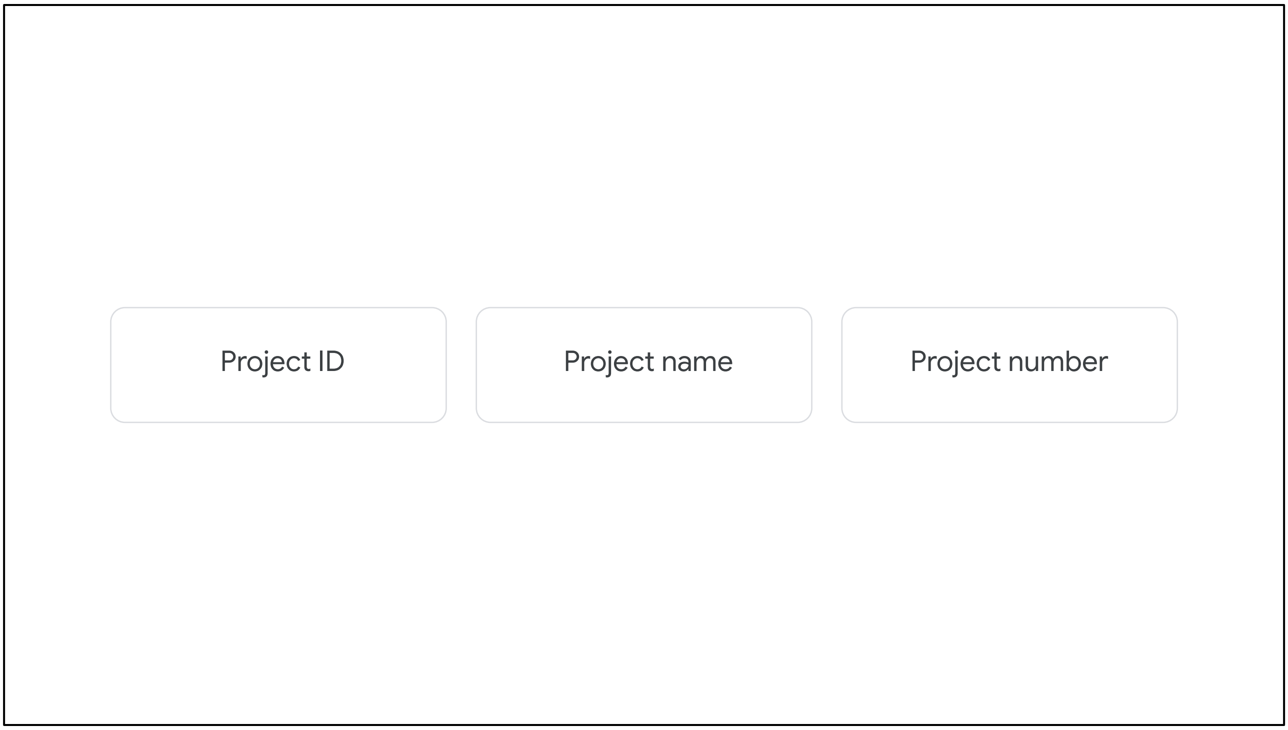
01 Projects are separate entities under the Organization node

02 Projects hold resources, each of which belongs to just one Project

03 Projects can have different owners and users

04 Projects are billed and managed separately

because they’re billed and managed separately.

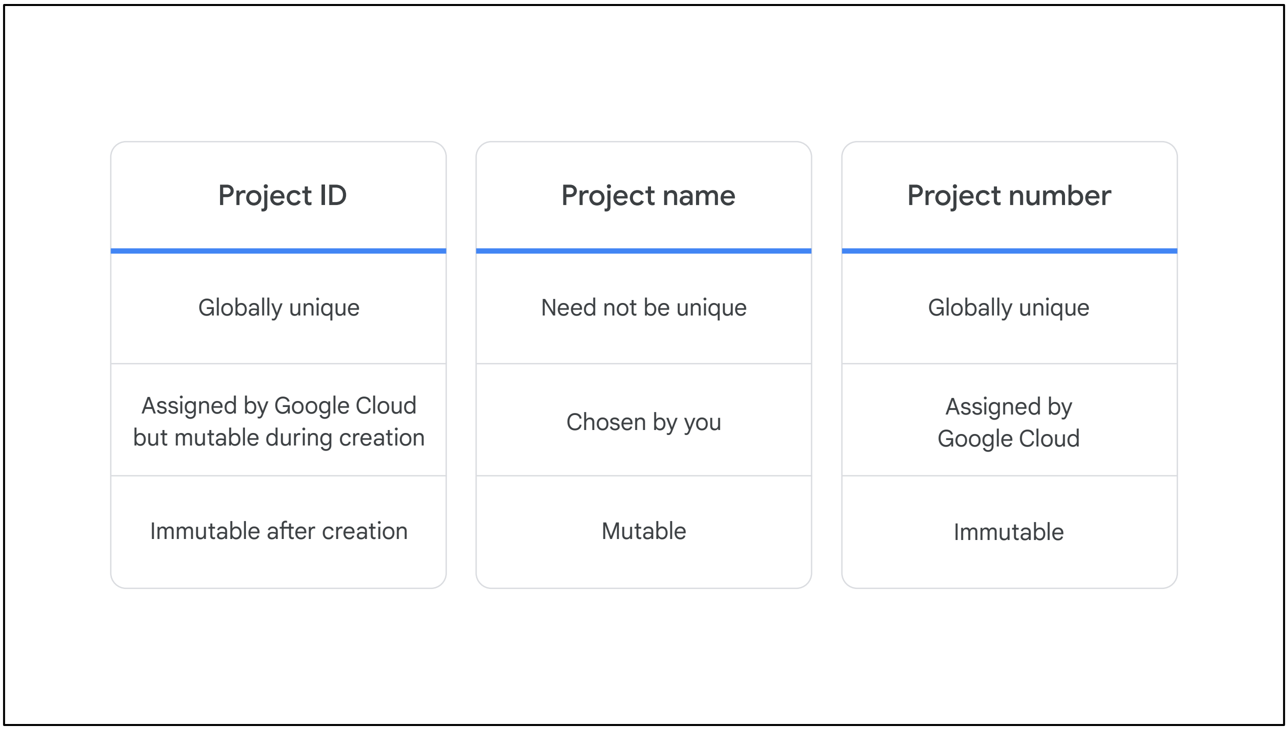


Project ID

Project name

Project number

Each Google Cloud project has three identifying attributes: a project ID, a project name, and a project number.

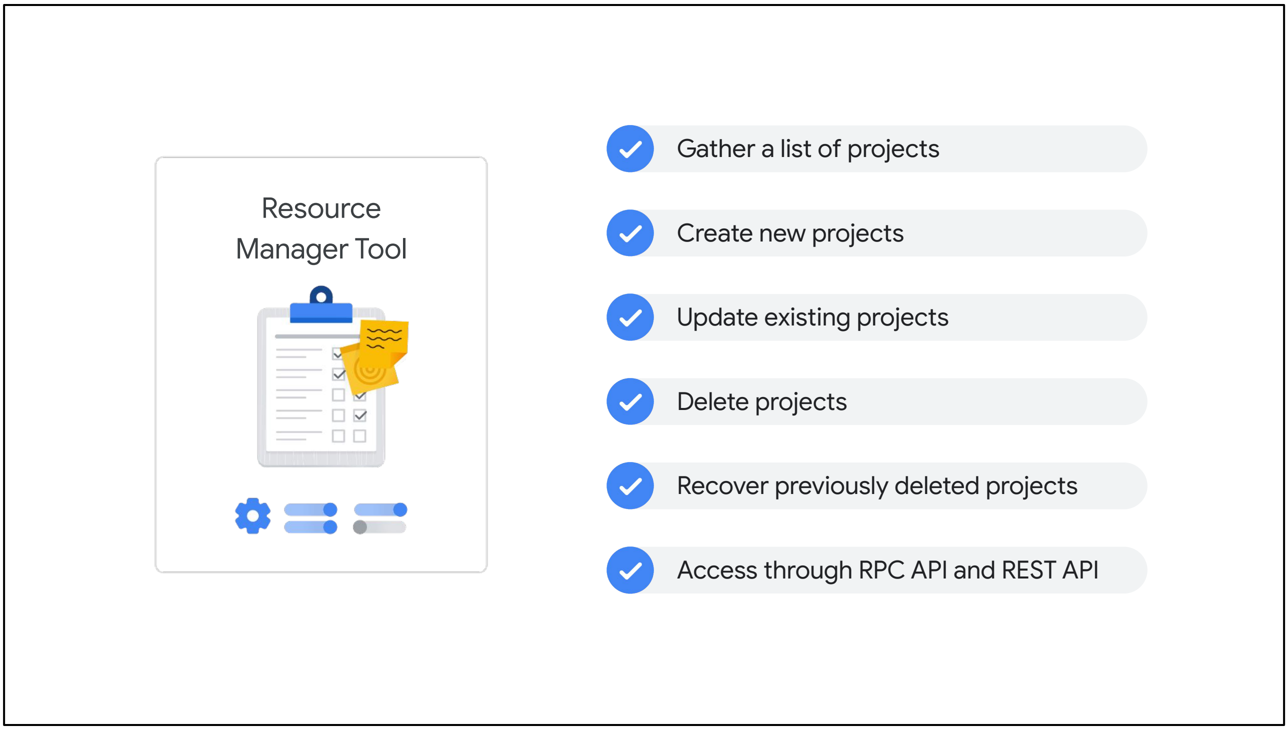


Project ID	Project name	Project number
Globally unique	Need not be unique	Globally unique
Assigned by Google Cloud but mutable during creation	Chosen by you	Assigned by Google Cloud
Immutable after creation	Mutable	Immutable

- The **project ID** is a globally unique identifier assigned by Google that can't be changed after creation. They're what we refer to as being *immutable*. Project IDs are used in different contexts to inform Google Cloud of the exact project to work with.

- **Project names**, however, are user-created. They don't have to be unique and they can be changed at any time, so they are not immutable.

- Google Cloud also assigns each project a unique **project number**. It's helpful to know that these Google-generated numbers exist, but we won't explore them much in this course. They're mainly used internally by Google Cloud to keep track of resources.

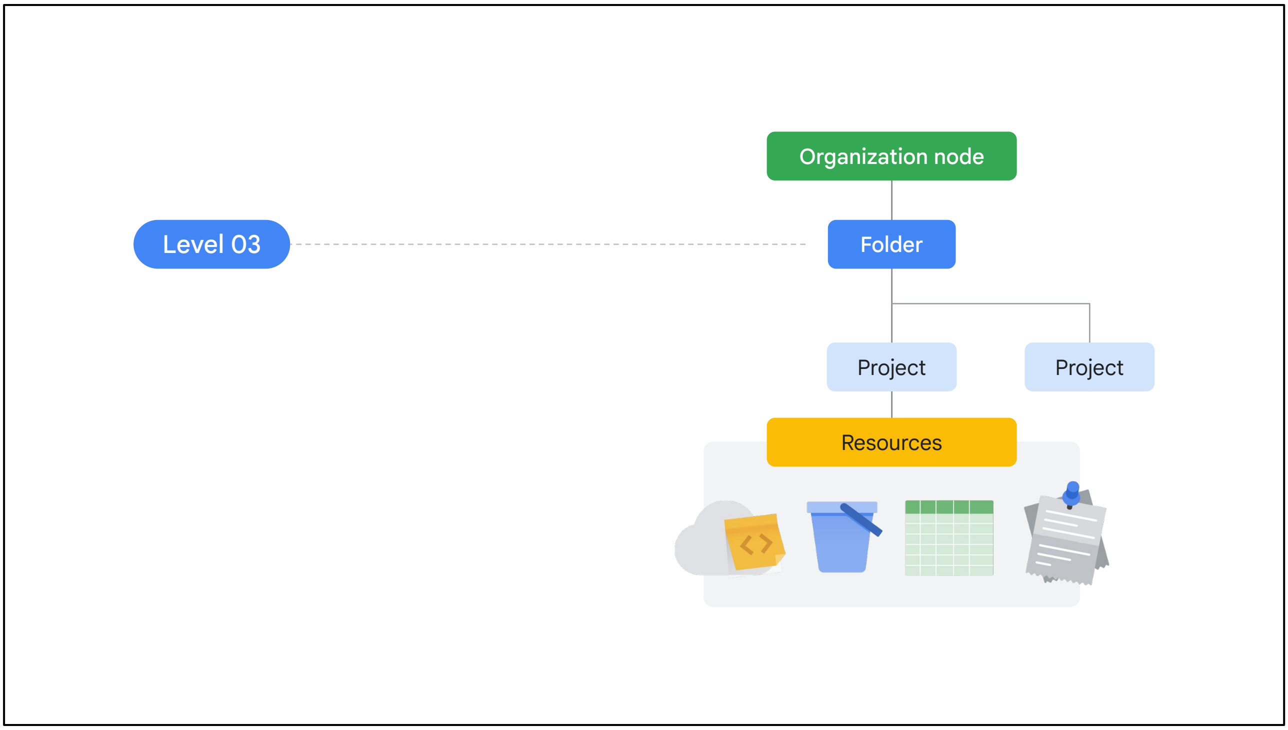


Resource Manager Tool

- ✓ Gather a list of projects
- ✓ Create new projects
- ✓ Update existing projects
- ✓ Delete projects
- ✓ Recover previously deleted projects
- ✓ Access through RPC API and REST API

Google Cloud's Resource Manager tool is designed to programmatically help you manage projects.

It's an API that can gather a list of all the projects associated with an account, create new projects, update existing projects, and delete projects. It can even recover projects that were previously deleted, and can be accessed through the RPC API and the REST API.



Level 03

Organization node

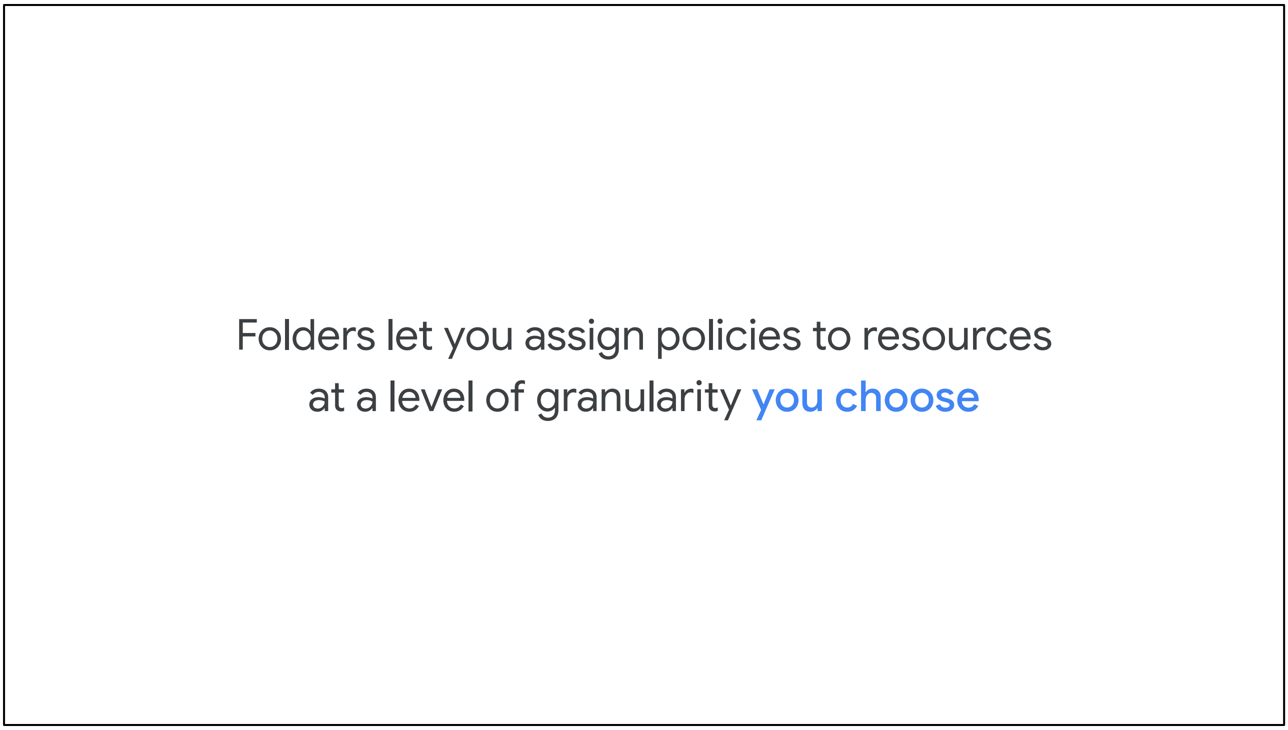
Folder

Project

Project

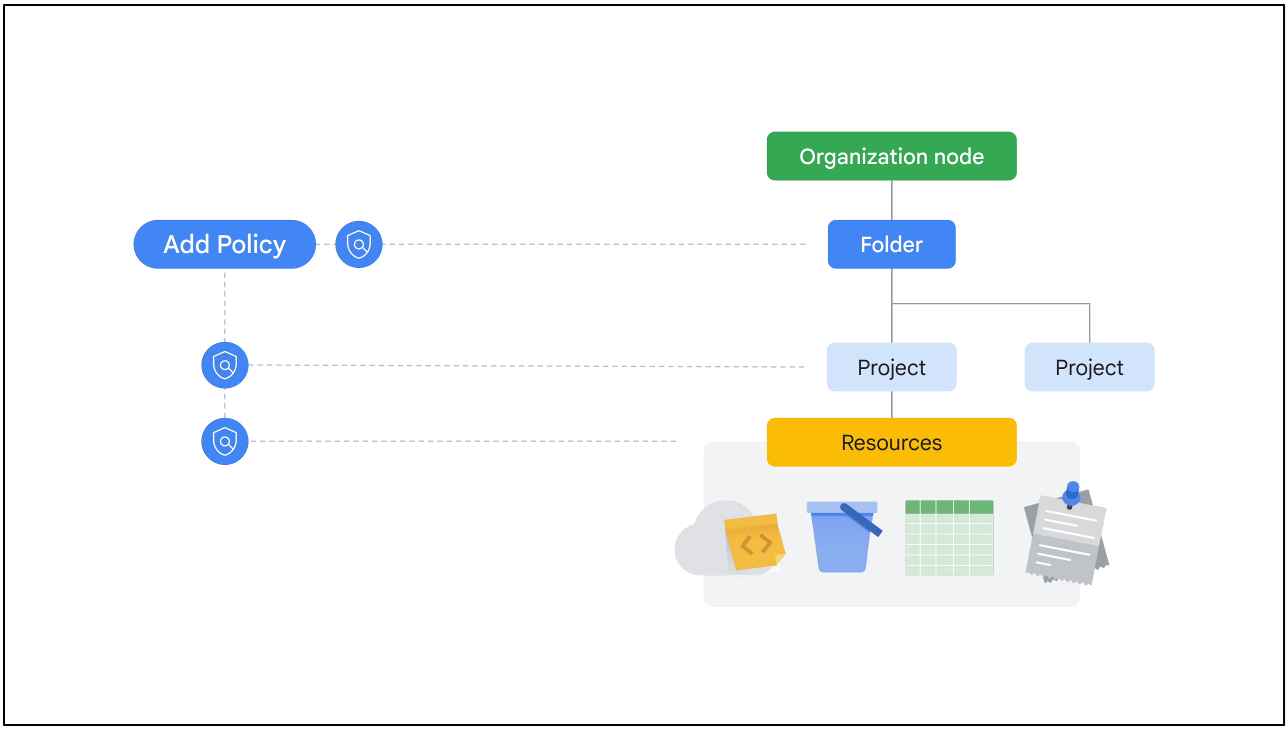
Resources

The third level of the Google Cloud resource hierarchy is folders.

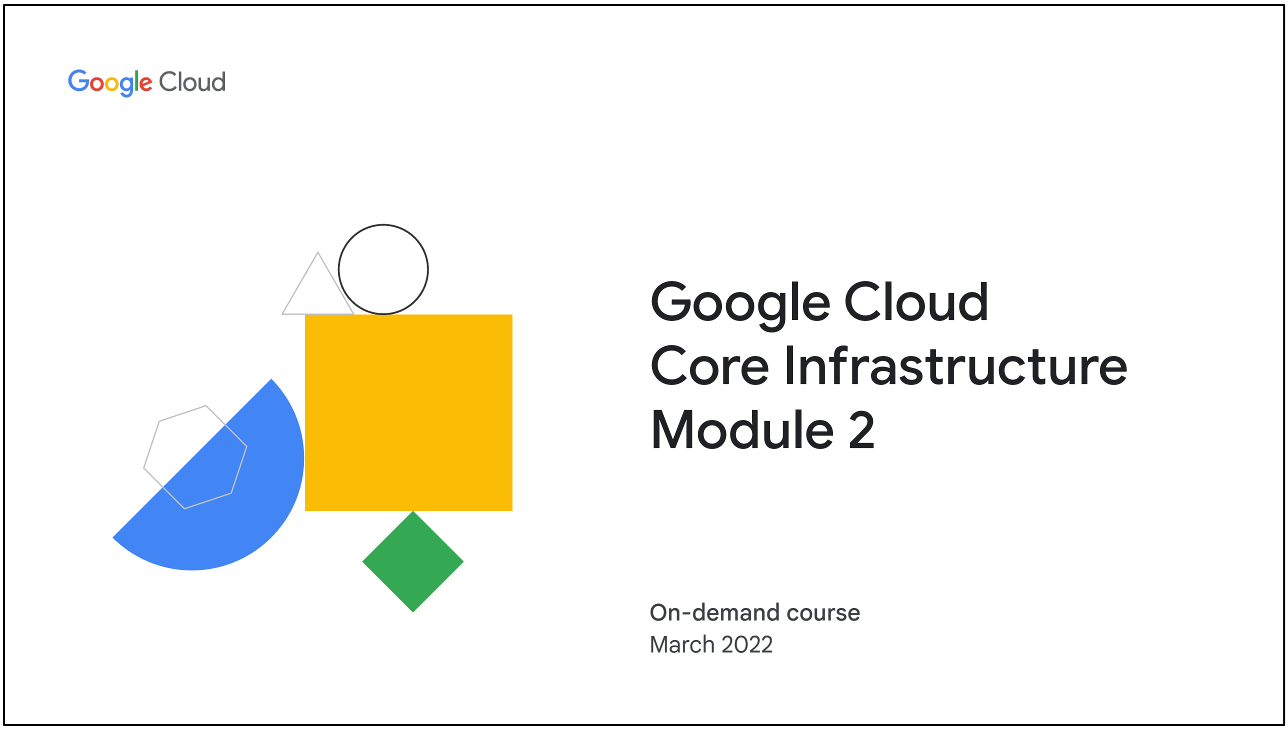


Folders let you assign policies to resources at a level of granularity you choose

Folders let you assign policies to resources at a level of granularity you choose.



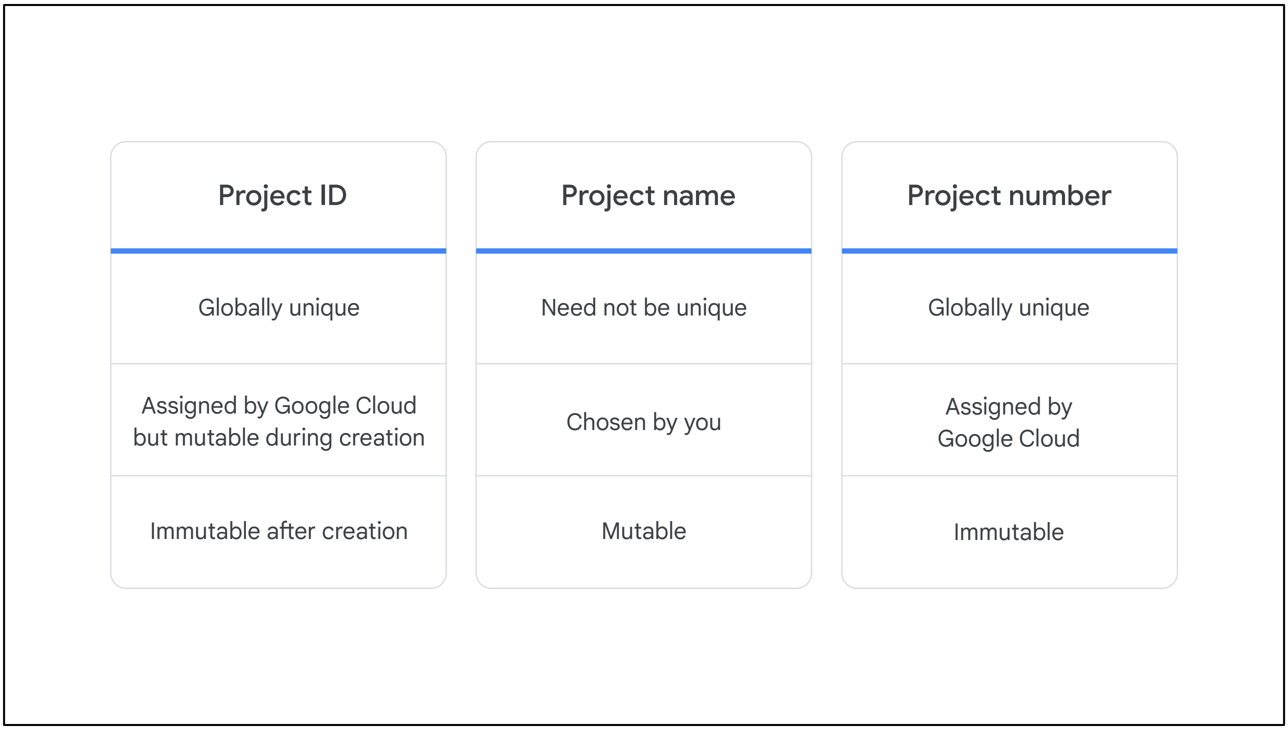
The resources in a folder inherit policies and permissions assigned to that folder.



