Resources and Access in the Cloud

Now that you've had a chance to explore what cloud computing is, the pricing structure and billing practices available with Google Cloud, and the ways that Google strives to make the platform secure and environmentally friendly, let’s now work to understand the functional structure of Google Cloud.

Resources and Access in the Cloud

01 Google Cloud resource hierarchy
02 Identity and Access Management (IAM)
03 IAM roles
04 Service accounts
05 Cloud Identity
06 Interacting with Google Cloud

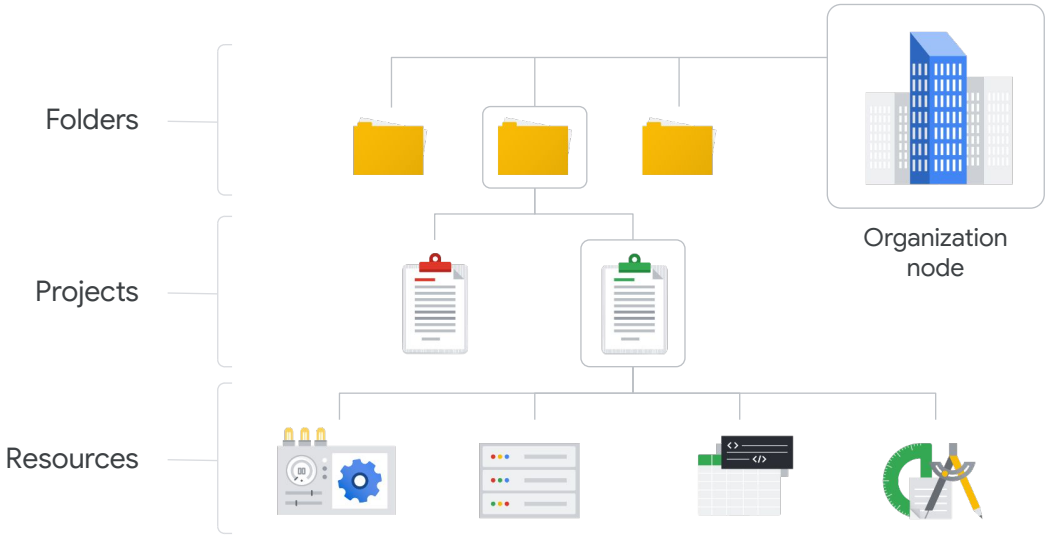
In this section of the course, we'll see how resources get organized with projects, and how access to those resources gets shared with the right part of a workforce through a tool called Identity and Access Management, or IAM.

We'll also look into the different ways in which you can interact with Google Cloud, including our web user interface, command-line interface, and our mobile apps.

Resources and Access in the Cloud

We'll begin with the Google Cloud Resource Hierarchy.

Resources are hierarchical



This hierarchy is made up of four levels, and starting from the bottom up they are: resources, projects, folders, and an organization node.

At the first level are **resources**. These represent virtual machines, Cloud Storage buckets, Virtual Private Networks (VPCs), tables in BigQuery, or anything else in Google Cloud.

Resources get organized into **projects**, which sit on the second level.

Projects can be organized into **folders**, or even subfolders. These sit at the third level.

And then at the top level is an **organization node**, which encompasses all the projects, folders, and resources in your organization.

Resource hierarchy determines policies

It's important to understand this resource hierarchy, as it directly relates to how policies are managed and applied when using Google Cloud.

Policies can be defined at the project, folder, and organization node levels. Some Google Cloud services allow policies to be applied to individual resources, too.

Policies are also inherited downward. This means that if you apply a policy to a folder, it will also apply to all of the projects within that folder.

Projects are the basis for using Cloud services

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 separately

Let's spend a little more time on the second level of the resource hierarchy, **projects**.

Projects are the basis for enabling and using Google Cloud services, like managing APIs, enabling billing, adding and removing collaborators, and enabling other Google services.

Each project is a separate compartment, and each resource belongs to exactly one project. Projects can have different owners and users, as they're billed and managed separately.

Project attributes vary in uniqueness and immutability

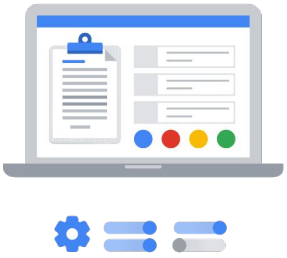
Project ID	Project name	Project number
Globally unique	Need not be unique	Globally unique
Assigned by Google Cloud but mutable during creation	Chosen by you	Assigned by Google Cloud
Immutable after creation	Mutable	Immutable

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

- The **project ID** is a globally unique identifier assigned by Google that cannot be changed–they are immutable–after creation. Project IDs are used in different contexts to inform Google Cloud of the exact project to work with.
- The **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 are mainly used internally, by Google Cloud, to keep track of resources.

