



# Google Cloud

---

GCP Fundamentals: Core Infrastructure

**Welcome**

## Facilities



Parking



Facilities



Food



## Course etiquette



---

Please silence  
your phone and  
take calls outside.



---

Recording  
this class  
is prohibited.

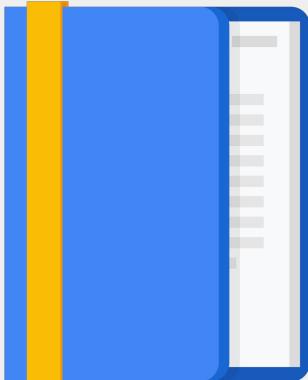


---

Ask questions  
interactively or  
via chat (online).



## Course objectives



- Identify the purpose and value of Google Cloud Platform products and services.
- Interact with Google Cloud Platform services.
- Use Google Cloud Platform application deployment environments.
- Use Google Cloud Platform storage options.
- Do automated deployment, monitoring, and data analysis on Google Cloud Platform.



This 1 day instructor-led class provides an overview of Google Cloud Platform. Through a combination of presentations and hands-on labs, participants learn the value of Google Cloud Platform and how cloud solutions factor into business strategies.

The intended target audience consists of solutions developers, systems operations professionals, and solution architects planning to deploy applications and create application environments on Google Cloud Platform. The course will also help business decision makers evaluating Google Cloud Platform.

The course has no prerequisites, although familiarity with application development, Linux operating systems, systems operations, and data analytics/machine learning is helpful in understanding the technologies covered.

## What's next in the Cloud Infrastructure track?



### Cloud Infrastructure

This track is designed for IT professionals who are responsible for implementing, deploying, migrating, and maintaining applications in the cloud.

1

Google Cloud Platform  
Fundamentals: Core Infrastructure

2

Architecting with Google  
Cloud Platform: Infrastructure

3

Architecting with Google  
Cloud Platform: Design and  
Process



## What's next in the Application Development track?



### Application Development

This track is designed for application programmers and software engineers who develop software programs in the cloud.

1

Google Cloud Platform Fundamentals: Core Infrastructure

2

Developing Applications with Google Cloud Platform



## Audience and prerequisites

### Target audiences

- Developers, DevOps and SysOps professionals, and solution architects planning to deploy applications and environments on GCP
- Decision-makers evaluating GCP

### Prerequisites and pre-work

None (although familiarity with Linux is helpful, as well as with the technologies covered)



## Agenda

Module	Lab
1 Introducing Google Cloud Platform	
2 Getting Started with Google Cloud Platform	Launch an app with Cloud Launcher
3 Virtual Machines in the Cloud	Create virtual machines and connect between them
4 Storage in the Cloud	Use Google Cloud Storage and Google Cloud SQL



### Notes:

½ day each

## Agenda, continued

Module	Lab
5 Containers in the Cloud	Launch a containerized application
6 Applications in the Cloud	Explore and launch a custom application
7 Development, Deployment, and Monitoring in the Cloud	Use Google Cloud Deployment Manager and Stackdriver
8 Big Data and Machine Learning in the Cloud	Use BigQuery
9 Summary and Review	



### Notes:

½ day each

# Introductions

## Your instructor

- Organization
- Background
- Course goals

## You

- Name
- Organization
- Job role
- Course goals



Welcome to your lab environment



For each lab, Qwiklabs offers:

- A set of resources for a fixed amount of time at no additional charge
- A clean environment with permissions



For each lab, Qwiklabs provides a temporary set of credentials and resources for a fixed amount of time at no cost.

It sets up a project in a clean state with an empty project or preconfigured resources.

## Open Qwiklabs

- 1** Open an incognito window  
(or private/anonymous window)
- 2** Go to Qwiklabs URL your instructor provides
- 3** Sign in and launch the course  
(with credentials you used to register for the course)

The image shows the 'Existing Account' sign-in page for Qwiklabs. At the top, there's a blue header bar with the Qwiklabs logo and a menu icon. Below the header, the page title 'Existing Account' is displayed. There are two input fields: 'E-mail' (labeled 'Username') and 'Password'. Below these fields is a 'Remember Me' checkbox. At the bottom of the form is a 'Sign In' button and a 'Forgot your password?' link.



Open an incognito window (or a private/anonymous window) so you know you haven't signed in with any Google or Gmail account.

If not available, make sure you're logged out of all Google and Gmail accounts. Otherwise, you may incur charges.

## View your labs

The screenshot shows a 'Class Details' page with a sidebar on the left containing tabs for 'Class Guides' and 'Labs'. The 'Labs' tab is selected, highlighted in dark grey. The main content area is titled 'Class Details' and lists six lab items. Each item has a small icon followed by a description and a status indicator. A vertical green line is positioned to the left of the first two items.

Lab Description	Status
Qwiklabs and Access to Google Cloud Platform	Lab Completed
Console and Cloud Shell	Active Lab
Projects	Currently Inactive
Infrastructure Preview	Currently Inactive
Bastion Host	Currently Inactive
Google App Engine Development	Currently Inactive



When you click on the **Labs** tab you'll see class labs and their status.

In the example above, the first lab has been completed, as indicated by the filled green circle. The second lab has been made active by the instructor and is available for your use. And the remaining labs are not yet available.

## Select a lab

The screenshot shows the Qwiklabs interface for selecting a lab. On the left, there's a sidebar with 'Class Details' and a 'Labs' section listing several labs: Owiklabs and Access to Google Cloud Platform, Console and Cloud Shell, Projects, Infrastructure, Bastion Host, and Google AI. The 'Console and Cloud Shell' lab is highlighted with a green arrow pointing to its 'Select' button. The main area displays the 'Console and Cloud Shell' lab details:

**Console and Cloud Shell**

In this lab you will become familiar with the GCP web-based interface including Console, the GUI (graphical user interface) environment, and Cloud Shell, the CLI (command line interface) environment.

**Duration:** 30 min.  
**Access Time:** 30 min.  
**Setup Time:** 0 min.  
**Level:** introductory

A yellow box at the bottom states: "You won't be able to pause and restart".

Note the lab's access time on the right and click **Select** to open the lab instructions.

Once you start a lab, you won't be able to pause and restart it, so you'll need a continuous block of time to complete the work.

## Run a lab

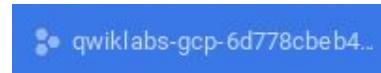
1. Click **Start Lab**

Google Console Details	
GCP Project ID:	qwiklabs-gcp-6d778cbeb42cdbda
Username:	gcpstaging1406-student@qwiklabs.net
Password:	yZ5ZP9WrH

2. Note the following

3. Click **Open Google Console** and sign in

4. Accept terms and note the following



When ready to begin:

1. Click **Start Lab**
2. Note the lab's username, password, and project ID
3. Click **Open Google Console** and sign in to Cloud Console with these credentials
4. Accept the terms and note the project set for you.

## End a lab

When done, click  to free your resources

Some labs may require you to NOT end the lab; the instructor will tell you.



When done, click End. The account will be wiped out and removed. You'll lose all work you have in the project.

Most labs are designed to be standalone, that is, you need to end lab when you finish each lab. Some labs may require you to NOT End lab, and the instructor will inform you.

## End of class: Materials

- 1 Click **Materials** on the top navigation bar.



- 2 Select the class from the **Course Materials** list.

A screenshot of a 'Course Materials' list. The list has a header row with columns: 'Name', 'Taken at', and 'Available until'. Below the header, there is one item listed: 'Data Engineering with Google Cloud Platform' (under 'Name'), '8 Apr 2017' (under 'Taken at'), and '-' (under 'Available until').

Name	Taken at	Available until
Data Engineering with Google Cloud Platform	8 Apr 2017	-

- (1) Available for 2 years following the completion of a course.
- (2) Labs currently available for reference. May not work outside of Qwiklabs environment. No support provided. Qwiklabs lab time not currently available for sale separate from the class.
- (3) class material will appear only at the end of class, provided the student has completed at least one lab in the course.



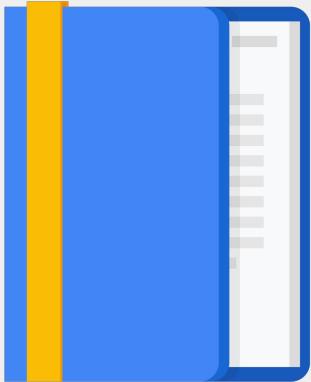
# Google Cloud

---

GCP Fundamentals: Core Infrastructure

**Introducing Google Cloud Platform**

## Agenda



- Introduction to Google Cloud Platform
- Quiz



## What is cloud computing?



On-demand self-service



Broad network access



Resource pooling



Rapid elasticity



Measured service

No human intervention needed to get resources

Access from anywhere

Provider shares resources to customers

Get more resources quickly as needed

Pay only for what you consume



Cloud computing has five fundamental attributes, according to the [definition of cloud computing](#) proposed by the United States National Institute of Standards and Technology.

First, customers get computing resources on-demand and self-service.

Cloud-computing customers use an automated interface and get the processing power, storage, and network they need, with no need for human intervention.