Google Cloud

Google Cloud Core Infrastructure Module 2

On-demand course
March 2022

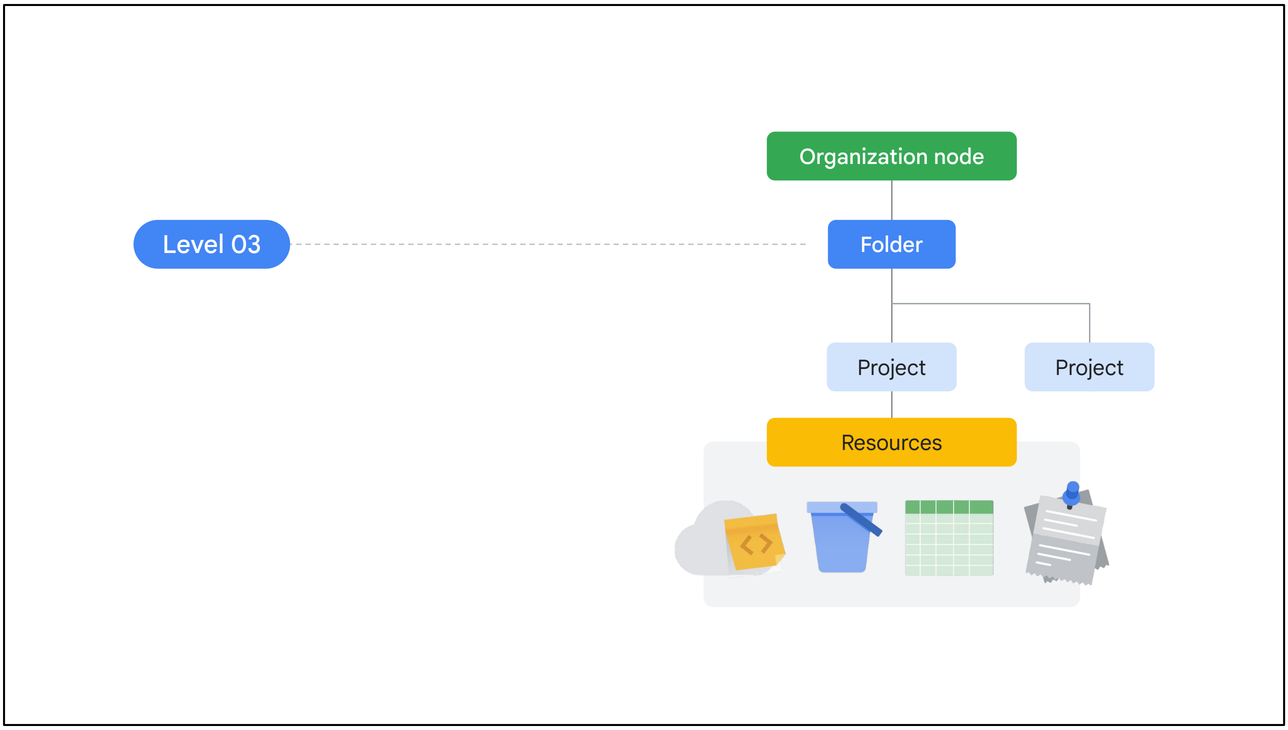


Project ID	Project name	Project number
Globally unique	Need not be unique	Globally unique
Assigned by Google Cloud but mutable during creation	Chosen by you	Assigned by Google Cloud
Immutable after creation	Mutable	Immutable

- The **project ID** is a globally unique identifier assigned by Google that can't be changed after creation. They're what we refer to as being *immutable*. Project IDs are used in different contexts to inform Google Cloud of the exact project to work with.

- **Project names**, however, are user-created. They don't have to be unique and they can be changed at any time, so they are not immutable.

- Google Cloud also assigns each project a unique **project number**. It's helpful to know that these Google-generated numbers exist, but we won't explore them much in this course. They're mainly used internally by Google Cloud to keep track of resources.



Level 03

Organization node

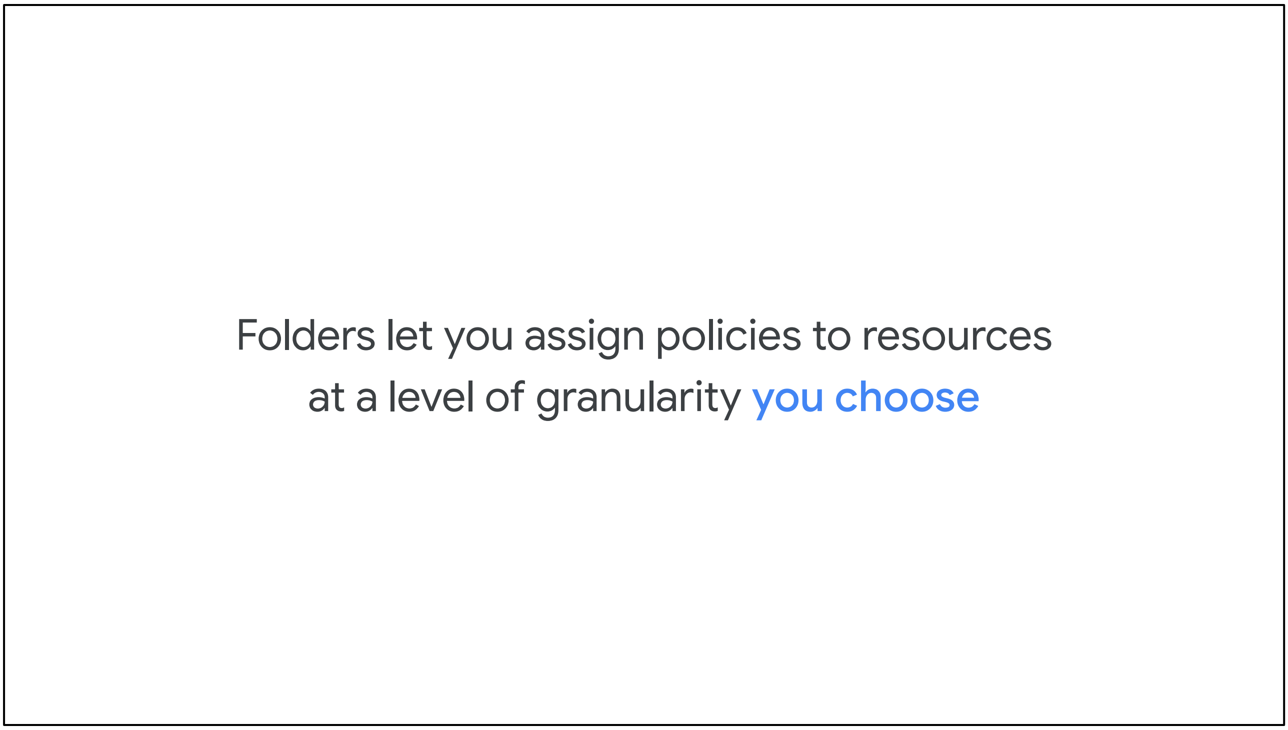
Folder

Project

Project

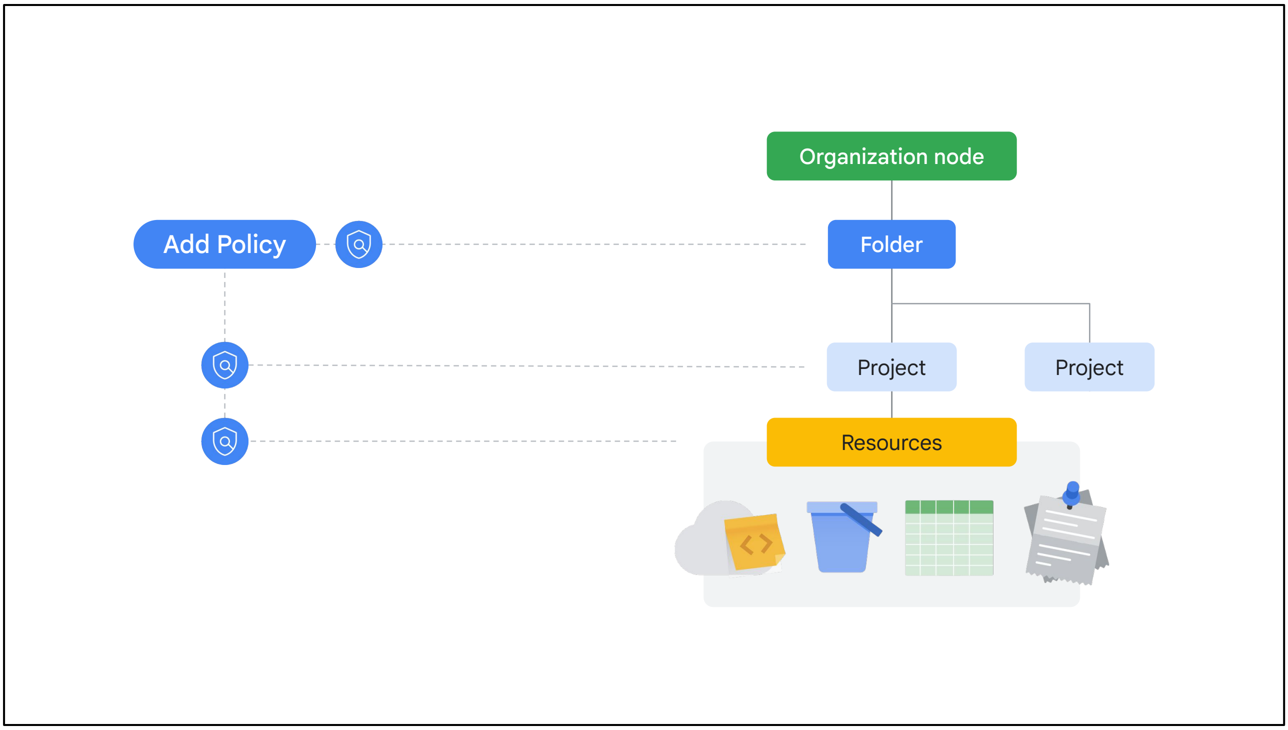
Resources

The third level of the Google Cloud resource hierarchy is folders.

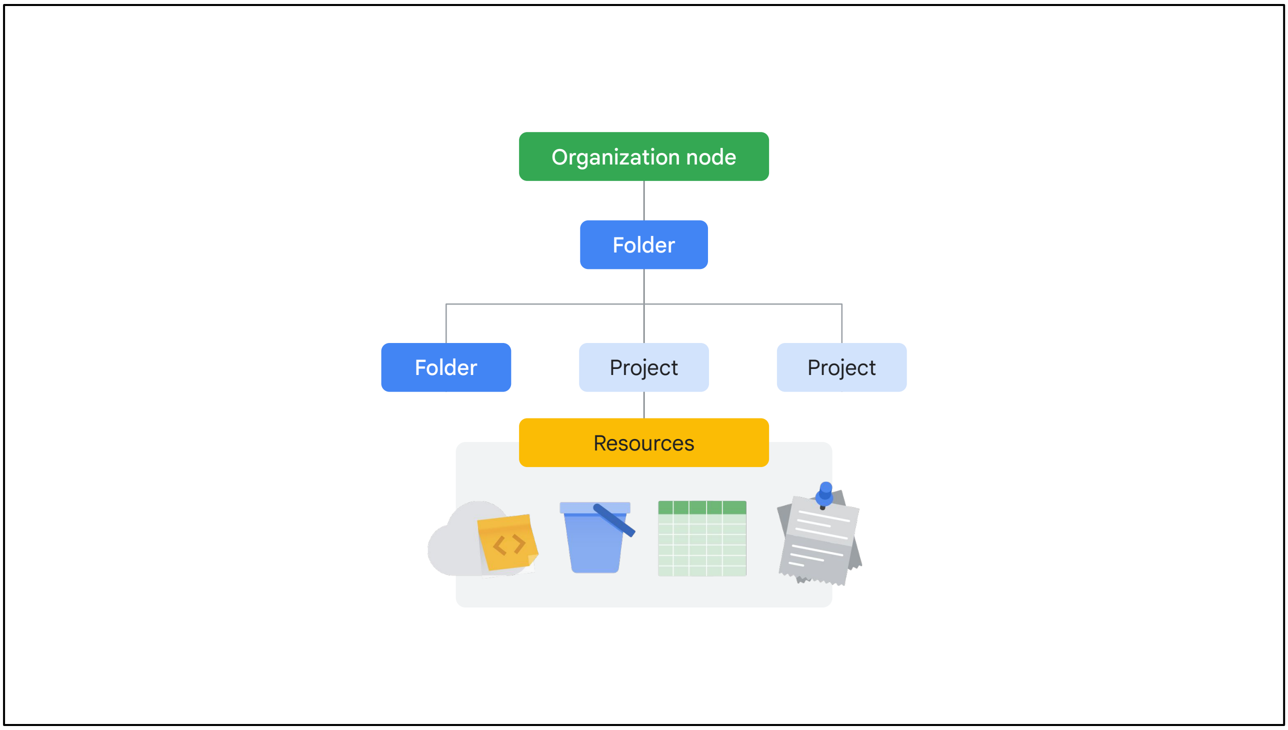


Folders let you assign policies to resources at a level of granularity **you choose**

Folders let you assign policies to resources at a level of granularity you choose.



The resources in a folder inherit policies and permissions assigned to that folder.



Organization node

Folder

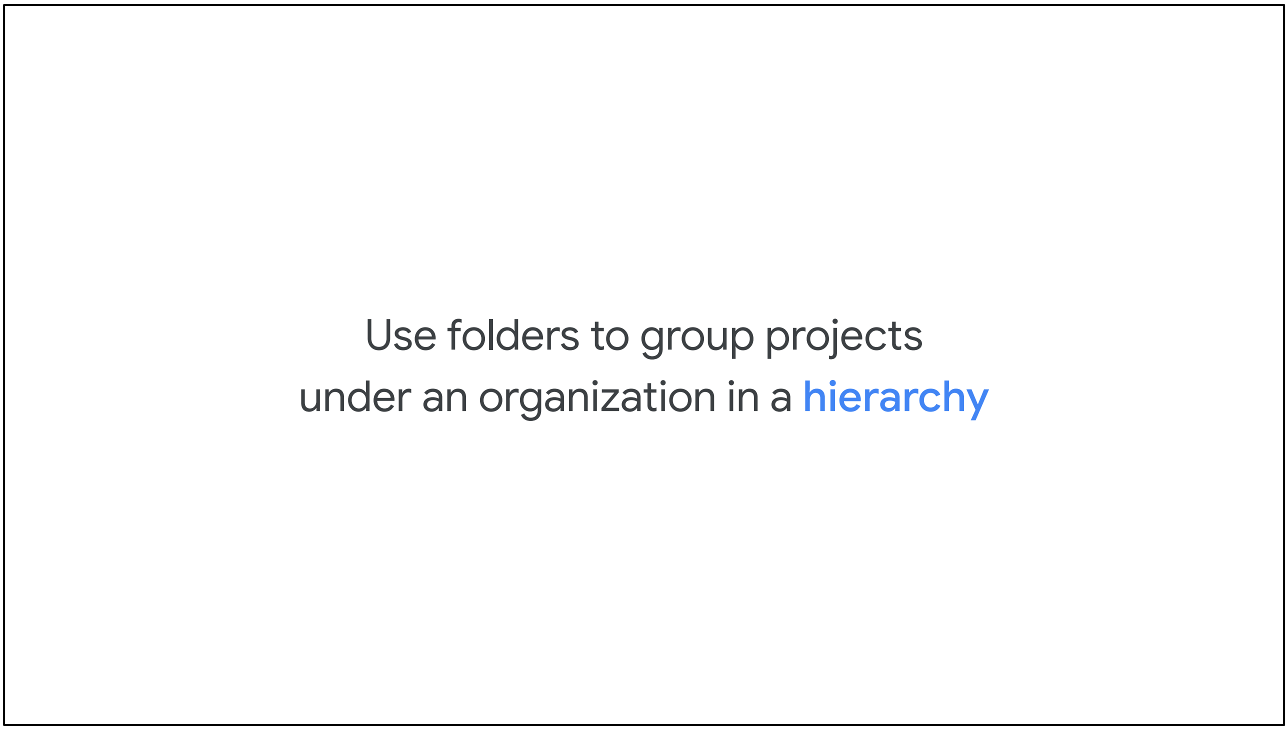
Folder

Project

Project

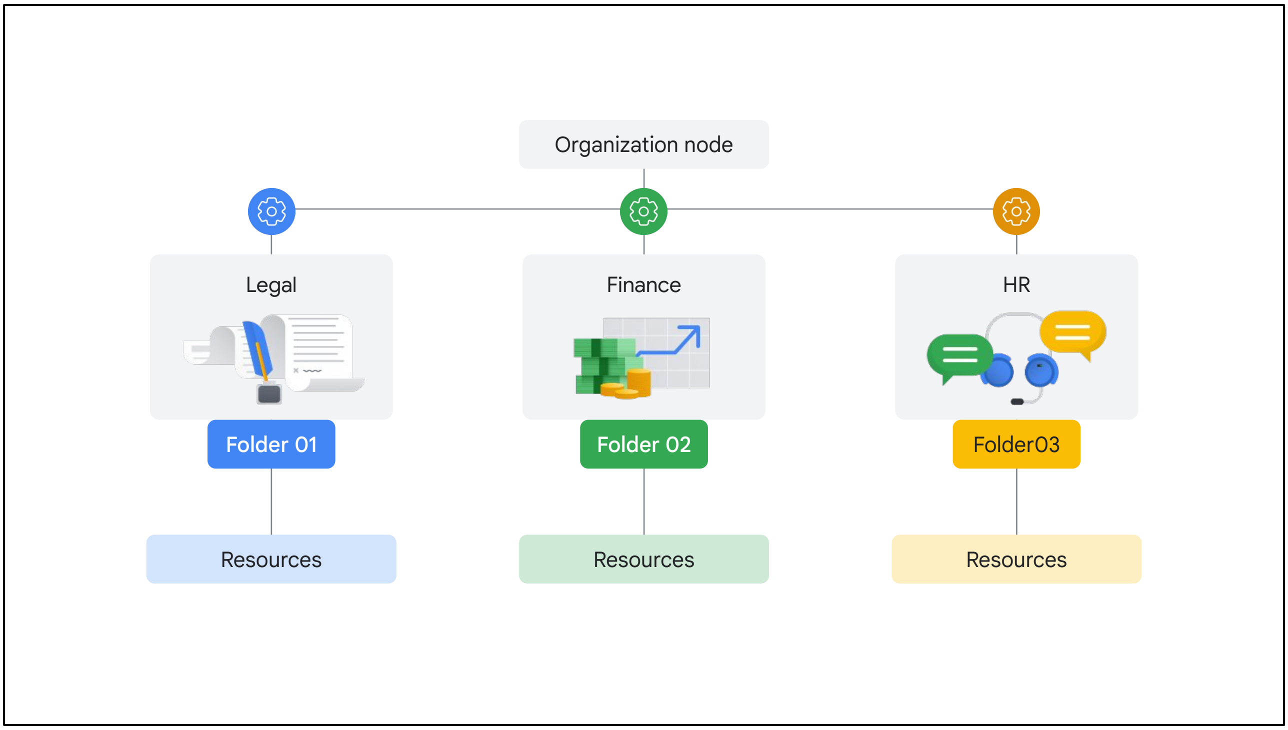
Resources

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



Use folders to group projects
under an organization in a **hierarchy**

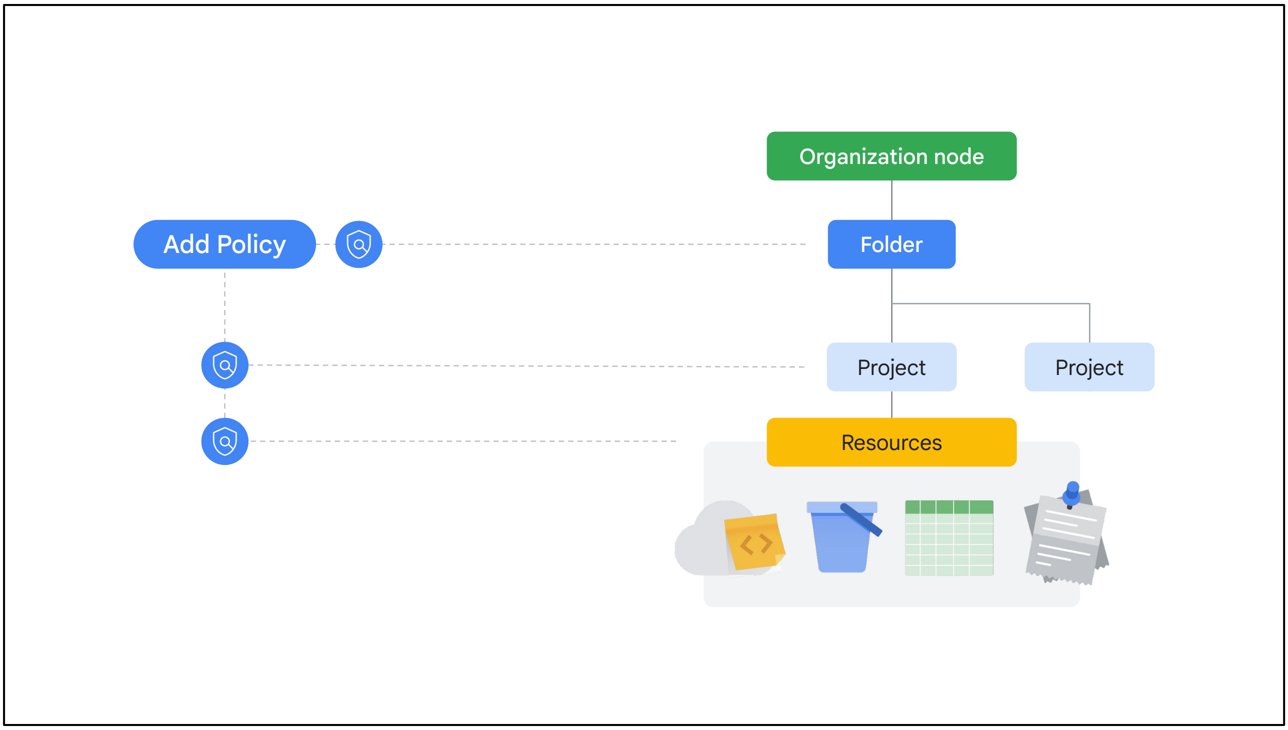
You can use folders to group projects under an organization in a hierarchy.



