



Google Cloud Fundamentals

Ran Shiloni



Welcome to the Google Cloud fundamentals module.



Learn how to...

Compare Google Cloud terminology to your source environment

Understand resource hierarchy

Control permissions with Cloud IAM

After discovering your source environment, analyzing your topology, and carefully selecting virtual machines to migrate to the cloud, it's time to learn about your destination environment, Google Cloud. We will compare the terminology that you are familiar with on-premises or in AWS to the corresponding terminology on Google Cloud, explain how resource hierarchy works in Google Cloud's environment, and discover ways to control permissions with Cloud IAM.



Learn how to...

Limit consumption with quotas and budgets

Predict cost and visualize spend over time

You will also learn how to limit consumption with quotas and budgets, predict cost and visualize spend over time.

All of this content will provide you with the foundational knowledge to start running your workloads in Google Cloud and will be the building blocks for the rest of this course.

Feel free to skip this module if you are already familiar with these concepts in Google Cloud.

Agenda

Google Cloud terminology

Google Cloud resource hierarchy

Cloud Identity and Access Management (Cloud IAM)

Lab

Identity

Interacting with Google Cloud

Lab

Billing, labels, and quotas

Lab

In this module, we compare the terminology you are used to from your source environment to Google Cloud's equivalent.

On-premises vs. Google Cloud: Compute

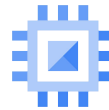
Data center

- Physical or virtualized hardware
- VMWare ESXi
- Hyper-V
- KVM
- XEN

AWS

Amazon Elastic Compute Cloud

Google Cloud



Compute Engine

When running virtual machines on-premises, you are responsible for the hypervisor that runs them: whether VMWare ESXi, Hyper-V, KVM, or XEN. You are also responsible for maintaining the underlying infrastructure such as networking, storage, and hardware lifecycles. In addition, when running on-premises, you might choose to not run a Hypervisor at all and install your Operating System of choice directly on bare metal hardware.

With Google Cloud, virtual machines are called Compute Engine instances, and they run on Google's Compute Engine service. With Compute Engine, you do not have to manage the underlying infrastructure or hypervisor, because Google maintains these services so that you can focus on the virtual machine instead of the underlying systems that run it.

On-premises vs. Google Cloud: Storage

Data center

- SAN
- NAS
- DAS

AWS

Amazon Elastic Block Store

Google Cloud



Persistent Disk

When providing storage on-premises, you might use directly attached storage (DAS), network attached storage (NAS), or a storage area network (SAN) to provide storage services to your machines. These storage options provide a wide array of benefits, from minimizing latency with DAS, to providing ease of use with NAS, to providing ultra-redundant, scalable, high-performance storage with a SAN.

Google Cloud has complementary storage offerings which map to the storage technologies that you are familiar with on-premises: whether it is Google's network-attached Persistent Disk technology, which supports standard spinning hard disks or SSDs, or high-performance, direct-attached Local SSDs, which support SCSI or NVMe interfaces.

Google Cloud provides you with VM Storage as a Service so that you do not have to manage these components yourself. We will cover these storage technologies in greater detail in the next module.

On-premises vs. Google Cloud: Identity

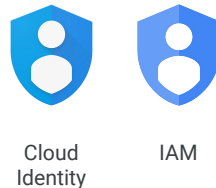
Data center

- Active Directory
- LDAP

AWS

Amazon IAM

Google Cloud



Typically, managing identity on-premises can be accomplished with Active Directory or LDAP servers, which must remain highly available.

Google Cloud provides a cloud-based Identity as a Service offering called Cloud Identity with which you can either extend your current directory services to the cloud via sync or create a new user directory for your cloud environment.

With Google Cloud's Identity and Access Management, or IAM, you can control and grant granular access to resources in your cloud environment. You can also sync your On-Premises Active Directory with Cloud Identity. We will explore this in detail later in the course.

Agenda

Google Cloud terminology

[Google Cloud resource hierarchy](#)

Identity and Access Management (IAM)

Lab

Identity

Interacting with Google Cloud

Lab

Billing, labels, and quotas

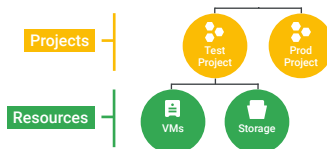
Lab

In this video, you will learn about Google Cloud's resource hierarchy and its characteristics.

Resource hierarchy levels define trust boundaries

Group your resources according to your organization structure.

Levels of the hierarchy provide trust boundaries and resource isolation.

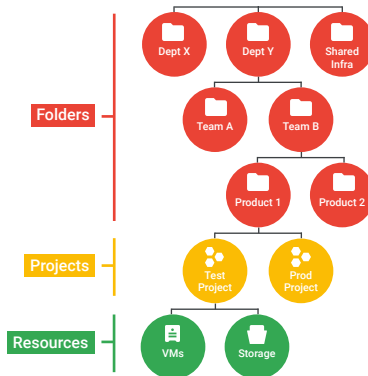


It is easier to understand the Google Cloud resource hierarchy from the bottom up. All of the resources you use, whether they are virtual machines, Cloud Storage buckets, tables in BigQuery, or anything else in Google Cloud, are organized into a logical construct known as a Project. In addition, Projects also contain the billing information for the resources in the Project, as well as maintain IAM permissions which dictate who can do what with the resources within your Project.

Resource hierarchy levels define trust boundaries

Group your resources according to your organization structure.

Levels of the hierarchy provide trust boundaries and resource isolation.

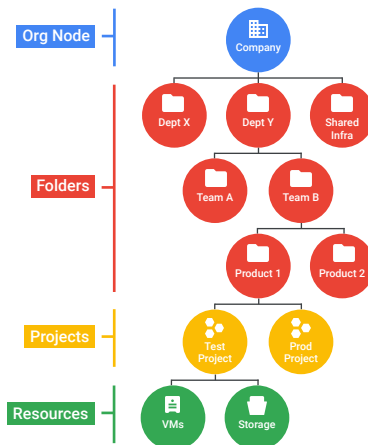


Optionally, these Projects may be organized into Folders, with Folders being able to contain other Folders.

Resource hierarchy levels define trust boundaries

Group your resources according to your organization structure.

Levels of the hierarchy provide trust boundaries and resource isolation.



All the Folders and Projects are brought together under an Organization node (known as an Org node). Projects, Folders, and Organization nodes are all places where policies can be defined and inherited. Some Google Cloud resources let you apply policies on individual resources too, like Cloud Storage buckets.

Let's explore these concepts in detail.

All Google Cloud services you use are associated with a Project

- Track resource and quota usage
- Enable billing
- Manage permissions and credentials
- Enable services and APIs
- Separate administrative control plane



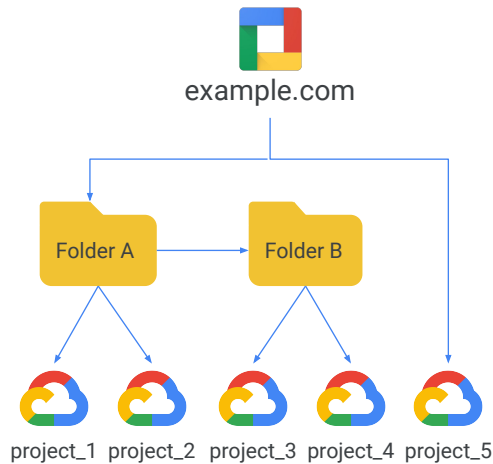
All Google Cloud resources belong to a Google Cloud Project. Projects are the basis for enabling and using Google Cloud services, like managing APIs, enabling billing, adding and removing collaborators and their assigned permissions, and enabling other Google services. Each Project is a separate logical compartment, with an individual cloud resource (Cloud Storage Bucket, Virtual Machine, etc.) belonging to exactly one Project. Different Projects can have different owners and users. They're billed separately, and they're managed separately.