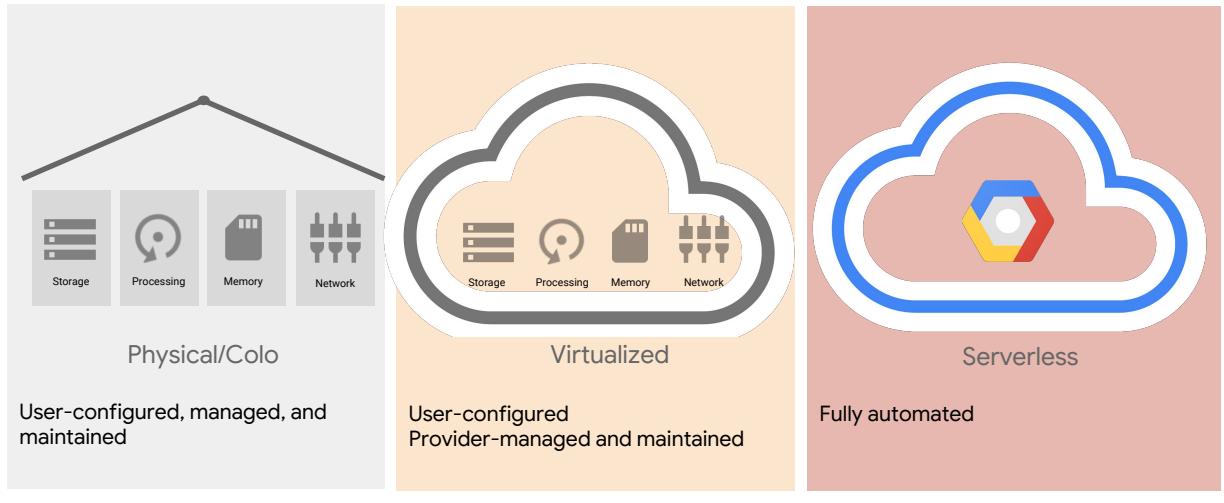
Second, they can access these resources over the network.

Third, the provider of those resources has a big pool of them, and allocates them to customers out of the pool. That allows the provider to get economies of scale by buying in bulk. Customers don't have to know or care about the exact physical location of those resources.

Fourth, the resources are elastic. Customers who need more resources can get more rapidly. When they need less, they can scale back.

And last, the customers pay only for what they use or reserve, as they go. If they stop using resources, they stop paying.

## How did we get here? Where are we going?



The first wave of the trend towards cloud computing was colocation. Colocation gave users the financial efficiency of renting physical space, instead of investing in data center real estate.

Virtualized data centers of today, the second wave, share similarities with the private data centers and colocation facilities of decades past. The components of virtualized data centers match the physical building blocks of hosted computing—servers, CPUs, disks, load balancers, and so on—but now they are virtual devices. Virtualization does provide a number of benefits: your development teams can move faster, and you can turn capital expenses into operating expenses. With virtualization you still maintain the infrastructure; it is still a user-controlled/user-configured environment.

About 10 years ago, Google realized that its business couldn't move fast enough within the confines of the virtualization model. So Google switched to a container-based architecture—a fully automated, elastic third-wave cloud that consists of a combination of automated services and scalable data. Services automatically provision and configure the infrastructure used to run applications.

Today Google Cloud Platform makes this third-wave cloud available to Google customers.

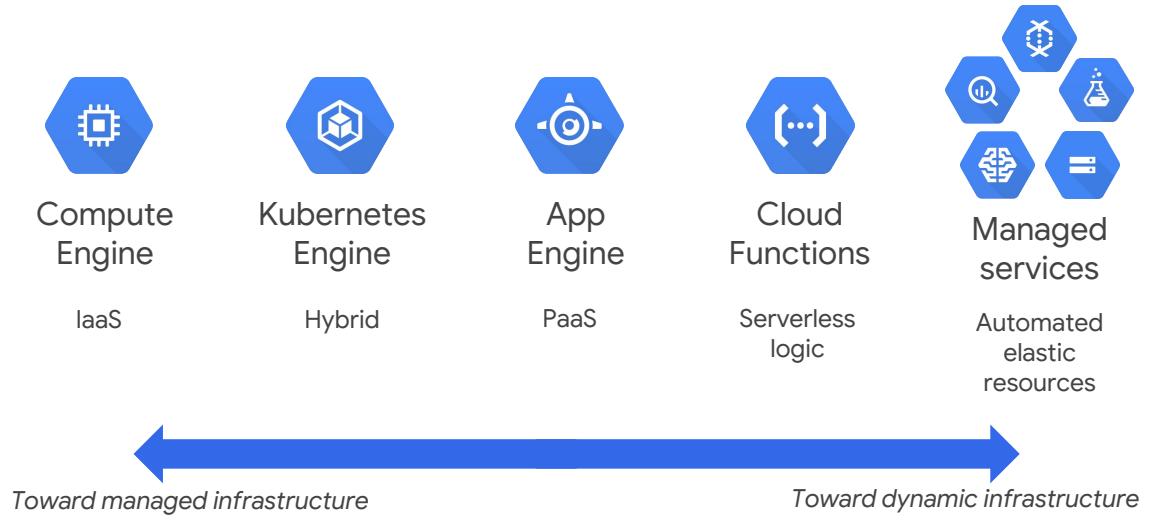
Every company is a  
data company



Google believes that, in the future, every company—regardless of size or industry—will differentiate itself from its competitors through technology. Largely, that technology will be in the form of software. Great software is centered on data. Thus, every company is or will become a data company.

Google Cloud provides a wide variety of services for managing and getting value from data at scale.

## GCP computing architectures meet you where you are

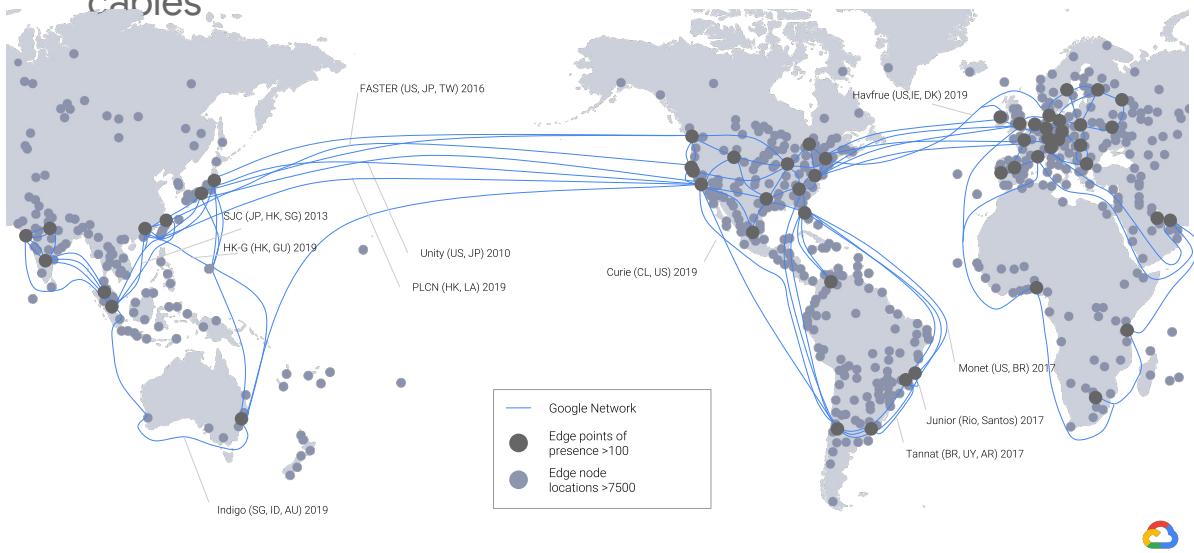


Virtualized data centers brought you infrastructure as a service (IaaS) and platform as a service (PaaS) offerings. IaaS offerings provide you with raw compute, storage, and network, organized in ways familiar to you from physical and virtualized data centers. PaaS offerings, on the other hand, bind your code to libraries that provide access to the infrastructure your application needs, thus allow you to focus on your application logic.

In the IaaS model, you pay for what you allocate. In the PaaS model, you pay for what you use.

As cloud computing has evolved, the momentum has shifted toward managed infrastructure and managed services.

## Google network: 100,000s of km of fiber cable, 8 subsea cables

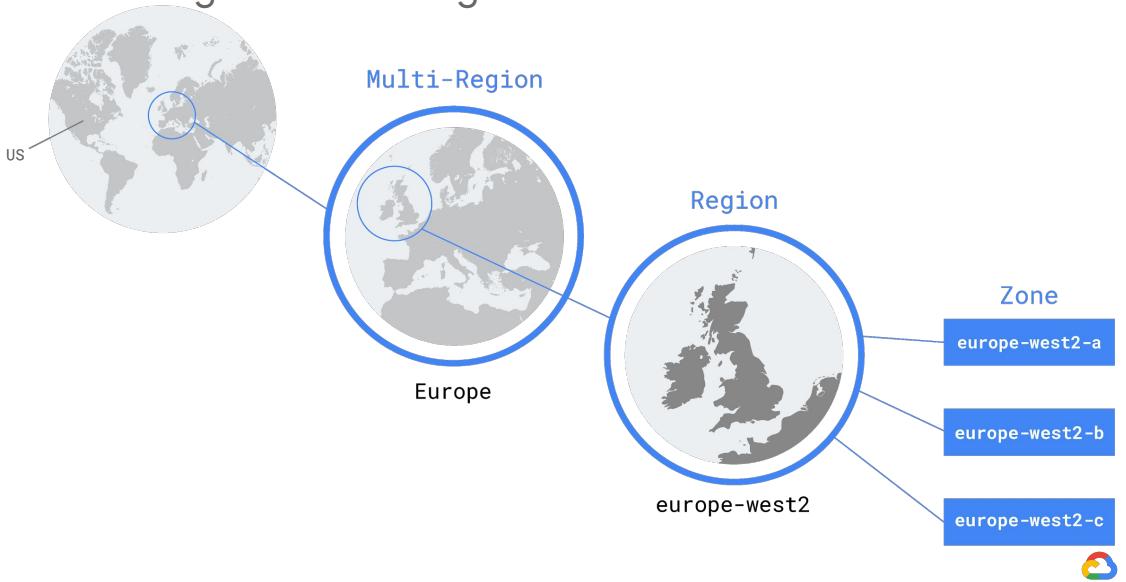


According to some publicly available estimates, Google's network carries as much as **40%** of the world's internet traffic every day. Google's network is the largest network of its kind on Earth. Google has invested billions of dollars over the years to build it.