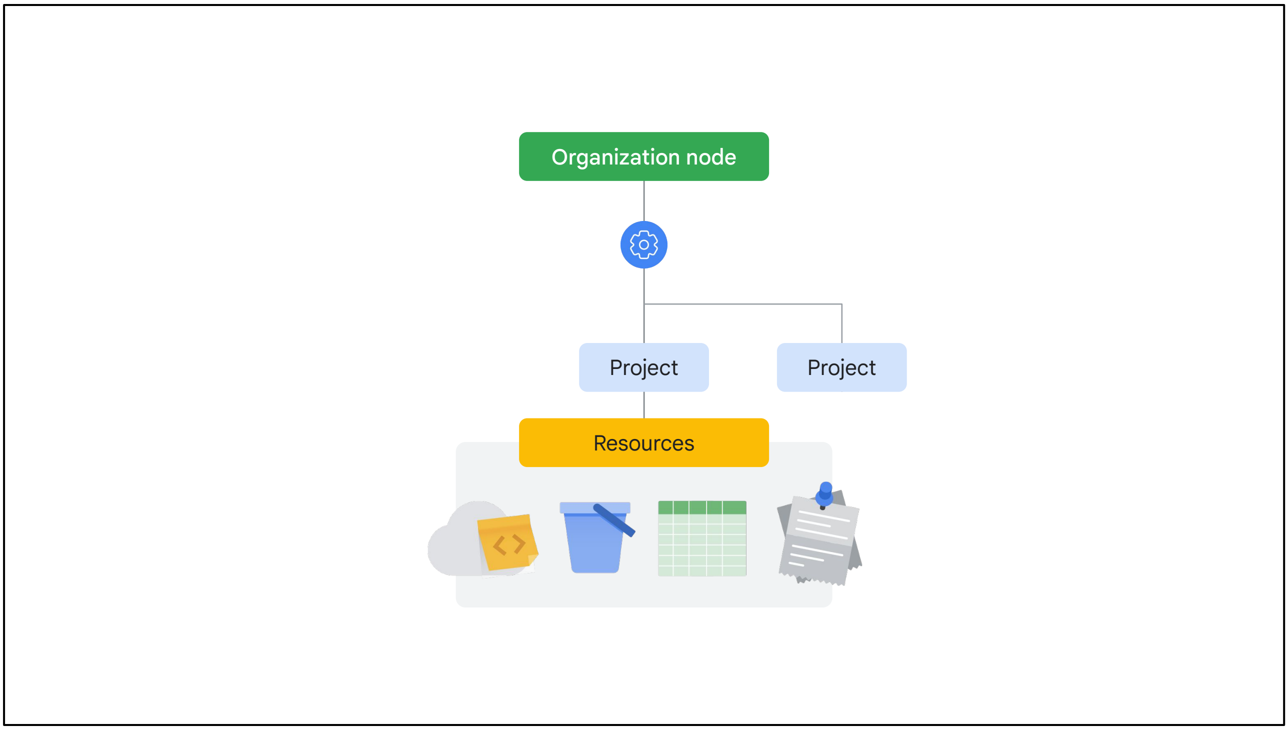
For example, your organization might contain multiple departments, each with its own set of Google Cloud resources.

Folders allow you to group these resources on a per-department basis.

Folders also give teams the ability to delegate administrative rights so that they can work independently.



As previously mentioned, the resources in a folder inherit policies and permissions from that folder.



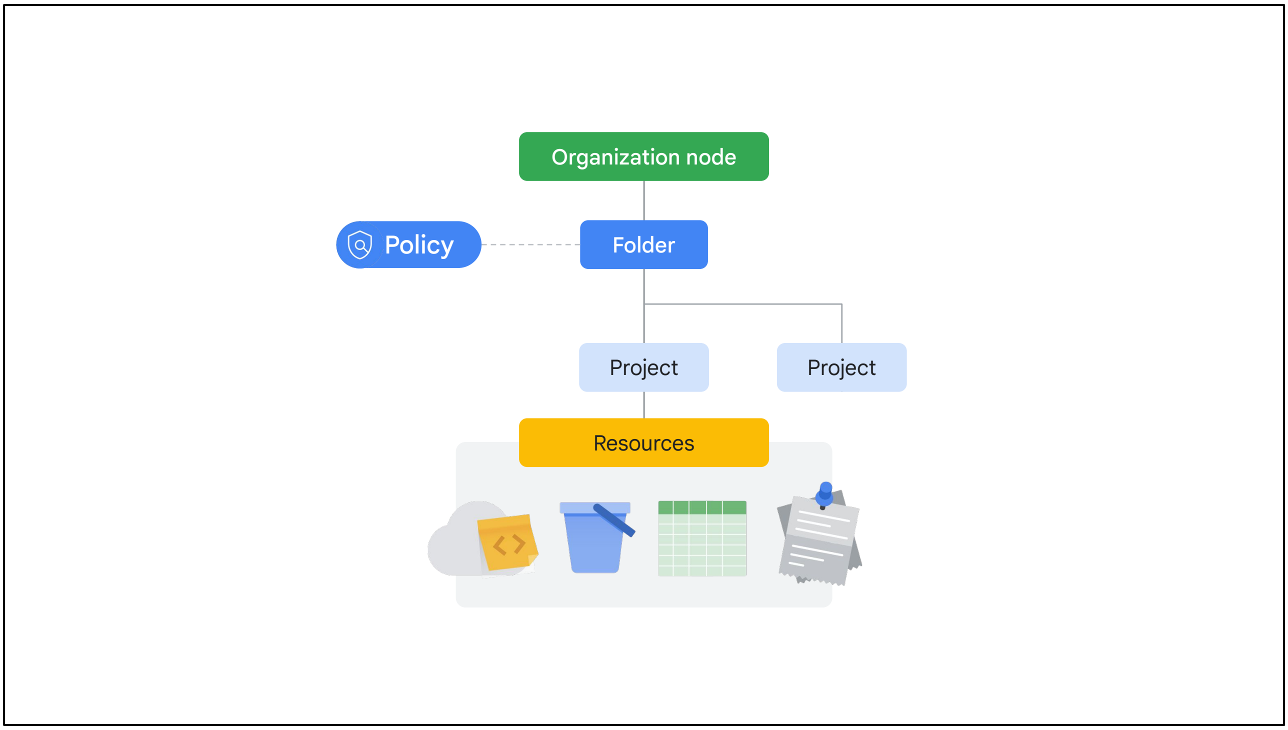
Organization node

Project

Project

Resources

For example, if you have two different projects that are administered by the same team,



Organization node

Policy

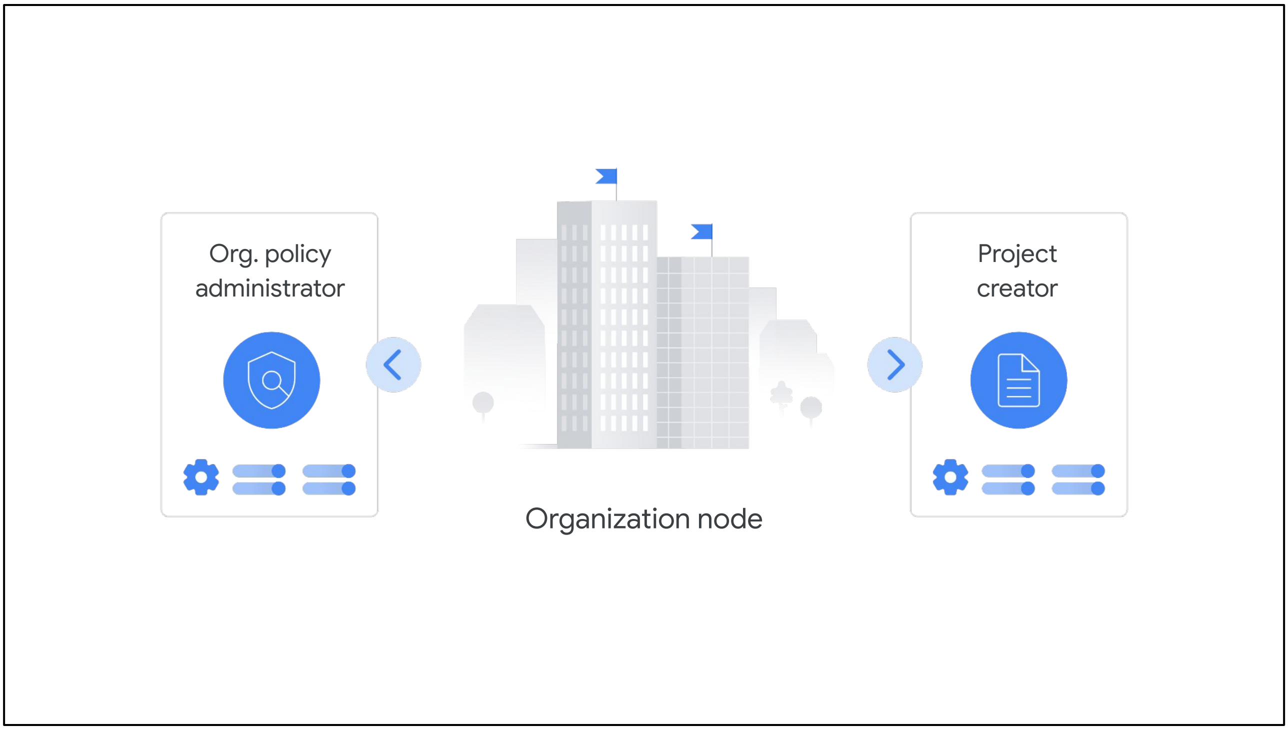
Folder

Project

Project

Resources

you can put policies into a common folder so they have the same permissions.



Org. policy administrator

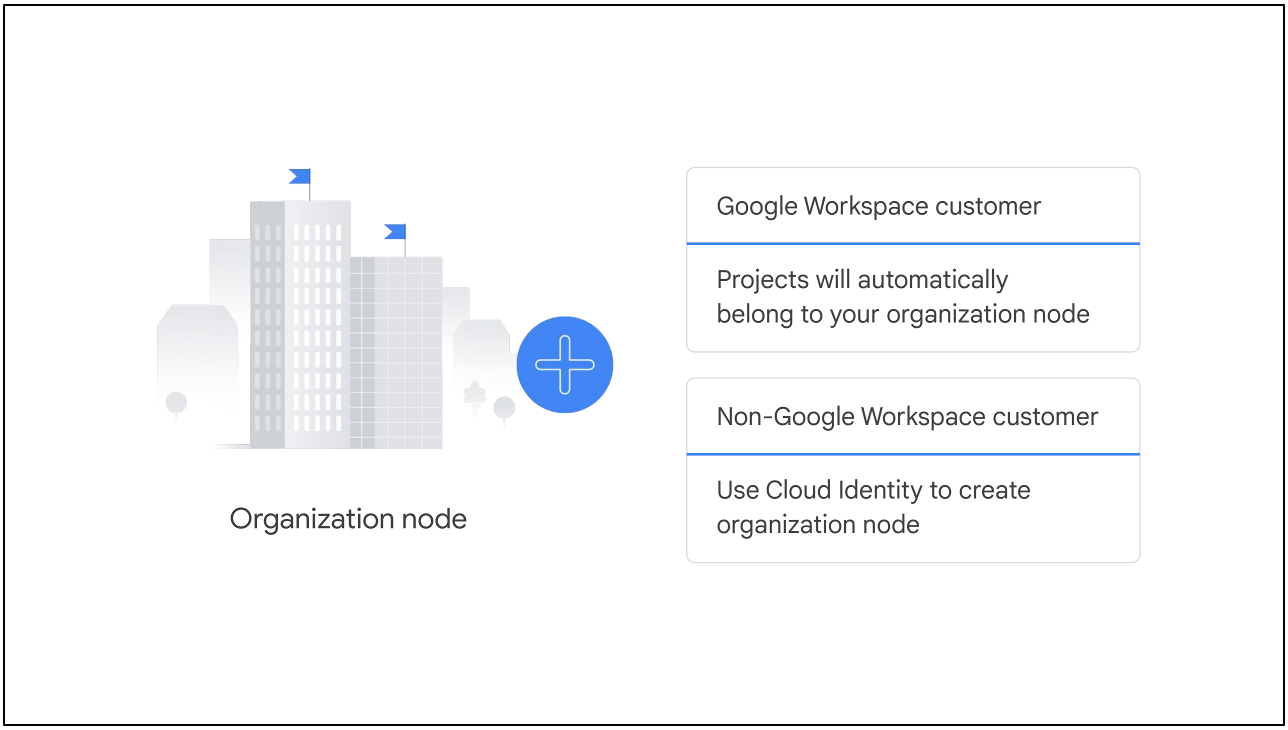
Organization node

Project creator

There are some special roles associated with this top-level organization node.

For example, you can designate an *organization policy administrator* so that only people with privilege can change policies.

You can also assign a *project creator* role, which is a great way to control who can create projects and, therefore, who can spend money.



Organization node

Google Workspace customer

Projects will automatically belong to your organization node

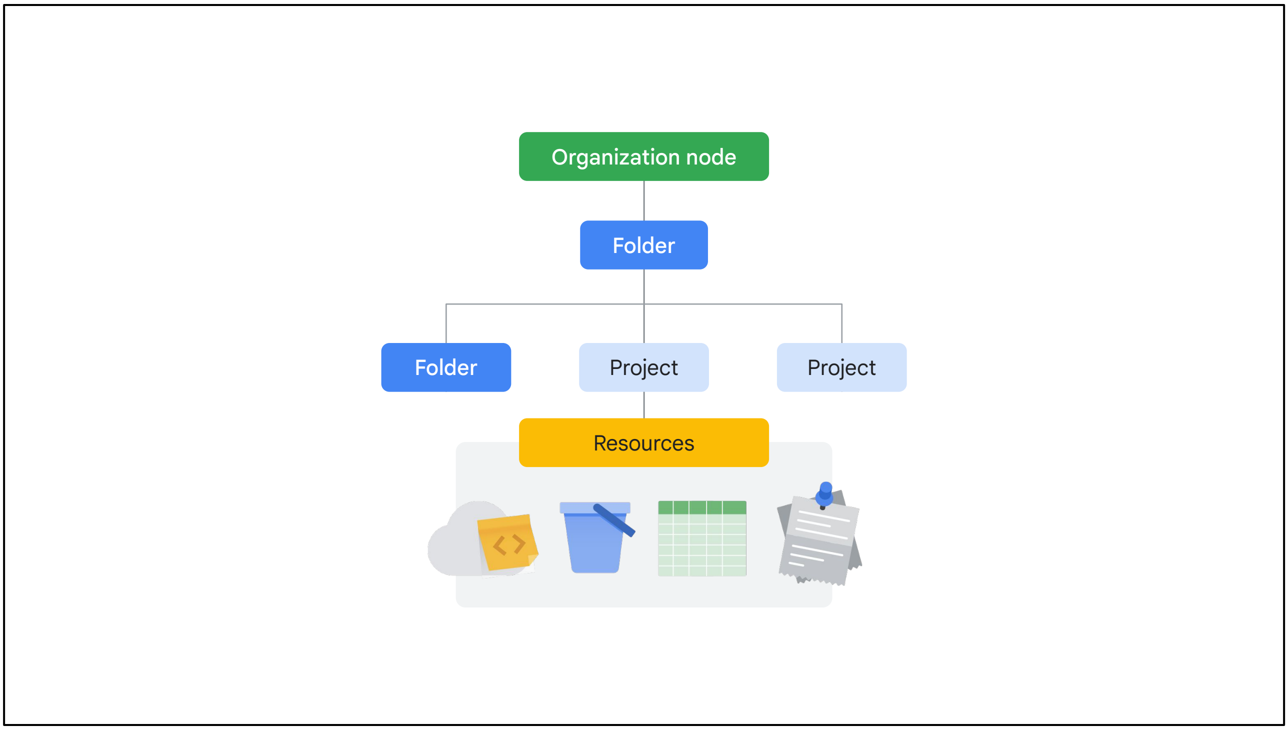
Non-Google Workspace customer

Use Cloud Identity to create organization node

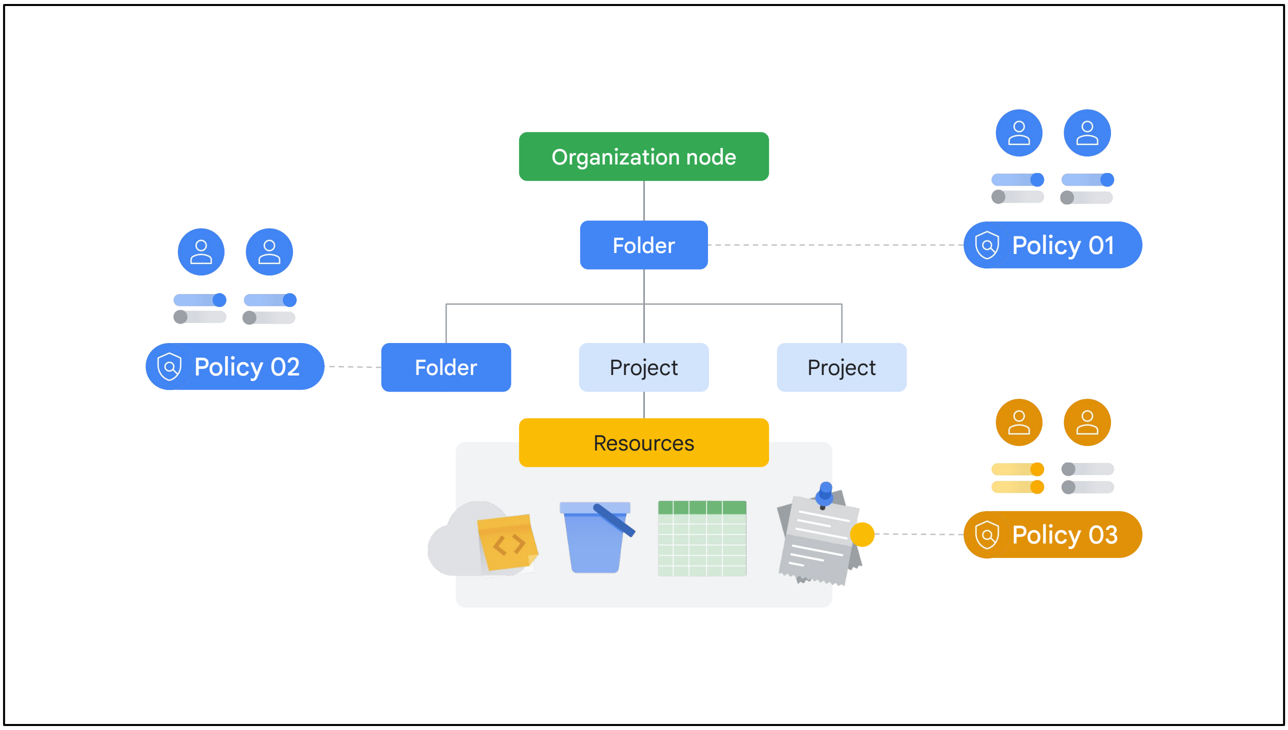
How a new organization node is created depends on whether your company is also a Google Workspace customer.

If you have a Workspace domain, Google Cloud projects will automatically belong to your organization node.

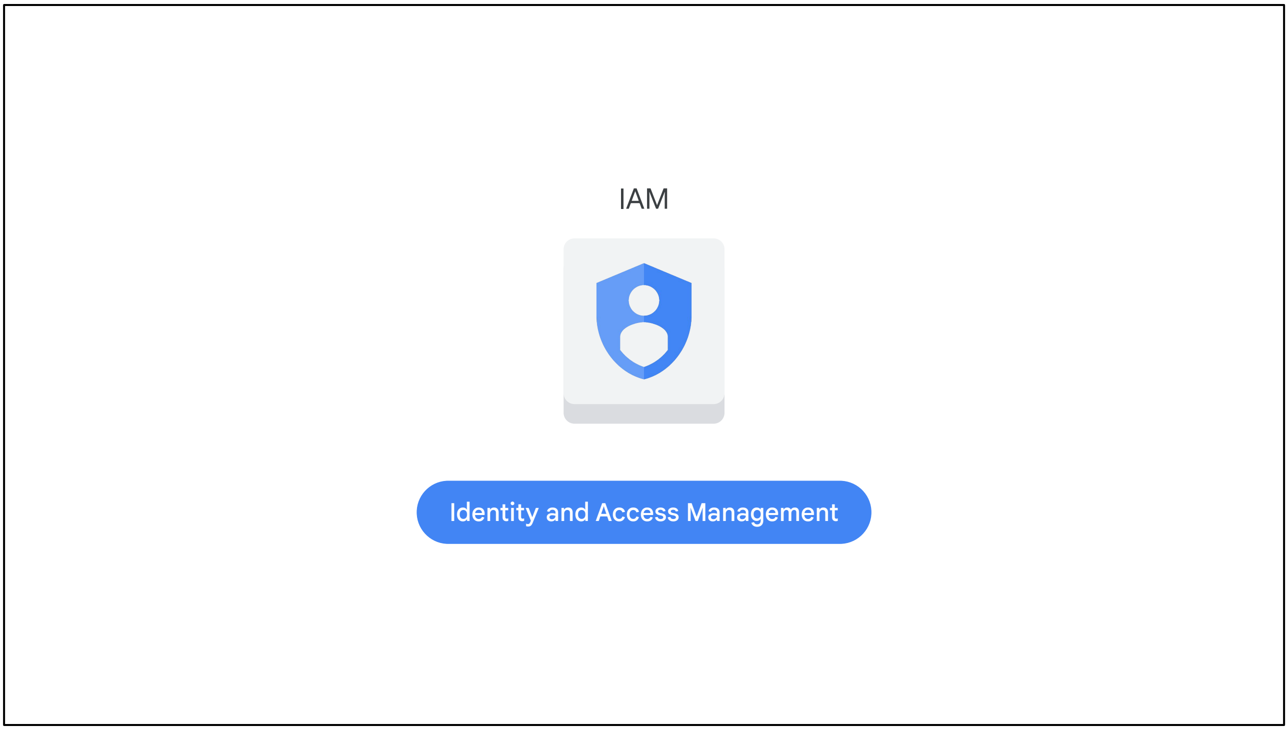
Otherwise, you can use Cloud Identity, Google’s identity, access, application, and endpoint management platform, to generate one.



When an organization node contains lots of folders, projects, and resources, a workforce



might need to restrict who has access to what.



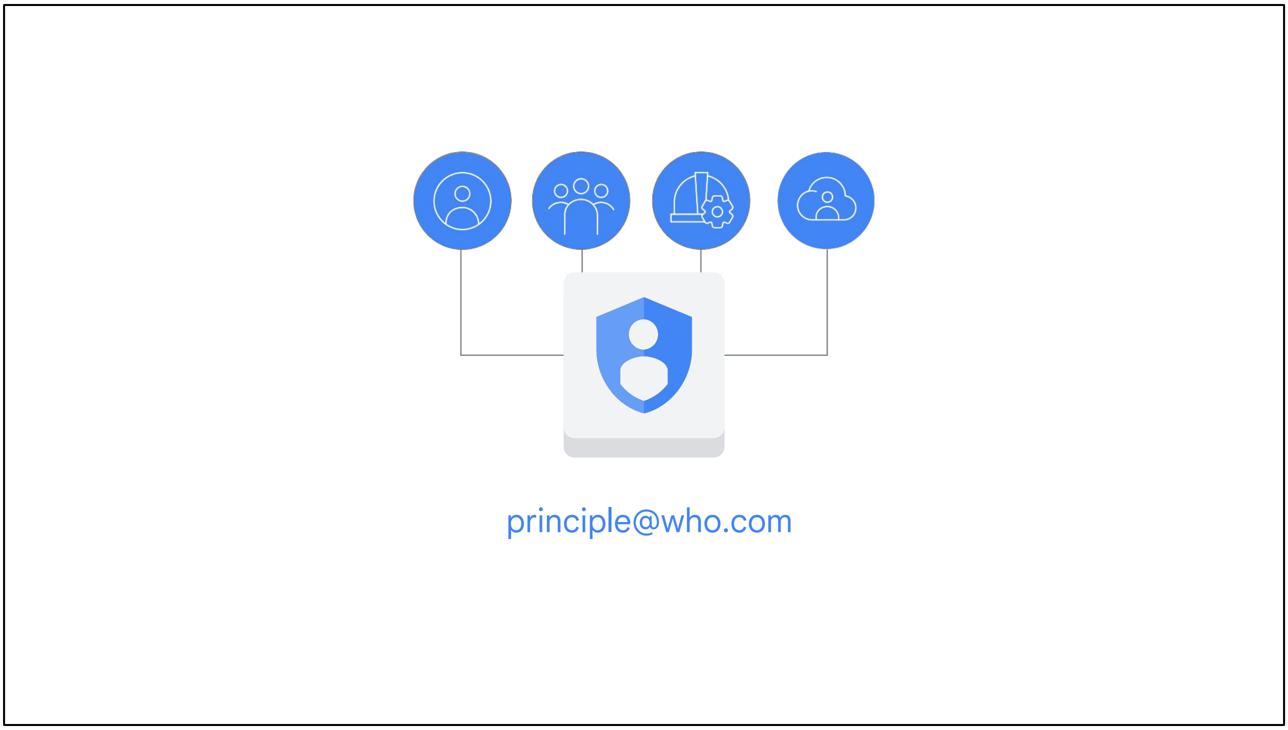
IAM

Identity and Access Management

To help with this task, administrators can use **Identity and Access Management**, or IAM.

