



Introduction to Google Cloud



Welcome to 'Introduction to Google Cloud', the first module of this course.

Learn how to ...

Define cloud computing.

Identify Google Cloud's compute services.

Understand regions and zones.

Understand the cloud resource hierarchy.

Administer your Google Cloud resources.



This first module shares a simple explanation of what cloud computing is, so that we start off with the right framework of concepts.

Here's a quick preview: cloud computing entails resources being provided to you as a service. Google Cloud offers several services that let you run general-purpose compute workloads on Google's hardware, and we will meet them.

Google Cloud has a global footprint. I will explain how Google Cloud's resources around the world are organized into regions and zones.

You can use these resources, but of course it's your job to define policies for what you use. Google Cloud offers a hierarchical structure for organizing your use of cloud resources, and we'll meet it in this module.

Finally, you'll meet the tools that let you connect to Google Cloud and allocate, change, and release resources.

Agenda

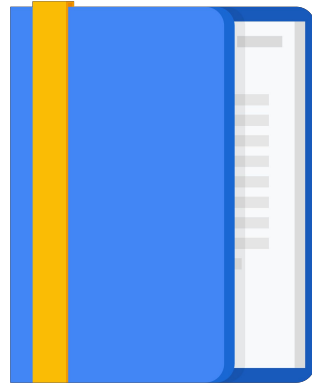
Cloud Computing and Google Cloud

Resource Management

Interacting with Google Cloud

Lab: Accessing the Cloud Console and Cloud Shell

Summary



Let's start by introducing the relationship between cloud computing and Google Cloud.

Cloud computing has five fundamental attributes



On-demand
self-service

No human
intervention
needed to get
resources



Broad network
access

Access from
anywhere



Resource
pooling

Provider shares
resources to
customers



Rapid
elasticity

Get more
resources
quickly as
needed



Measured
service

Pay only for what
you consume



Cloud computing has five fundamental attributes:

First, computing resources are on-demand and self-service. Cloud-computing customers use an automated interface and get the processing power, storage, and network they need, with no human intervention.

Second, resources are accessible over a network from any location.

Providers allocate resources to customers from a large pool, allowing them to benefit from economies of scale. Customers don't have to know, or care, about the exact physical location of those resources.

Resources are elastic. Customers who need more resources can get them rapidly. And when they need less, they can scale back.

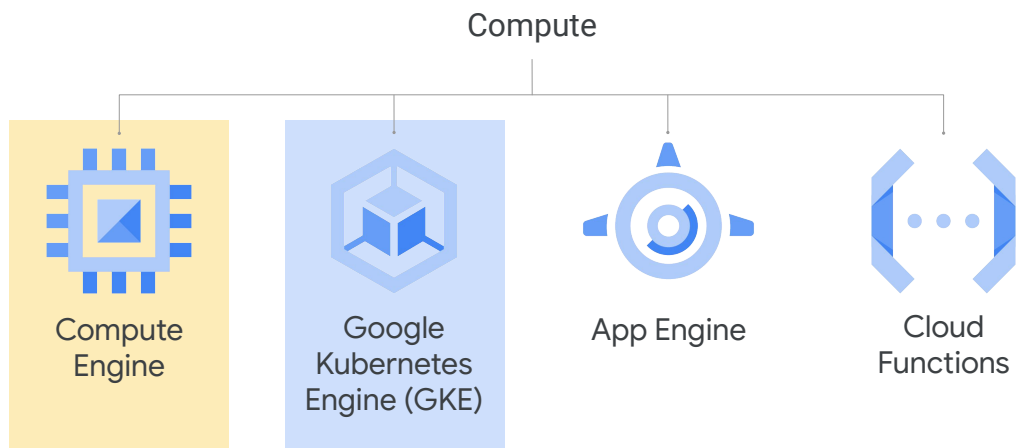
Finally, customers pay only for what they use or reserve, as they go. If they stop using resources, they simply stop paying.

*** Source: According to the [definition of cloud computing](#) proposed by the United States National Institute of Standards and Technology.



That's how public clouds such as Google Cloud work. Google Cloud offers a variety of cloud services, and architects and developers like you can choose among them to build your solutions. Many of these cloud services are very familiar, such as virtual machines on demand. Other services represent an entirely new paradigm. One of those services, Google Kubernetes Engine, is the focus of this course and the ones that follow. By the way, we informally call the service GKE, and you can too.

Google Cloud offers a range of services



The first thing many people ask of Google Cloud is: “Please run some code in the cloud for me” Google Cloud provides a range of services for doing that, each aimed at satisfying a different set of user preferences. Here’s a quick summary of the choices. In a later module we will compare them in more detail.

The service that might be most familiar to newcomers is Compute Engine, which lets you run virtual machines on demand in the cloud. It’s Google Cloud’s Infrastructure-as-a-Service solution. It provides maximum flexibility for people who prefer to manage server instances themselves.

GKE is different. It lets you run containerized applications on a cloud environment that Google manages for you, under your administrative control. What’s a “containerized application”? And what’s Kubernetes? We’ll learn a lot about those topics later in this course. For now, think of containerization as a way to package code that’s designed to be highly portable and to use resources very efficiently. And think of Kubernetes as a way to orchestrate code in containers.

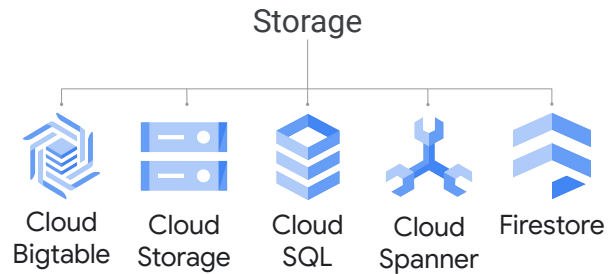
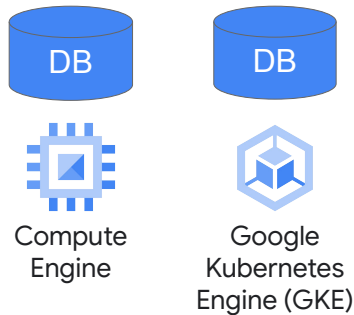
App Engine is Google Cloud’s fully managed Platform-as-a-Service framework. That means it’s a way to run code in the cloud without having to worry about infrastructure. You just focus on your code, and let Google deal with all the provisioning and resource management. You can learn a lot more about App Engine in the

specialization “Developing Applications in Google Cloud.”

Cloud Functions is a completely serverless execution environment, or Functions-as-a-Service. It executes your code in response to events, whether those occur once a day or many times per second. Google scales resources as required, but you only pay for the service while your code runs. The specialization “Developing Applications in Google Cloud” also discusses Cloud Functions.