Resource Manager manages projects

Resource Manager tool



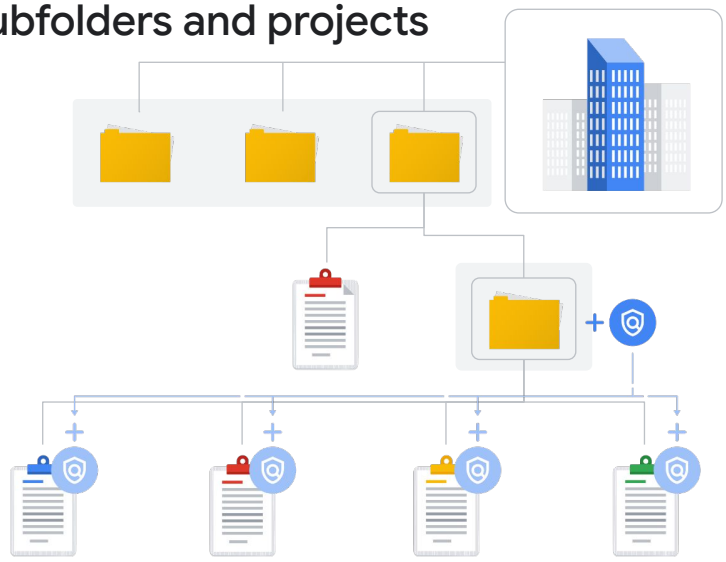
- Gather a list of projects
- Create new projects
- Update existing projects
- Delete projects
- Recover previously deleted projects
- Access through RPC API and REST API

So, how is one expected to manage projects? Google Cloud has the **Resource Manager** tool, designed to programmatically help do just that.

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.

Folders can contain subfolders and projects

The resources inherit policies and permissions assigned to folders



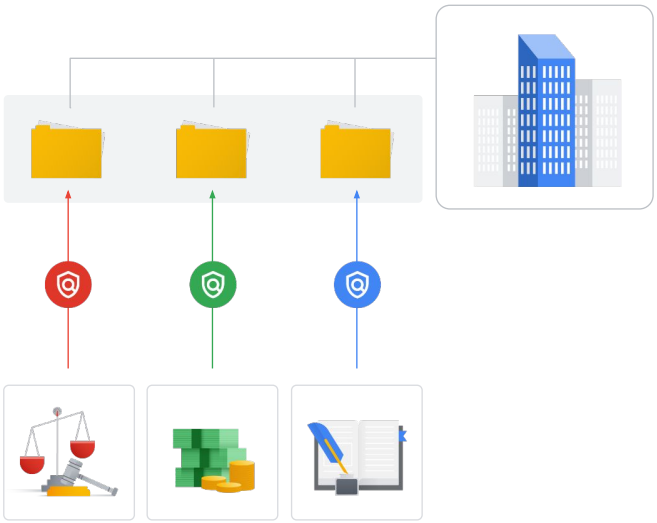
The third level of the Google Cloud resource hierarchy is folders.

Folders let you assign policies to resources at a level of granularity you choose. The projects and subfolders in a folder contain resources that inherit policies and permissions assigned to that folder.

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

Folders group projects

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



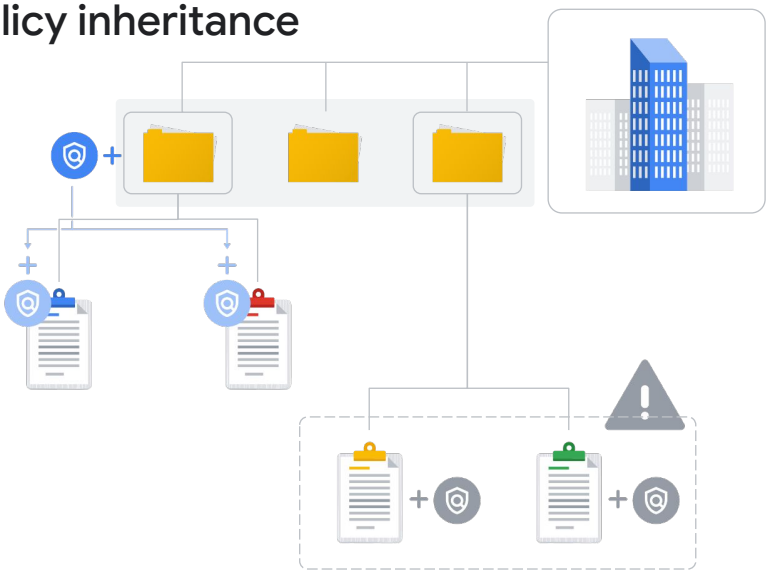
You can use folders to group projects under an organization in a hierarchy.

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

Folders facilitate policy inheritance

The projects
inherit policies
assigned to a folder



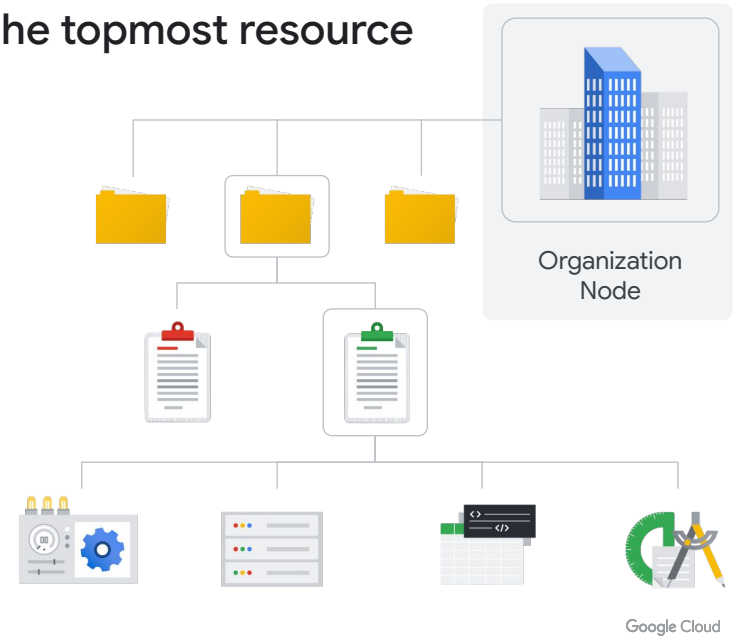
As previously mentioned, the resources in a folder inherit policies and permissions from that folder.