Administrators can apply policies that define
who can do what on which resources

With IAM, administrators can apply policies that define who can do what and on which resources.

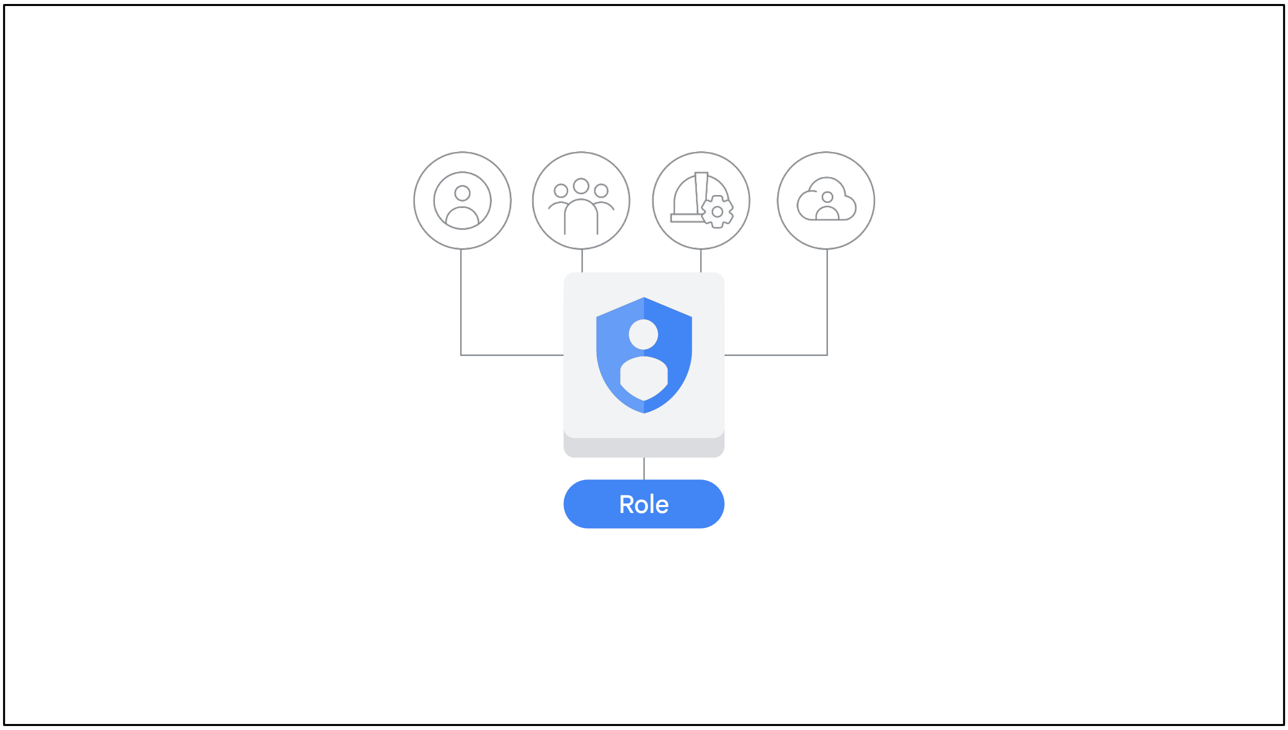


principle@who.com

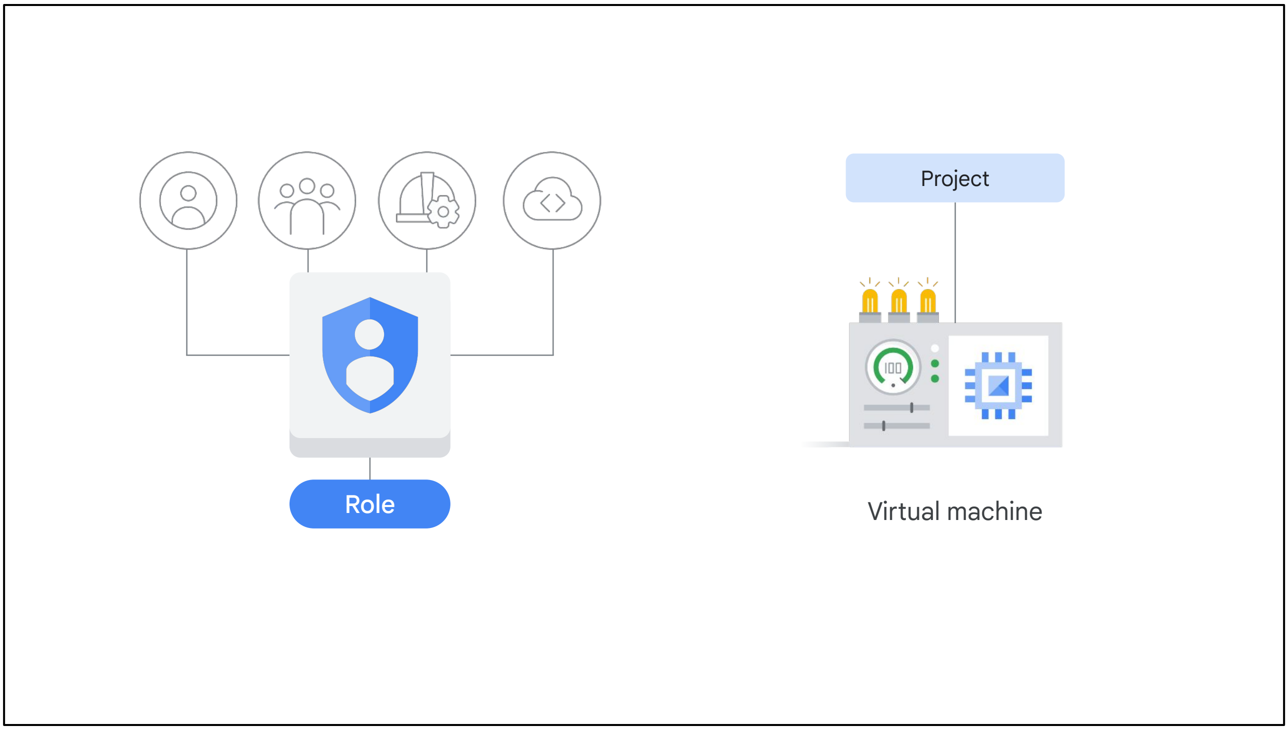
The “who” part of an IAM policy can be:

- a Google account,
- a Google group,
- a service account, or
- a Cloud Identity domain.

A “who” is also called a “principal.” Each principle has its own identifier, usually an email address.



The “can do what” part of an IAM policy is defined by a role. An IAM role is a collection of permissions. When you grant a role to a principal, you grant all the permissions that the role contains.

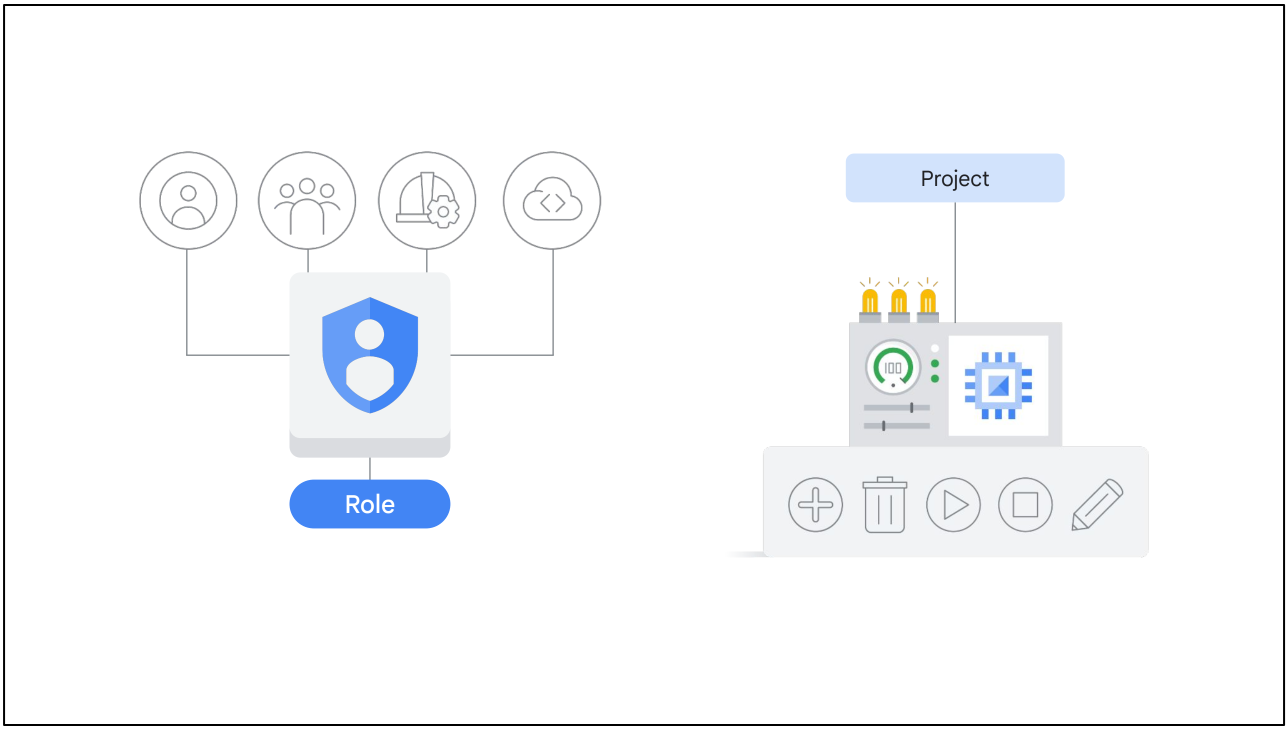


Role

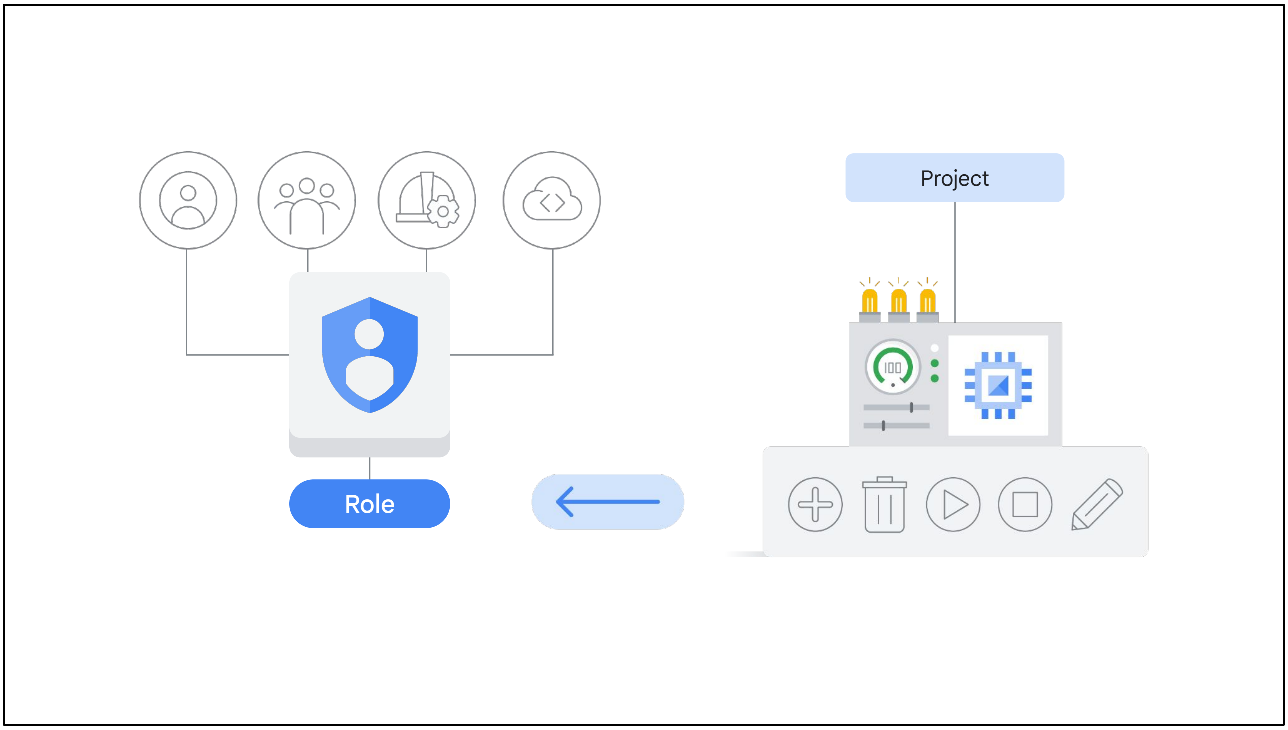
Project

Virtual machine

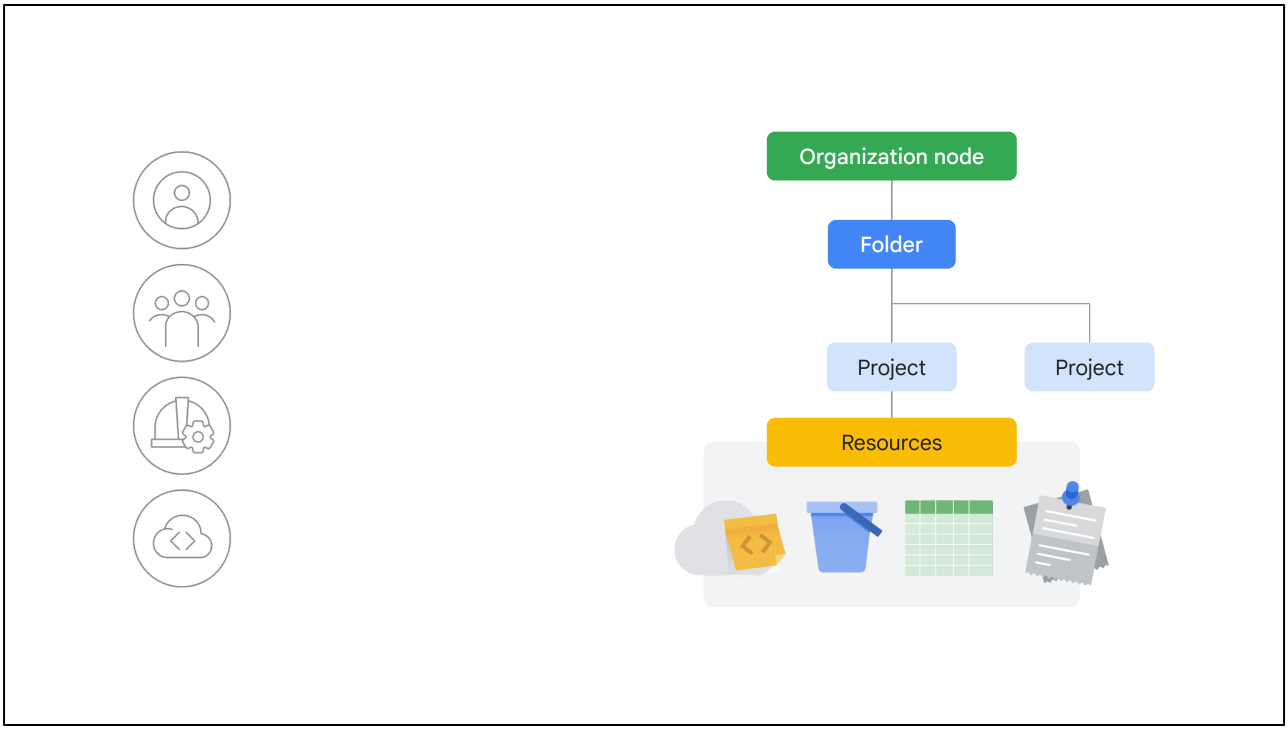
For example, to manage virtual machine instances in a project, you must be able to to



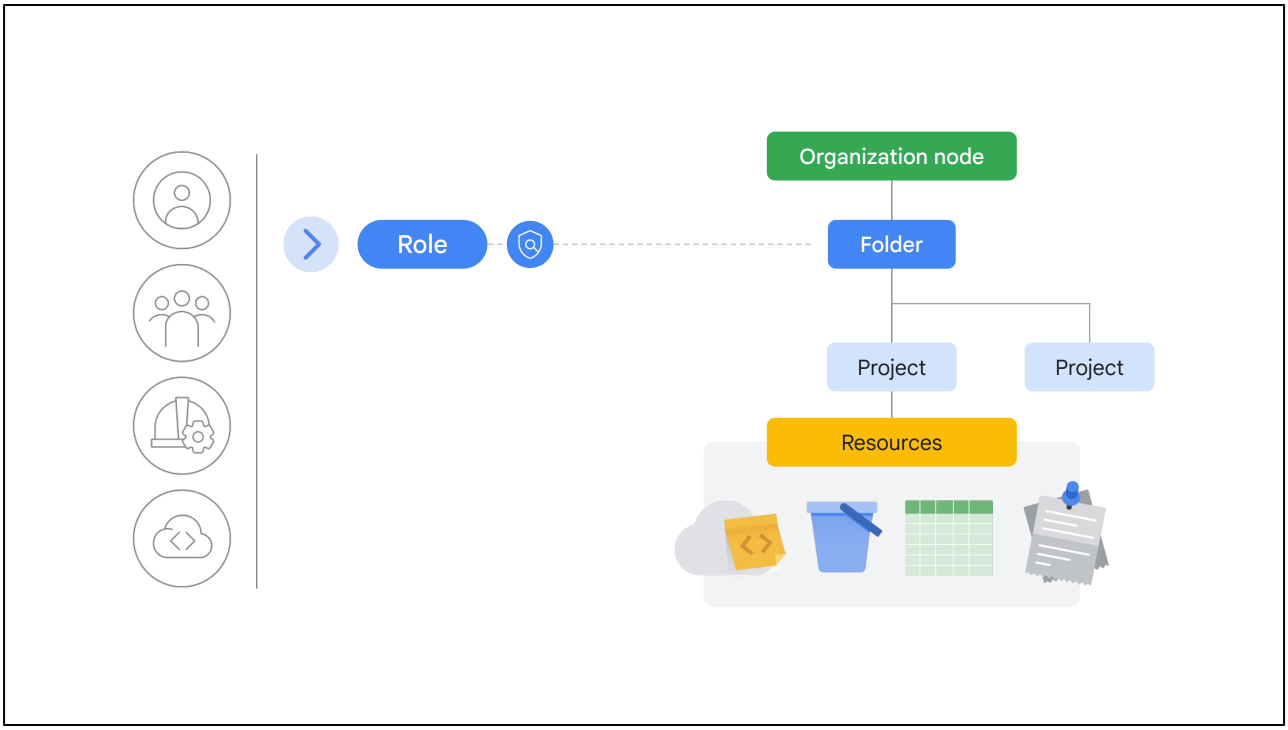
create, delete, start, stop and change virtual machines. So these permissions are



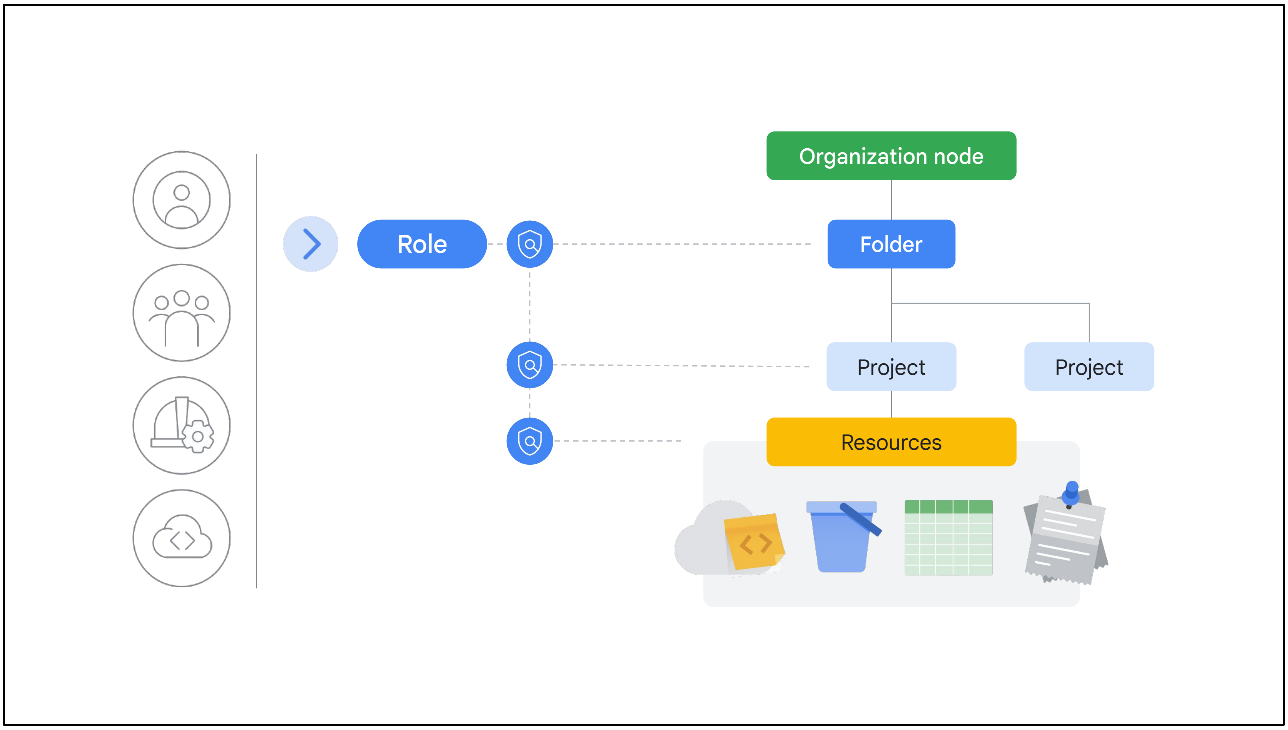
grouped into a role to make them easier to understand and easier to manage.