It is designed to give customers the highest possible throughput and lowest possible latencies for their applications.

The network interconnects 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 latency. Google's edge caching network sites content close to end users to minimize latency.

GCP is organized into regions and zones



## Regions and zones

[Regions](#) are independent geographic areas that consist of [zones](#). Locations within regions tend to have round-trip network latencies of under 5 milliseconds on the 95th percentile.

A zone is a deployment area for Google Cloud Platform resources within a region. Think of a zone as a single failure domain within a region. In order to deploy fault-tolerant applications with high availability, you should deploy your applications across multiple zones in a region to help protect against unexpected failures.

To protect against the loss of an entire region due to natural disaster, you should have a disaster recovery plan and know how to bring up your application in the unlikely event that your primary region is lost.

For more information on the specific resources available within each location option, see Google's [Global Data Center Locations](#).

Google Cloud Platform's services and resources can be [zonal](#), [regional](#), or [managed by Google across multiple regions](#). For more information on what these options mean for your data, see [geographic management of data](#).

## Zonal resources

Zonal resources operate within a single zone. If a zone becomes unavailable, all of the zonal resources in that zone are unavailable until service is restored.

- Google Compute Engine VM instance resides within a specific zone.

## **Regional resources**

Regional resources are deployed with redundancy within a region. This gives them higher availability relative to zonal resources.

## **Multi-regional resources**

A few Google Cloud Platform services are managed by Google to be redundant and distributed within and across regions. These services optimize availability, performance, and resource efficiency. As a result, these services require a trade-off on either latency or the consistency model. These trade-offs are documented on a product-specific basis. The following services have one or more multi-regional deployments in addition to any regional deployments:

- Google App Engine and its features
- Google Cloud Datastore
- Google Cloud Storage
- Google BigQuery

## Google Cloud Platform's regions and zones around the world



At the time of this writing, Google Cloud Platform has 18 regions, with more to come.



## Google is committed to environmental responsibility

100% carbon neutral  
since 2007

One of the world's  
largest corporate  
purchasers of  
renewable energy

First data centers to  
achieve ISO 14001  
certification

This image shows Google's data center in Hamina, Finland. The facility is one of the most advanced and efficient data centers in the Google fleet. Its cooling system, which uses sea water from the Bay of Finland, reduces energy use and is the first of its kind anywhere in the world.

Google is one of the world's largest corporate purchasers of wind and solar energy. Google has been 100% carbon neutral since 2007, and will shortly reach 100% renewable energy sources for its data centers.

The virtual world is built on physical infrastructure, and all those racks of humming servers use vast amounts of energy. Together, all existing data centers use roughly 2% of the world's electricity. So Google works to make data centers run as efficiently as possible. Google's data centers were the first to achieve ISO 14001 certification, a standard that maps out a framework for improving resource efficiency and reducing waste.

## Google offers customer-friendly pricing

Billing in sub-hour increments	Discounts for sustained use	Discounts for committed use	Discounts for preemptible use	Custom VM instance types
For compute, data processing and other services	Automatically applied to virtual machine use over 25% of a month	Pay less for steady, long-term workloads	Pay less for interruptible workloads	Pay only for the resources you need for your application



Google was the first major cloud provider to deliver per-second billing for its Infrastructure-as-a-Service compute offering, Google Compute Engine. Per-second billing is offered for users of Compute Engine, Kubernetes Engine (container infrastructure as a service), Cloud Dataproc (the open-source Big Data system Hadoop as a service), and App Engine flexible environment VMs (a Platform as a Service).

Google Compute Engine offers automatically applied sustained-use discounts, which are automatic discounts that you get for running a virtual-machine instance for a significant portion of the billing month. Specifically, when you run an instance for more than 25% of a month, Compute Engine automatically gives you a discount for every incremental minute you use for that instance.

Custom virtual machine types allow Google Compute Engine virtual machines to be fine-tuned for their applications, so that you can tailor your pricing for your workloads.

Try the [online pricing calculator](#) to help estimate your costs.

## Open APIs and open source mean customers can leave

Open APIs;  
compatibility with  
open-source services



Cloud Bigtable



Cloud Dataproc

Open source for a  
rich ecosystem



TensorFlow    Kubernetes



Forseti Security

Multi-vendor-friendly  
technologies



Google Stackdriver



Kubernetes Engine



Google gives customers the ability to run their applications elsewhere if Google becomes no longer the best provider for their needs.

This includes:

- Using Open APIs. Google services are compatible with open-source products. For example, Google Cloud Bigtable, a horizontally scalable managed database: Bigtable uses the Apache HBase interface, which gives customers the benefit of code portability. Another example: Google Cloud Dataproc offers the open-source big data environment Hadoop as a managed service.
- Google publishes key elements of its technology, using open-source licenses, to create ecosystems that provide customers with options other than Google. For example, TensorFlow, an open-source software library for machine learning developed inside Google, is at the heart of a strong open-source ecosystem.
- Google provides interoperability at multiple layers of the stack. Kubernetes and Google Kubernetes Engine give customers the ability to mix and match microservices running across different clouds. Google Stackdriver lets customers monitor workloads across multiple cloud providers.

## Security is designed into Google's technical infrastructure

Layer	Notable security measures (among others)
Operational security	Intrusion detection systems; techniques to reduce insider risk; employee U2F use; software development practices
Internet communication	Google Front End; designed-in Denial of Service protection
Storage services	Encryption at rest
User identity	Central identity service with support for U2F
Service deployment	Encryption of inter-service communication
Hardware infrastructure	Hardware design and provenance; secure boot stack; premises security



**Hardware design and provenance:** Both the server boards and the networking equipment in Google data centers are custom-designed by Google. Google also designs custom chips, including a hardware security chip that is currently being deployed on both servers and peripherals.

**Secure boot stack:** Google server machines use a variety of technologies to ensure that they are booting the correct software stack, such as cryptographic signatures over the BIOS, bootloader, kernel, and base operating system image.

**Premises security:** Google designs and builds its own data centers, which incorporate multiple layers of physical security protections. Access to these data centers is limited to only a very small fraction of Google employees. Google additionally hosts some servers in third-party data centers, where we ensure that there are Google-controlled physical security measures on top of the security layers provided by the data center operator.

**Encryption of inter-service communication:** Google's infrastructure provides cryptographic privacy and integrity for remote procedure call ("RPC") data on the network. Google's services communicate with each other using RPC calls. The infrastructure automatically encrypts all infrastructure RPC traffic which goes between data centers. Google has started to deploy hardware cryptographic accelerators that will allow it to extend this default encryption to all infrastructure RPC traffic inside Google data centers.

User identity: Google's central identity service, which usually manifests to end users as the Google login page, goes beyond asking for a simple username and password. The service also intelligently challenges users for additional information based on risk factors such as whether they have logged in from the same device or a similar location in the past. Users also have the option of employing second factors when signing in, including devices based on the Universal 2nd Factor (U2F) open standard

Encryption at rest: Most applications at Google access physical storage indirectly via storage services, and encryption (using centrally managed keys) is applied at the layer of these storage services. Google also enables hardware encryption support in hard drives and SSDs.

Google Front End (“GFE”): Google services that want to make themselves available on the Internet register themselves with an infrastructure service called the Google Front End, which ensures that all TLS connections are terminated using correct certificates and following best practices such as supporting perfect forward secrecy. The GFE additionally applies protections against Denial of Service attacks.

Denial of Service (“DoS”) protection: The sheer scale of its infrastructure enables Google to simply absorb many DoS attacks. Google also has multi-tier, multi-layer DoS protections that further reduce the risk of any DoS impact on a service running behind a GFE.

Intrusion detection: Rules and machine intelligence give operational security engineers warnings of possible incidents. Google conducts Red Team exercises to measure and improve the effectiveness of its detection and response mechanisms.

Reducing insider risk: Google aggressively limits and actively monitors the activities of employees who have been granted administrative access to the infrastructure.

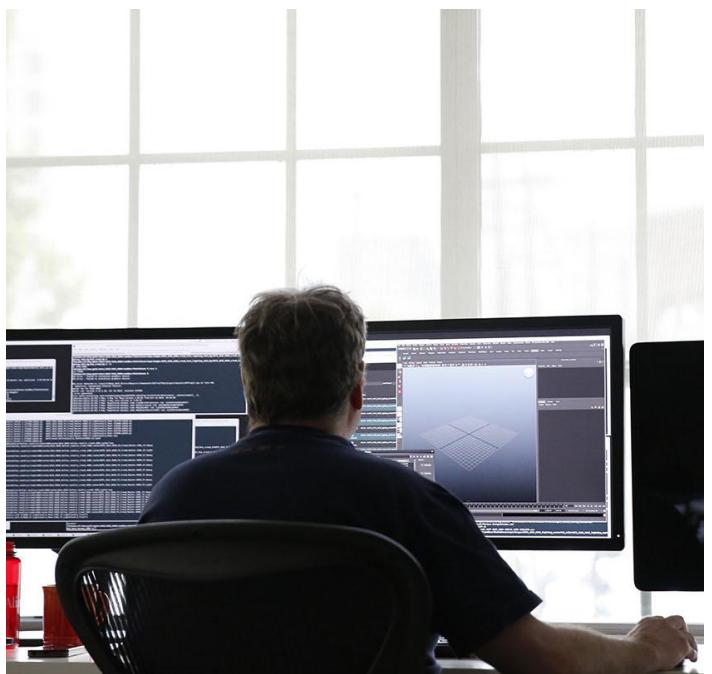
Employee U2F use: To guard against phishing attacks against Google employees, employee accounts require use of U2F-compatible Security Keys.