In this specialization, GKE is our main focus. And GKE is built on top of Compute Engine, so we’ll learn more about that service too along the way.

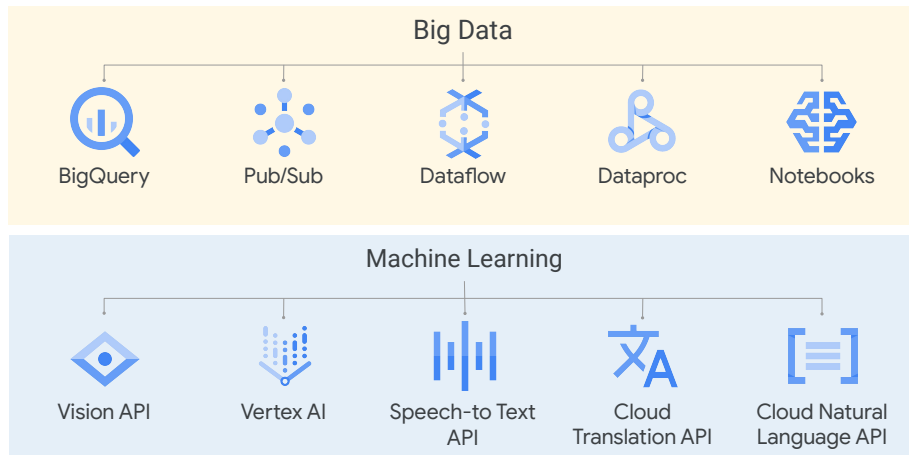
Build your own database solution or use a managed service



Most applications need a database of some kind. If you've built a cloud application, you can install and run your own database for it on a virtual machine in Compute Engine. You simply start up a virtual machine, install your database engine, and set it up, just like in your data center. Or you can run your own database server in Google Kubernetes Engine too. However, with either approach, you'll have to manage and support the database yourself.

Alternatively, you can use Google's fully managed database and storage services. What all these have in common is that they reduce the work it takes to store all kinds of data. Google Cloud offers relational and non-relational databases, and worldwide object storage. You will learn more about these later in this specialization.

Google Cloud offers fully managed big data and machine learning services



Just as with storage and database services, you could build and implement these services yourself, and some Google Cloud customers do.

But the fact that they're available as a service means that you can get started faster and deal with a lot less routine work along the way.