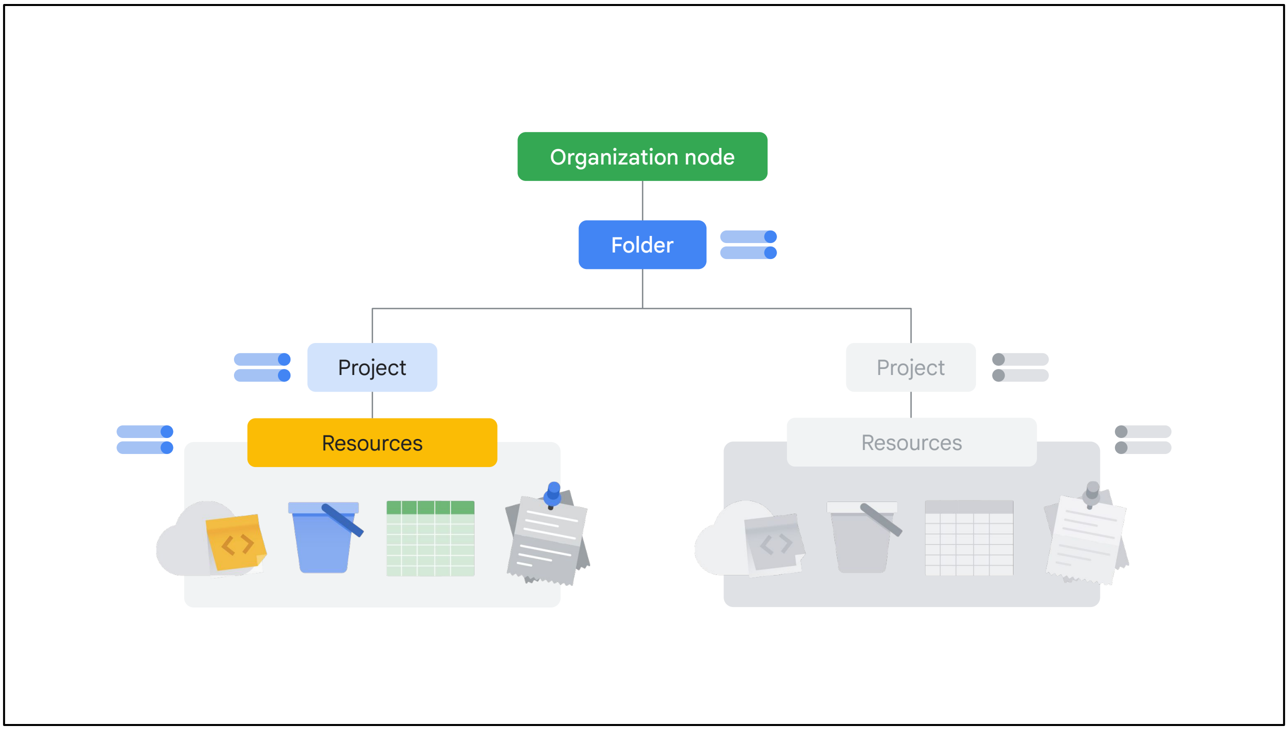
When a principal



is given a role on a specific element of the resource hierarchy, the resulting policy



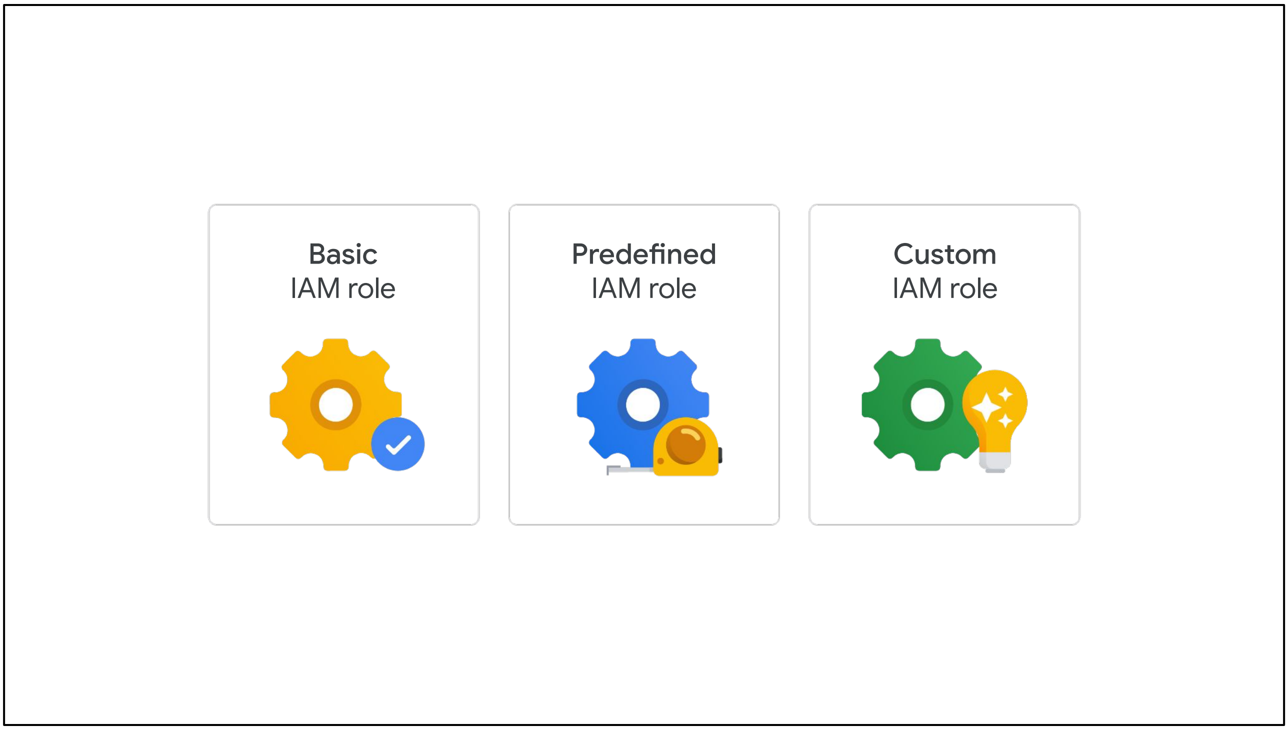
applies to both the chosen element and all the elements below it in the hierarchy.



You can define deny rules that prevent certain principals from using certain permissions, regardless of the roles they're granted.

This is because IAM always checks relevant deny policies before checking relevant allow policies.

Deny policies, like allow policies, are inherited through the resource hierarchy.

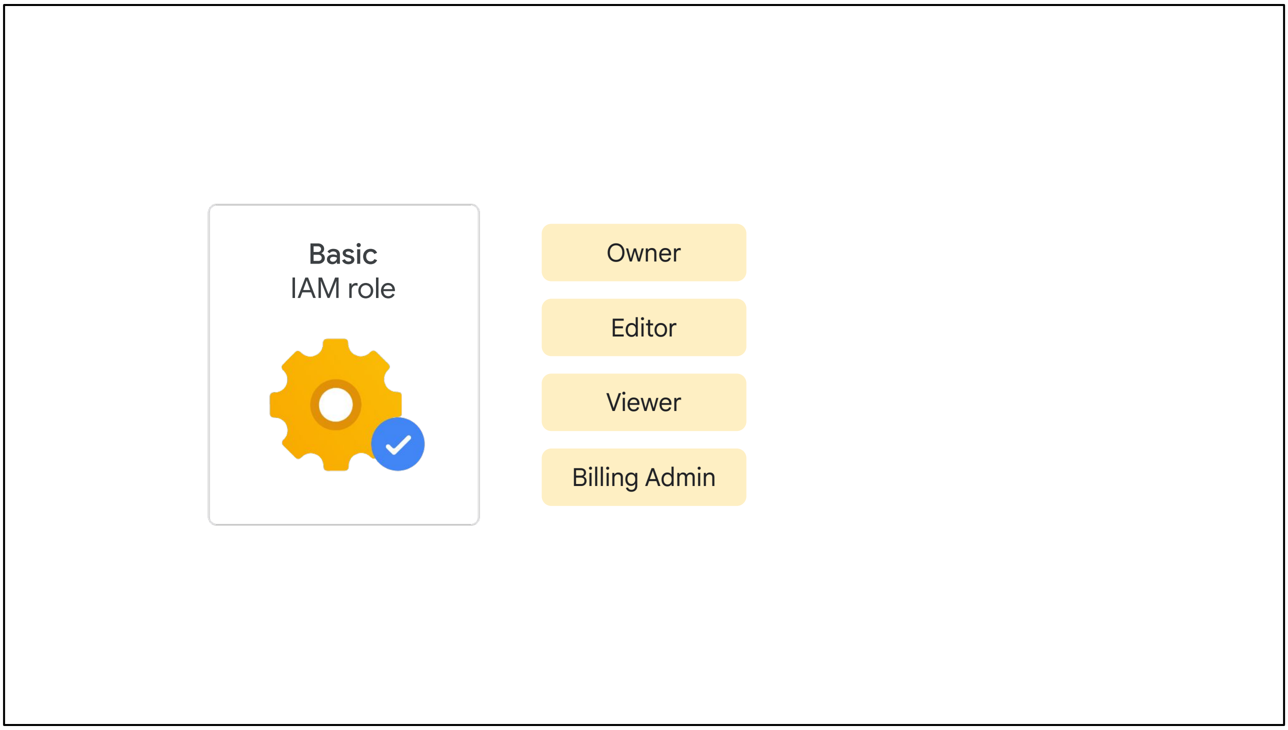


Basic
IAM role

Predefined
IAM role

Custom
IAM role

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



Basic
IAM role

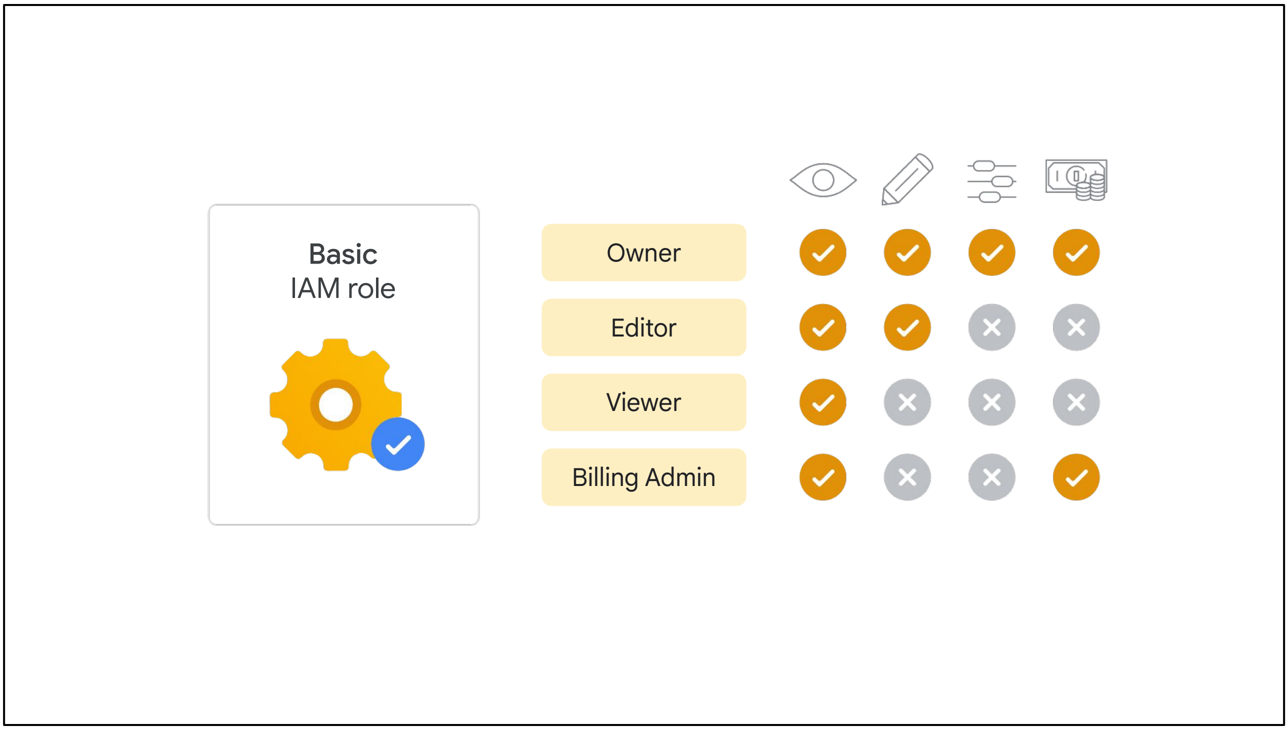
Owner

Editor

Viewer

Billing Admin

Basic roles include owner, editor, viewer, and billing administrator. Let’s look at these basic roles in a bit more detail.



Basic
IAM role

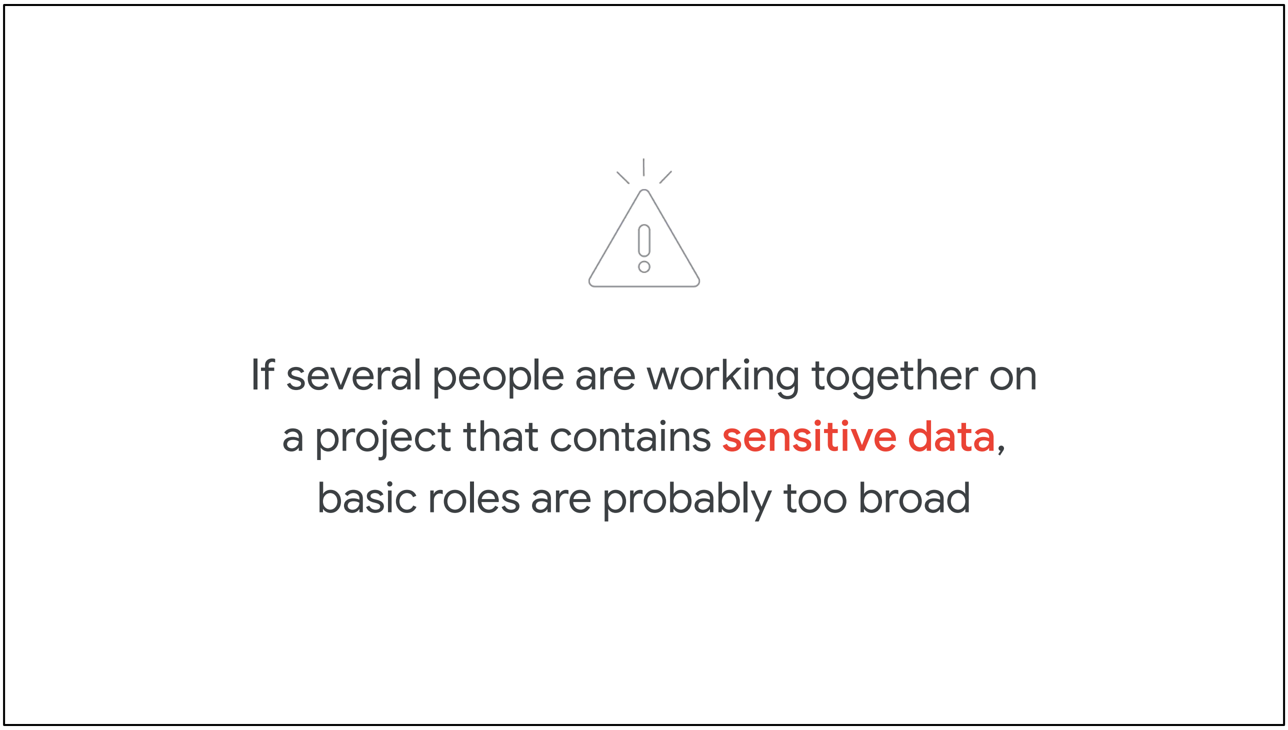
Owner	✓	✓	✓	✓
Editor	✓	✓	✗	✗
Viewer	✓	✗	✗	✗
Billing Admin	✓	✗	✗	✓

Project **viewers** can access resources but can’t make changes.

Project **editors** can access and make changes to a resource.

And project **owners** can also access and make changes to a resource. In addition, project owners can manage the associated roles and permissions and set up billing.

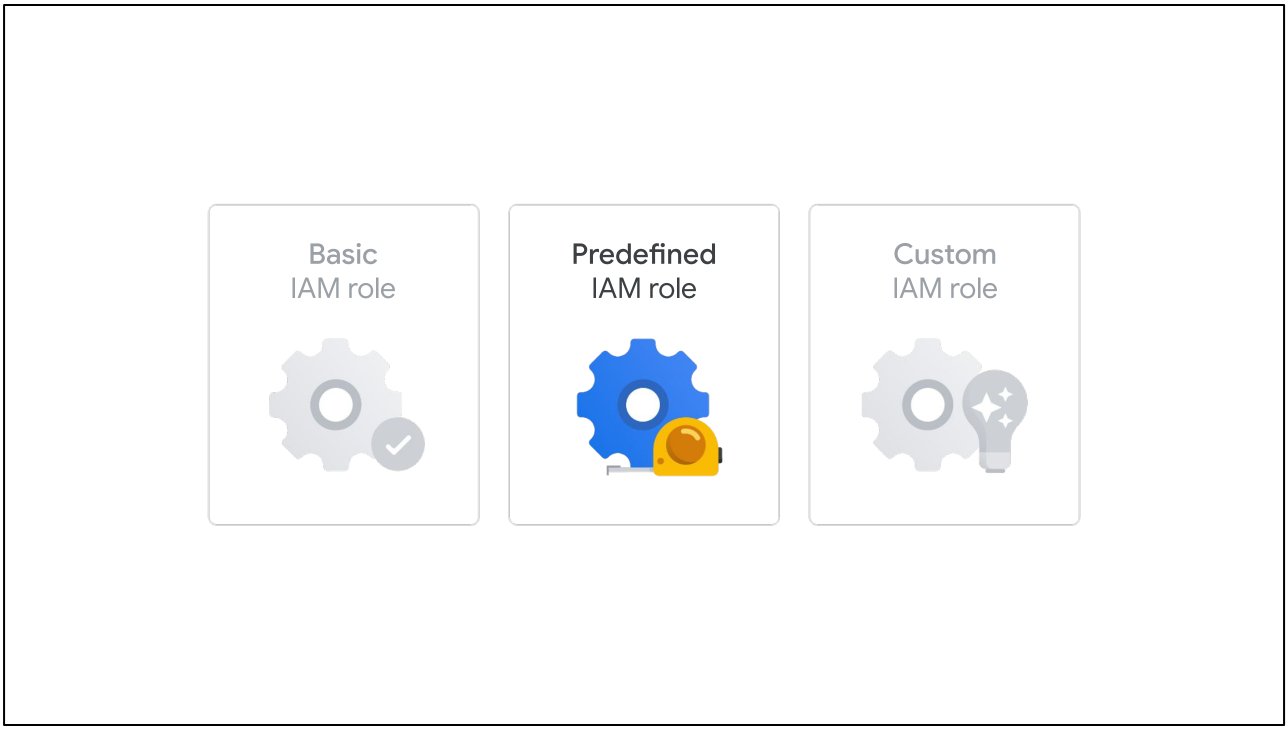
Often companies want someone to control the billing for a project but not be able to change the resources in the project. This is possible through a **billing administrator** role.



If several people are working together on a project that contains **sensitive data**, basic roles are probably too broad

A word of caution: If several people are working together on a project that contains sensitive data, basic roles are probably too broad.

Fortunately, IAM provides other ways to assign permissions that are more specifically tailored to meet the needs of typical job roles.



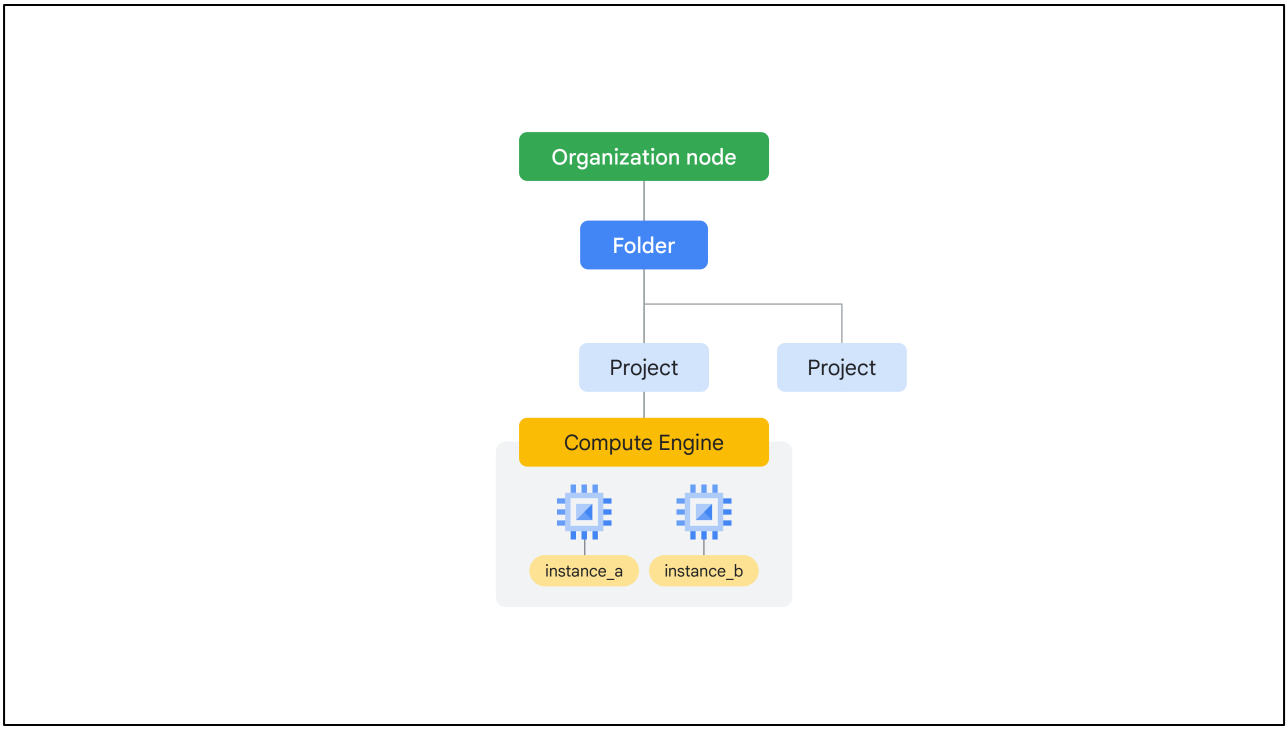
Basic
IAM role

Predefined
IAM role

Custom
IAM role

This brings us to the second type of role, **predefined** roles.

Specific Google Cloud services offer sets of predefined roles, and they even define where those roles can be applied.



Organization node

Folder

Project

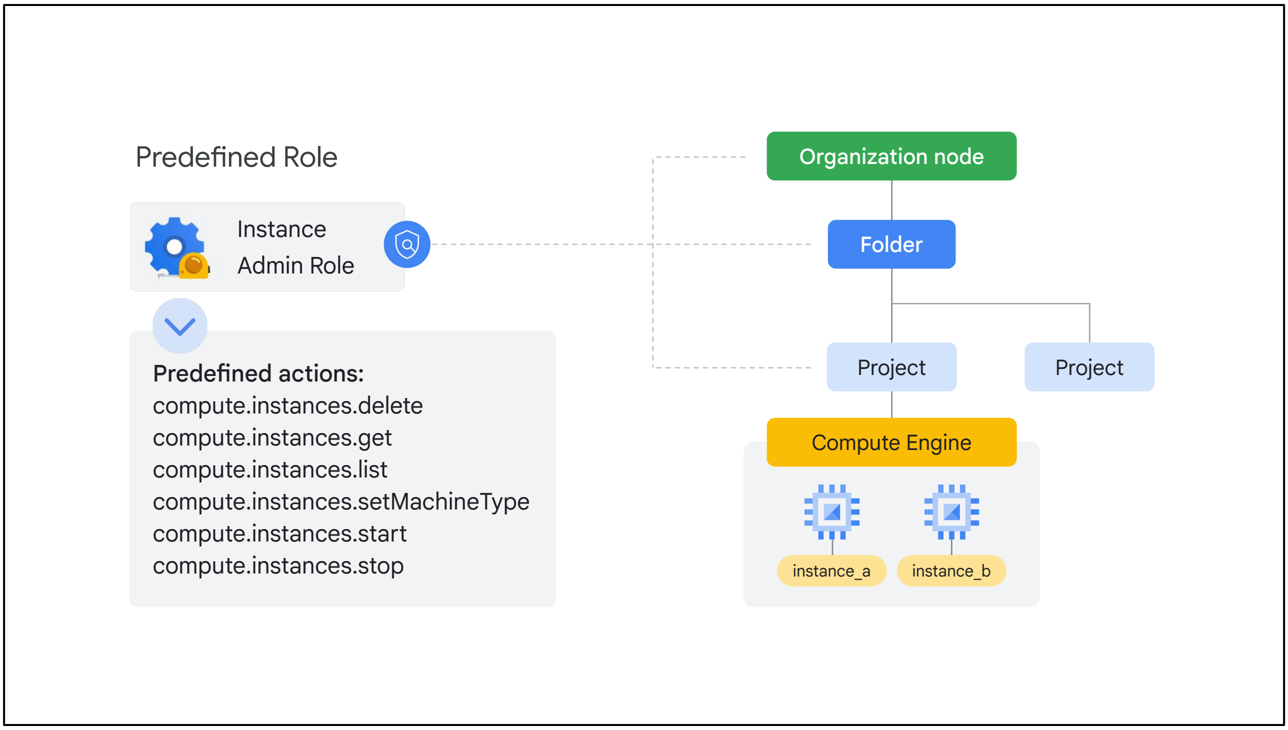
Project

Compute Engine

instance_a

instance_b

Let’s look at Compute Engine, for example, a Google Cloud product that offers virtual machines as a service.