Software development practices: Google employs central source control and requires two-party review of new code. Google also provides its developers libraries that prevent them from introducing certain classes of security bugs. Google also runs a Vulnerability Rewards Program where we pay anyone who is able to discover and inform us of bugs in our infrastructure or applications.

For more information about Google's technical-infrastructure security, see  
<https://cloud.google.com/security/security-design/>

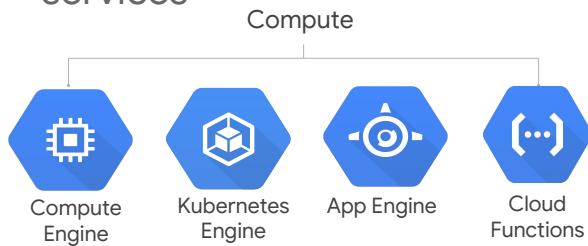
## Why choose Google Cloud Platform?

Google Cloud Platform enables developers to **build, test, and deploy** applications on Google's **highly secure, reliable, and scalable** infrastructure.



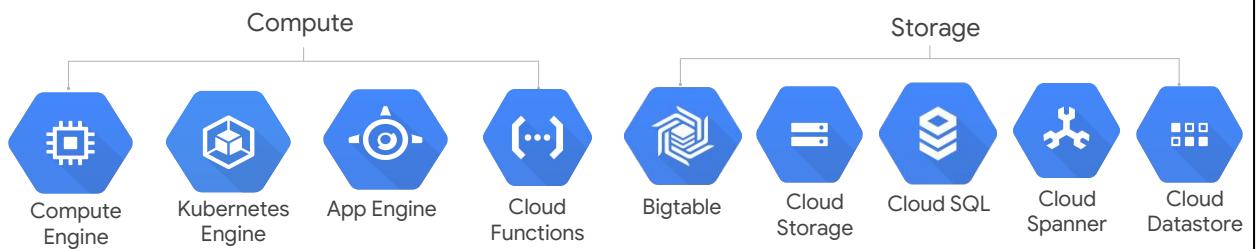
Google Cloud Platform lets you choose from computing, storage, big data/machine learning, and application services for your web, mobile, analytics, and backend solutions.

Review: Google Cloud Platform offers a range of compute services



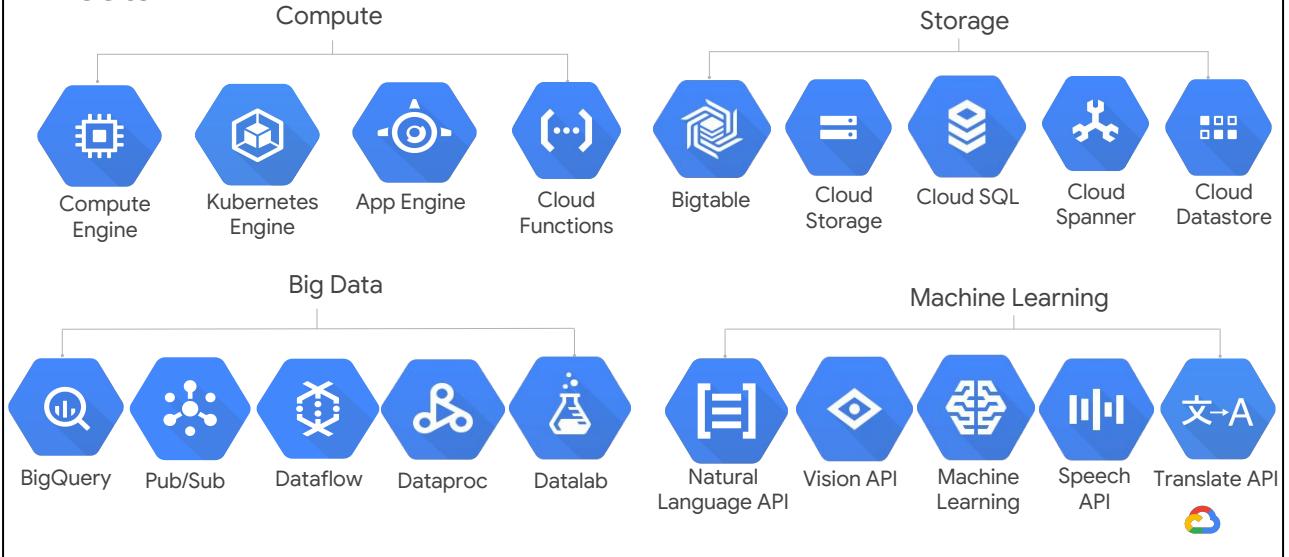
Google Cloud Platform's products and services can be broadly categorized as Compute, Storage, Big Data, Machine Learning, Networking, and Operations/Tools. This course considers each of the compute services and discuss why customers might choose each.

## Google Cloud Platform offers a range of storage services



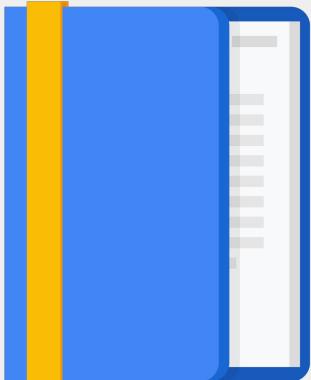
This course will examine each of Google Cloud Platform's storage services: how it works and when customers use it. To learn more about these services, you can participate in the training courses in Google Cloud's [Data Analyst learning track](#).

## Google Cloud Platform offers services for getting value from data



This course also examines the function and purpose of Google Cloud Platform's big data and machine-learning services. More details about these services are also available in the training courses in Google Cloud's [Data Analyst learning track](#).

## Agenda



- Introduction to Google Cloud Platform
- Quiz



# Question #1

---

Name some of Google Cloud Platform's pricing innovations.



# Question #1

---

Name some of Google Cloud Platform's pricing innovations.

1. Sub-hour billing
2. Sustained-use discounts
3. Compute Engine custom machine types



## Question #2

---

Name some benefits of using Google Cloud Platform other than its pricing.



# Question #2

---

Name some benefits of using Google Cloud Platform other than its pricing.

1. Commitment to environmental responsibility
2. Commitment to open-source technologies
3. Robust infrastructure



# More resources

---

Why Google Cloud Platform? <https://cloud.google.com/why-google/>

Pricing philosophy <https://cloud.google.com/pricing/philosophy/>

Data centers <https://www.google.com/about/datacenters/>

Google Cloud Platform product overview <http://cloud.google.com/products/>

Google Cloud Platform solutions <http://cloud.google.com/solutions/>





# Google Cloud

---

GCP Fundamentals: Core Infrastructure

**Getting Started with Google Cloud Platform**

# Cloud security requires collaboration

- 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.

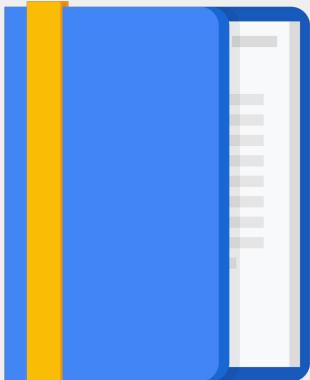
Customer-managed    Google-managed



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. When you move an application to Google Cloud Platform, Google handles many of the lower layers of security. Because of its scale, Google can deliver a higher level of security at these layers than most of its customers could afford to do on their own.

The upper layers of the security stack remain the customer's responsibility. Google provides tools, such as IAM, to help customers implement the policies they choose at these layers.

## Agenda

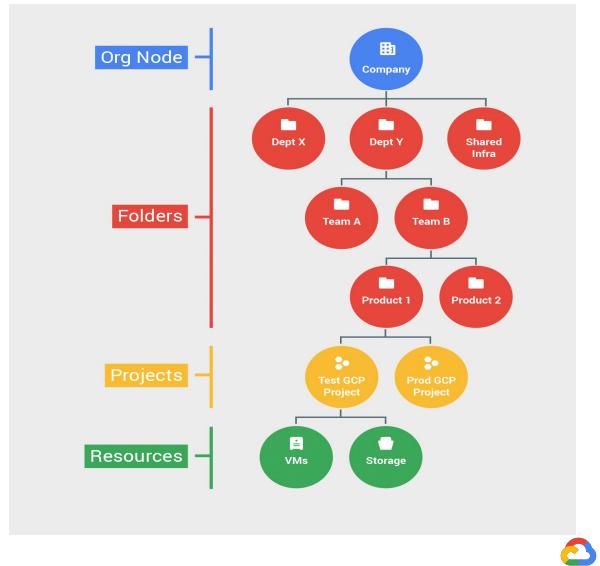


- **Google Cloud Platform resource hierarchy**
- Identity and Access Management (IAM)
- Cloud Identity
- Interacting with Google Cloud Platform
- Cloud Marketplace
- Quiz and Lab



## Resource hierarchy levels define trust boundaries

- Group your resources according to your organization structure.
- Levels of the hierarchy provide trust boundaries and resource isolation.



You may find it easiest to understand the GCP resource hierarchy from the bottom up. All the resources you use--whether they're virtual machines, Cloud Storage buckets, tables in BigQuery, or anything else in GCP--are organized into projects. Optionally, these projects may be organized into folders; folders can contain other folders. All the folders and projects used by your organization can be brought together under an organization node. Projects, folders, and organization nodes are all places where policies can be defined. Some GCP resources let you put policies on individual resources too, like Cloud Storage buckets. (This course discusses Cloud Storage buckets later in the course.)