Agenda

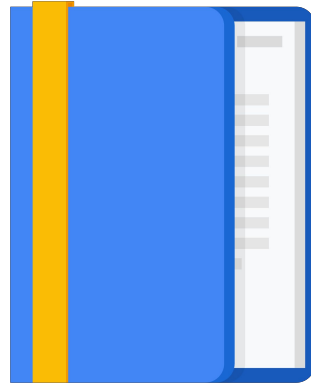
Cloud Computing and Google Cloud

Resource Management

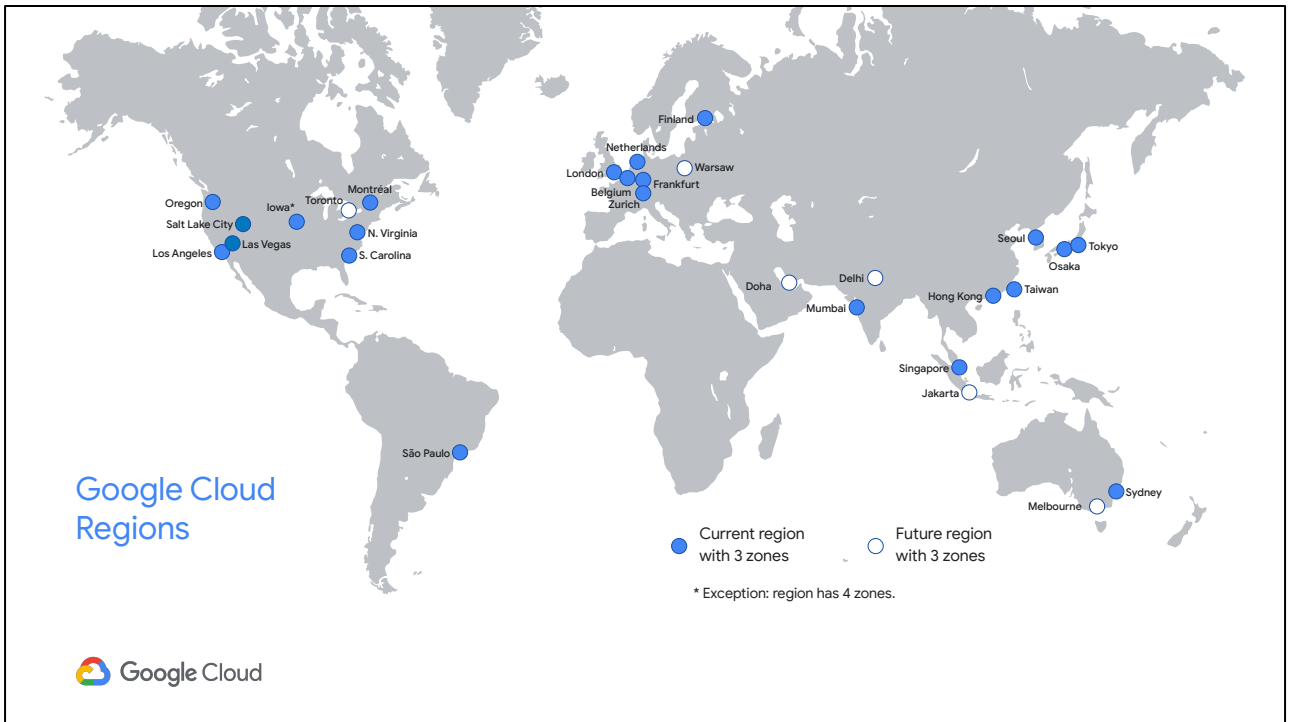
Interacting with Google Cloud

Lab: Accessing the Cloud Console and Cloud Shell

Summary

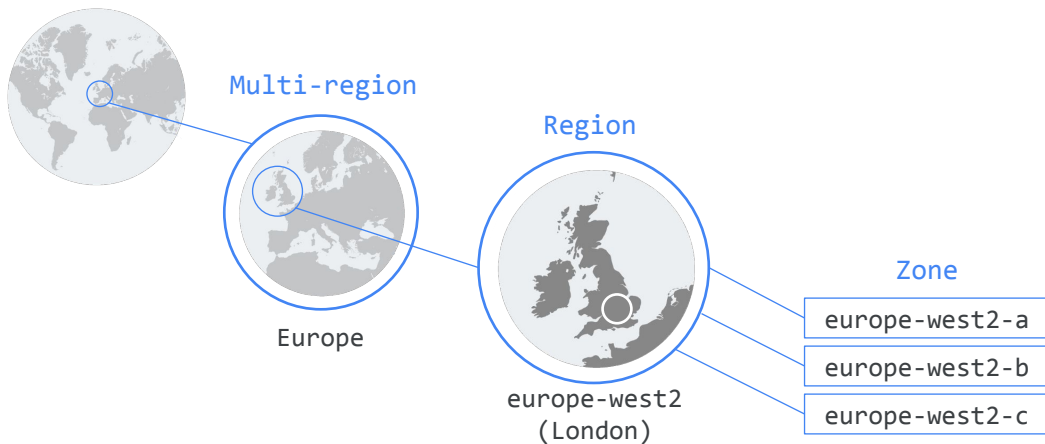


In this lesson you will learn about the topology of Google Cloud services and how they are distributed geographically in zones, regions and at a global scale. You will also learn how you structure the resources you use hierarchically, using Organizations, folders and projects, to manage your access control and your billing.



Behind the services provided by Google Cloud lie a huge range of Google Cloud resources: physical assets, such as physical servers and hard disk drives; and virtual resources, such as virtual machines and containers. These resources are managed by Google within its global data centers. These data centers are located in 18 regions, 55 zones, and more than 100 points of presence across 35 countries.

Google Cloud provides resources in multi-regions, regions, and zones



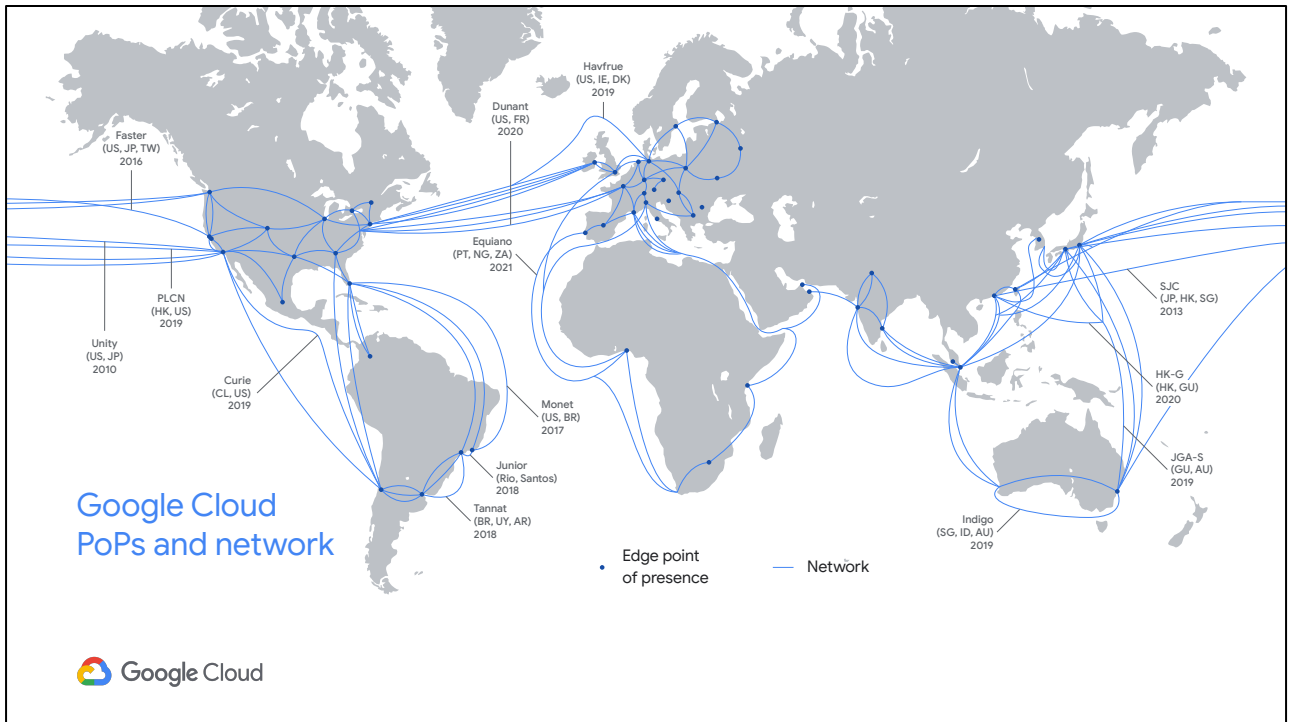
It divides the world into three multi-regional areas: the Americas, Europe, and Asia Pacific.

Next, the three multi-regional areas are divided into regions, which are independent geographic areas on the same continent. Within a region, there's fast network connectivity: generally round-trip network latencies of under 1 millisecond (that is, at the 95th percentile).

Finally, regions are divided into zones, which are deployment areas for Google Cloud resources within a focused geographic area. You can think of a zone as a data center within a region, although strictly speaking a zone isn't necessarily a single data center.

We mentioned Compute Engine earlier. Compute Engine virtual-machine instances reside within a specific zone. If that zone became unavailable, so would your virtual machine and the workload running on it. And GKE uses Compute Engine, so your GKE workloads could be affected similarly.

Deploying applications across multiple zones enables fault tolerance and high availability, and in this specialization, we will learn how to do that.

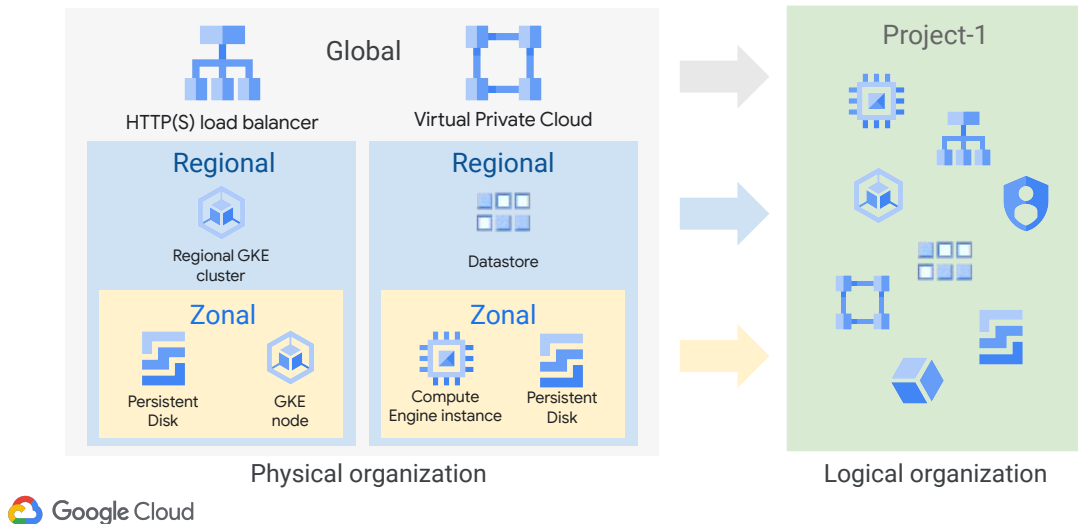


Google's data centers around the world are interconnected by the Google network, which, by some publicly available estimates, carries as much as 40% of the world's internet traffic every day. This is the largest network of its kind on Earth, and it continues to grow.