For example, if you have two different projects that are administered by the same team, you can put policies into a common folder so they have the same permissions.

Doing it the other way--putting duplicate copies of those policies on both projects–could be tedious and error-prone. if you needed to change permissions on both resources, you would now have to do that in two places instead of just one.

Organization node is the topmost resource

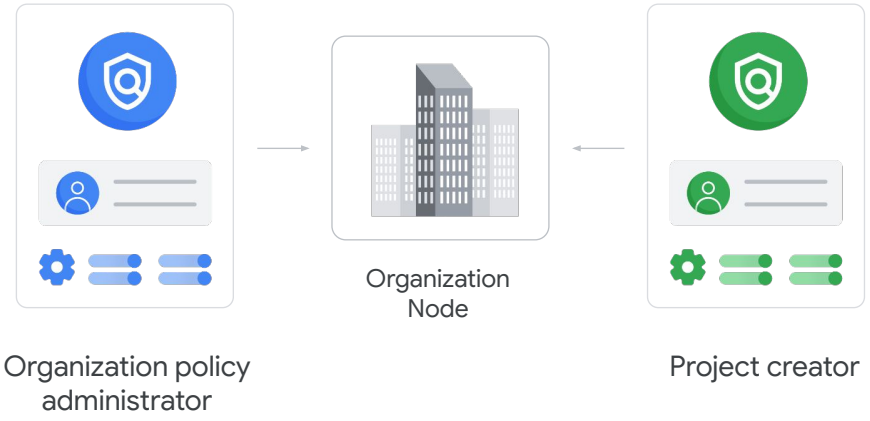
Everything attached to the account goes under the organization node



To use folders, you must have an organization node, which is the very topmost resource in the Google Cloud hierarchy.

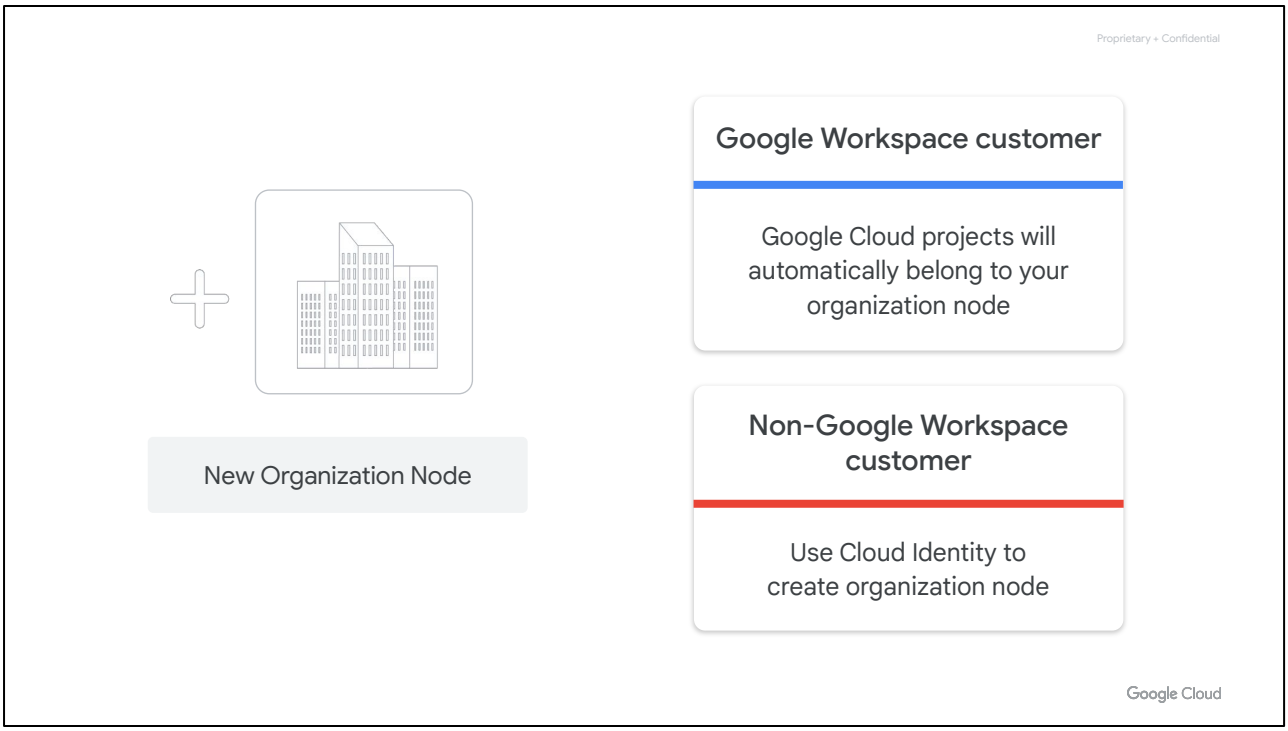
Everything else attached to that account goes under this node, which includes projects, folders, and other resources.

Special roles are associated with the Organization Node



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.



So how do you create an organization node? In part the answer 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 create one.

When you get a new organization node, it lets anyone in the domain create projects and billing accounts, just as they could before. That's to avoid surprises and disruption.