Projects have three identifying attributes

Project ID	Globally unique	Chosen by you	Immutable
Project name	Need not be unique	Chosen by you	Mutable
Project number	Globally unique	Assigned by Google Cloud	Immutable

Each Google Cloud project has a name and project ID you can choose. The project ID is a permanent, unchangeable identifier and it has to be unique across Google Cloud. You will use Project IDs in several contexts to tell Google Cloud which Project you want to work with. On the other hand, Project names are for your convenience and you can change them. Google Cloud also assigns each of your Projects a unique Project number, which you will see displayed in various contexts, but using it is mostly outside the scope of this course. In general, Project IDs are made to be human-readable strings, and you will use them frequently to refer to Projects.

Folders offer flexible management



- Folders group projects under an organization.
- Folders can contain projects, other folders, or both.
- Use folders to assign policies.

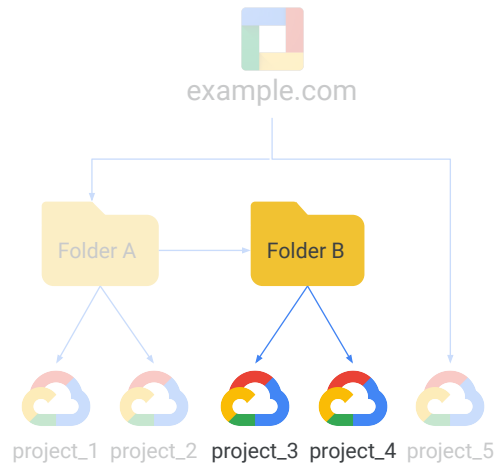
Projects can be nested inside Folders.

Folders let you assign policies to resources at the level of granularity you choose. The resources in a Folder inherit IAM policies assigned to the Folder.

A Folder can contain projects, other folders, or a combination of both. 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 or in a structure that maps to your organization's business or operational model.

Folders give teams the ability to delegate administrative rights, so that they can work independently but with consistency with regard to enforcing proper cloud governance.

Folders offer flexible management

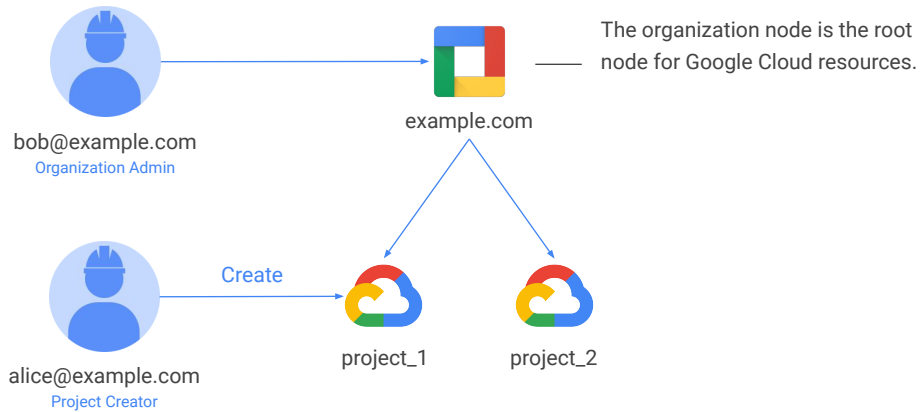


- Folders group projects under an organization.
- Folders can contain projects, other folders, or both.
- Use folders to assign policies.

The resources in a Folder inherit IAM policies from the Folder. So, if “project_3” and “project_4” are administered by the same team, by design, you can apply IAM policies to Folder B instead, which applies these policies within all Projects within Folder B. Without the ability to apply policies at the Folder level, you would have to apply duplicate copies of the policies to both “project_3” and “project_4” individually, which would be a tedious and error-prone process.

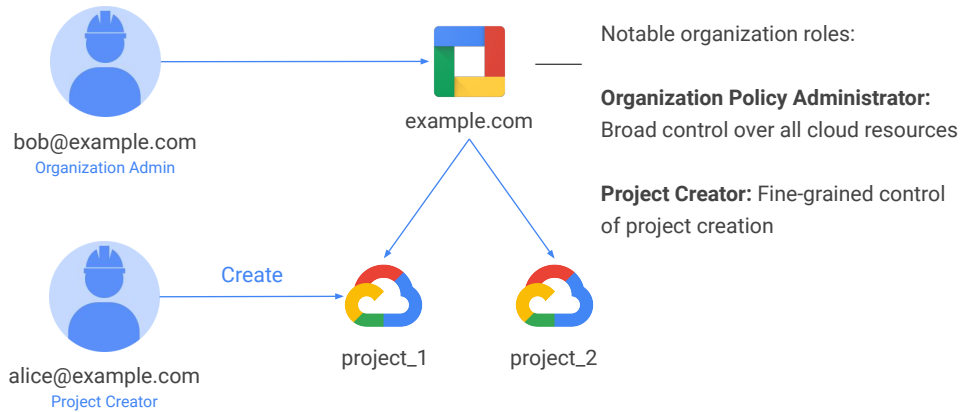
It is important to note that to use folders, you need an Organization node at the top of the hierarchy.

The organization node organizes projects



You probably want to organize all the projects in your company into a single structure. Most companies want to have centralized visibility of how resources are being used, and also to apply policies centrally. This is exactly what the organization node is designed for. It is the top of the Google Cloud hierarchy.

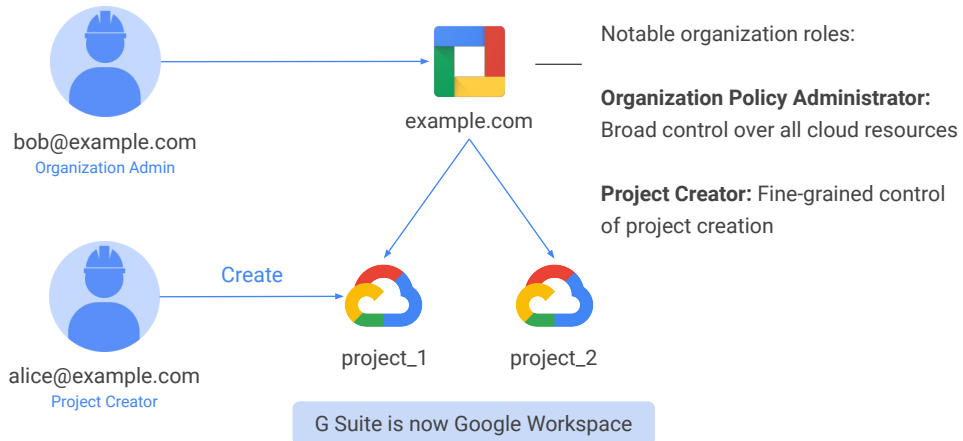
The organization node organizes projects



There are some special roles associated with the 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 spend money and delegate permissions.

So how do you get an Organization node?