It's designed to provide the highest possible throughput and the lowest possible latencies for applications, including yours.

The network interconnects with the public internet at more than 90 internet exchanges and more than 100 points of presence worldwide. When an internet user sends traffic to a Google resource, Google responds to the user's request from an Edge Network location that will provide the lowest delay or latency. Google's edge caching network places content close to end users to minimize latency. Your applications in Google Cloud, including those running in GKE, can take advantage of this edge network too.

Resources are organized both physically and logically



When you take advantage of Google Cloud services and resources, you get to specify those resources' geographical locations. In many cases you can also specify whether you're doing so on a zonal level, a regional level, or a multi-regional level. Zonal resources operate within a single zone, which means that if a zone becomes unavailable, the resources will not be available either. A simple example would be a Compute Engine virtual machine instance and its persistent disks. GKE has a component called a "node," and these are zonal too.

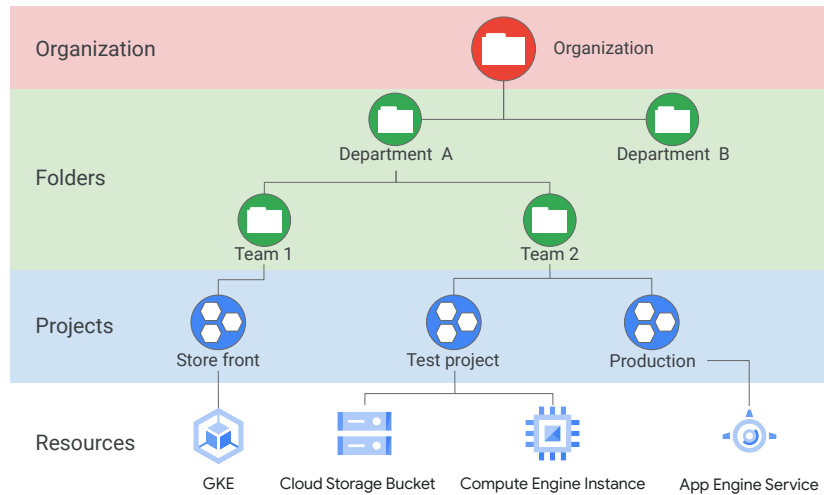
Regional resources operate across multiple zones, but within one region. An application using these resources can be redundantly deployed to improve its availability. Later in this specialization, you'll learn how to use GKE so that it has resources spread across different zones in a region. Datastore is another Google Cloud service that can be deployed in a similarly redundant way.

Finally, global resources can be managed across multiple regions. These resources can further improve the availability of an application. Examples of such resources include HTTPS load balancers and Virtual Private Cloud networks, which GKE users benefit from too.

The Google Cloud resources you use, no matter where they reside, must belong to a project. So what is a project? A project is the base-level organizing entity for creating and using resources and services and managing billing, APIs, and permissions. Zones and regions *physically* organize the Google Cloud resources you use, and projects *logically* organize them. Projects can be easily created, managed, deleted, or

even recovered from accidental deletions.

Resources have hierarchy



Each project is identified by a unique project ID and project number. You can name your project and apply labels for filtering. These labels are changeable, but the project ID and project number remain fixed.

Projects can belong to a “folder” which is another grouping mechanism.

You should use folders to reflect the hierarchy of your enterprise and apply policies at the right levels in your enterprise. You can nest folders inside folders.

For example, you can have a folder for each department, and within each department’s folder, you can have subfolders for each of the teams that make it up. Each team’s projects belong to its folder.

A single organization owns the folders beneath it.

An organization is the root node of a Google Cloud resource hierarchy. Although you are not required to have an organization to use Google Cloud, organizations are very useful. Organizations let you set policies that apply throughout your enterprise. Also, having an organization is required to use folders.

If you are already a Google Workspace customer, you have an organization already,

and if not, you can get one for free through Google Cloud Identity.

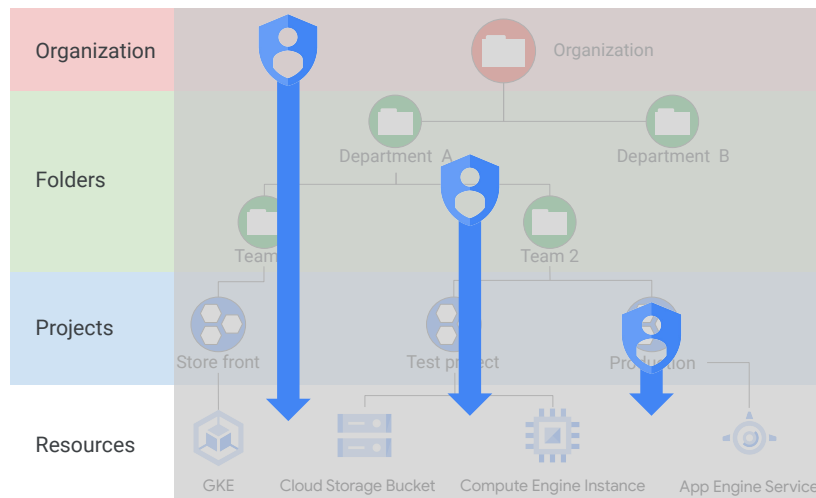
Your organization will have a fixed organization ID and a changeable display name.

The Google Cloud resource hierarchy helps you manage resources across multiple departments and multiple teams within an organization.

You can define a hierarchy that creates trust boundaries and resource isolation. For example, should members of your Human Resources team be able to delete running database servers? And should your engineers be able to delete the database containing employees' salaries? Probably not, in either case.

Cloud Identity and Access Management, also called Cloud IAM, lets you fine-tune access control to all the Google Cloud resources you use. You define Cloud IAM policies that control user access to resources.