Predefined Role

Instance Admin Role

Predefined actions:
compute.instances.delete
compute.instances.get
compute.instances.list
compute.instances.setMachineType
compute.instances.start
compute.instances.stop

Organization node

Folder

Project

Project

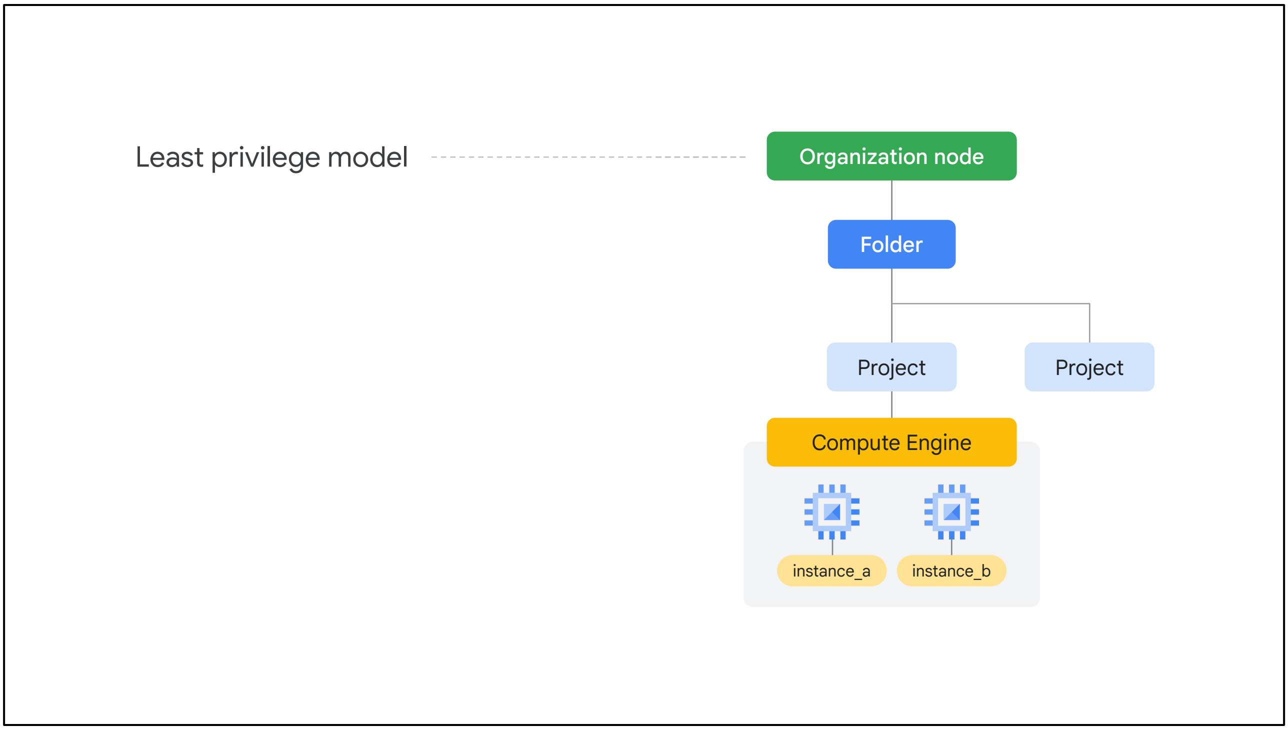
Compute Engine

instance_a

instance_b

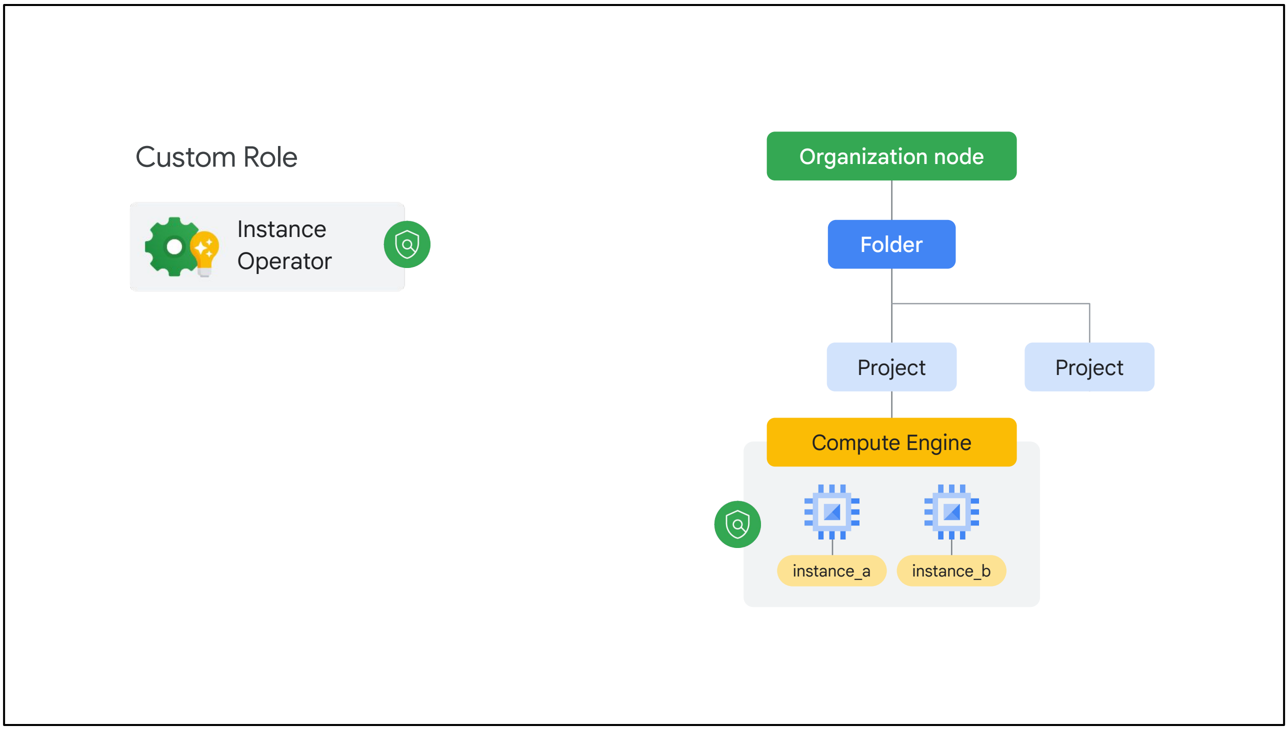
With Compute Engine, you can apply specific predefined roles—such as “instanceAdmin”—to Compute Engine resources in a given project, a given folder, or an entire organization.

This then allows whoever has these roles to perform a specific set of predefined actions.

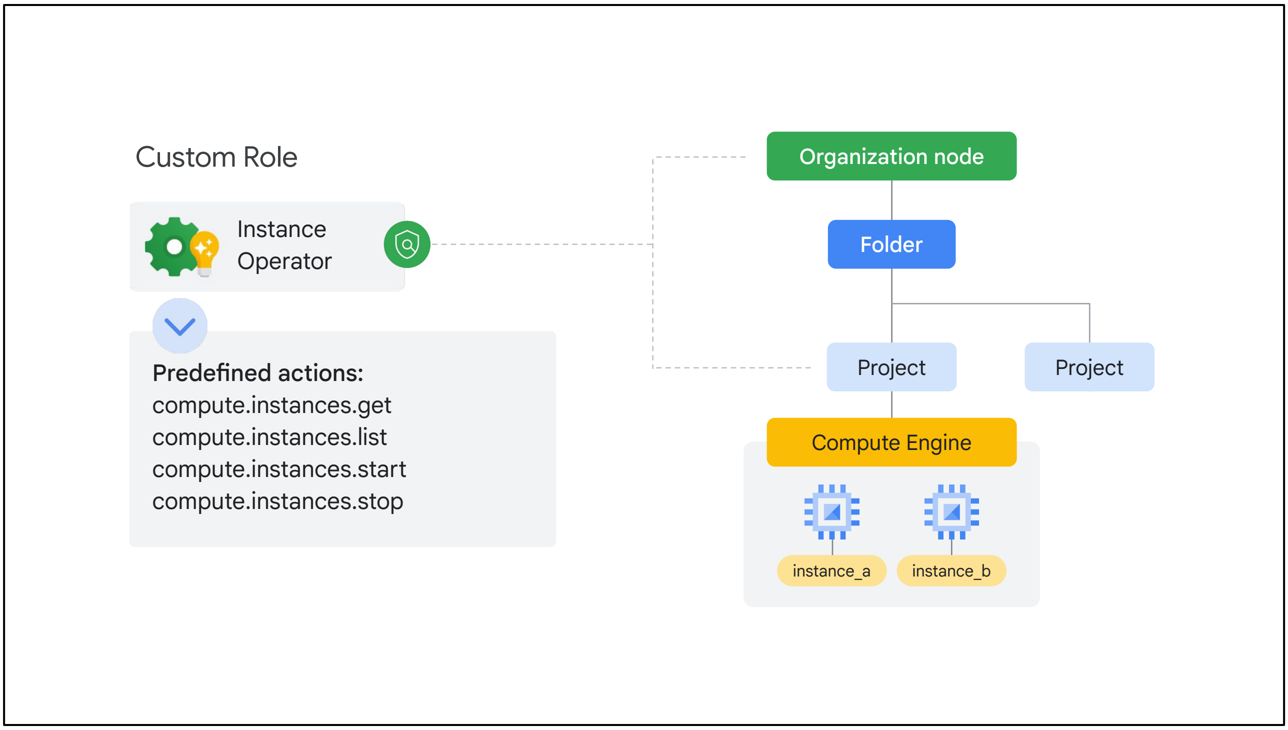


Least privilege model

Many companies use a “least-privilege” model in which each person in your organization is given the *minimal amount of privilege needed* to do their job.



So, for example, maybe you want to define an “instanceOperator” role to allow

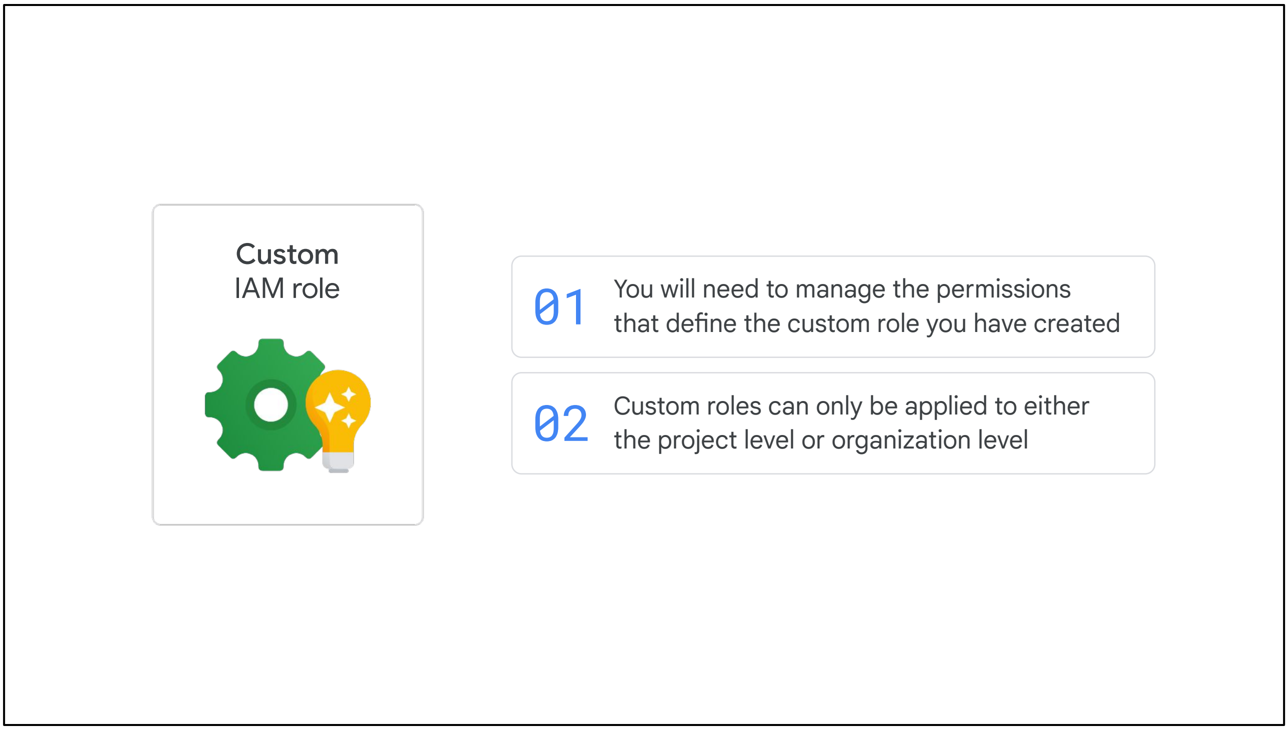


Custom Role

Instance Operator

Predefined actions:
compute.instances.get
compute.instances.list
compute.instances.start
compute.instances.stop

some users to stop and start Compute Engine virtual machines, but not reconfigure them. Custom roles will allow you to define those exact permissions.



Custom IAM role

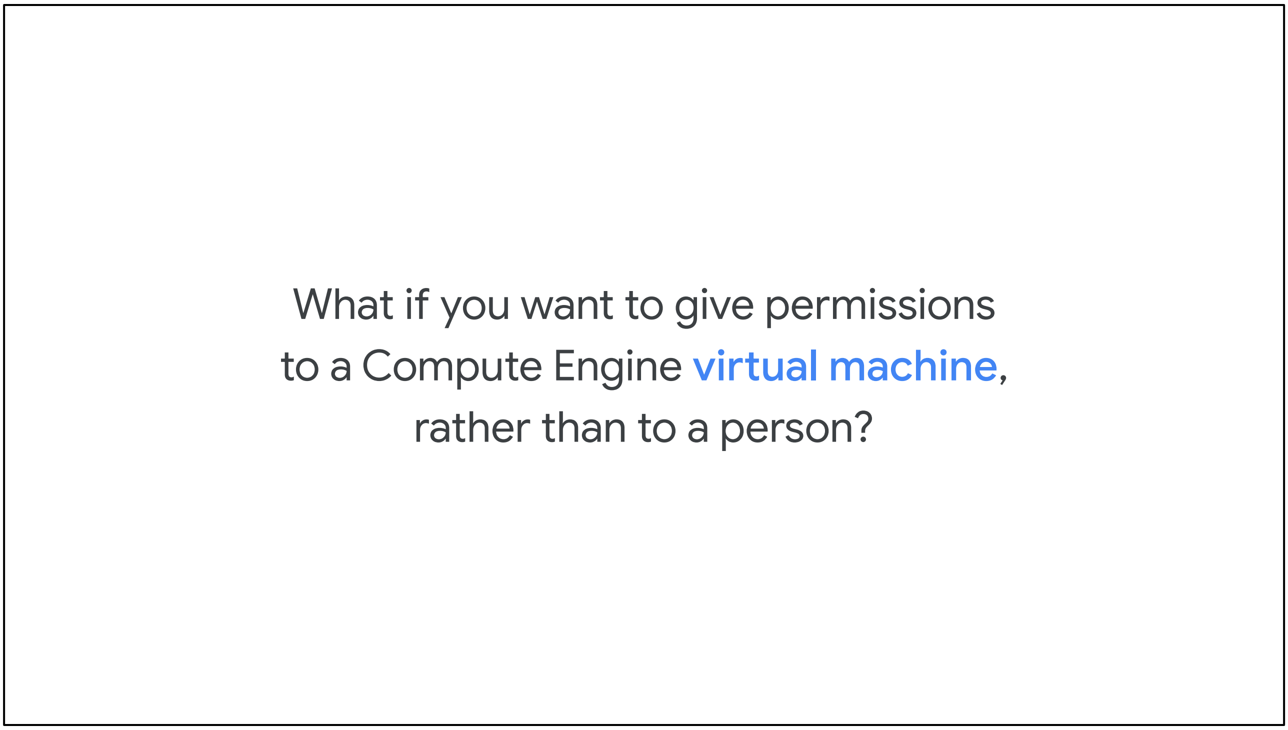
01 You will need to manage the permissions that define the custom role you have created

02 Custom roles can only be applied to either the project level or organization level

Before you start creating custom roles, please note two important details.

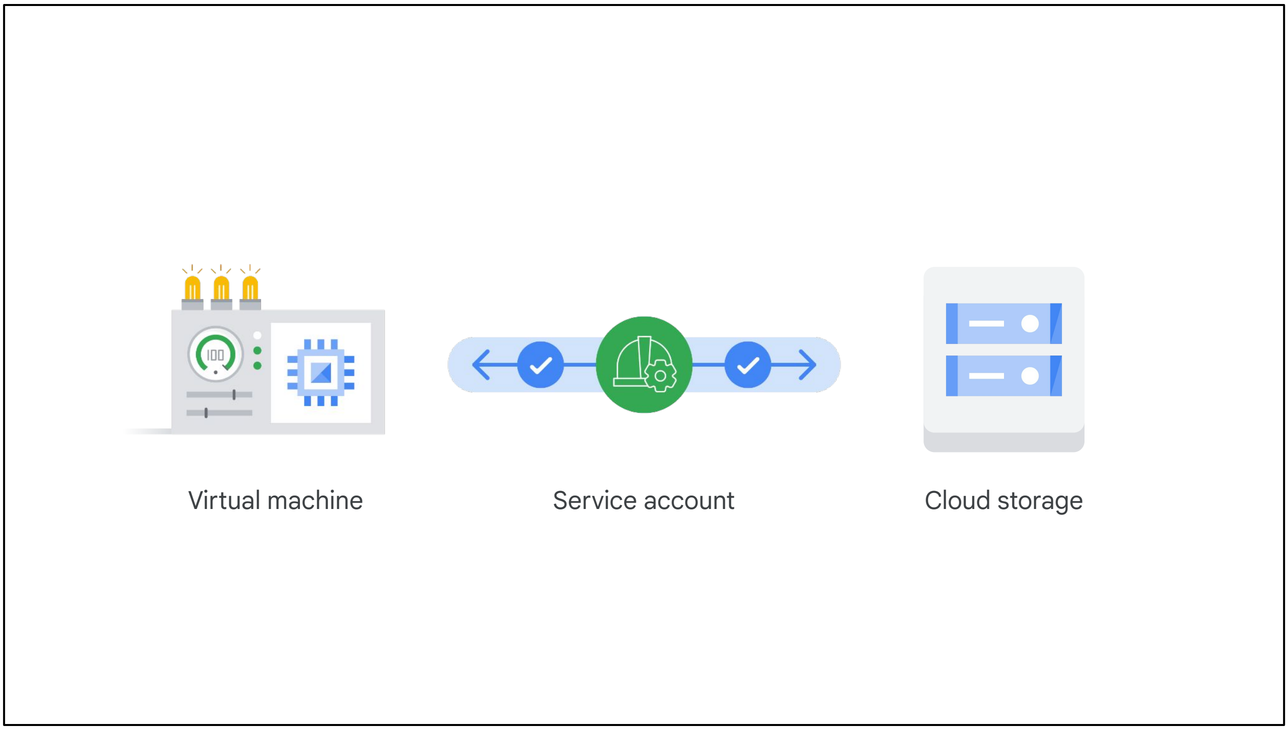
- First, you'll need to manage the permissions that define the custom role you've created. Because of this, some organizations decide they'd rather use the predefined roles.

- And second, custom roles can only be applied to either the project level or organization level. They can't be applied to the folder level.



What if you want to give permissions to a Compute Engine virtual machine, rather than to a person?

What if you want to give permissions to a Compute Engine virtual machine, rather than to a person? Well, that’s what **service accounts** are for.

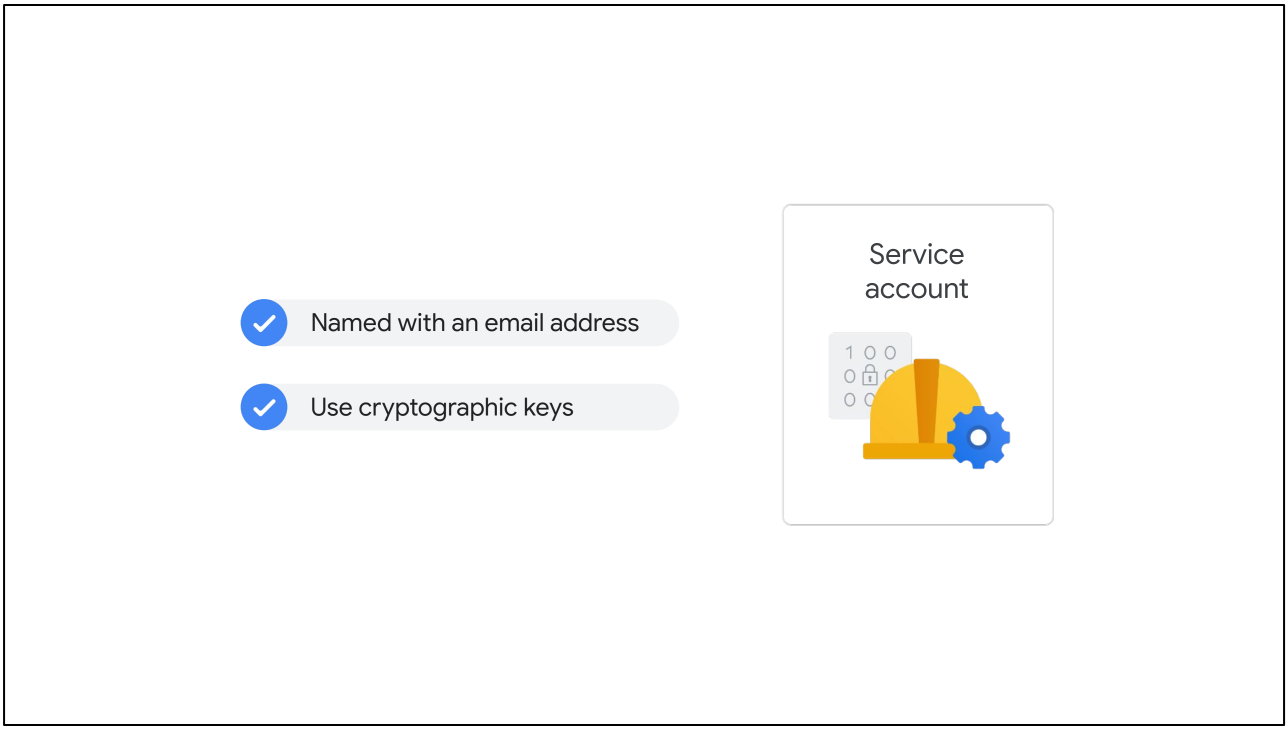


Virtual machine

Service account

Cloud storage

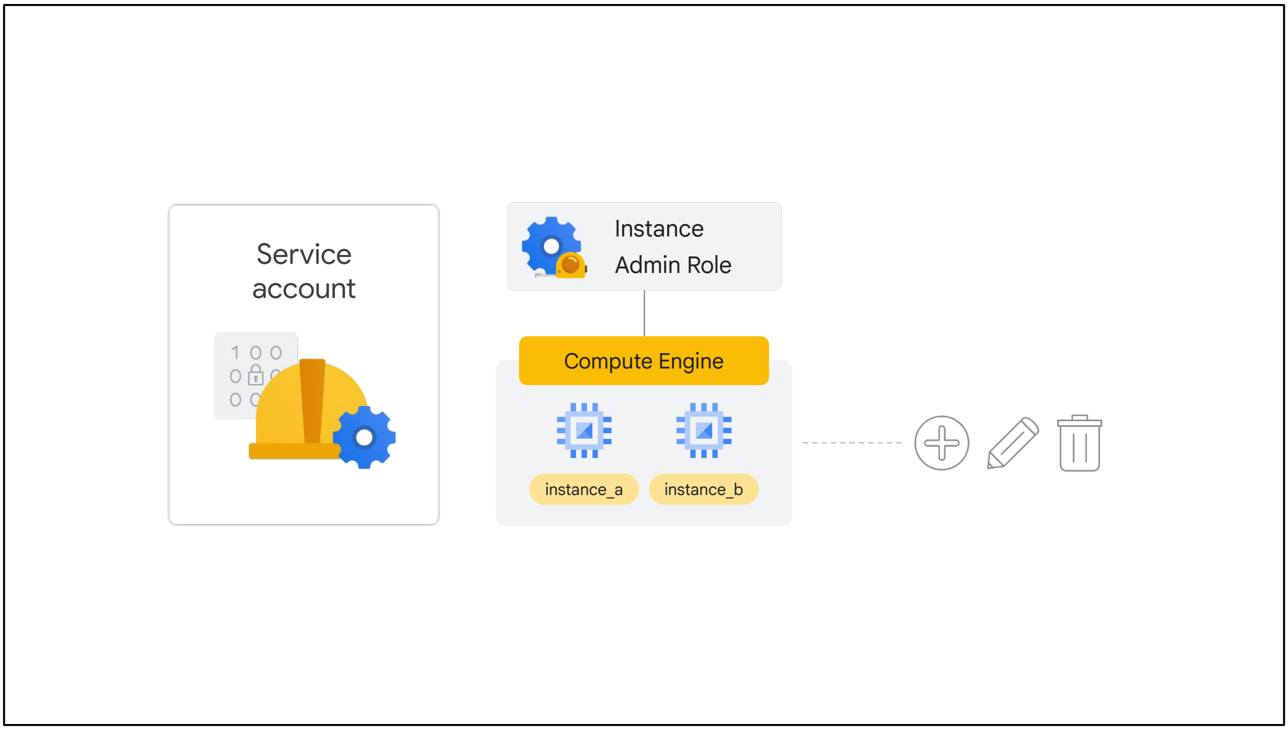
You can create a service account to authenticate that VM to Cloud Storage.