After you have an organization node, you can create folders underneath it and put projects into it. Both folders and projects are considered to be "children" of the organization node.

Resources and Access in the Cloud

01 Google Cloud resource hierarchy

02 **Identity and Access Management (IAM)**

03 IAM roles

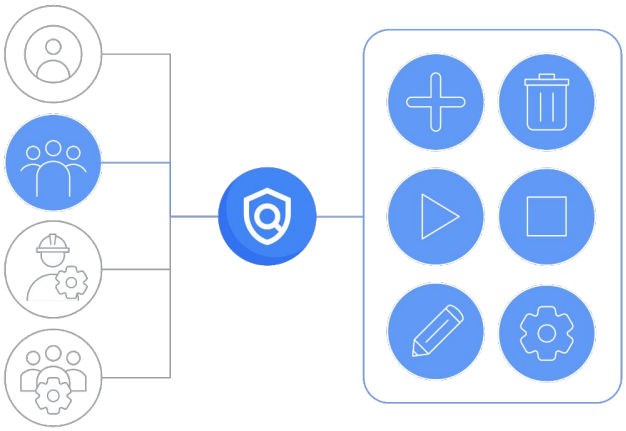
04 Service accounts

05 Cloud Identity

06 Interacting with Google Cloud

Identity and Access Management applies policies

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

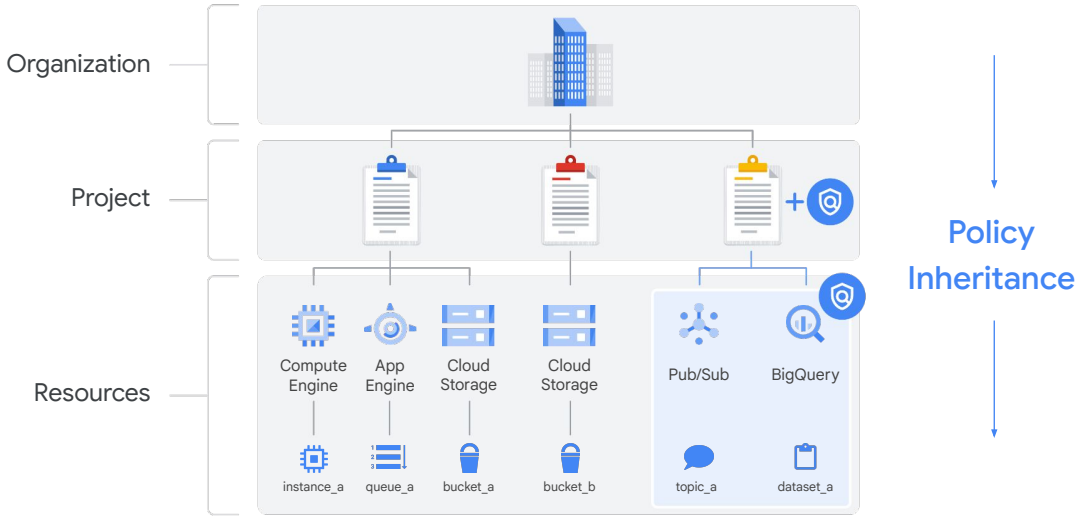


When an organization node contains lots of folders, projects, and resources, it's likely that a workforce might need to restrict who has access to what.

To help with this task, administrators can use Identity and Access Management, or IAM. With IAM, administrators can apply policies that define who can do what on which resources.

- The "who" part of an IAM policy can be a Google account, a Google group, a service account, or Cloud Identity domain. A "who" is also called a "principal." Each principal 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. For example, to manage virtual machine instances in a project, you have to be able to create, delete, start, stop and change virtual machines. So these permissions are grouped together into a role to make them easier to understand and easier to manage.

Policies are managed and applied by IAM



When a principal (user, group, or service account) is given a role on a specific element of the resource hierarchy, the resulting policy applies to the chosen element, as well as to all of the elements below it in the hierarchy.

Deny policies prevent specific IAM permissions

A deny policy overrides any existing allow policy regardless of the IAM role granted

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.

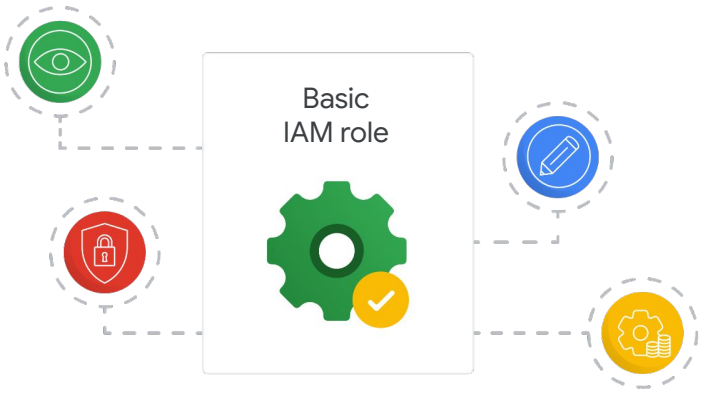
Deny policies, like allow policies, are inherited through the resource hierarchy.