Policies are inherited downwards in the hierarchy.

## All GCP services you use are associated with a project

- Track resource and quota usage.
- Enable billing.
- Manage permissions and credentials.
- Enable services and APIs.



All Google Cloud Platform resources belong to a Google Cloud Platform Console project. Projects are the basis for enabling and using GCP 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. Projects can have different owners and users. They're billed separately, and they're managed separately.

The Cloud Resource Manager provides methods that you can use to programmatically manage your projects in Google Cloud Platform. With this API, you can do the following:

- Get a list of all projects associated with an account.
- Create new projects.
- Update existing projects.
- Delete projects.
- Undelete, or recover, projects that you don't want to delete.

You can access Cloud Resource Manager in either of the following ways:

- Through the [RPC API](#)
- Through the [REST API](#)

## Projects have three identifying attributes

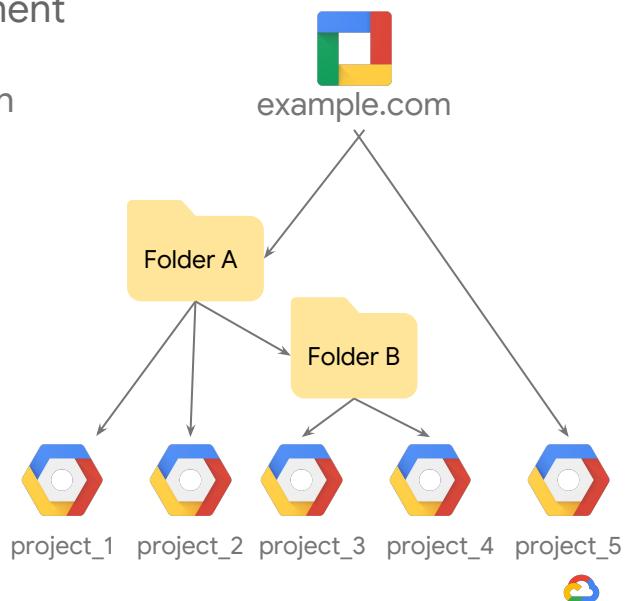
<b>Project ID</b>	Globally unique	Chosen by you	Immutable
<b>Project name</b>	Need not be unique	Chosen by you	Mutable
<b>Project number</b>	Globally unique	Assigned by GCP	Immutable



Each GCP project has a name and project ID you assign. The project ID is a permanent, unchangeable identifier, and it has to be unique across GCP. You'll use project IDs in several contexts to tell GCP which project you want to work with. On the other hand, project names are for your convenience, and you can change them. GCP also assigns each of your projects a unique project number, and you'll see it displayed to you 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'll 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.



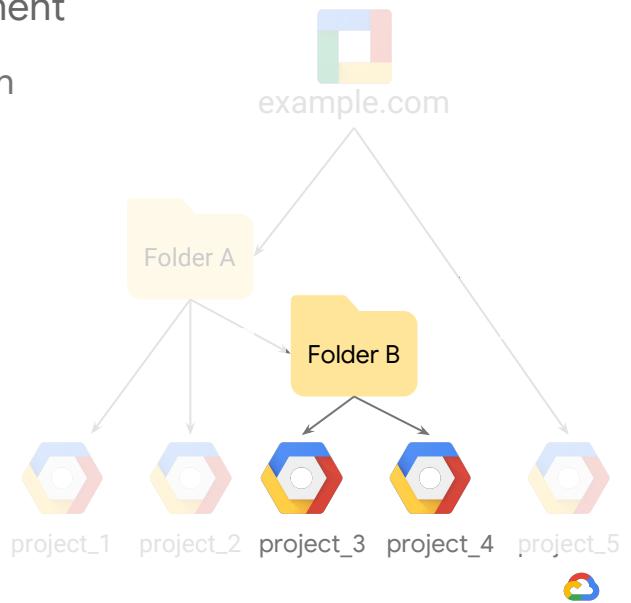
The Cloud IAM Folders feature lets you assign policies to resources at a level of granularity you choose. The resources in a folder inherit IAM policies assigned to the folder.

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

Folders let teams have the ability to delegate administrative rights, so that they can work independently.

## Folders offer flexible management

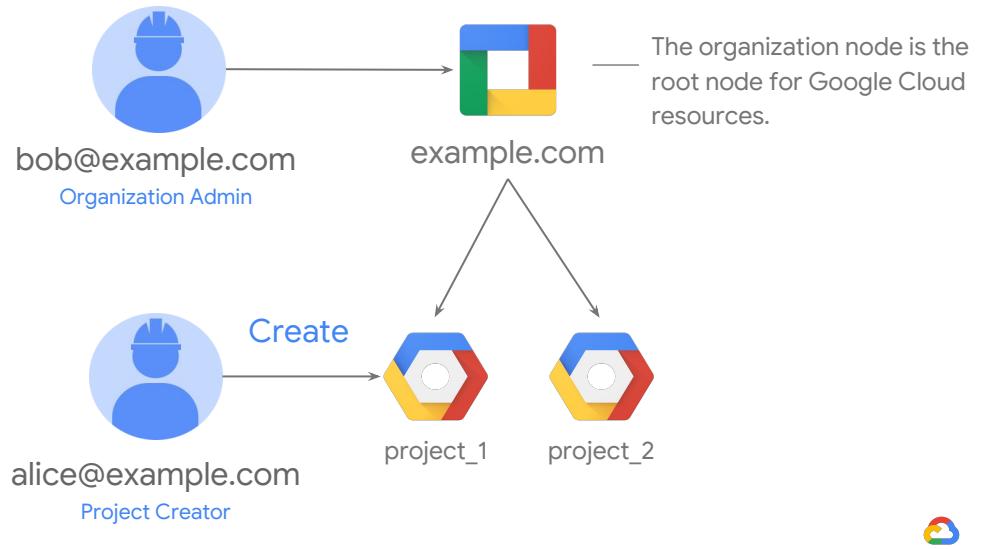
- Folders group projects under an organization.
- Folders can contain projects, other folders, or both.
- Use folders to assign policies.



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 put IAM policies onto Folder B instead. Doing it the other way-- putting duplicate copies of those policies on project 3 and project 4-- would be tedious and error-prone.

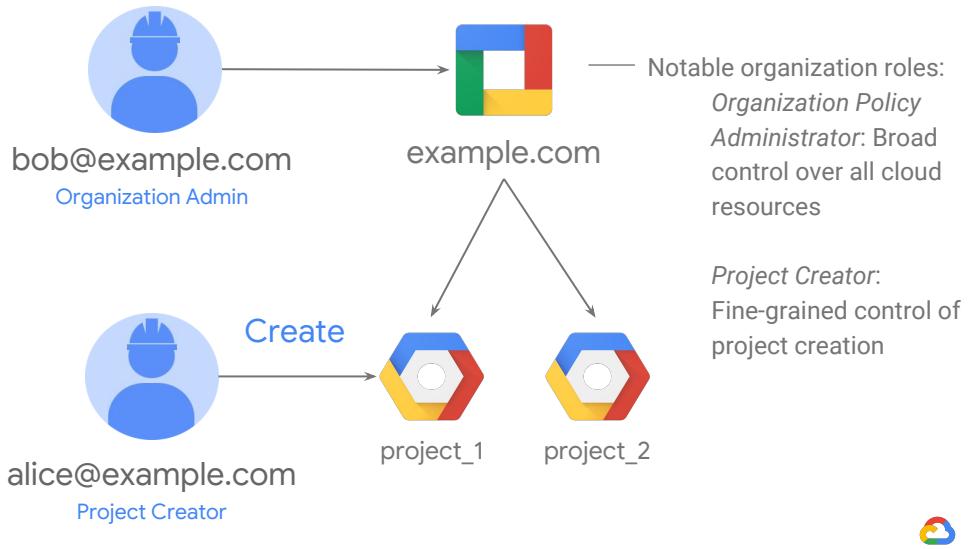
One word of caution: to use folders, you need an organization node at the top of the hierarchy.

## The organization node organizes projects



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

## The organization node organizes projects



There are some special roles associated with it. 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.

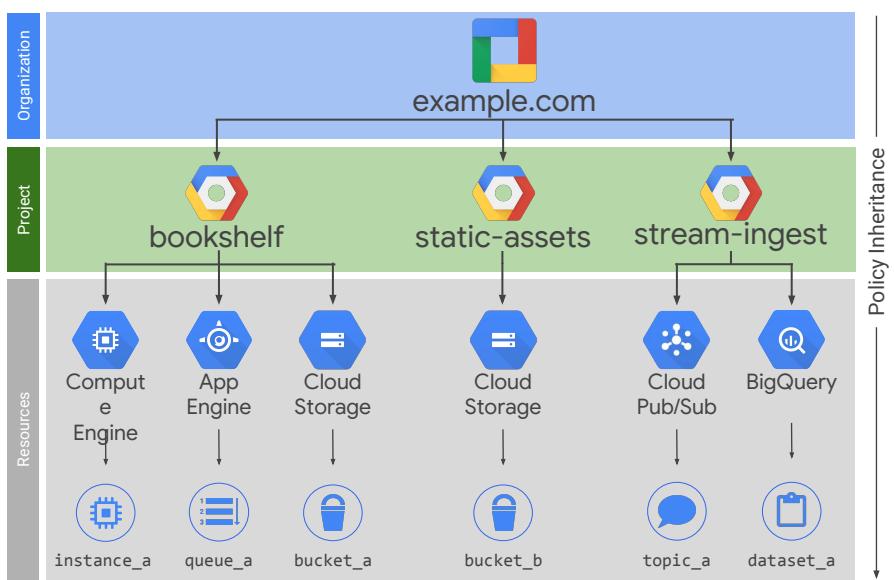
So how do you get an organization node? In part the answer depends on whether your company is also a G Suite customer. If you have a G Suite domain, GCP projects will automatically belong to your organization node. Otherwise, you can use Google Cloud Identity to create one.