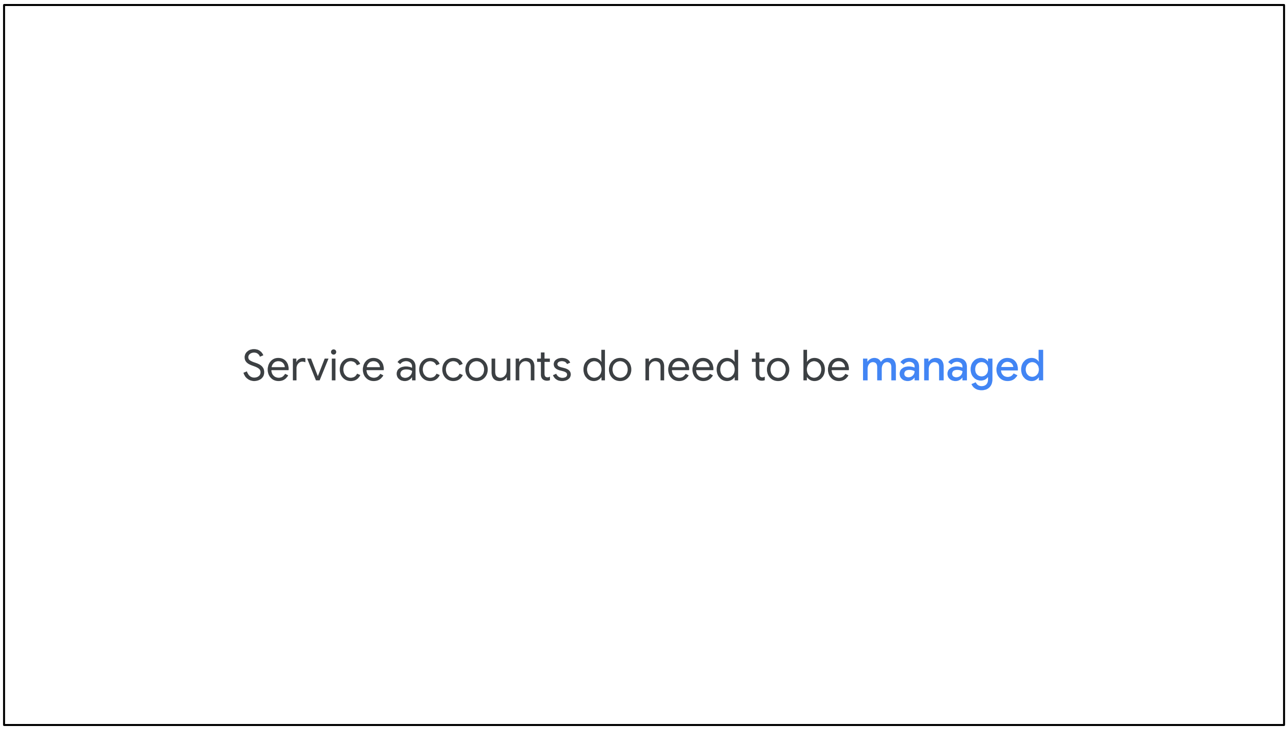
- ✓ Named with an email address
- ✓ Use cryptographic keys

Service account

Service accounts are named with an email address, but instead of passwords they use cryptographic keys to access resources.

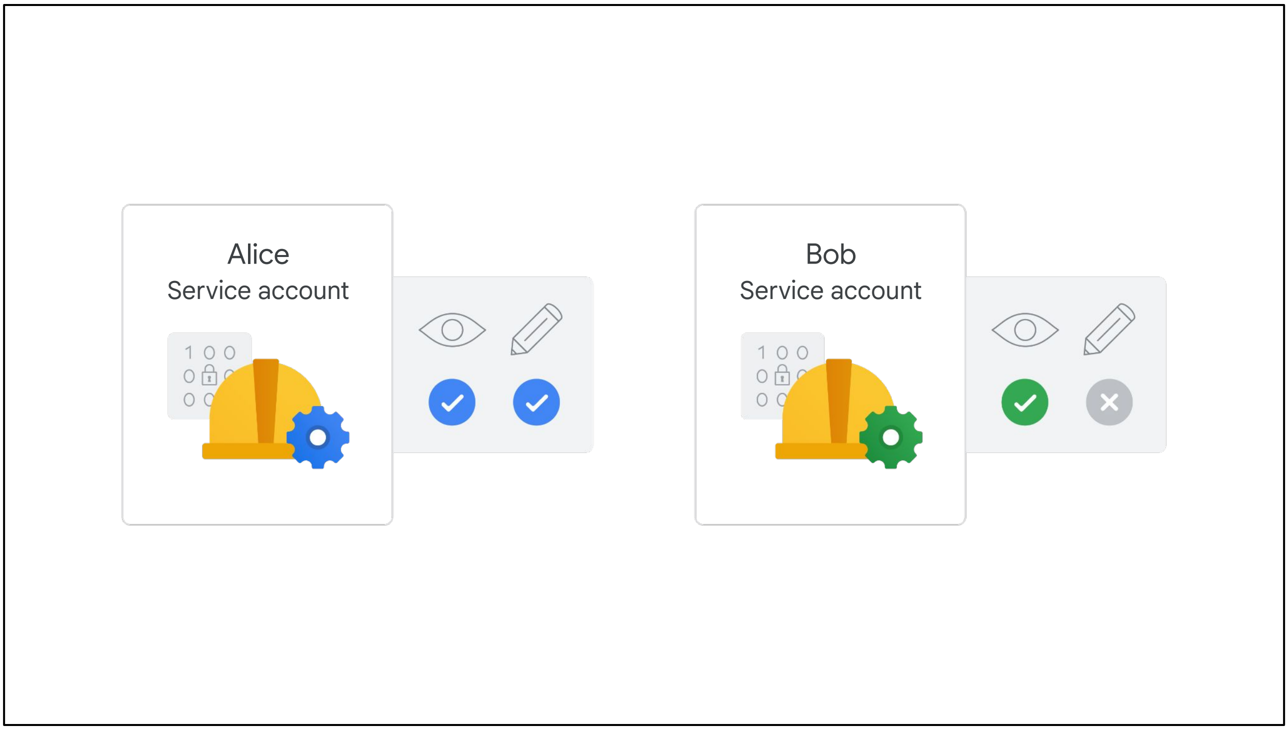


So, if a service account has been granted Compute Engine’s Instance Admin role, this would allow an application running in a VM with that service account to create, modify, and delete other VMs.

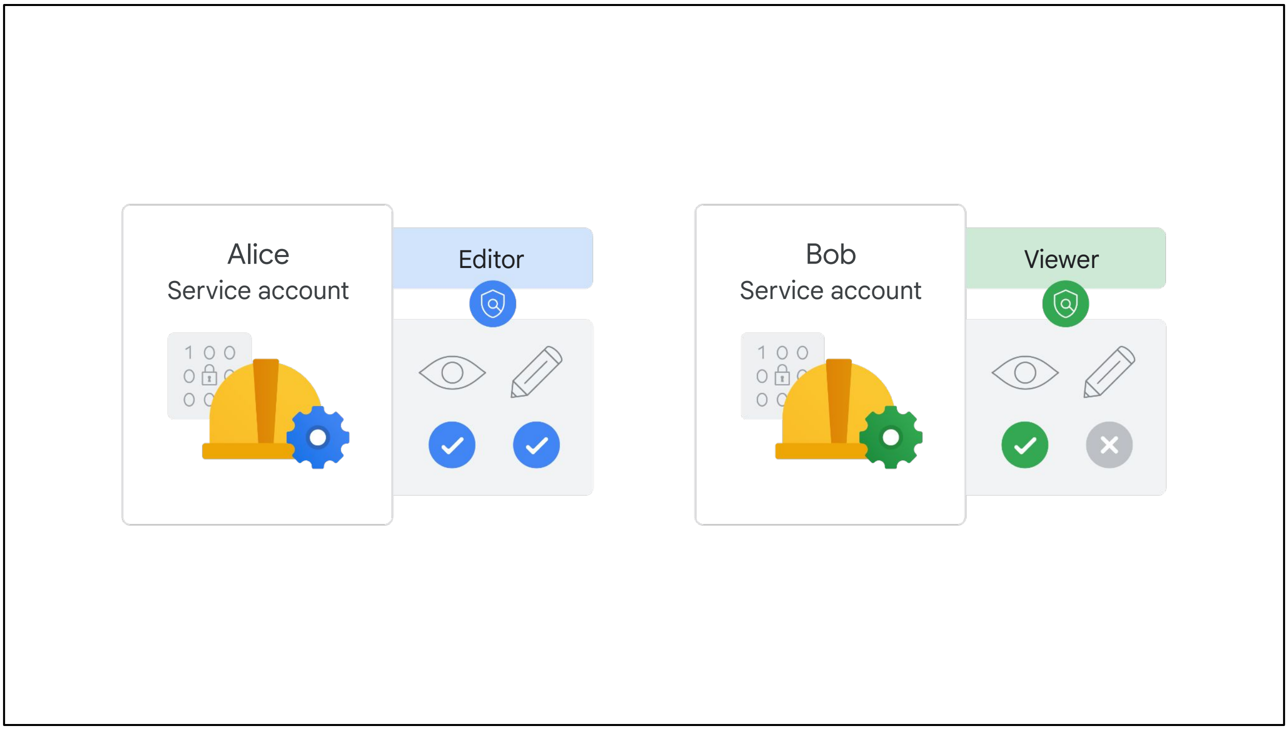


Service accounts do need to be managed

Service accounts *do* need to be managed.

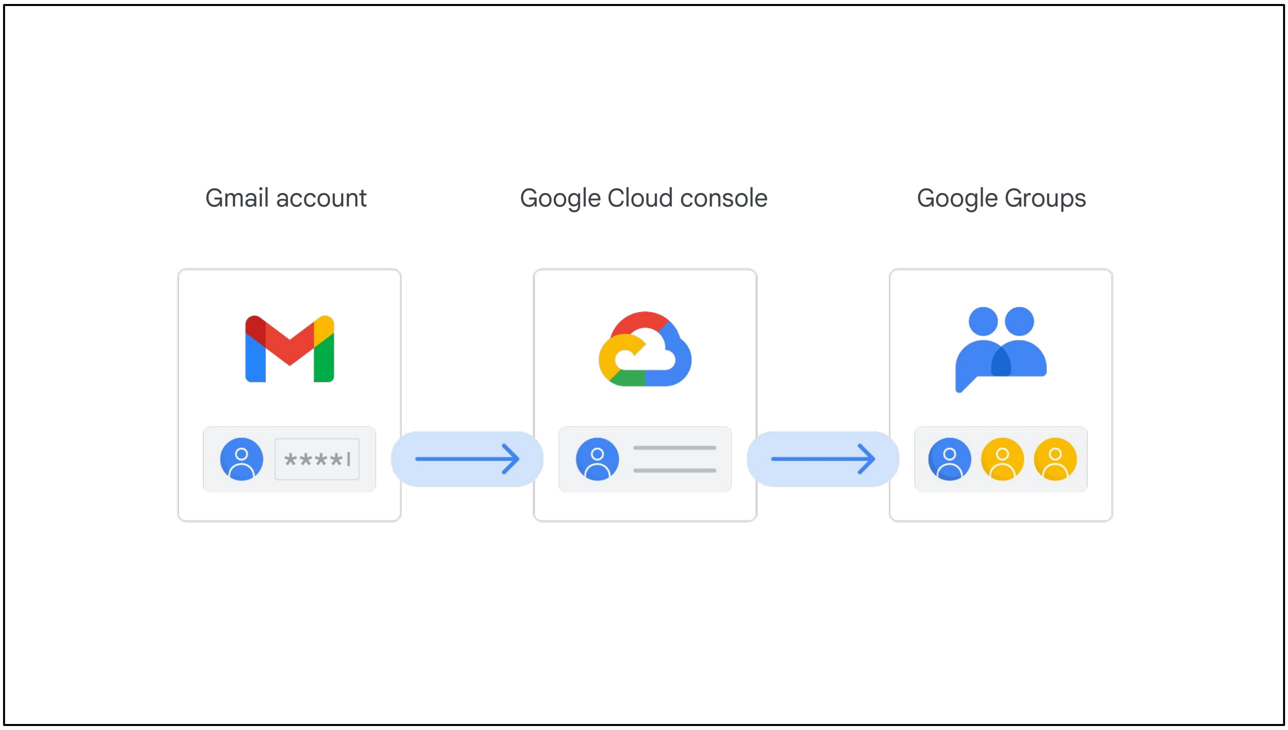


For example, maybe Alice needs to *manage* which Google accounts can act as service accounts, while Bob just needs to be able to *view* a list of service accounts.

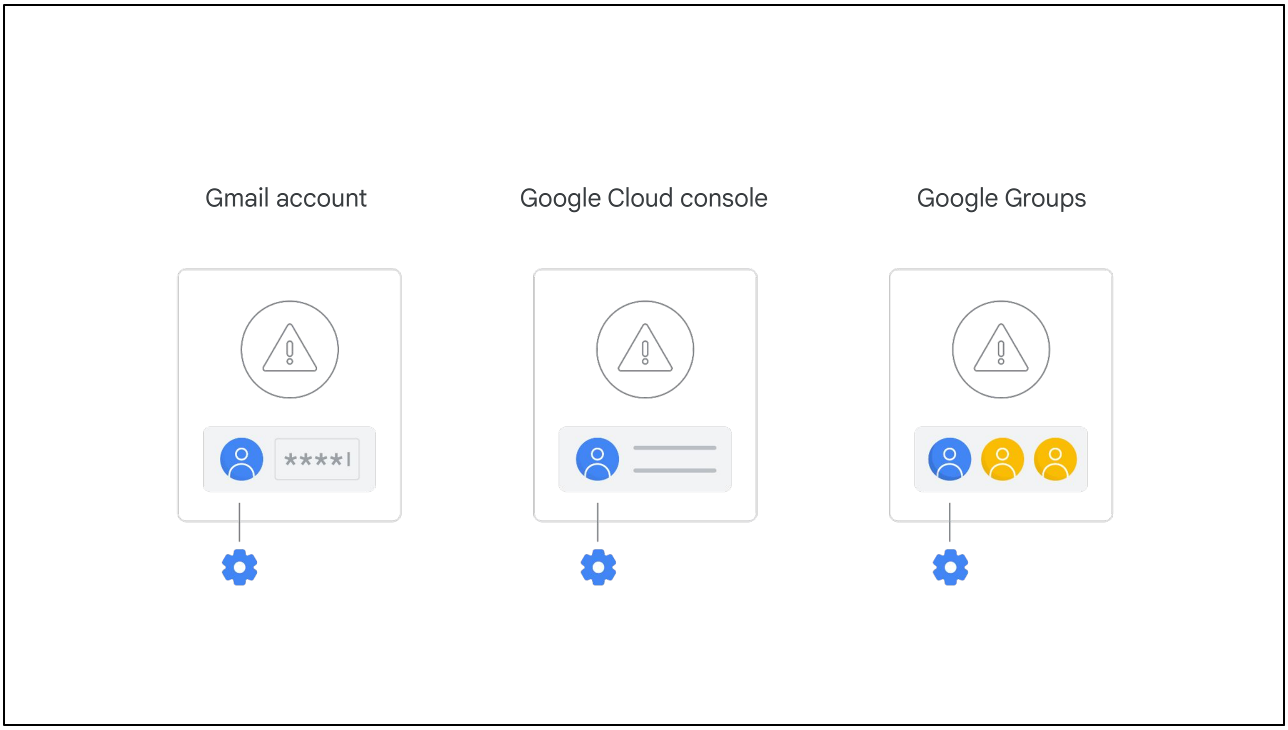


This means that Alice can have the editor role on a service account, and Bob can have the viewer role.

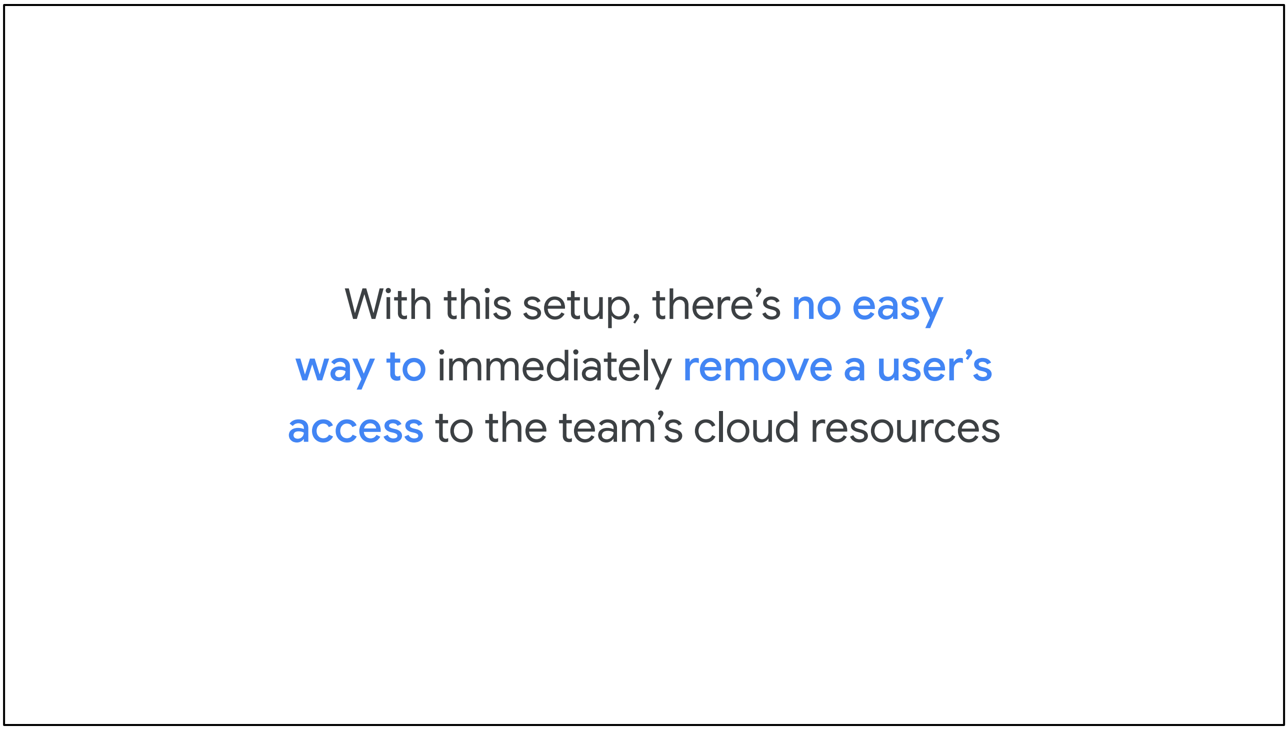
This is just like granting roles for any other Google Cloud resource.



When new Google Cloud customers start using the platform, it’s common to log in to the Google Cloud console with a Gmail account and then use Google Groups to collaborate with teammates who are in similar roles. Although this approach is easy to start with, it can

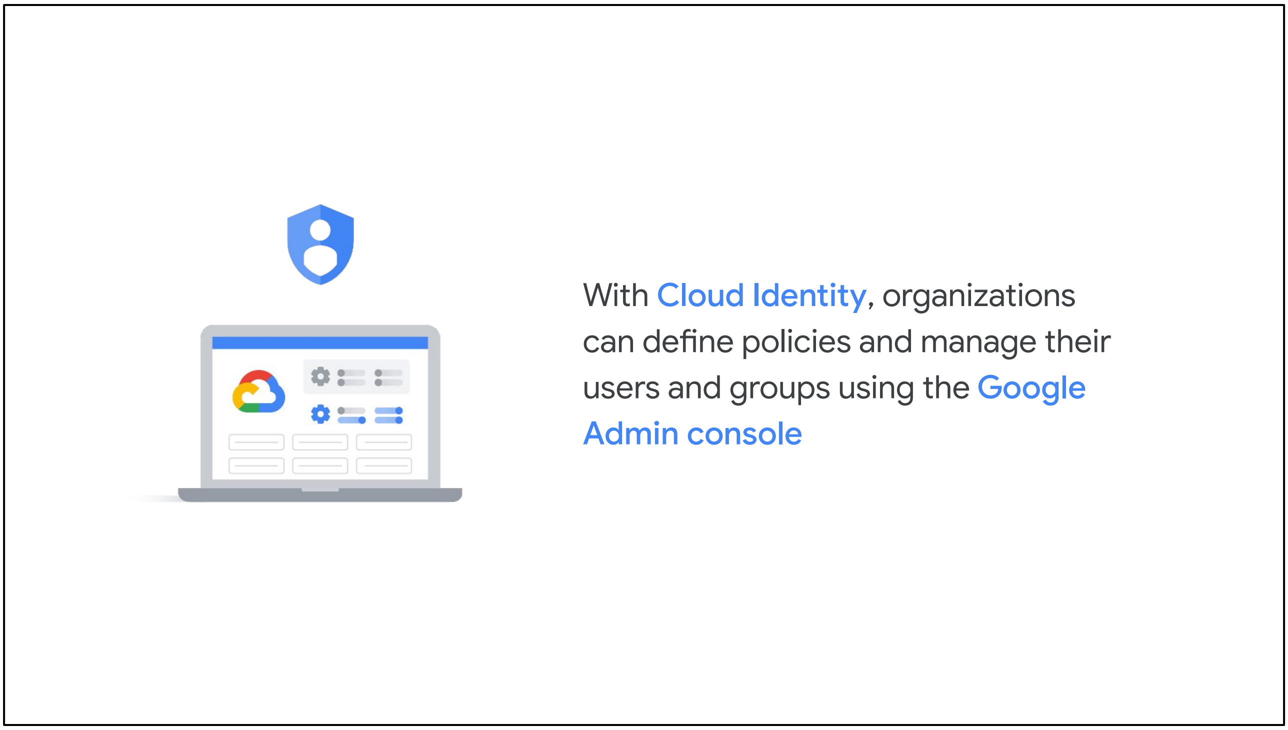


present challenges later because the team's identities are not centrally managed. This can be problematic if, for example,



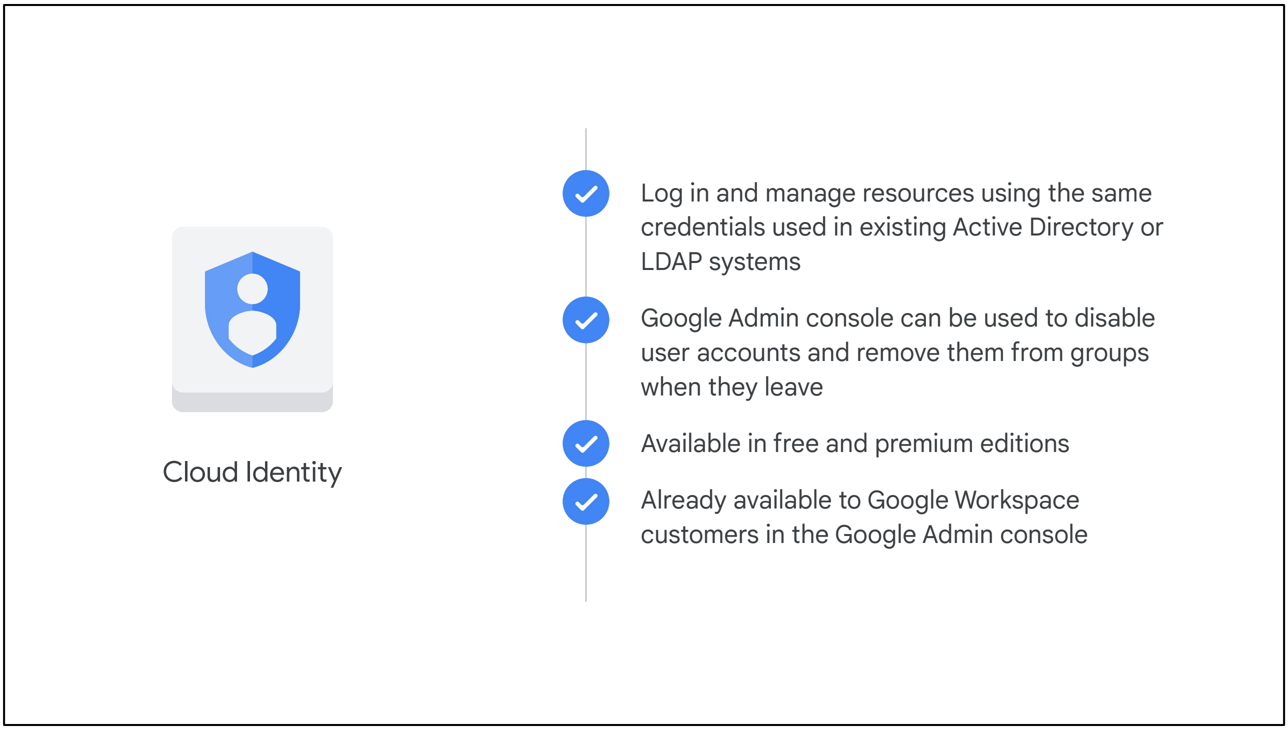
With this setup, there's no easy way to immediately remove a user's access to the team's cloud resources

With this setup, there's no easy way to immediately remove a user's access to the team's cloud resources.