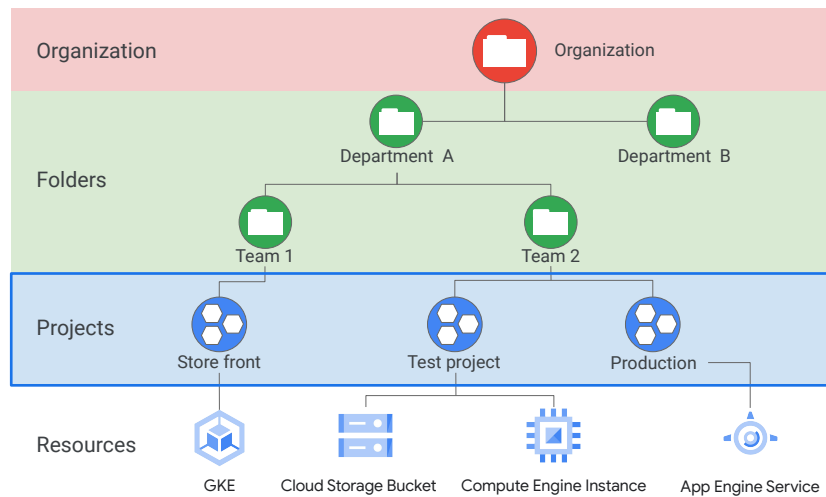
Resources have hierarchy



 Google Cloud

You apply these policies at the level you choose, and those policies inherit downwards. For example, a Cloud IAM policy applied at the organization level will be inherited by a folder, the project, and even the resources beneath it. Additional policies at lower levels of the hierarchy can grant additional permissions.

Resources have hierarchy

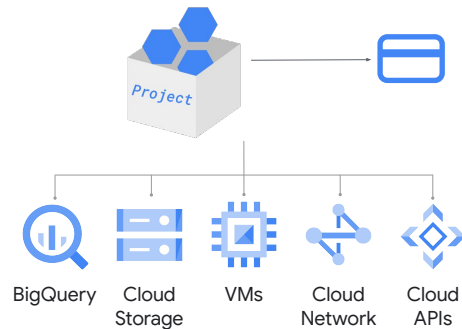


 Google Cloud

Billing, on the other hand, accumulates at the project level. Most Google Cloud customers have a resource hierarchy that looks like their employee organization chart, while their project billing looks like their cost-center structure.

How billing works

- Billing account pays for project resources.
- A billing account is linked to one or more projects.
- Charged automatically or invoiced every month or at threshold limit.
- Subaccounts can be used for separate billing for projects.



Billing in Google Cloud is set up at the project level. When you define a Google Cloud project, you link a billing account to it. This billing account is where you will configure all your billing information, including your payment option.

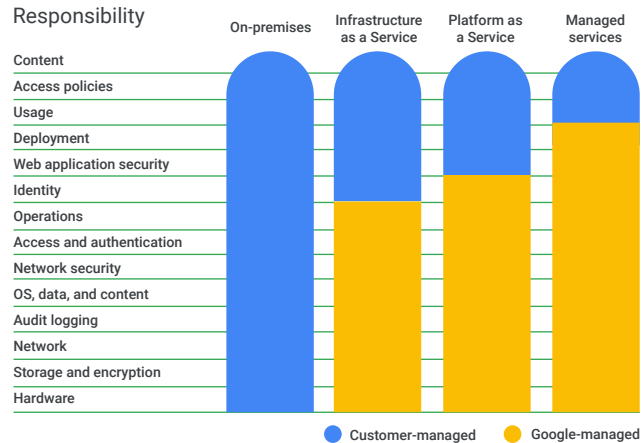
You can link your billing account to one or more projects. Projects that you don't link to any billing account can only use free Google Cloud services.

Your billing account can be charged automatically and invoiced every month, or at every threshold limit.

You can separate project billings by setting up billing subaccounts. Some Google Cloud customers who resell Google Cloud services use subaccounts for each of their own clients.

Resource hierarchy matters because of Google Cloud's shared security 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's security: from the physical security of the hardware and the premises in which they are housed, through the encryption of the data on disk, the integrity of your network, and all the way up to securing the content stored in those applications.

But when you move an application to Google Cloud, Google handles many of the lower layers of security, like the physical security of the hardware and its premises, the encryption of data on disk, and the integrity of the physical network. Because of its scale, Google can deliver a higher level of security at these layers than most customers could afford on their own.

The upper layers of the security stack, including the securing of data, remain your responsibility. Google provides tools to help you implement the policies you define at these layers. The resource hierarchy we just explored is one of those tools. Cloud IAM is another. You will learn more about security in depth later in this specialization.

How to keep your billing under control

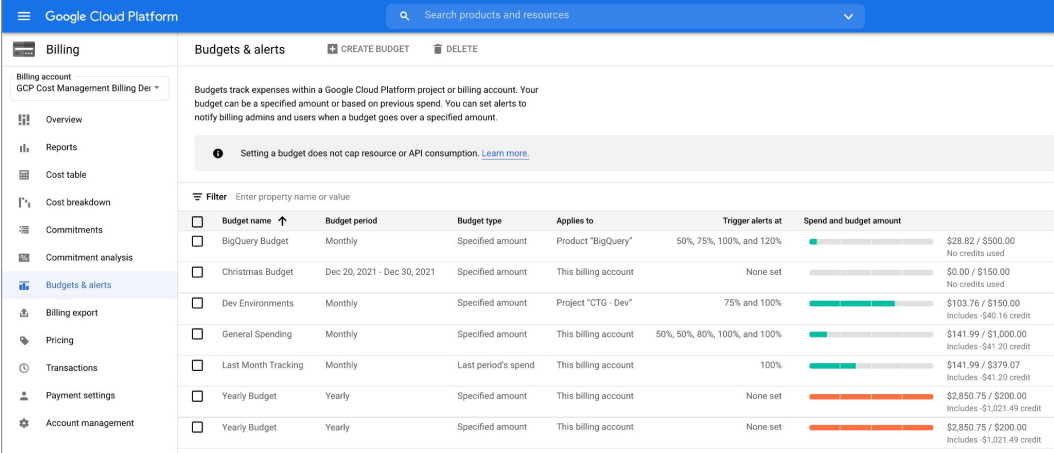
- Budgets and alerts
- Billing export
- Reports



You're probably thinking, "How can I make sure I don't accidentally run up a big Google Cloud bill?" Google Cloud provides three tools to help:

- Budgets and alerts,
- Billing export, and,
- Reports.