Here's a tip: 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. But it'd be a great first step with a new organization node to decide who on your team really should be able to do those things.

Once you have an organization node, you can create folders underneath it and put projects in.

## 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.

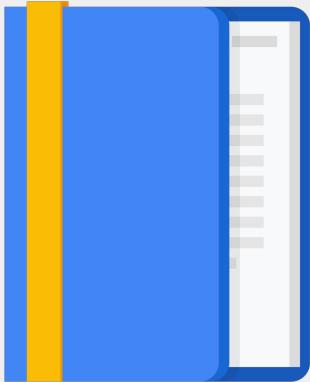


Here's an example of how you might organize your resources. There are three projects, each of which uses resources from several GCP services. In this example, we haven't used any folders, although we always could move projects into folders if that became helpful.

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.

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.

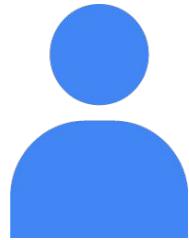
## Agenda



- Google Cloud Platform resource hierarchy
- **Identity and Access Management (IAM)**
- Cloud Identity
- Interacting with Google Cloud Platform
- Cloud Marketplace
- Quiz and Lab



Google Cloud Identity and Access Management defines...



Who

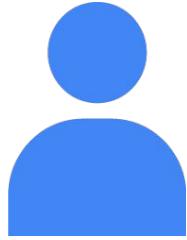
can do what

on which resource



IAM lets administrators authorize who can take action on specific resources. An IAM policy has a “who” part, a “can do what” part, and an “on which resource” part.

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



Who



Google account or Cloud Identity user  
test@gmail.com      test@example.com



Service account  
test@project\_id.iam.gserviceaccount.com



Google group  
test@googlegroups.com

G Suite

Cloud Identity or G Suite domain  
example.com



The “who” part of an IAM policy can be a Google account, a Google group, a service account, or an entire G Suite or Cloud Identity domain.

Can do what: IAM roles are collections of related permissions



can do what



InstanceAdmin  
Role

Service	Resource	Verb
compute	instances	<i>list</i>
compute	instances	<i>delete</i>
compute	instances	<i>start</i>
...		

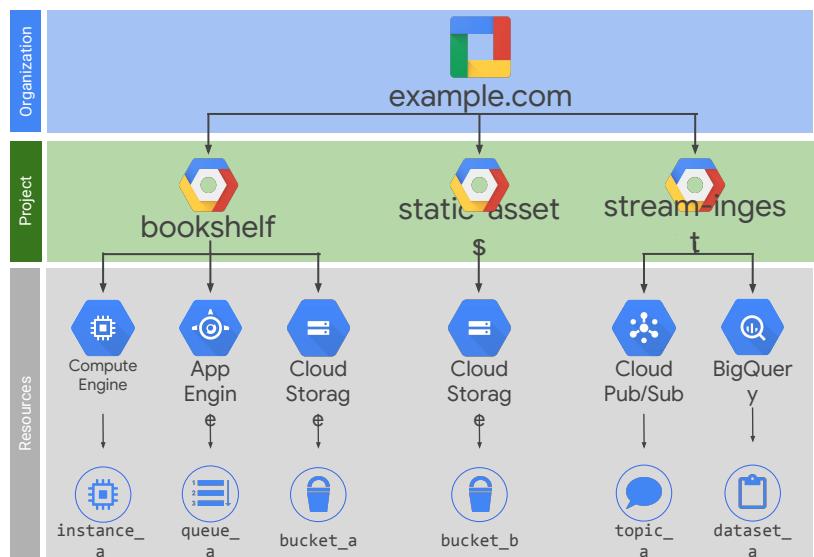


The “can do what” part is defined by an IAM role. An IAM role is a collection of permissions. Most of the time, to do any meaningful operations, you need more than 1 permission. For example, to manage instances in a project, you need to create, delete, start, stop and change an instance. So the permissions are grouped together into a role to make them easier to manage.

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



on which resource



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.

There are three types of IAM roles

Primitive



Predefined



Custom



The “can do what” part of an IAM policy is defined by a role. An IAM role is a collection of permissions, because, most of the time you need more than 1 permission to do meaningful work. 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.

There are three kinds of roles in Cloud IAM. Let’s talk about each in turn.

IAM **primitive** roles apply across all GCP services in a project



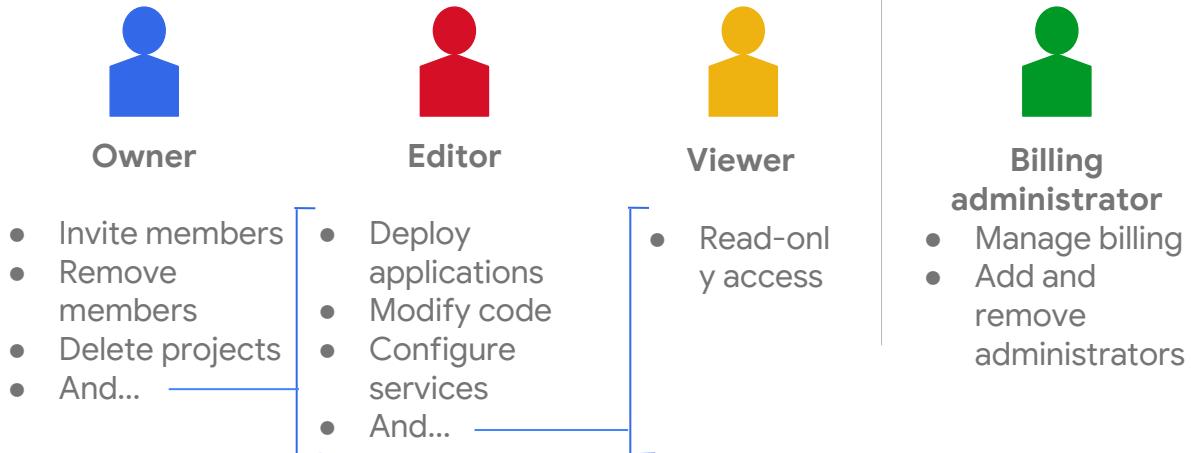
can do what

on all resources



Primitive roles are broad. You apply them to a GCP project, and they affect all resources in that project.

## IAM primitive 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, primitive roles are probably too coarse a tool. Fortunately, GCP IAM provides finer-grained types of roles.

IAM **predefined** roles apply to a particular GCP service in a project



can do what

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



GCP 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.

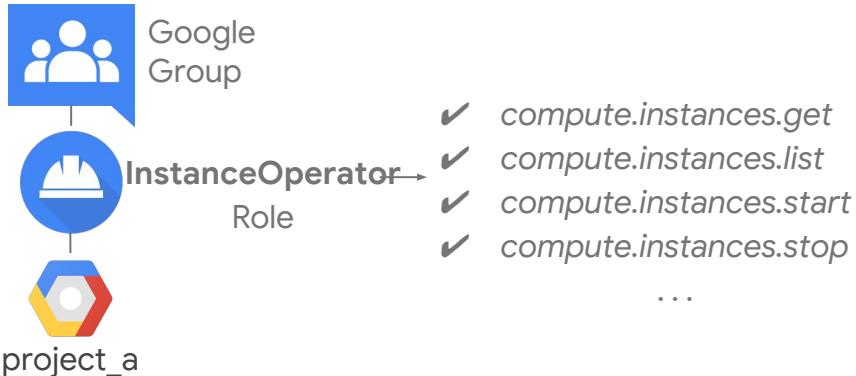
IAM predefined roles offer more fine-grained permissions on particular services



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.

IAM custom roles let you define a precise set of permissions



What if you need something even finer-grained? That's what custom roles permit. A lot of companies use a "least-privilege" model, in which each person in your organization 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.

## Service Accounts control server-to-server interactions

- Provide an identity for carrying out **server-to-server** interactions in a project
- Used to **authenticate** from one service to another
- Used to **control privileges** used by resources
  - So that applications can perform actions on behalf of authenticated end users
- Identified with an **email** address:

**PROJECT\_NUMBER**-compute@developer.gserviceaccount.com

**PROJECT\_ID**@appspot.gserviceaccount.com



What if you want to give permissions to a Compute Engine virtual machine rather than to a person? That's what service accounts are for. For instance, maybe you have an application running in a virtual machine that needs to store data in Google Cloud Storage. But you don't want to let just anyone on the Internet have access to that data; only that virtual machine. So you'd create a service account to authenticate your VM to Cloud Storage. Service accounts are named with an email address, but instead of passwords they use cryptographic keys to access resources.

## Service Accounts and IAM

- Service accounts authenticate using keys.
  - Google manages keys for Compute Engine and App Engine.
- You can assign a predefined or custom IAM role to the service account.