With Cloud Identity, organizations can define policies and manage their users and groups using the Google Admin console

With a tool called **Cloud Identity**, organizations can define policies and manage their users and groups using the Google Admin console.



Cloud Identity

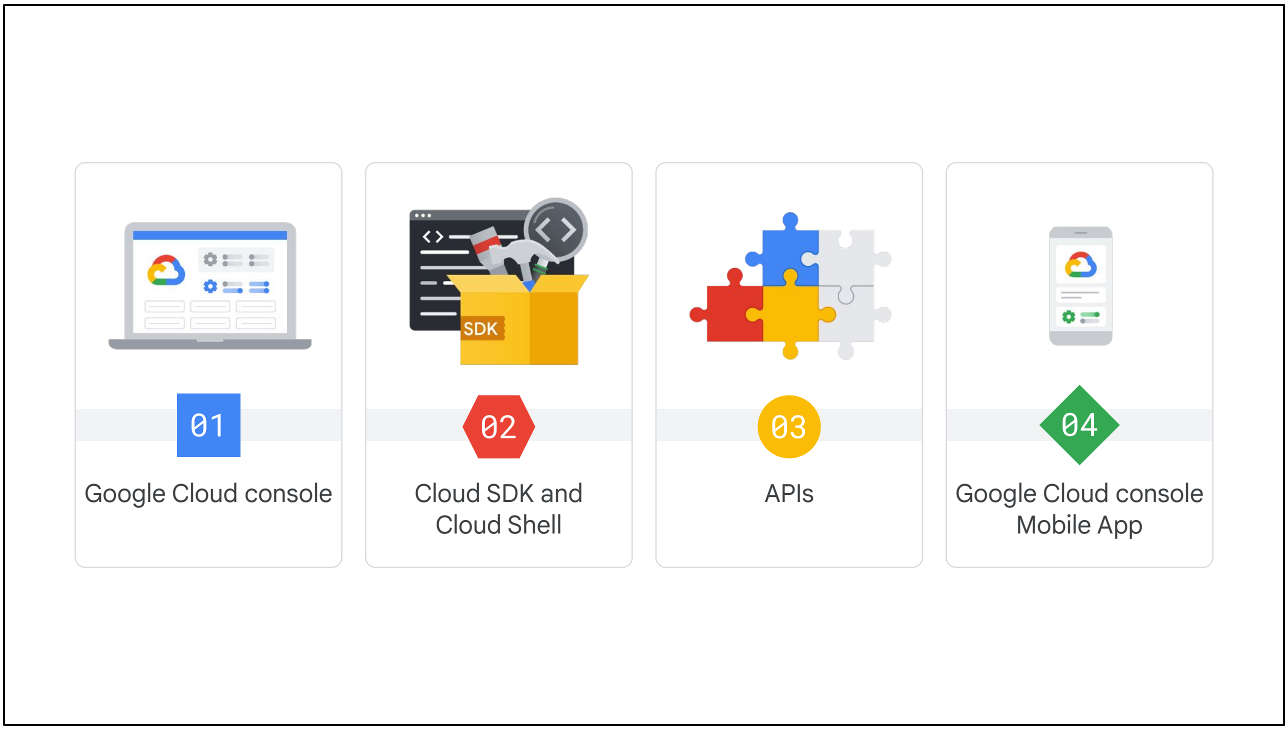
- ✓ Log in and manage resources using the same credentials used in existing Active Directory or LDAP systems
- ✓ Google Admin console can be used to disable user accounts and remove them from groups when they leave
- ✓ Available in free and premium editions
- ✓ Already available to Google Workspace customers in the Google Admin console

- Admins can log in and manage Google Cloud resources using the same usernames and passwords they already use in existing Active Directory or LDAP systems.

- Using Cloud Identity also means that when someone leaves an organization, an administrator can use the Google Admin console to disable their account and remove them from groups.

- Cloud Identity is available in a free edition and also in a premium edition that provides capabilities to manage mobile devices.

- If you’re a Google Cloud customer who is also a Google Workspace customer, this functionality is already available to you in the Google Admin console.



01 Google Cloud console

02 Cloud SDK and Cloud Shell

03 APIs

04 Google Cloud console Mobile App

There are four ways to access and interact with Google Cloud:

- The **Google Cloud console**,
- the **Cloud SDK** and **Cloud Shell**,
- the **APIs**,
- and the **Google Cloud console Mobile App**.

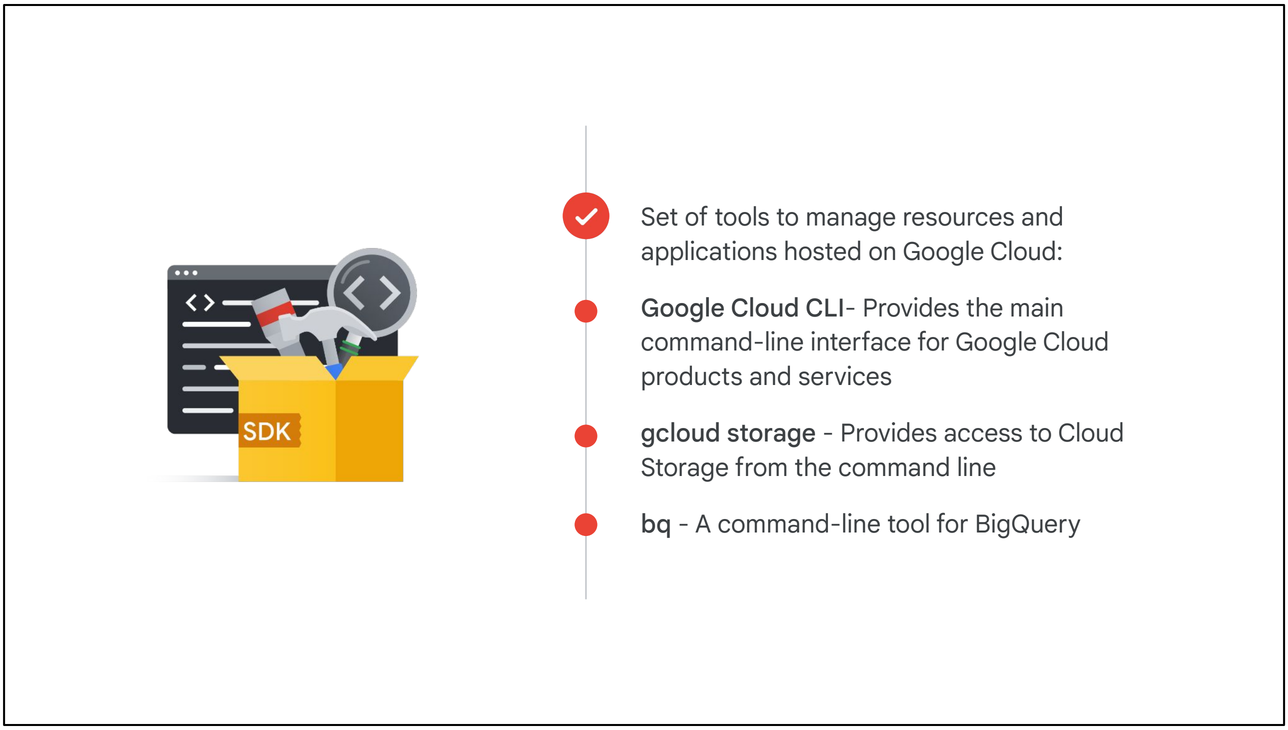
Let’s explore each of those now.

- ✓ Simple web-based graphical user interface
- ✓ Easily find resources, check their health, have full management control over them, and set budgets
- ✓ Provides a search facility to quickly find resources and connect to instances via SSH in the browser

Graphical User Interface, GUI, that helps you deploy, scale, and diagnose production issues in a simple web-based interface.

With the Google Cloud console, you can easily find your resources, check their health, have full management control over them, and set budgets to control how much you spend on them.

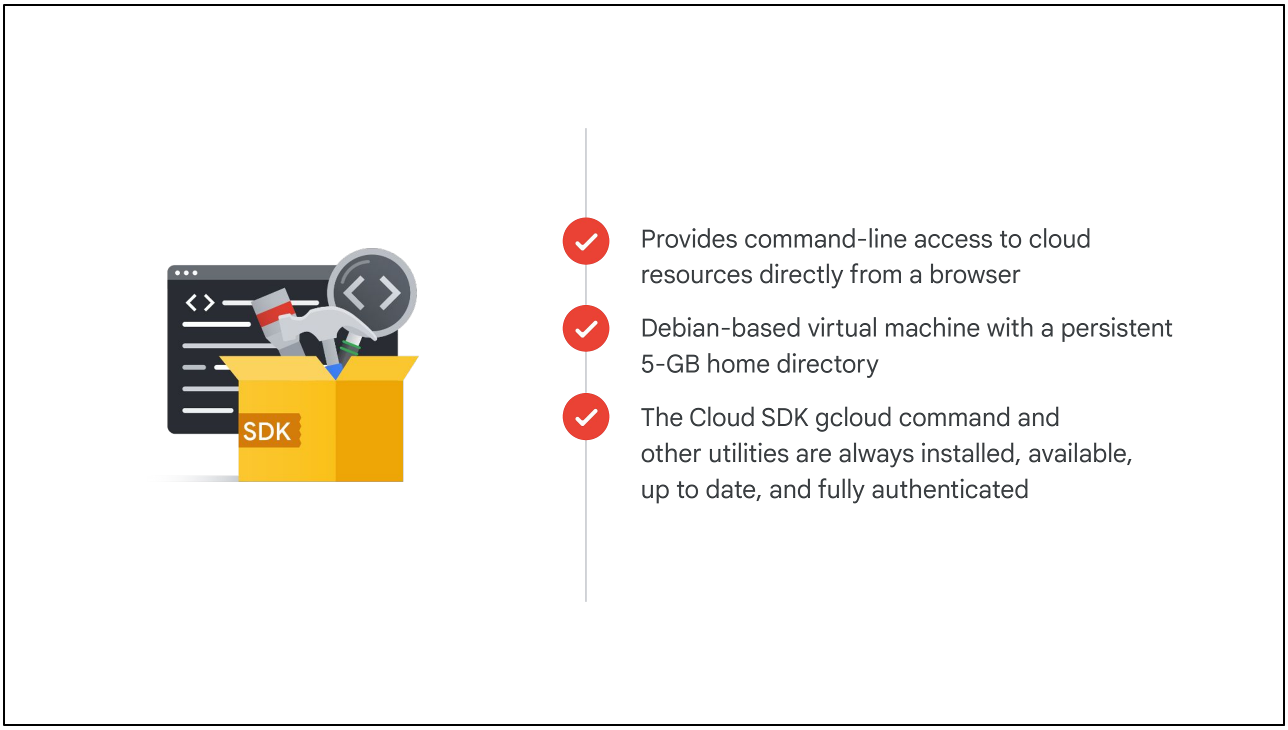
The Google Cloud console also provides a search facility to quickly find resources and connect to instances via SSH in the browser.



- ✓ Set of tools to manage resources and applications hosted on Google Cloud:
- **Google Cloud CLI**- Provides the main command-line interface for Google Cloud products and services
- **gcloud storage** - Provides access to Cloud Storage from the command line
- **bq** - A command-line tool for BigQuery

The **Cloud SDK** is a set of tools that you can use to manage resources and applications hosted on Google Cloud. These include the *Google Cloud CLI*, which provides the main command-line interface for Google Cloud products and services, *gcloud storage*, which lets you access Cloud Storage from the command line, and *bq*, a command-line tool for BigQuery.

When installed, all of the tools within the Cloud SDK are located under the *bin* directory.

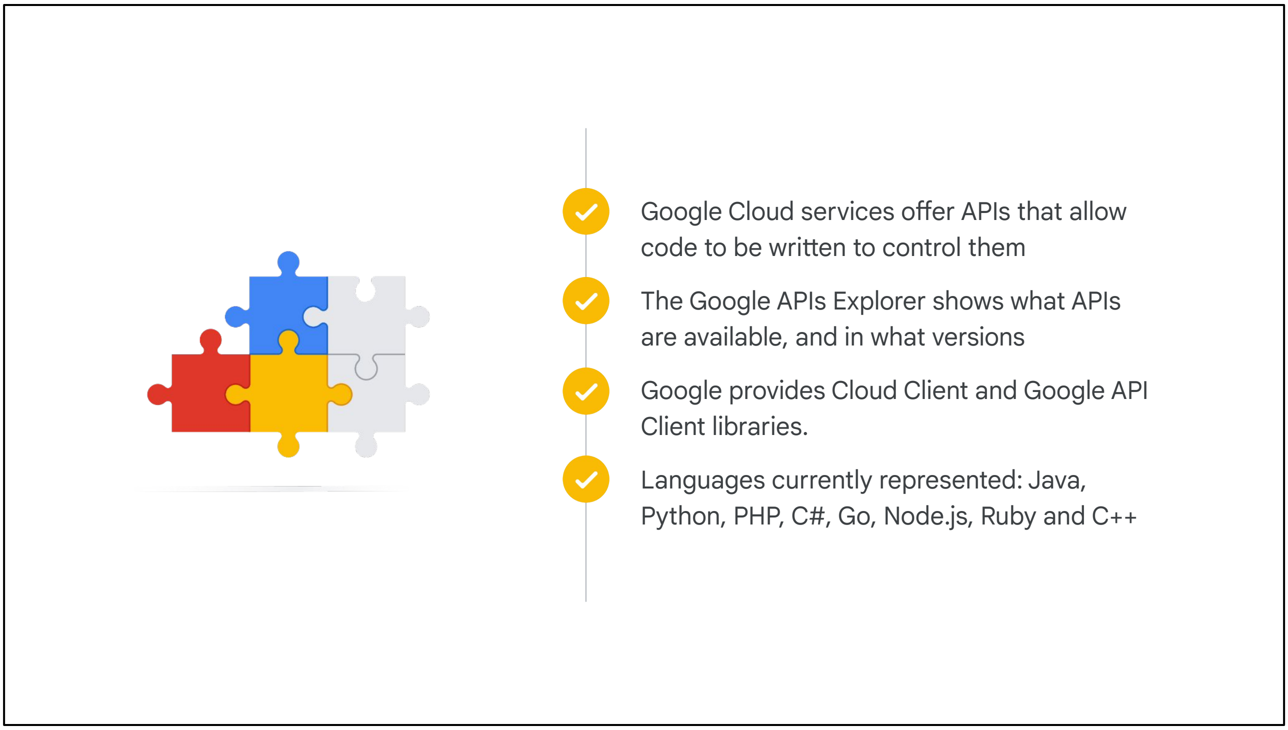


- Provides command-line access to cloud resources directly from a browser
- Debian-based virtual machine with a persistent 5-GB home directory
- The Cloud SDK gcloud command and other utilities are always installed, available, up to date, and fully authenticated

Cloud Shell provides command-line access to cloud resources directly from a browser.

Cloud Shell is a Debian-based virtual machine with a persistent 5 gigabyte home directory, which makes it easy to manage Google Cloud projects and resources.

With Cloud Shell, the Cloud SDK gcloud command and other utilities are always installed, available, up to date, and fully authenticated.



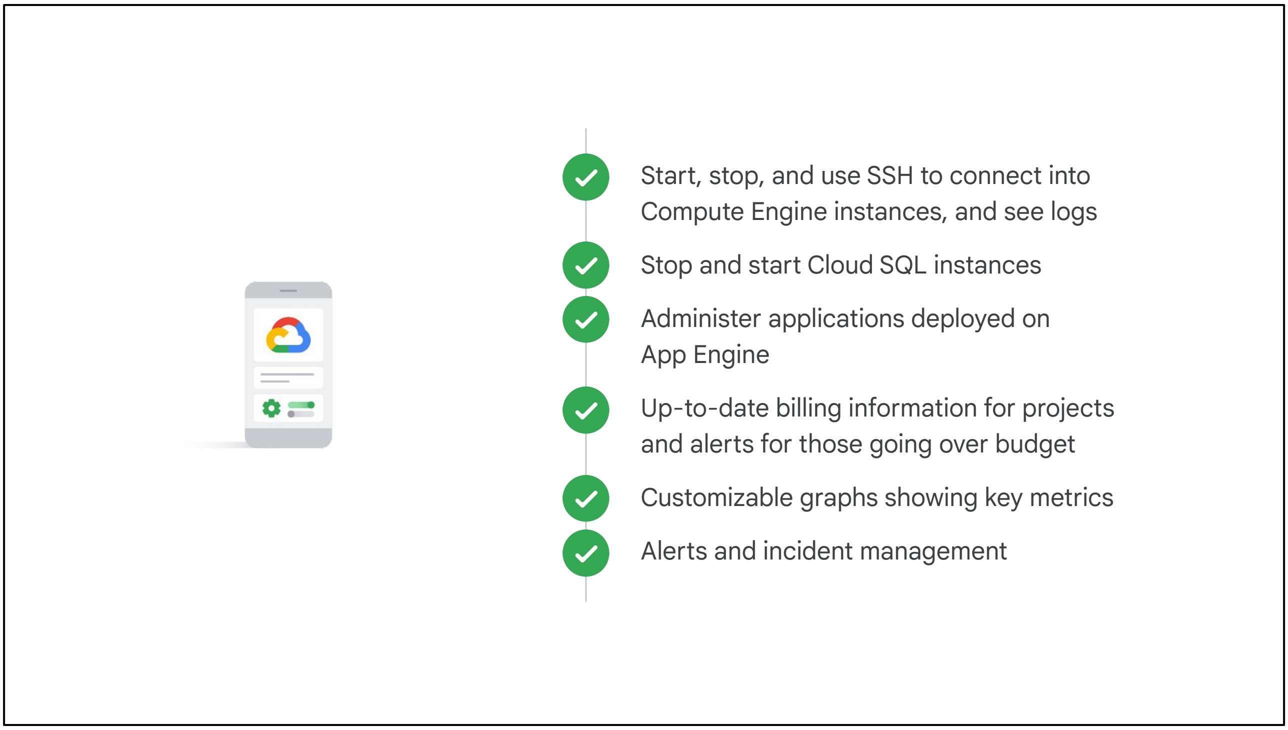
- Google Cloud services offer APIs that allow code to be written to control them
- The Google APIs Explorer shows what APIs are available, and in what versions
- Google provides Cloud Client and Google API Client libraries.
- Languages currently represented: Java, Python, PHP, C#, Go, Node.js, Ruby and C++

The services that make up Google Cloud offer **APIs** so that code you write can control them.

The Google Cloud console includes a tool called the *Google APIs Explorer* that shows which APIs are available, and in which versions. You can try these APIs interactively, even those that require user authentication.

Suppose you've explored an API, and you're ready to build an application that uses it. Do you have to start coding from scratch? No. Google provides *Cloud Client libraries* and *Google API Client libraries* in many popular languages to take a lot of the drudgery out of the task of calling Google Cloud from your code.

Languages currently represented in these libraries are Java, Python, PHP, C#, Go, Node.js, Ruby, and C++.



- ✓ Start, stop, and use SSH to connect into Compute Engine instances, and see logs
- ✓ Stop and start Cloud SQL instances
- ✓ Administer applications deployed on App Engine
- ✓ Up-to-date billing information for projects and alerts for those going over budget
- ✓ Customizable graphs showing key metrics
- ✓ Alerts and incident management

to start, stop, and use SSH to connect to Compute Engine instances and see logs from each instance.

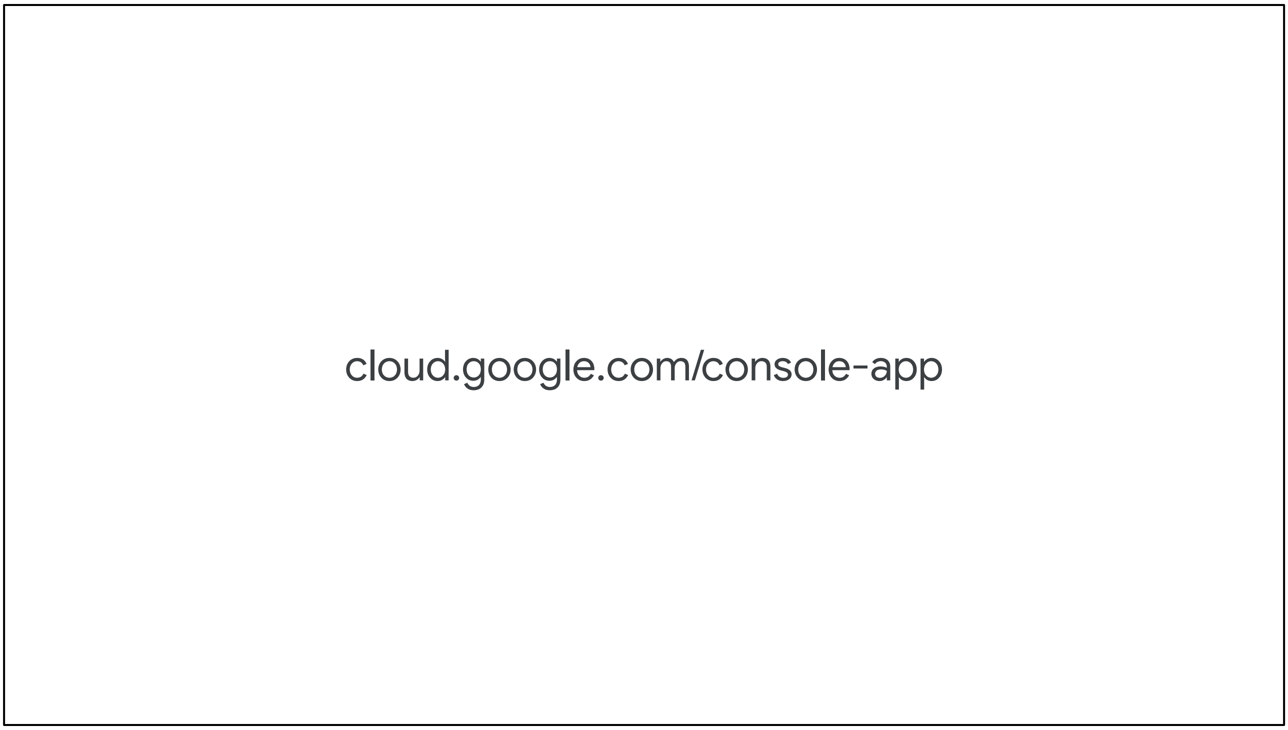
It also lets you stop and start Cloud SQL instances.

Additionally, you can administer applications deployed on App Engine by viewing errors, rolling back deployments, and changing traffic splitting.

The Google Cloud console Mobile App provides up-to-date billing information for your projects and billing alerts for projects that are going over budget.

You can set up customizable graphs showing key metrics such as CPU usage, network usage, requests per second, and server errors.

The mobile app also offers alerts and incident management.



cloud.google.com/console-app

You can download the Google Cloud console Mobile App at cloud.google.com/console-app.