Budgets and alerts keep your billing under control



Budget name	Budget period	Budget type	Applies to	Trigger alerts at	Spend and budget amount
BigQuery Budget	Monthly	Specified amount	Product "BigQuery"	50%, 75%, 100%, and 120%	\$28.82 / \$500.00 No credits used
Christmas Budget	Dec 20, 2021 - Dec 30, 2021	Specified amount	This billing account	None set	\$0.00 / \$150.00 No credits used
Dev Environments	Monthly	Specified amount	Project "CTG - Dev"	75% and 100%	\$103.76 / \$150.00 Includes -\$40.16 credit
General Spending	Monthly	Specified amount	This billing account	50%, 50%, 80%, 100%, and 100%	\$141.99 / \$1,000.00 Includes -\$41.20 credit
Last Month Tracking	Monthly	Last period's spend	This billing account	100%	\$141.99 / \$379.07 Includes -\$41.20 credit
Yearly Budget	Yearly	Specified amount	This billing account	None set	\$2,850.75 / \$200.00 Includes -\$1,021.49 credit
Yearly Budget	Yearly	Specified amount	This billing account	None set	\$2,850.75 / \$200.00 Includes -\$1,021.49 credit



You can define budgets at the billing account level or at the project level.

To be notified when costs approach your budget limit, you can create an alert. For example, with a budget limit of \$20,000 and an alert set at 90%, you'll receive a notification alert when your expenses reach \$18,000.

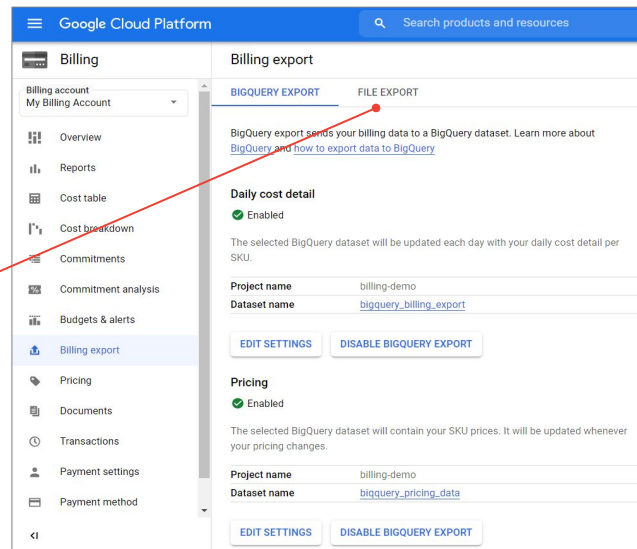
You can also set up a webhook to be called in response to an alert. This webhook can control automation based on billing alerts. For example, you could trigger a script to shut down resources when a billing alert occurs. Or you could use this webhook to file a trouble ticket for your team.

Source:

<https://cloud.google.com/solutions/best-practices-for-iam-and-billing-in-higher-education>

Billing export allows you to send your billing data to a BigQuery dataset

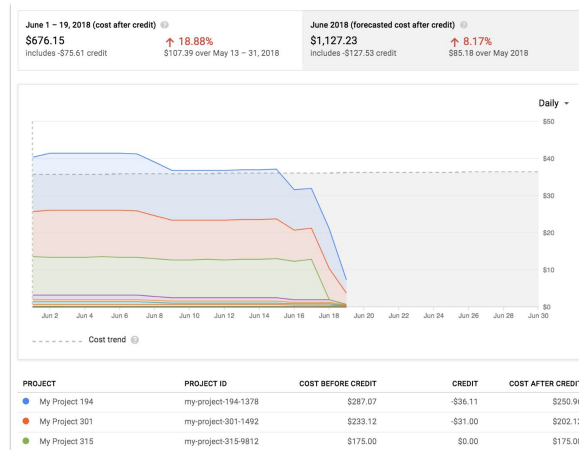
File export of billing data to CSV and JSON is deprecated.



Cloud Billing export to BigQuery enables you to export detailed Google Cloud billing data (such as usage, cost estimates, and pricing data) automatically throughout the day to a BigQuery dataset that you specify. Then you can access your Cloud Billing data from BigQuery for detailed analysis, or use a tool like Google Data Studio to visualize your data.

Note that the exporting of Cloud Billing data to a file is now deprecated and available only to existing customers using this feature.

Reports is a visual tool to monitor expenditure



And Reports is a visual tool in the Console that allows you to monitor expenditure based on a project or services.

Source:

<https://cloud.google.com/billing/docs/how-to/reports>

Quotas are helpful limits



Rate quota

GKE API: 1,000 requests per 100 seconds

Allocation quota

5 networks per project

Many quotas are changeable



Google Cloud also implements quotas, which limit unforeseen extra billing charges. Quotas are designed to prevent the over-consumption of resources because of an error or a malicious attack.

Quotas apply at the level of the Google Cloud project.

There are two types of quotas: rate quotas and allocation quotas. Rate quotas reset after a specific time. For example, by default, the GKE service implements a quota of 1,000 calls to its API from each Google Cloud project every 100 seconds. After that 100 seconds, the limit is reset. This doesn't limit the rate of calls to your *applications running in GKE*, but, rather, calls to the administrative configuration of your GKE clusters themselves. It would be very unusual to make that many calls in such a short period of time. The quota might well catch and stop erroneous behavior.

Allocation quotas govern the number of resources you can have in your projects. This count doesn't reset at intervals; instead you need to free up resources to stay within them. For example, by default, each Google Cloud project has a quota allowing it no more than 5 Virtual Private Cloud networks.

These quotas are not the same for all projects. Although projects start with the same quotas, you can change some of them by requesting an increase from Google Cloud

Support. Some quotas may increase automatically, based on your use of a product. And you can use the Cloud Console to explicitly lower some of them for your own projects, say, if you want to put a more stringent cap on your consumption.

Finally, some quotas are fixed for all Google Cloud customers. Regardless, in addition to their benefits to customers, Google Cloud quotas also protect the community of Google Cloud users by reducing the risk of unforeseen spikes in usage.

Agenda

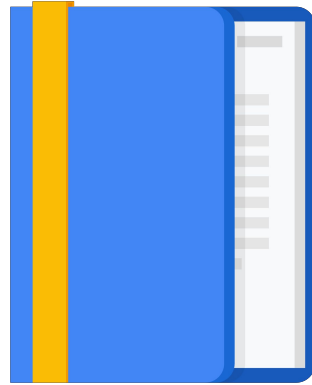
Cloud Computing and Google Cloud

Resource Management

Interacting with Google Cloud

Lab: Accessing the Cloud Console and Cloud Shell

Summary



This next lesson looks at how you interact with Google Cloud. You'll learn about the Google tools and interfaces that you can use to manage and configure Google Cloud resources.