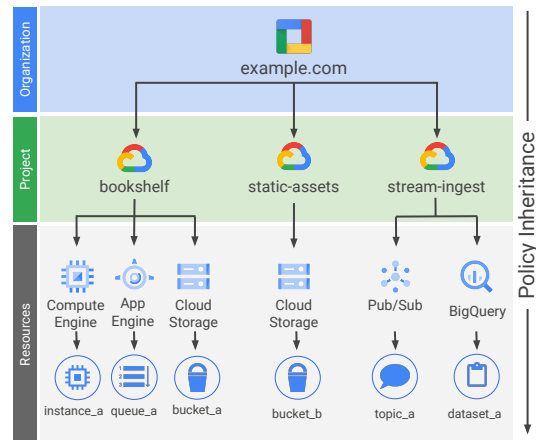
The organization node organizes projects



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, which is typically the domain name you use for Workspace. If you do not use Workspace, you can leverage Google Cloud Identity to create an Organization node. We will introduce Cloud Identity later in this course.

An example IAM resource hierarchy

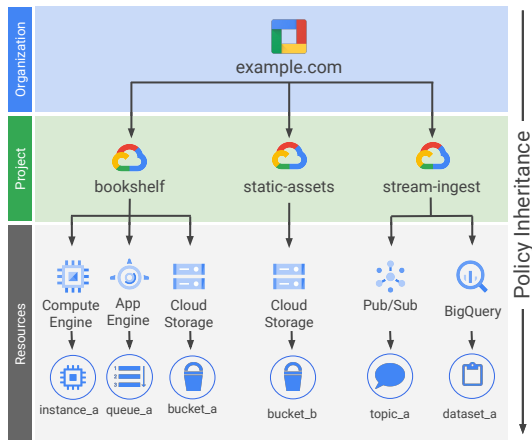
- A policy is set on a resource.
 - Each policy contains a set of roles and role members.
- Resources inherit policies from parent.
 - Resource policies are a union of parent and resource.
- A less restrictive parent policy overrides a more restrictive resource policy.



Here is an example of how you might organize your resources. In this example, there are three Projects, each of which uses resources from several Google Cloud services. You will notice in the diagram that we have not used any Folders in the current organization structure. If the use of Folders would be helpful in the future, we can always implement them and apply policies as needed.

Resources inherit the policies of their parent resource. For instance, if you set a policy at the Organization level, it is automatically inherited by all its children projects. And this inheritance is transitive, which means that all the resources in those projects inherit the policy too.

An example IAM resource hierarchy



There's one important rule to keep in mind. The policies implemented at a higher level in this hierarchy can't take away access that's granted at lower level. For example, suppose that a policy applied on the "bookshelf" project gives user Pat the right to modify a Cloud Storage bucket. But a policy at the organization level says that Pat can only view Cloud Storage buckets, not change them. The more generous policy takes effect. Keep this in mind as you design your policies.

Agenda

Google Cloud terminology

Google Cloud resource hierarchy

Identity and Access Management (IAM)

Lab

Identity

Interacting with Google Cloud

Lab

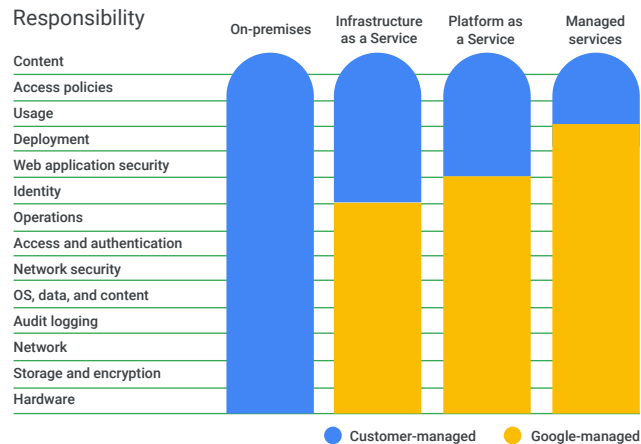
Billing, labels, and quotas

Lab

In this video, I will introduce Cloud Identity and Access Management and how to use it to control and secure your cloud environment.

Shared Responsibilities Model

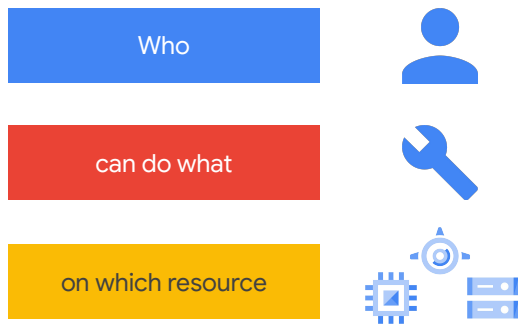
- Google is responsible for managing its infrastructure security.
- You are responsible for securing your data.
- Google helps you with best practices, templates, products, and solutions.



When you build an application on your on-premises infrastructure, you're responsible for the entire stack. When you move an application to Google Cloud, Google handles many of the lower layers of security. This concept of shared responsibilities is called the Shared Responsibilities Model and clearly defines which responsibilities are handled by the Cloud Provider (Google) and which responsibilities are handled by the customer. Because of its scale, Google can deliver a higher level of operational efficiency and security at these layers than most of its customers could afford to do on their own.

As shown in the slide, the upper (blue) layers of the responsibility model remain the customer's responsibility. Google provides tools, such as IAM, to help customers implement the policies they choose at these layers.

Identity and Access Management



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

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

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

Who: IAM policies can apply to any of four types of principals



Google account
test@gmail.com



Service account
test@project_id.iam.gserviceaccount.com



Google group
test@googlegroups.com



Cloud Identity
example.com

The “who” part of an IAM policy can be a Google account, a Google group, a service account, or a Cloud Identity domain. We will explore all of these identities later in the module.

Can do what: IAM roles are collections of related permissions

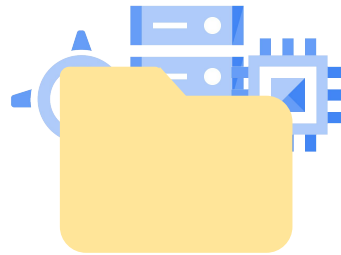
Basic



Predefined



Custom



The “can do what” part of an IAM policy is defined by a role. There are three kinds of roles in Cloud IAM. Let’s explore each in turn.

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



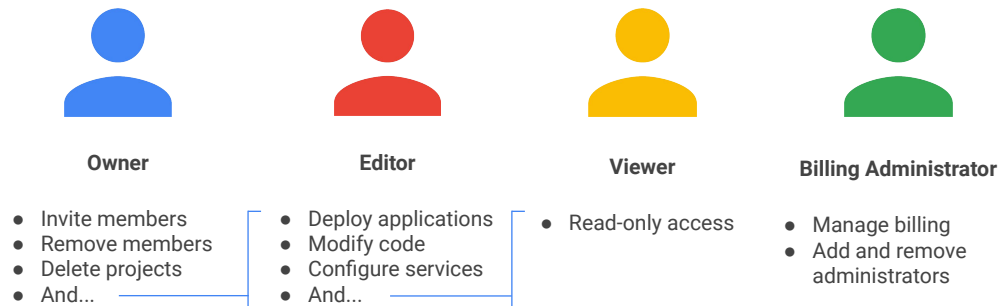
can do what



on all resources

The basic roles are broad. You apply them to a Google Cloud project, and they affect all resources in that project, from virtual machines to firewall rules, databases, and logs.

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



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

These are the Owner, Editor, and Viewer roles. If you're a viewer on a given resource, you can examine it but not change its state. If you're an editor, you can do everything a viewer can do plus change its state. And if you're an owner, you can do everything an editor can do plus manage roles and permissions on the resource. The owner role on a project lets you do one more thing too: you can set up billing. Often companies want someone to be able to control the billing for a project without the right to change the resources in the project, and that's why you can grant someone the billing administrator role.