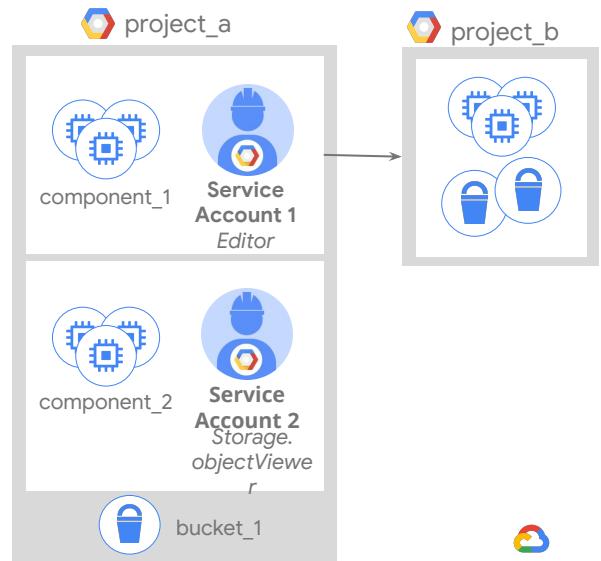


In this simple example, a service account has been granted Compute Engine's Instance Admin role. This would allow an application running in a VM with that service account to create, modify, and delete other VMs.

Incidentally, service accounts need to be managed too! For example, maybe Alice needs to manage what can act as a given service account, while Bob just needs to be able to view what can. 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. For instance, 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 GCP resource.

## Example: Service Accounts and IAM

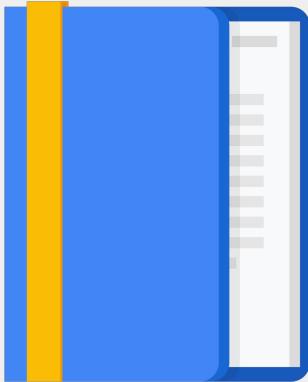
- VMs running component\_1 are granted **Editor** access to project\_b using *Service Account 1*.
- VMs running component\_2 are granted **objectViewer** access to bucket\_1 using *Service Account 2*.
- Service account permissions can be changed without recreating VMs.



You can grant different groups of VMs in your project different identities. This makes it easier to manage different permissions for each group. You also can change the permissions of the service accounts without having to recreate the VMs.

Here's a more complex example. Say you have an application that's implemented across a group of Compute Engine virtual machines. One component of your application needs to have an editor role on another project, but another component doesn't. So you would create two different service accounts, one for each subgroup of virtual machines. Only the first service account has privilege on the other project. That reduces the potential impact of a miscoded application or a compromised virtual machine.

## Agenda



- Google Cloud Platform resource hierarchy
- Identity and Access Management (IAM)
- **Cloud Identity**
- Interacting with Google Cloud Platform
- Cloud Marketplace
- Quiz and Lab



What can you use to manage your GCP administrative users?



Gmail accounts and  
Google Groups



**G Suite**

Users and groups in  
your G Suite domain



Users and groups in  
your Cloud Identity  
domain



Many new GCP customers get started by logging into the GCP console with a Gmail account. To collaborate with their teammates, they use Google Groups to gather together people who are in the same role. 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.

GCP customers who are also G Suite customers can define GCP policies in terms of G Suite users and groups. This way, when someone leaves your organization, an administrator can immediately disable their account and remove them from groups using the Google Admin Console.

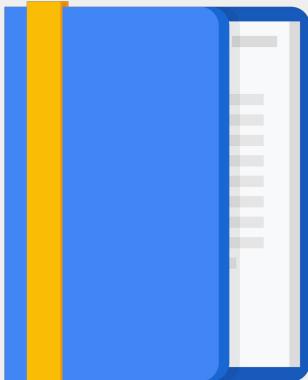
GCP customers who are not G Suite customers can get these same capabilities through Cloud Identity. Cloud Identity lets you manage users and groups using the Google Admin Console, but you do not pay for or receive G Suite's collaboration products such as Gmail, Docs, Drive, and Calendar. Cloud Identity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

## What if you already have a different corporate directory?



Using Google Cloud Directory Sync, your administrators can log in and manage GCP resources using the same usernames and passwords they already use. This tool synchronizes users and groups from your existing Active Directory or LDAP system with the users and groups in your Cloud Identity domain. The synchronization is one-way only; no information in your Active Directory or LDAP map is modified. Google Cloud Directory Sync is designed to run scheduled synchronizations without supervision, after its synchronization rules are set up.

## Agenda



- Google Cloud Platform resource hierarchy
- Identity and Access Management (IAM)
- Cloud Identity
- **Interacting with Google Cloud Platform**
- Cloud Marketplace
- Quiz and Lab



There are four ways to interact with GCP

Cloud Platform Console	Cloud Shell and Cloud SDK	Cloud Console Mobile App	REST-based API
Web user interface	Command-line interface	For iOS and Android	For custom applications



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

## Google Cloud Platform Console

- Centralized 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



Google Cloud Source Repositories provides Git version control to support collaborative development of any application or service, including those that run on Google App Engine and Google Compute Engine. If you are using the Stackdriver Debugger, you can use Cloud Source Repositories and related tools to view debugging information alongside your code during application runtime. Cloud Source Repositories also provides a source editor that you can use to browse, view, edit, and commit changes to repository files from within the Cloud Platform Console.

Google Cloud Shell provides you with command-line access to your cloud resources directly from your browser. You can easily manage your projects and resources without having to install the Google Cloud SDK or other tools on your system. With Cloud Shell, the Cloud SDK gcloud command and other utilities you need are always available, up to date, and fully authenticated when you need them.

## Google Cloud SDK

- SDK includes CLI tools for Cloud Platform products and services
  - gcloud, gsutil (Cloud Storage), bq (BigQuery)
- Available as Docker image
- Available via Cloud Shell
  - Containerized version of 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 Platform. These include the gcloud tool, which provides the main command-line interface for Google Cloud Platform products and services, as well as gsutil and bq. All of the tools are located under the bin directory.

For more information on the SDK command-line tools, see:  
<https://cloud.google.com/sdk/cloudplatform>

Note: Currently, the App Engine SDKs are separate downloads. For more information, see: <https://cloud.google.com/appengine/downloads>

Cloud Shell provides the following:

- A temporary Compute Engine virtual machine instance running a Debian-based Linux operating system
- Command-line access to the instance from a web browser using terminal windows in the Cloud Platform Console
- 5 GB of persistent disk storage per user, mounted as your \$HOME directory in Cloud Shell sessions across projects and instances
- Google Cloud SDK and other tools pre-installed on the Compute Engine instance

- Language support, including SDKs, libraries, runtime environments and compilers for Java, Go, Python, Node.js, PHP and Ruby
- Web preview functionality, which allows you to preview web applications running on the Cloud Shell instance through a secure proxy
- Built-in authorization for access to projects and resources

You can use Cloud Shell to:

- Create and manage Google Compute Engine instances.
- Create and access Google Cloud SQL databases.
- Manage Google Cloud Storage data.
- Interact with hosted or remote Git repositories, including Google Cloud Source Repositories.
- Build and deploy Google App Engine applications.

You can also use Cloud Shell to perform other management tasks related to your projects and resources, using either the gcloud command or other available tools.

## 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 Platform 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 GCP offer Application Programming Interfaces, so that code you write can control them. These APIs are what's called "RESTful"; in other words, they follow the "Representational state transfer" paradigm. In a broad sense, that means that your code can use Google services in much the same way that web browsers talk to web servers. The APIs name resources in GCP with URLs. Your code can pass information to the APIs using JSON, which is a very popular way of passing textual information over the Web. And there's an open system, OAuth2, for user login and access control.

## Use APIs Explorer to help you write your code

- The [APIs Explorer](#) is an interactive tool that lets you easily try Google APIs using a browser.
- With the APIs Explorer, you can:
  - Browse quickly through available APIs and versions.
  - See methods available for each API and what parameters they support along with inline documentation.
  - Execute requests for any method and see responses in real time.
  - Easily make authenticated and authorized API calls.



The GCP Console includes a tool called the APIs Explorer that helps you learn about the APIs interactively. It lets you see what APIs are available, and in what versions. These APIs expect parameters, and documentation on them is built-in. You can try the APIs interactively, even with 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 client libraries to take a lot of the drudgery out of the task of calling GCP from your code.

Use client libraries to control GCP resources from within your code

- [Cloud Client Libraries](#)
  - Community-owned, hand-crafted client libraries
- [Google API Client Libraries](#)
  - Open source, generated
  - Support various languages
    - Java, Python, JavaScript, PHP, .NET, Go, Node.js, Ruby, Objective-C, Dart



There are two kinds of libraries. The Cloud Client libraries are Google Cloud's latest and recommended libraries for its APIs. They adopt the native styles and idioms of each language. On the other hand, sometimes a Cloud Client library doesn't support the newest services and features. In that case, you can use the Google API Client library for your desired languages. These libraries are designed for generality and completeness.

## 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 mobile app allows you to start, stop, and SSH into Compute Engine instances, and to see logs from each instance. You can stop and start Cloud SQL instances. You can also administer applications deployed on Google App Engine, by viewing errors, rolling back deployments, and changing traffic splitting.

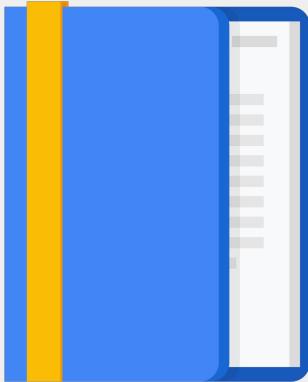
You can also get up-to-date billing information for your projects and get billing alerts for projects that are going over budget.

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

The mobile app also offers alerts and incident management.

Download the Google Cloud Console Mobile App from Google Play or from the iOS App Store.

## Agenda



- Google Cloud Platform resource hierarchy
- Identity and Access Management (IAM)
- Cloud Identity
- Interacting with Google Cloud Platform
- **Cloud Marketplace**
- Quiz and Lab



## Cloud Marketplace gives quick access to solutions

- A solution marketplace containing pre-packaged, ready-to-deploy solutions
  - Some offered by Google
  - Others by third-party vendors
- You pay for the underlying GCP resource usage.
  - Some solutions also assess third-party license fees.