There are four ways to interact with Google Cloud



Google Cloud 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

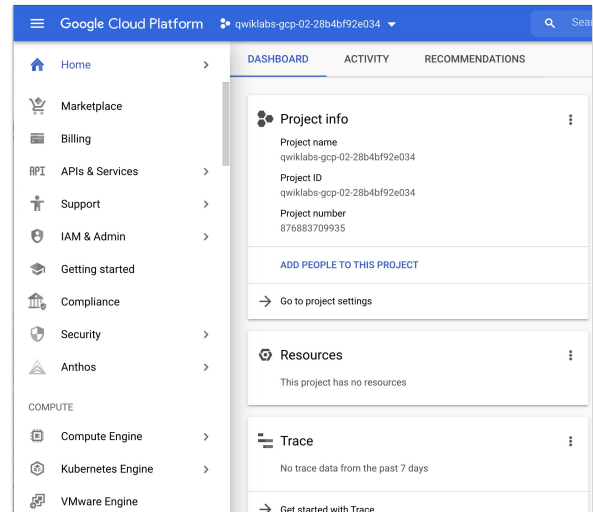


The Google Cloud Console, the Cloud Shell and Cloud SDK, the Cloud Console mobile app, and REST-based APIs.

We will not focus very much on these APIs in this specialization. Developers use them to build applications that allocate and manage Google Cloud resources. But our present focus is on letting Kubernetes manage resources for us.

The Cloud Console

- Web-based GUI to manage all Google Cloud resources.
- Executes common tasks using simple mouse clicks.
- Provides visibility into Google Cloud projects and resources.

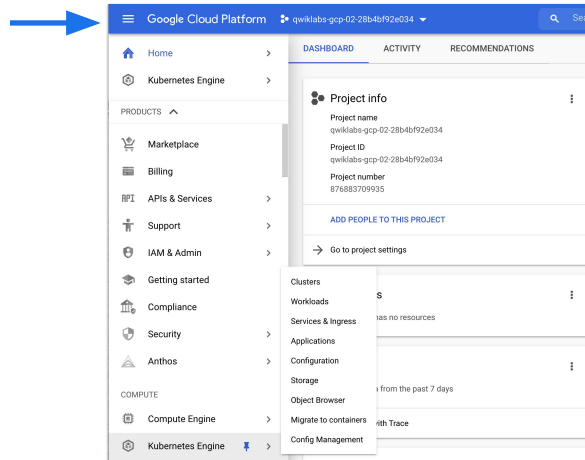


The Cloud Console is a web-based, graphical user interface from where you manage your Google Cloud resources.

It lets you execute common tasks using simple mouse clicks with no need to remember commands or avoid typos.

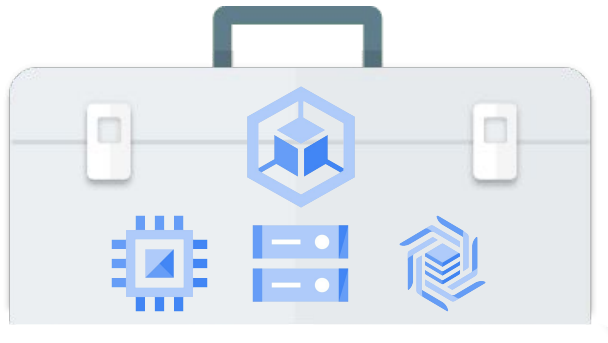
It also provides visibility into your Google Cloud project and its resources.

Interacting with the Cloud Console



You can sign in to the Cloud Console from a web browser at console.cloud.google.com. All Google Cloud services are accessible through the simple Navigation Menu button in the top-left corner. You can pin frequently used services to this menu. You'll learn how to use the Cloud Console during an upcoming lab.

The Cloud SDK contains a set of command-line tools for Google Cloud



- gcloud
- kubectl
- gsutil
- bq



Alternatively, you can download and install the Cloud SDK onto a computer of your choice. The Cloud SDK contains a set of command-line tools for Google Cloud. Most notably, it contains the gcloud and kubectl commands, which we will use a lot in this course. It also contains the gsutil and bq utilities. You can run these tools interactively, or in your automated scripts. The Cloud SDK contains client libraries for various programming languages too.

Cloud Shell is an alternative to installing the Cloud SDK

- Command-line access to your cloud resources directly from your browser.
- Constant availability of gcloud command-line tool and other utilities.
- Ephemeral Compute Engine virtual machine instance.
- Built-in authorization for access to Cloud Console projects and resources.



But what if it isn't convenient to install the Cloud SDK on the machine you're working with? Cloud Shell provides command-line access to your cloud resources directly from within your browser.

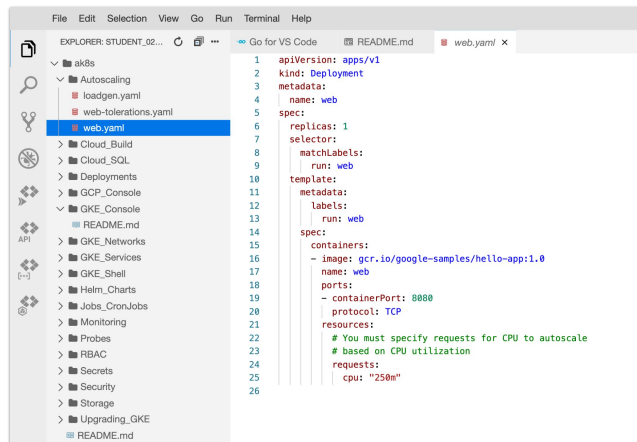
Using Cloud Shell, you can manage your projects and resources easily without having to install the Cloud SDK or other tools locally. The Cloud SDK gcloud and kubectl command-line tools and other utilities are always available, up to date, and fully authenticated.

So how does Cloud Shell do that? It's built using a docker container running on a Compute Engine virtual machine instance that you're not billed for. Each Google Cloud user has one.

Your Cloud Shell virtual machine is ephemeral, which means that it will be stopped whenever you stop using it interactively, and it'll be restarted when you re-enter Cloud Shell. So you wouldn't want to run a production web server in your Cloud Shell, for example. You also get five gigabytes of persistent disk storage that is reattached for you every time a new Cloud Shell session is started.

It also provides web preview functionality and built-in authorization for access to Cloud Console projects and resources, including your GKE resources.

Cloud Shell Editor is a tool for editing files inside your Cloud Shell environment



Cloud Shell Editor is a tool for editing files inside your Cloud Shell environment in real time within the web browser. You can also use text editors from the Cloud Shell command prompt.

This tool is extremely convenient when working with code-first applications or container-based workloads, because you can edit files easily without the need to download and upload changes. But is the easy way *always* the best way? Of course not. Later in the specialization, we'll talk about best management practices.