Be careful! If you have several people working together on a project that contains sensitive data, basic roles are probably too coarse a tool. Fortunately, Cloud IAM provides finer-grained types of roles.

IAM predefined roles apply to a particular Google Cloud service in a project



can do what

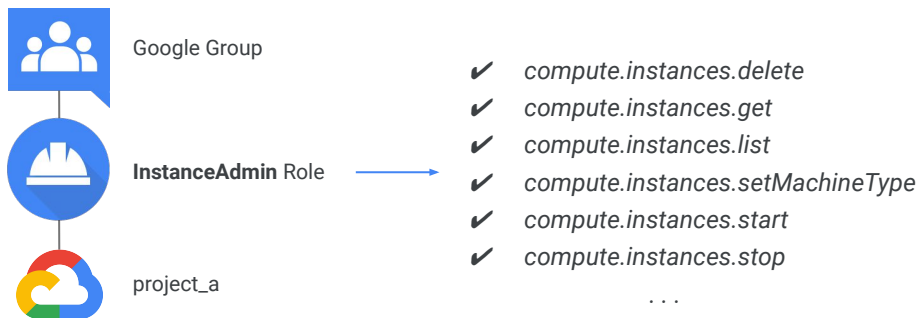


on Compute Engine resources in
this project, or folder, or org

Google Cloud services offers their own sets of predefined roles, and they define where those roles can be applied. For example, later in this course, we'll talk more about Compute Engine, which offers virtual machines as a service. Compute Engine offers a set of predefined roles, and you can apply them to Compute Engine resources in a given project, a given folder, or an entire organization.

Another example: consider Cloud Bigtable, which is a managed database service. Cloud Bigtable offers roles that can apply across an entire organization, to a particular project, or even to individual Bigtable database instances.

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



Compute Engine's `instanceAdmin` role lets whoever has it perform a certain set of actions on virtual machines. What set of actions? Those listed here: listing them, reading and changing their configurations, and starting and stopping them. And which virtual machines? Well, that depends on where the role is applied.

In this example, all the users of a certain Google group have the role, and they have it on all the virtual machines in project A.

Compute Engine IAM roles

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

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

Compute Engine IAM roles

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

The Compute Admin role provides full control of all Compute Engine resources. This includes all permissions that start with *compute*, which means that every action for any type of Compute Engine resource is permitted.

Compute Engine IAM roles

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

The Network Admin role contains permissions to create, modify, and delete networking resources, *except* for firewall rules and SSL certificates.

In other words, the network admin role allows read-only access to firewall rules, SSL certificates, and instances to view their ephemeral IP addresses.

Compute Engine IAM roles

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

The Storage Admin role contains permissions to create, modify, and delete disks, images, and snapshots.

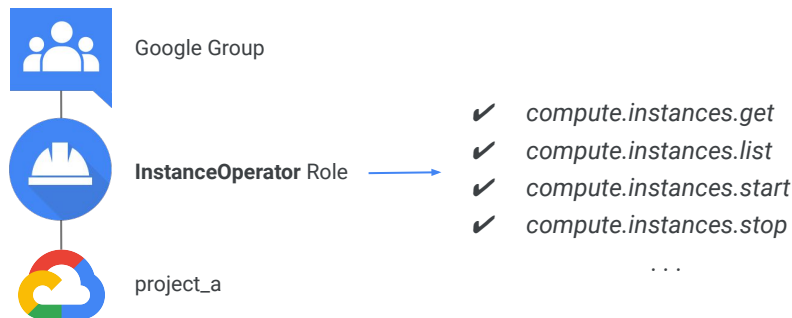
For example, if your company has someone who manages project images and you don't want them to have the editor role on the project, grant their account the Storage Admin role on the project.

Compute Engine IAM roles

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

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

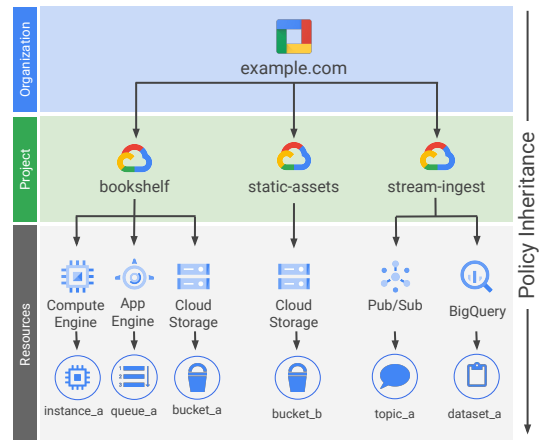
IAM custom roles let you define a precise set of permissions



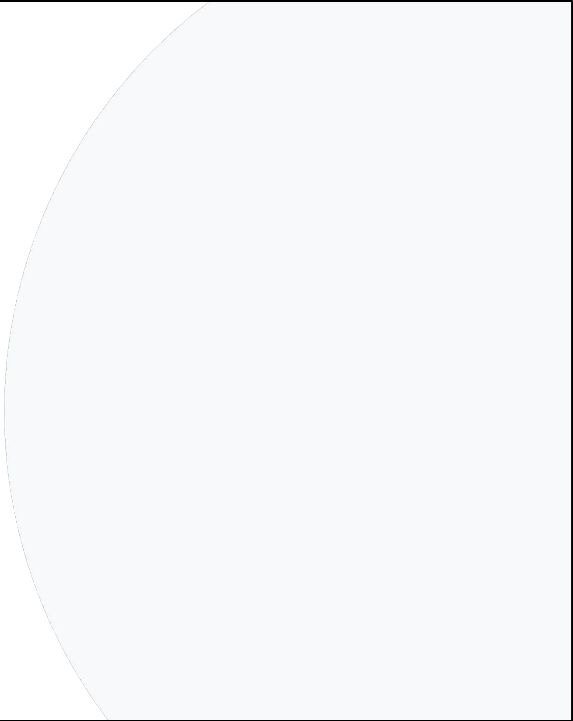
That's what custom roles permit. A lot of companies use a "least-privilege" model, in which each person in your organization has the minimal amount of privilege needed to do his or her job. So, for example, maybe I want to define an "instanceOperator" role to allow some users to stop and start Compute Engine virtual machines but not reconfigure them. Custom roles allow me to do that.

A couple of cautions about custom roles. First, if you decide to use custom roles, you'll need to manage the permissions that make them up. Some companies decide they'd rather stick with the predefined roles. Second, custom roles can only be used at the project or organization levels. They can't be used at the folder level.

On which resource: Users get roles on specific items in the hierarchy



Remember that when you give a user, group, or service account a role on a specific element of the resource hierarchy, the resulting policy applies to the element you chose, as well as to elements below it in the hierarchy.