Resources and Access in the Cloud

There are three kinds of IAM roles

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

Basic IAM roles are broad in scope

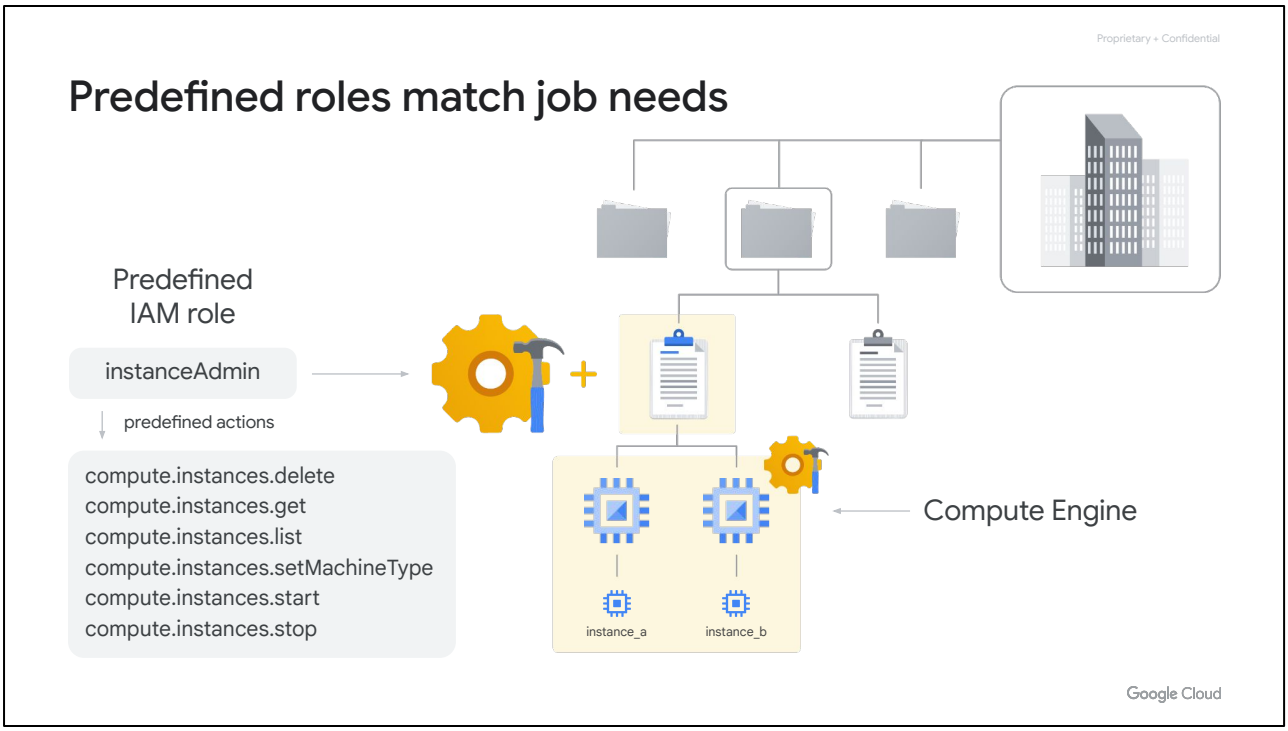


The first role type is **basic**. Basic roles are quite broad in scope. When applied to a Google Cloud project, they affect *all* resources in that project. Basic roles include owner, editor, viewer, and billing administrator.

Let's have a look at these basic roles:

- Project **viewers** can examine resources, but can make no changes.
- Project **editors** can examine and make changes to a resource.
- And project **owners** can also examine and make changes to a resource. In addition, project owners can manage the associate roles and permissions, as well as set up billing.
- Often companies want someone to be able to control the billing for a project, but not have permissions to change the resources in the project. This is possible through a **billing administrator** role.

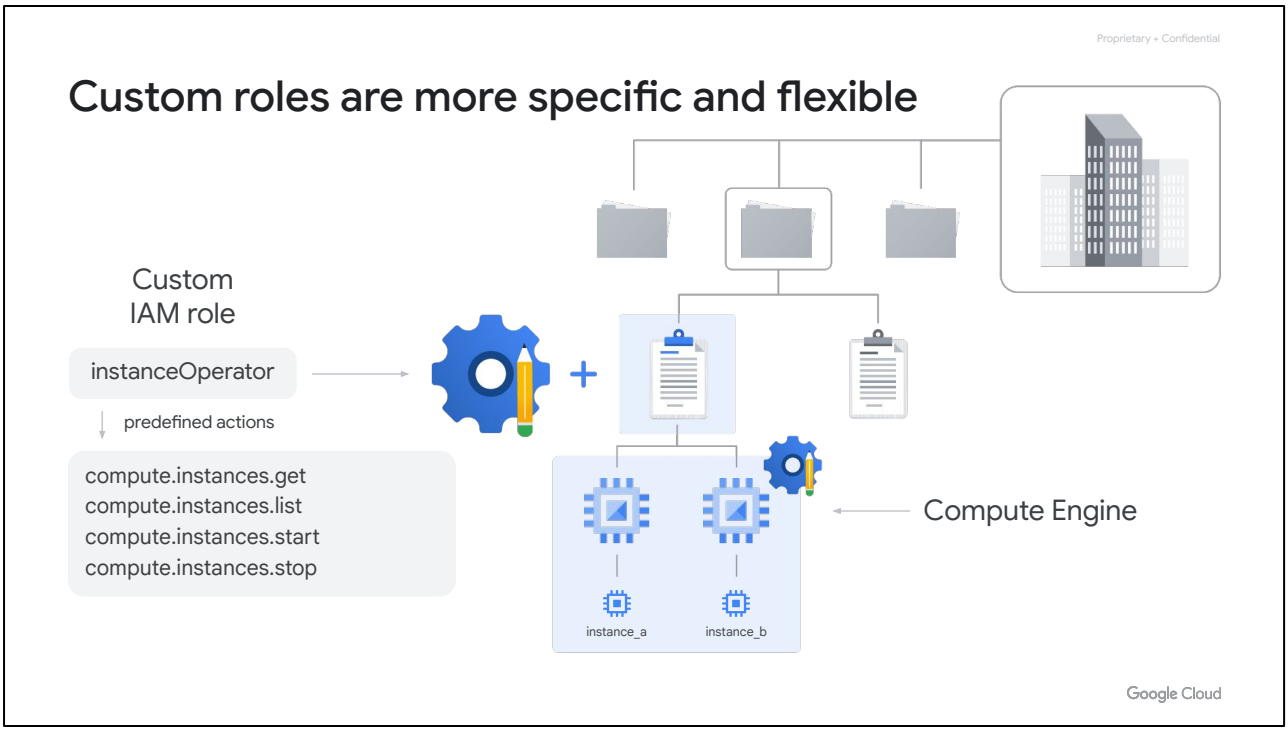
A word of caution: If several people are working together on a project that contains sensitive data, basic roles are probably too broad. Fortunately, IAM provides other ways to assign permissions that are more specifically tailored to meet the needs of typical job roles.