The Cloud Console and Cloud Shell

The screenshot shows the Google Cloud Platform console interface for the 'Kubernetes Engine' section. The top navigation bar includes the Google Cloud logo, the project name 'qwklabs-gcp-03-135971e70bda', and a search bar. The left sidebar contains a menu with options like Clusters, Workloads, Services & Ingress, Applications, Configuration, Storage, Marketplace, Release Notes, and Cloud Shell. The main content area is titled 'Kubernetes clusters' and includes tabs for 'OVERVIEW', 'COST OPTIMIZATION', and 'PERFORMANCE'. A table lists the clusters, with one cluster named 'standard-cluster-1' in the 'us-central1-a' region, having 2 nodes, 4 vCPUs, and 8 GB of memory. Below the table, there is a 'CLOUD SHELL' section with a terminal window. The terminal shows the output of the following commands:

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwklabs-gcp-03-135971e70bda.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
student_02_aea05c1a2c8@cloudshell:~ (qwklabs-gcp-03-135971e70bda) $ kubectl get service web
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
web       NodePort    10.11.255.177    <none>            8080:32452/TCP    34m
student_02_aea05c1a2c8@cloudshell:~ (qwklabs-gcp-03-135971e70bda) $ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
web-96c9d4b64-j2ctr   1/1     Running   0           34m
student_02_aea05c1a2c8@cloudshell:~ (qwklabs-gcp-03-135971e70bda) $
```



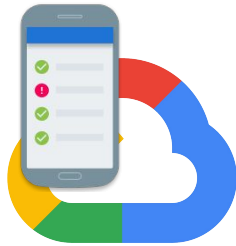
Here's a screenshot of the Cloud Console's GKE area, showing you its web-based interface for administering your GKE resources. The bottom third of the screenshot is your Cloud Shell in operation, where you can launch commands to administer those resources as well. Some of those commands are from the Cloud SDK, and others will be specific to your workload. Later in this course, we'll learn about the `kubectl` command, and you can see it being launched from Cloud Shell here.

Cloud Console mobile app

Start, stop, and SSH into Compute Engine instances.

Get up-to-date billing information and alerts.

Set up customizable graphs, showing key metrics.



Finally, there's the Cloud Console mobile app, available for iOS and Android. It offers many capabilities, like managing virtual machines, and viewing their logs, getting up-to-date billing information for your projects, and getting billing alerts for projects that are going over budget, and setting up customizable graphs, showing key metrics such as CPU usage, network usage, requests per second, and server errors. We will not use it in this course, but it's a resource you might find convenient, and of course it's at no additional charge.

Agenda

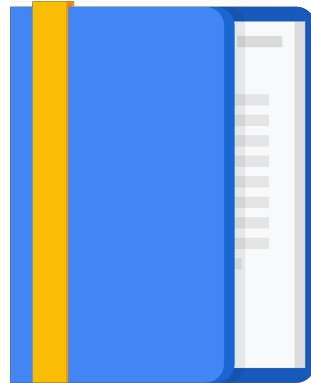
Cloud Computing and Google Cloud

Resource Management

Interacting with Google Cloud

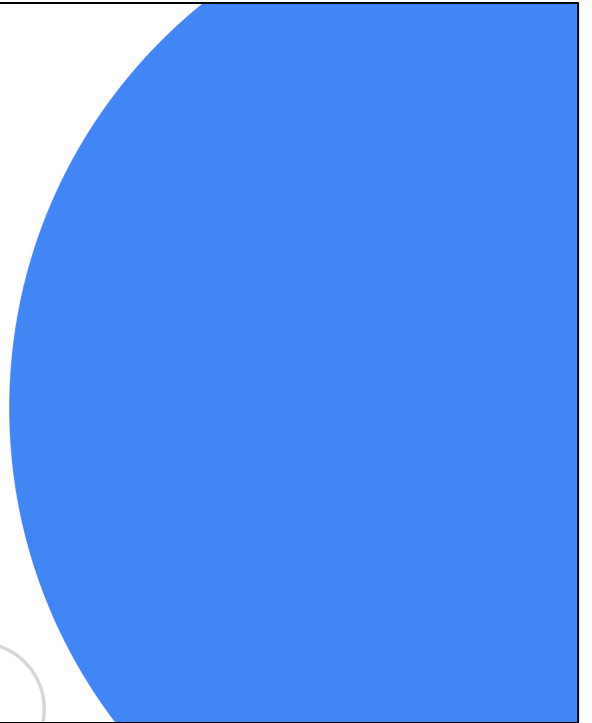
[Lab: Accessing the Cloud Console and Cloud Shell](#)

Summary



Lab Intro

Accessing the Cloud Console and
Cloud Shell



In this lab, you'll access and become familiar with the Cloud Console. You'll also become familiar with features of the Cloud Shell, including the Cloud Shell code editor.

You'll create buckets, VMs, and service accounts using the Cloud Console and the Cloud Shell and execute several other commands through the Cloud Shell.

Agenda

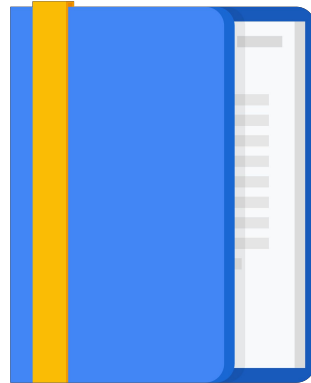
Cloud Computing and Google Cloud

Resource Management

Interacting with Google Cloud

Lab: Accessing the Cloud Console and Cloud Shell

[Summary](#)



Summary

Cloud computing means on-demand, pay-as-you-go resources.

Google Cloud offers 4 compute services.

Google Cloud is organized into regions and zones.

The resource hierarchy helps you manage your Google Cloud use.

Use the Cloud Console and the Cloud Shell for access.



That concludes the 'Introduction to Google Cloud' module. Let me remind you of what you learned.

Cloud computing is a way to organize your use of IT in which a provider gives you on-demand access over the network to resources from a pool they maintain. You pay for what you use or reserve. The provider maintains the infrastructure for you, and you can turn it off when you're done.

Google Cloud offers a lot of different kinds of cloud computing services, including four that run your code for you on hardware Google maintains. In this course, we are focusing on Kubernetes Engine.

All the resources offered by Google Cloud are organized into regions and zones. You can use resources in several zones in a region to increase your application's resiliency.

Google Cloud has a shared security model. You are responsible for defining security policies for your Google Cloud resources, and the cloud resource management hierarchy helps you do that in a manageable way.

Two important tools you can use to administer your use of Google Cloud resources are the Cloud Console and Cloud Shell. We'll use both throughout this specialization.