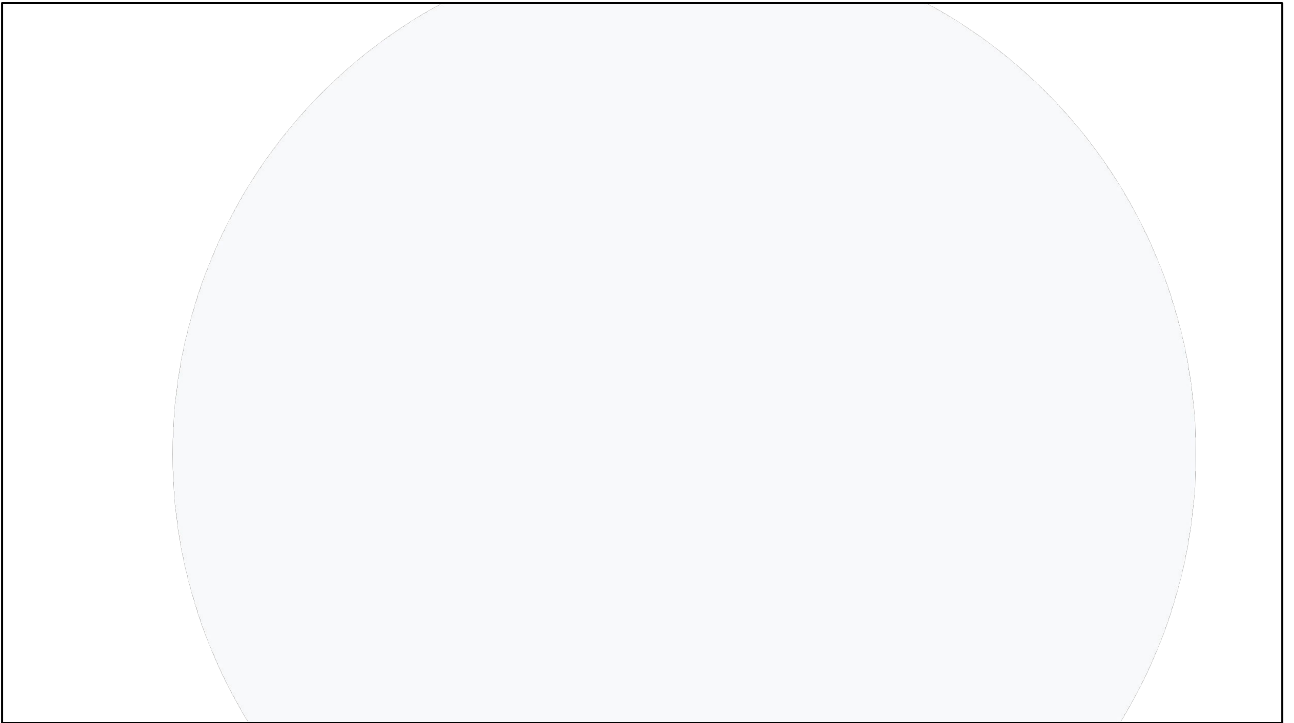


Service accounts provide an identity for carrying out server-to-server interactions

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

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

For example, if you write an application that interacts with Cloud Storage, it must first authenticate to either the Cloud Storage APIs.



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

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

Lab Intro

Identity and Access Management



It's time to apply what you learned.

In this lab, you'll use Cloud IAM to manage access control to grant access to employees and external users. You will also create a service account and assign it to a virtual machine.

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

Lab Solution

Identity and Access Management



In this lab you used Identity and Access Management (IAM) to grant access to both a cloud identity user (@qwiklabs.net) and an external Gmail user. You also created a service account, granted it minimal permissions, and assigned the service account to a Compute Engine virtual machine.

Agenda

Google Cloud terminology

Google Cloud resource hierarchy

Identity and Access Management (IAM)

Lab

Identity

Interacting with Google Cloud

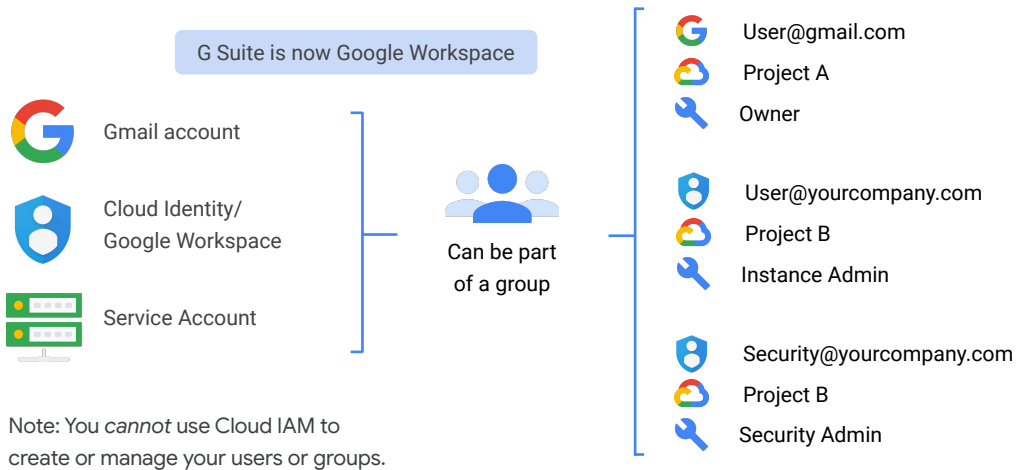
Lab

Billing, labels, and quotas

Lab

In this video, you'll learn how identities are being managed in Google Cloud. The topic of Identity will be introduced in more detail in module 6.

Identity



Many new Google Cloud customers get started by logging into the Cloud Console with a Google Gmail account. This approach is easy to get started with, but its disadvantage is that your team's identities are not centrally managed. For example, if someone leaves your organization, there is no centralized way to remove their access to your cloud resources immediately.

Google Cloud customers who are also Workspace customers can define Google Cloud permissions in terms of Workspace users and groups. This way, when someone leaves your organization, an administrator can immediately disable their account and remove them from any associated groups using the Google Admin Console.

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

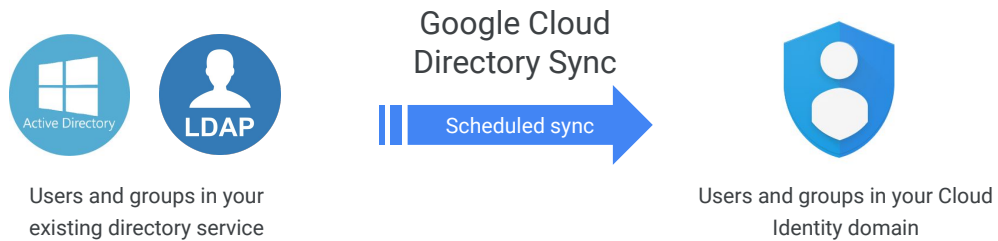
A service account is a special kind of account that belongs to an application or a virtual machine (VM) instance, not a person. Applications use service accounts to make authorized API calls. You can create as many service accounts as needed to represent the different logical components and security boundaries of your

application.

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

It is important to note that you cannot use Cloud IAM to create or manage your users or groups. Instead, you use Cloud Identity or Workspace within the Google Admin panel, to create and manage users.

What if you already have a different corporate directory?



Using Google Cloud Directory Sync, also known as GCDS, your administrators can enable the capability to leverage Google Cloud resources using the same usernames and passwords your company already uses for popular Directory Services platforms like Microsoft Active Directory or LDAP. We will go into more details on GCDS in Module 6.

Agenda

Google Cloud terminology

Google Cloud resource hierarchy

Identity and Access Management (IAM)

Lab

Identity

[Interacting with Google Cloud](#)

Lab

Billing, labels, and quotas

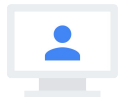
Lab

In this video, I will introduce the various ways you can interact with Google Cloud.

There are four ways to interact with Google Cloud

Cloud Platform Console

Web user interface



Cloud SDK and Cloud Shell

Command-line interface



Cloud Console Mobile App

For iOS and Android



REST-based API

For custom applications



There are four ways you can interact with Google Cloud, and we'll talk about each in turn: the Console, the SDK and Cloud Shell, the mobile app, and the APIs.