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.

Let's take Compute Engine for example, a Google Cloud product that offers virtual machines as a service. 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 pre-defined actions.



But what if you need to assign a role that has even more specific permissions? That's when you'd use a **custom role**.

A lot of 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 some users to stop and start Compute Engine virtual machines, but not reconfigure them. Custom roles allow for that.

Custom roles are created at the project or organization level using IAM

- Permissions need to be managed
- Can be applied to project or organization level

You can create a custom role at the project or organization level. You cannot define custom roles at the folder level. If you need to use a custom role within a folder, define the custom role at the organization level.

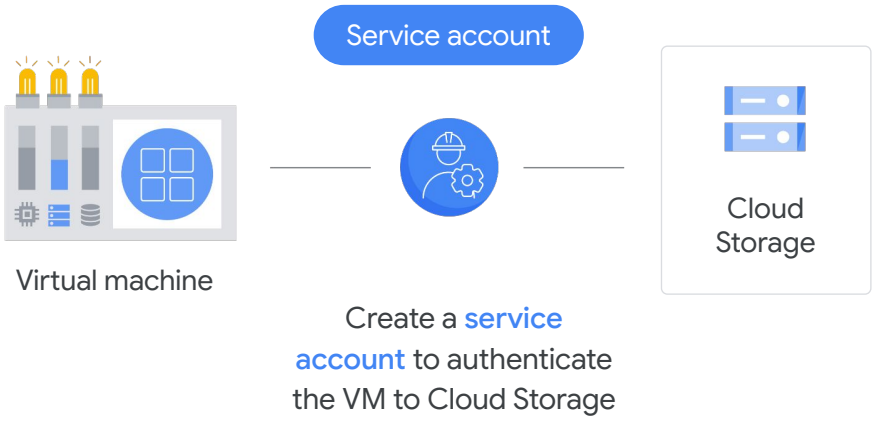
An organization-level custom role can include any of the IAM permissions that are supported in custom roles.

A project-level custom role can contain any supported permission except for permissions that can only be used at the organization or folder level, such as `resourcemanager.organizations.get`.

Resources and Access in the Cloud

Imagine you have a Compute Engine virtual machine running a program that needs to access other cloud services regularly. Instead of requiring a person to manually grant access each time the program runs, you can give the virtual machine itself the necessary permissions. This is where **service accounts** come in. Service accounts allow you to assign specific permissions to a virtual machine, so it can interact with other cloud services without human intervention.

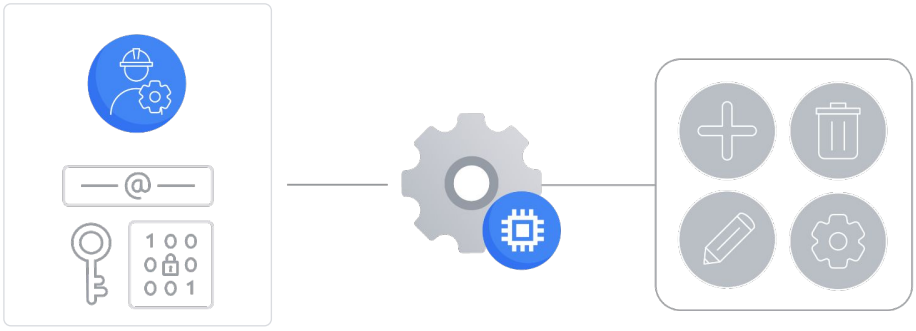
Service accounts are assigned roles



Let's say you have an application running in a virtual machine that needs to store data in Google Cloud Storage, but you don't want just anyone on the Internet to have access to that data–just *that particular* virtual machine.

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

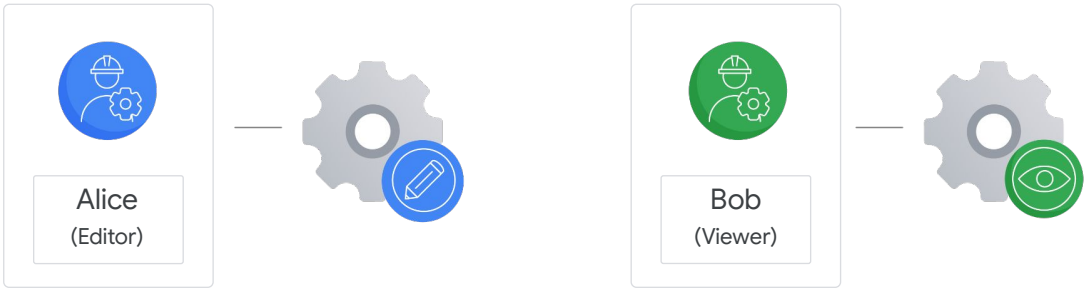
Service accounts are identified with email addresses



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 are also managed by IAM



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. Fortunately, in addition to being an identity, a service account is also a resource!