Google Cloud Console



- Centralized GUI-based console for all project data
- Developer tools
 - Cloud Source Repositories
 - Cloud Shell
 - Test Lab (mobile app testing)
- Access to product APIs
- Manage and create projects

Cloud Console is Google Cloud's Graphical User Interface (GUI) which helps you deploy, scale, and diagnose production issues in a simple web-based interface. With 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. Search to quickly find resources and connect to instances via SSH in the browser. Master the most complex tasks with Cloud Shell, your admin machine in the cloud, where you can use the built-in SDK.

The Cloud SDK and Cloud Shell



- The Cloud SDK includes CLI tools for Google Cloud products and services
 - `gcloud`, `gsutil` (Cloud Storage), `bq` (BigQuery)
- Available as Docker image
- Available via Cloud Shell
 - Containerized version of the Cloud SDK running on Compute Engine instance

The Google Cloud SDK is a set of tools that you can use to manage resources and applications hosted on Google Cloud. These include the [`gcloud` tool](#), which provides the main command-line interface for Google Cloud products and services, as well as [`gsutil`](#) and [`bq`](#). When installed, all of the tools within the SDK are located under the `bin` directory.

Google Cloud Shell provides you with command-line access to your cloud resources directly from your browser. Cloud Shell is a Debian-based virtual machine with a persistent 5-GB home directory, which makes it easy for you to manage your Google Cloud projects and resources. With Cloud Shell, the Cloud SDK `gcloud` command and other utilities you need are always installed, available, up to date, and fully authenticated when you need them.

RESTful APIs

- Programmatic access to products and services
 - Typically use JSON as an interchange format
 - Use OAuth 2.0 for authentication and authorization
- Enabled through the Google Cloud Console
- To help you control spend, most include daily quotas and rates (limits)
 - Quotas and rates can be raised by request

The services that make up Google Cloud offer Application Programming Interfaces (APIs), which allow you to programmatically control your cloud environment through code. Cloud APIs provide functionality similar to Cloud SDK and Cloud Console and allow you to automate your workflows by using your favorite language. Use these Cloud APIs with REST calls or client libraries in popular programming languages.

Cloud Console Mobile App

- Manage virtual machines and database instances
- Manage apps in Google App Engine
- Manage your billing
- Visualize your projects with a customizable dashboard



The Cloud Console Mobile App gives you a convenient way to discover, understand, and respond to production issues. Monitor and make changes to Google Cloud resources from your iOS or Android device. Manage Google Cloud resources such as projects, billing, App Engine apps, and Compute Engine VMs. Receive and respond to alerts helping you quickly address production-affecting issues.

Lab Intro

Getting Started with Cloud Shell
and gcloud Command Line
Interface



In this lab, you'll explore the features of Cloud Shell and Google Cloud SDK command line interface, gcloud.

Lab Solution

Getting Started with Cloud Shell
and gcloud Command Line
Interface



In this lab you explored the cloud shell environment and the gcloud command line interface, explored the interactive mode, and used Cloud Shell's developer tools to test a simple web application.

Agenda

Google Cloud terminology

Google Cloud resource hierarchy

Identity and Access Management (IAM)

Lab

Identity

Interacting with Google Cloud

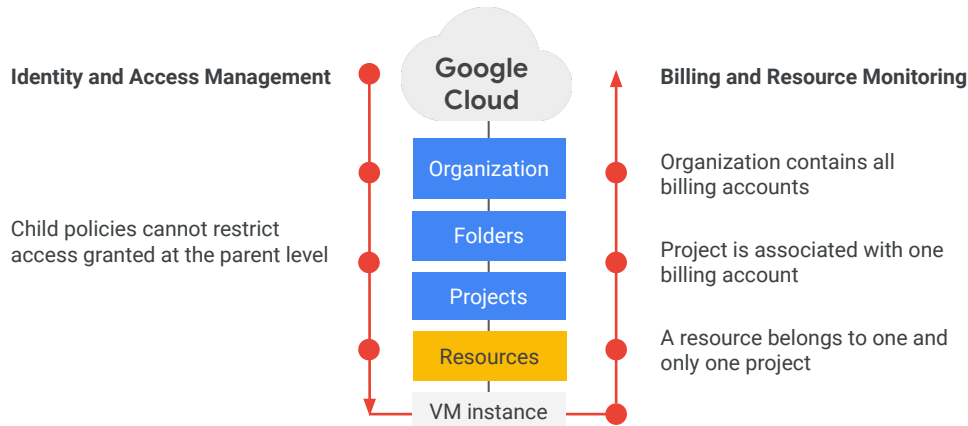
Lab

[Billing, labels, and quotas](#)

Lab

In this video, I will present the way billing works in Google Cloud, how to control spend with quotas, and how to leverage the power of labels.

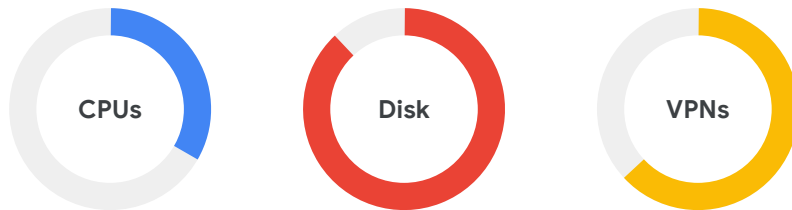
Resource Manager lets you hierarchically manage resources



Although IAM policies are inherited top-to-bottom, billing is accumulated from the bottom up, as you can see on the right. Resource consumption is measured in quantities, like rate of use or time, number of items, or feature use. Because a resource belongs to only one Project, a Project accumulates the consumption of all its resources.

Each Project is associated with one billing account, which means that an Organization node contains all billing accounts. Let's explore organizations, projects, and resources more.

Resource quotas



- Quotas **provide protection** against:
 - Cost overruns
 - Can be indicators of bad code
 - Other poorly behaved Google Cloud customers
- **Default quotas** may **increase** as **your use** of Google Cloud expands over time
- Most quotas are applied **per project**, based on **resource type and location**

Cloud resources have near unlimited capacity, and since you pay for what you consume, quotas protect you from unintentional expenditure. That's the reason all resources in Google Cloud are subject to Project quotas or limits where their purpose is to encourage you to make capacity planning a priority by setting upper limits for resources which can be consumed within your Project. A good example is decreasing a quota on the amount of vCPUs from the default 24 to 6 for a proof of concept or test Project, therefore controlling the monthly bill as a result. If your Project exceeds a particular quota while using a service, the platform will return an error.

Given these quotas, you may be wondering, how do I spin up one of those 96-core VMs?

As your use of Google Cloud expands over time, your quotas may increase accordingly. If you expect a notable upcoming increase in usage, you can proactively request quota adjustments from the Quotas page in the Cloud Console. This page will also display your current quotas.

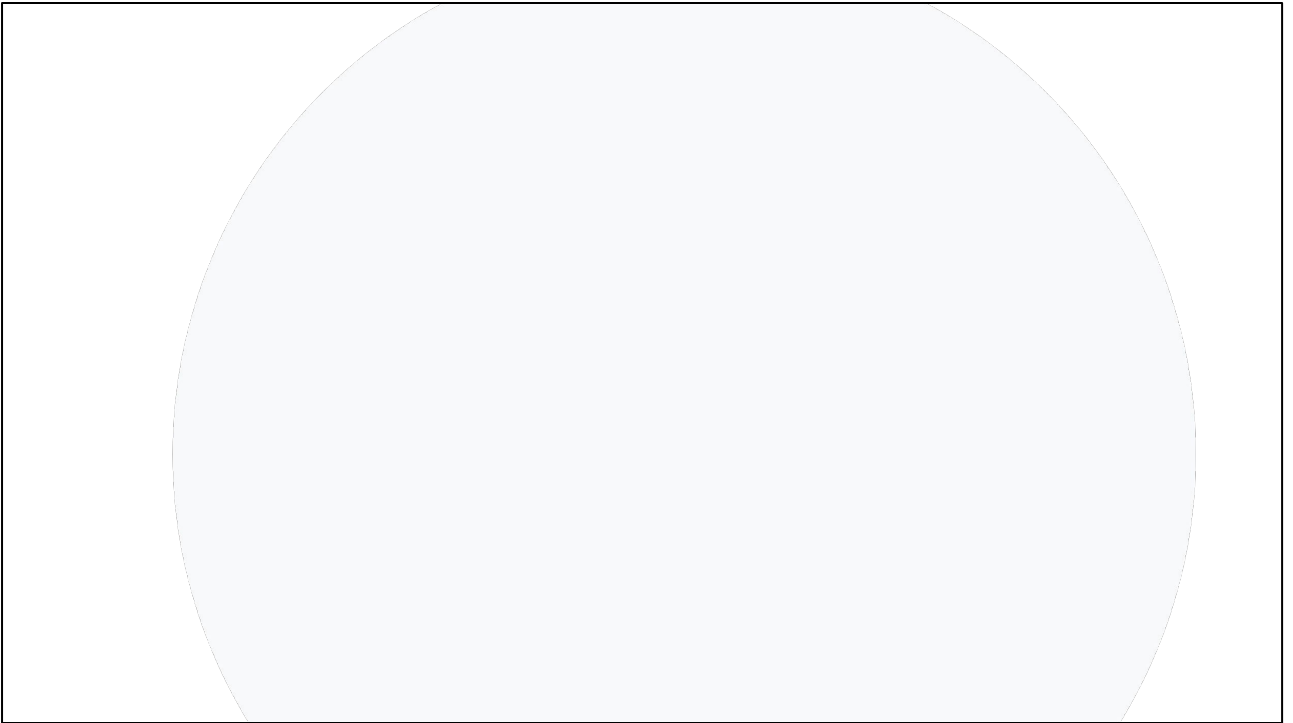
Why use Project quotas?

- Prevent runaway consumption in case of an error or malicious attack
- Prevent billing spikes or surprises
- Forces sizing consideration and periodic review

Project quotas prevent runaway consumption in case of an error or malicious attack. For example, imagine you accidentally create 100 instead of 10 Compute Engine instances using the `gcloud` command line. Having quotas in place can protect you from this scenario.

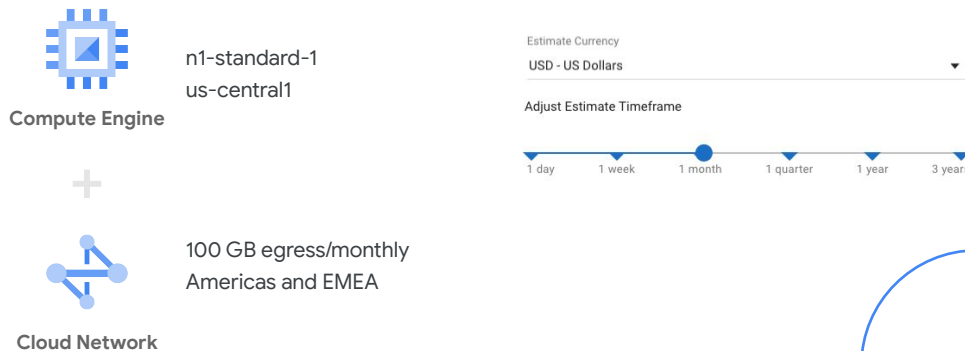
Quotas also prevent billing spikes or surprises. Quotas are related to billing, but we will go through how to set up budgets and alerts later, which will really help you manage billing efficiently.

Finally, quotas force sizing consideration and periodic review. For example, do you really need a 96-core instance, or can you go with a smaller and cheaper alternative?



It is also important to mention that quotas are the maximum amount of resources you can create for that resource type *as long as those resources are available*. Quotas do not guarantee that resources will be available at all times. For example, if a region is out of local SSDs, you cannot create local SSDs in that region, even if you still had quota for local SSDs.

Estimate costs with the Google Cloud Pricing Calculator



The screenshot displays the Google Cloud Pricing Calculator interface. On the left, two services are added to the estimate: 'Compute Engine' (n1-standard-1, us-central1) and 'Cloud Network' (100 GB egress/monthly, Americas and EMEA). A plus sign is between them. On the right, there are controls for 'Estimate Currency' (set to USD - US Dollars) and 'Adjust Estimate Timeframe' (a slider with markers for 1 day, 1 week, 1 month, 1 quarter, 1 year, and 3 years, with 1 month selected). A large blue circle is overlaid on the right side of the interface.

cloud.google.com/products/calculator/

Because each Google Cloud service has its own pricing model, we recommend using the Google Cloud pricing calculator to estimate the cost of a collection of resources. The pricing calculator is a web-based tool that allows you to specify the expected consumption of certain services and resources. You will receive an estimated cost for the utilization of these resources as the output from the pricing calculator.

For example, you can specify an n1-standard-1 VM instance in us-central1 along with 100 GB of egress traffic to Americas and EMEA. The pricing calculator then returns the total estimated cost. You can adjust the currency and time frame to meet your needs, and when you are done, you can email the estimate or save it to a specific URL for future reference.

Budgets and email alerts

1 Scope

Name *

Budget Name

Projects

All projects (2)

2 Amount

Budget type

Specified amount

A fixed amount that your spend will be compared against.

Target amount

\$ 500

3 Actions

Percent of budget	Amount	Trigger on
50 %	\$ 250	Actual
90 %	\$ 450	Actual
100 %	\$ 500	Actual

Specified amount

Last month's spend

Actual

Forecasted

To help with project planning and controlling costs, you can set a budget. Setting a budget lets you track how your spend is growing toward that amount. This screenshot shows the budget creation interface:

1. Set a budget name and specify which project this budget applies to.
2. Set the budget at a specific amount or match it to the previous month's spend.
3. Determine your budget percentage alert. These alerts send emails to billing admins after spend exceeds a percent of the budget or a specified amount.

In our case, it would send an email when spending reaches 50%, 90%, and 100% of the budget amount. You can even choose to send an alert when the spend is forecasted to exceed the percent of the budget amount by the end of the budget period.

Example budget alert email

Billing Alert Notification

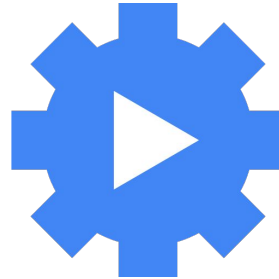
Dear Google customer,

You are receiving this email because you are a Google Cloud, Firebase, or API customer.

This is an automated notification to inform you that the project: **arch-gce** has exceeded **50%** of the monthly budget of **\$500.00**.