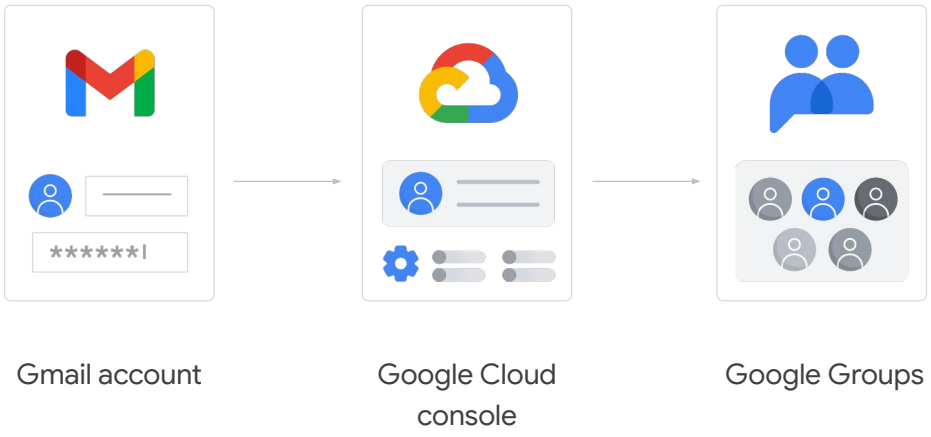
So it can have IAM policies of its own attached to it.

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.

Resources and Access in the Cloud

01	Google Cloud resource hierarchy
02	Identity and Access Management (IAM)
03	IAM roles
04	Service accounts
05	Cloud Identity
06	Interacting with Google Cloud

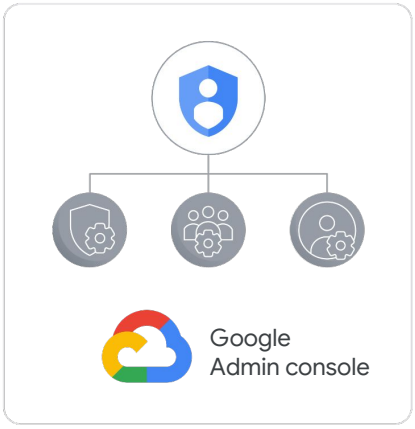
Cloud Identity manages team and organization access



When new Google Cloud customers start using the platform, it's common to log into the Google Cloud console with a Gmail account, and then use Google Groups to collaborate with teammates who are in similar roles.

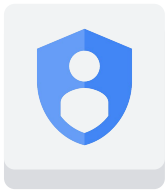
Although this approach is easy to start with, it can present challenges later because the team's identities are not centrally managed. This can be problematic if, say, if someone leaves the organization. With this setup, there is no easy way to immediately remove a user's access to the team's cloud resources.

Cloud Identity defines user and group policies



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.

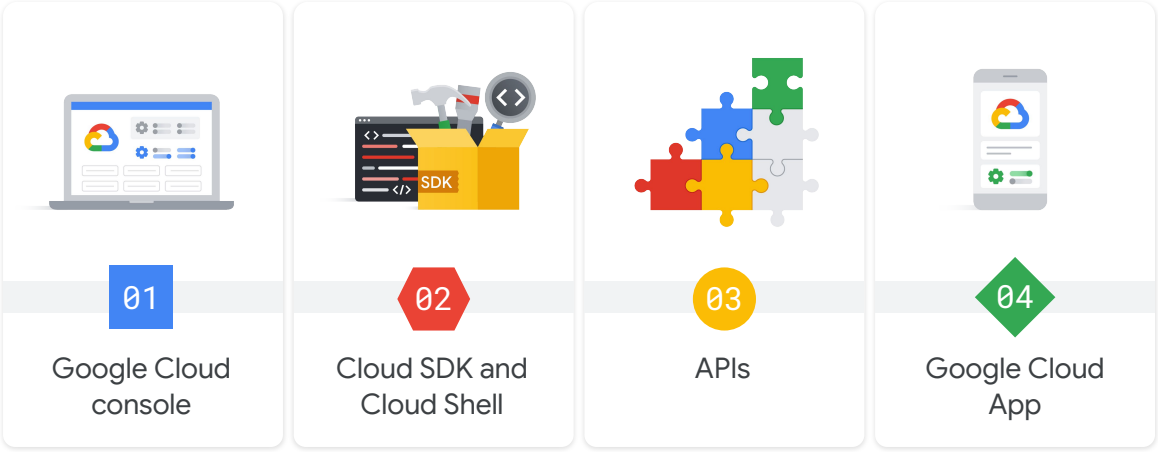
Using Cloud Identity also means that when someone leaves an organization, an administrator can use the Google Admin console to disable their account and remove them from groups.

Cloud Identity is available in a free edition, and also 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.