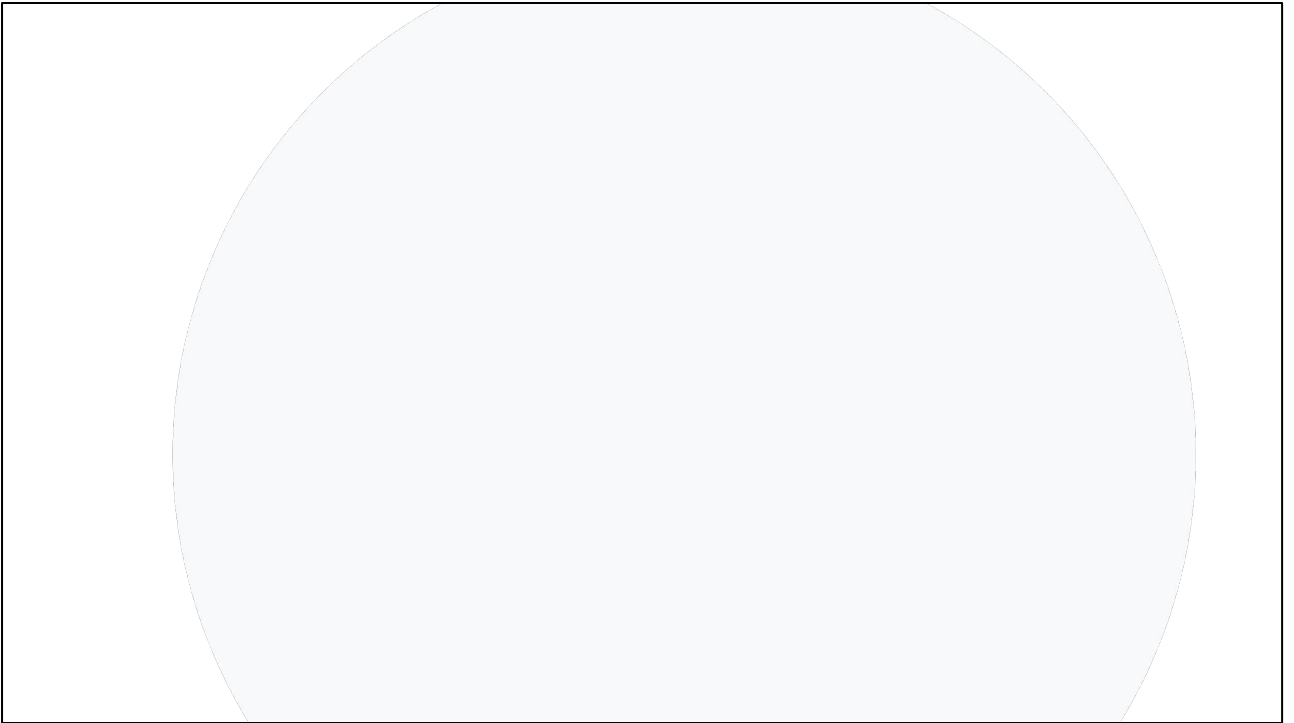
You are receiving this message because there is an alert configured on this project's budget. To disable this alert or modify the [budget's](#) threshold, please edit [your budget](#).

OR



Here is an example of an email notification. The email contains the project name, the percent of the budget that was exceeded, and the budget amount. It's worth mentioning that you can also respond to budget notification programmatically using webhooks, so you can develop your own solution.

Email isn't always the best way to stay up to date on your cloud costs, particularly if your budget is critical and time-sensitive. You can use programmatic notifications to forward your budget messages to other mediums and to automate cost management.



Labels are a utility for organizing Google Cloud resources. Labels are key-value pairs that you can attach to your resources, like VMs, disks, snapshots and images. You can create and manage labels using the Cloud Console, gcloud, or the Resource Manager API, and each resource can have up to 64 labels.

Labels are a utility for organizing Google Cloud resources

- Attached to resources: VM, disk, snapshot, image
 - Cloud Console, gcloud, or API
- *Example uses of labels:*
 - Inventory
 - Filter resources
 - In scripts
 - Help analyze costs
 - Run bulk operations

Key	Value
department	website-deve
engineering	development
owner	bobzalman
project	account-1569

[+ Add label](#)

[Save](#) [Cancel](#)

For example, you could create a label to define the environment of your virtual machines. Then you define the label for each of your instances as either production or test. Using this label, you could search and list all your production resources for inventory purposes.

Labels can also be used in scripts to help analyze costs or to run bulk operations on multiple resources. The screenshot on the right shows an example of 4 labels that are created on an instance.

Use labels for ...

- Team or Cost Center

team:marketing
team:research

- Components

component: redis
component: frontend

- Environment or stage

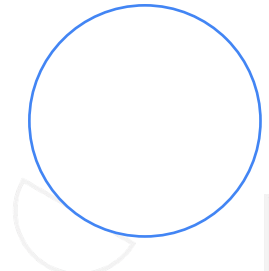
environment: prod
environment: test

- Owner or contact

owner:gaurav
contact:opm

- State

state:inuse
state:readyfordeletion



Let's go over some examples of what to use labels for:

- We recommend adding labels based on team or cost center to distinguish instances owned by different teams. You can use this type of label for cost accounting or budgeting. For example, team:marketing and team:research.
- You can also use labels to distinguish components. For example, component:redis, component:frontend.
- Again, you can label based on environment or stage.
- You should also consider using labels to define an owner or a primary contact for a resource. For example, owner:gaurav, contact:opm.
- Or add labels to your resources to define their state. For example, state:inuse, state:readyfordeletion

Visualize Google Cloud spend with Data Studio



Billing Dashboard



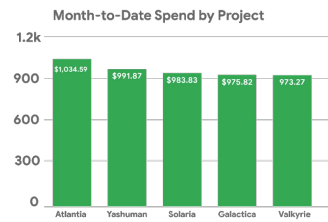
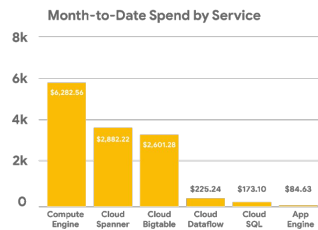
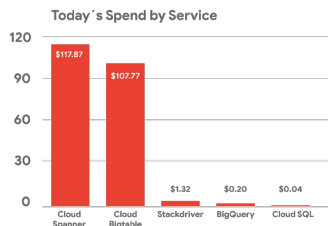
Daily View



Monthly View



Overall



You can even visualize spend over time with Data Studio. Data Studio turns your data into informative dashboards and reports that are easy to read, easy to share, and fully customizable. For example, you can slice and dice your billing reports using your labels.

Lab Intro

Understanding and Analyzing your
Costs with Google Cloud



In this lab, you will explore the Billing report section of Google Cloud Console, gaining insight into current and forecasted costs, and help analyze a billing report based on different parameters.

Lab Solution

Understanding and Analyzing your
Costs with Google Cloud



In this lab, you learned how to View Billing reports in the Google Cloud Console using a sample billing account.

In addition, you viewed your current and forecasted Google Cloud costs at project, product, and SKU level.

Lastly, you analyzed costs using report filters to identify cost drivers and trends. Examples of report filters include Projects, Products, SKUs, Locations, and Credits.



Google Cloud Fundamentals - Review

In this module, we compared the terminology from your source environment to Google Cloud's equivalent, for example that virtual machines are called Compute Engine instances on Google Cloud. You also learned how resource hierarchy levels define trust boundaries in the Google Cloud environment. Finally, you were introduced to Cloud Identity and Access Management and how you can use it to control and secure your cloud environment.

In the next module, we will show you how you can leverage Google's physical infrastructure by creating and configuring your own virtual private cloud network. In addition to explaining how to control access to your network with firewall rules and how to create subnets, we will identify the types of virtual machines that you can create in Compute Engine, and discuss how to choose the right configuration for your needs based on configuration and cost. Before you can start a migration to Google Cloud, you need to create a secure connection between your on-premises and your VPC. We will introduce you to the range of interconnect options offered by Google Cloud.

Move on to the next module to learn more.