Resources and Access in the Cloud

You can interact with Google Cloud in four ways



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

Let's explore the list, which includes: the **Google Cloud console**, the **Cloud SDK** and **Cloud Shell**, the **APIs**, and the **Google Cloud App**.

Google Cloud console provides web-based interaction

- 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

First is the **Google Cloud console**, which is Google Cloud's Graphical User Interface (GUI) which helps you deploy, scale, and diagnose production issues in a simple web-based interface. With the 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 console also provides a search facility to quickly find resources and connect to instances via SSH in the browser.

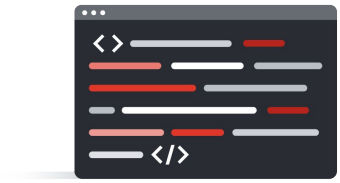
Cloud SDK is a collection of command line tools

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

Second is through the Cloud SDK and Cloud Shell.

The **Cloud SDK** is a set of tools that you can use to manage resources and applications hosted on Google Cloud. These include the gcloud CLI, which provides the main command-line interface for Google Cloud products and services, and bq, a command-line tool for BigQuery. When installed, all of the tools within the Cloud SDK are located under the bin directory.

Cloud Shell provides command line access to resources



- 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-GB 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.

APIs allow code to control your Cloud resources

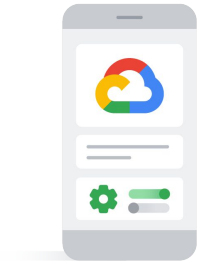
- Google Cloud services offer APIs that allow code to be written to control them
- The Google APIs Explorer shows what APIs are available, and in what versions
- Google provides Cloud Client and Google API Client libraries
  - Languages currently represented: Java, Python, PHP, C#, Go, Node.js, Ruby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

The **third** way to access Google Cloud is through application programming interfaces, or **APIs**.

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 what APIs are available, and in what 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 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++.

Manage your resources with the Google Cloud App



cloud.google.com/app

- 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

And finally, the **fourth** way is with the **Google Cloud App,** which can be used to start, stop, and use ssh to connect to Compute Engine instances, and to 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.

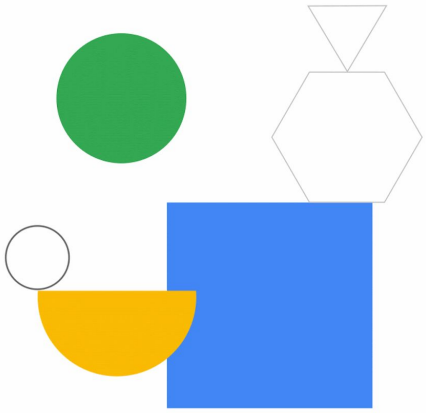
The Google Cloud App provides up-to-date billing information for your projects, and billing alerts for projects that are going over budget.

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

The Google Cloud App also offers alerts and incident management.

Download the Google Cloud App at cloud.google.com/app.

Module Quiz



Quiz | Question 1

Question

Choose the correct completion: Services and APIs are enabled on a per-_____ basis.

A. Folder

B. Project

C. Organization

D. Billing account

Quiz | Question 1

Answer

Choose the correct completion: Services and APIs are enabled on a per-_____ basis.

A. Folder

B. Project 

C. Organization

D. Billing account

Choose the correct completion: Services and APIs are enabled on a per-_____ basis.

A: Folder
Feedback: Enable or disable APIs on a per-project basis.

B: Project
Feedback: Correct!

C: Organization
Feedback: Enable or disable APIs on a per-project basis.

D: Billing account
Feedback: Enable or disable APIs on a per-project basis.

Quiz | Question 2

Question

Order these IAM role types from broadest to finest-grained.

A. Custom roles, predefined roles, basic roles

B. Predefined roles, custom roles, basic roles

C. Basic roles, predefined roles, custom roles

Quiz | Question 2

Answer

Order these IAM role types from broadest to finest-grained.

A. Custom roles, predefined roles, basic roles

B. Predefined roles, custom roles, basic roles

C. Basic roles, predefined roles, custom roles 

Order these IAM role types from broadest to finest-grained.

A: Custom roles, predefined roles, basic roles
Feedback: Basic roles affect all resources in a project.

B: Predefined roles, custom roles, basic roles
Feedback: Basic roles affect all resources in a project.

C: Basic roles, predefined roles, custom roles
Feedback: Correct!

Quiz | Question 3

Question

Which of these values is globally unique, permanent, and unchangeable, but can be modified by the customer during creation?

A. The project name

B. The project ID

C. The project number

D. The project's billing credit-card number

Quiz | Question 3

Answer

Which of these values is globally unique, permanent, and unchangeable, but can be modified by the customer during creation?

A. The project name

B. The project ID 

C. The project number

D. The project's billing credit-card number

Which of these values is globally unique, permanent, and unchangeable, but can be modified by the customer during creation?

A: The project name
Feedback: Review the lecture "The Google Cloud resource hierarchy."

B: The project ID
Feedback: Correct! The project ID is immutable (cannot be changed) *after* creation, but can be changed *during* creation.

C: The project number
Feedback: Review the lecture "The Google Cloud resource hierarchy."

D: The project's billing credit-card number
Feedback: Review the lecture "The Google Cloud resource hierarchy.

Lab Intro

Getting Started with Cloud Marketplace

In this lab you use Cloud Marketplace to quickly and easily deploy a LAMP stack on a Compute Engine instance.

The Bitnami LAMP Stack provides a complete web development environment for Linux that can be launched in one click.