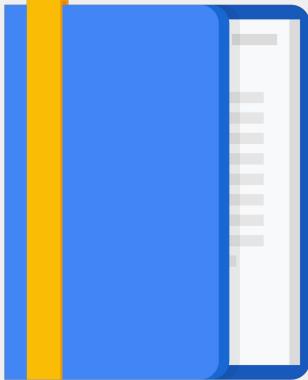
Google Cloud Marketplace lets you quickly deploy functional software packages that run on Google Cloud Platform. You can easily start up a familiar software package without having to manually configure the software, virtual machine instances, storage, or network settings.

Many software packages in Cloud Marketplace are free. The only costs to deploy these solutions are the normal usage fees for Google Cloud Platform resources. Estimated costs are based on the minimum recommended instance and storage configuration. The estimate does not include networking costs. You can modify the instance and storage configuration when you deploy the configuration.

Google Cloud Platform updates the images for these software packages to fix critical issues and vulnerabilities, but doesn't update software that you have already deployed.

Some Cloud Marketplace images assess usage fees, particularly those published by third parties and containing commercially licensed software. If an image does incur a usage fee, the fee appears on your monthly Google Cloud Platform invoice as a separate line item. See the [Cloud Marketplace documentation](#) for details.

## Agenda



- Google Cloud Platform resource hierarchy
- Identity and Access Management (IAM)
- Cloud Identity
- Interacting with Google Cloud Platform
- Cloud Marketplace
- **Quiz and Lab**



# Question #1

---

True or False: If a Google Cloud IAM policy gives you Owner permissions at the project level, your access to a resource in the project may be restricted by a more restrictive policy on that resource.



# Question #1

---

True or False: If a Google Cloud IAM policy gives you Owner permissions at the project level, your access to a resource in the project may be restricted by a more restrictive policy on that resource.

**False:** Policies are a union of the parent and the resource. If a parent policy is less restrictive, it overrides a more restrictive resource policy.



## Question #2

---

True or False: All Google Cloud Platform resources are associated with a project.



## Question #2

---

True or False: All Google Cloud Platform resources are associated with a project.

**True:** All Google Cloud Platform resources are associated with a project.



# Question #3

---

Service accounts are used to provide which of the following?

- Authentication between Google Cloud Platform services
- Key generation and rotation when used with App Engine and Compute Engine
- A way to restrict the actions a resource (such as a VM) can perform
- A way to allow users to act with service account permissions
- All of the above



# Question #3

---

Service accounts are used to provide which of the following?

- Authentication between Google Cloud Platform services
- Key generation and rotation when used with App Engine and Compute Engine
- A way to restrict the actions a resource (such as a VM) can perform
- A way to allow users to act with service account permissions
- All of the above



# Lab

---

Deploy a virtual development environment using Cloud Marketplace.

## Objectives

- Deploy a Bitnami LAMP stack to Compute Engine using Cloud Marketplace.
- Verify the deployment.



In this lab, you create two nginx web servers and control external HTTP access to the web servers using tagged firewall rules. Then, you explore IAM roles and service accounts.

# More resources

---

Google Cloud Platform security <https://cloud.google.com/security/>

Configuring permissions <https://cloud.google.com/docs/permissions-overview>

Identity and Access Management (IAM) <https://cloud.google.com/iam/>

Cloud SDK installation and quick start [https://cloud.google.com/sdk/#Quick\\_Start](https://cloud.google.com/sdk/#Quick_Start)

gcloud tool guide <https://cloud.google.com/sdk/gcloud/>





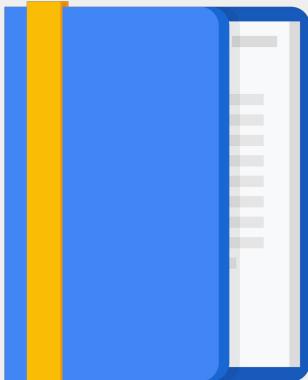
# Google Cloud

---

GCP Fundamentals: Core Infrastructure

## **Virtual Machines in the Cloud**

## Agenda



- **Virtual Private Cloud (VPC) Network**
- Compute Engine
- Important VPC capabilities
- Quiz and lab



## Virtual Private Cloud Networking

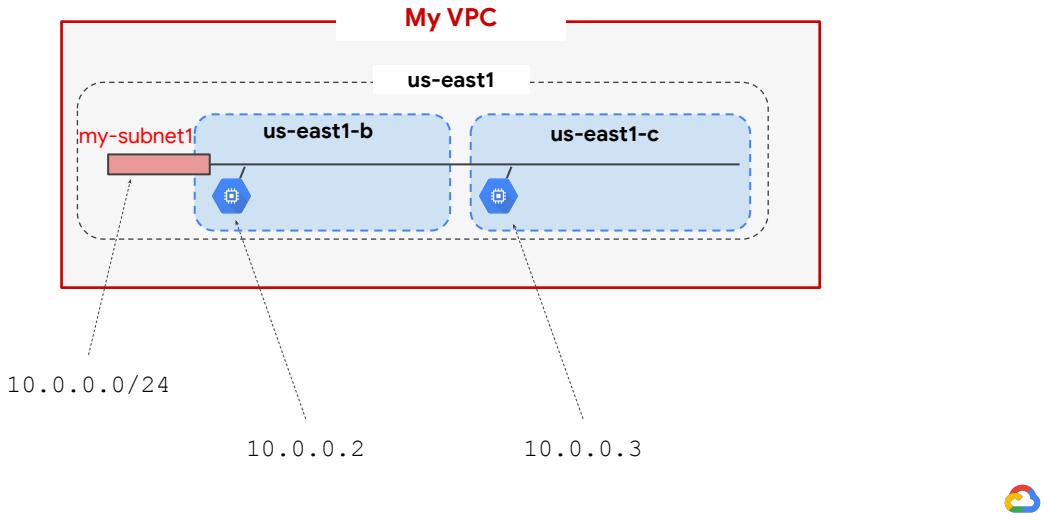
- Each VPC network is contained in a GCP project.
- You can provision Cloud Platform resources, connect them to each other, and isolate them from one another.



Your VPC networks connect your Google Cloud Platform resources to each other and to the internet. You can segment your networks, use firewall rules to restrict access to instances, and create static routes to forward traffic to specific destinations.

Many users get started with GCP is to define their own Virtual Private Cloud inside their first GCP project. Or they can simply choose the default VPC and get started with that.

Google Cloud VPC networks are **global**; subnets are **regional**

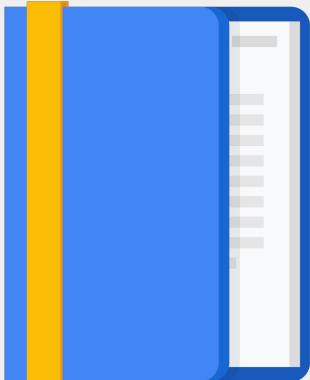


Google Virtual Private Cloud networks that you define have global scope. They can have subnets in any GCP region worldwide. Subnets can span the zones that make up a region. This architecture makes it easy for you to define your own network layout with global scope. You can also have resources in different zones on the same subnet.

You can dynamically increase the size of a subnet in a custom network by expanding the range of IP addresses allocated to it. Doing that doesn't affect already configured VMs.

In this example, your VPC has one network. So far, it has one subnet defined, in GCP's us-east1 region. Notice that it has two Compute Engine VMs attached to it. They're neighbors on the same subnet even though they are in different zones! You can use this capability to build solutions that are resilient but still have simple network layouts.

## Agenda



- Virtual Private Cloud (VPC) Network
- **Compute Engine**
- Important VPC capabilities
- Quiz and lab



## Compute Engine offers managed virtual machines

- High CPU, high memory, standard and shared-core machine types
- Persistent disks
  - Standard, SSD, **local SSD**
  - Snapshots
- **Resize disks with no downtime**
- Instance metadata and startup scripts



Virtual machines have the power and generality of a full-fledged operating system in each. You configure a virtual machine much like you build out a physical server: by specifying its amounts of CPU power and memory, its amounts and types of storage, and its operating system. Compute Engine lets you create and run virtual machines on Google infrastructure. There are no upfront investments, and you can run thousands of virtual CPUs on a system that is designed to be fast and to offer consistent performance.

You can flexibly reconfigure Compute Engine virtual machines. And a VM running on Google's cloud has unmatched worldwide network connectivity.

You can create a virtual machine instance by using the Google Cloud Platform Console or the gcloud command-line tool. A Compute Engine instance can run Linux and Windows Server images provided by Google or any customized versions of these images. You can also build and run images of other operating systems.

## Compute Engine offers customer friendly pricing

- Per-second billing, sustained use discounts, committed use discounts
- Preemptible instances
- High throughput to storage at no extra cost
- Custom machine types: Only pay for the hardware you need



Compute Engine bills by the second for use of virtual machines, with a one-minute minimum. And discounts apply automatically to virtual machines that run for substantial fractions of a month. For each VM that you run for more than 25% of a month, Compute Engine automatically gives you a discount for every incremental minute. You can get up to a 30% net discount for VMs that run the entire month.

Compute Engine offers the ability to purchase committed use contracts in return for deeply discounted prices for VM usage. These discounts are known as committed use discounts. If your workload is stable and predictable, you can purchase a specific amount of vCPUs and memory for up to a 57% discount off of normal prices in return for committing to a usage term of 1 year or 3 years.

Suppose you have a workload that no human being is sitting around waiting to finish. Say, a batch job analyzing a large dataset. You can save money by choosing Preemptible VMs to run the job. A Preemptible VM is different from an ordinary Compute Engine VM in only one respect: you've given Compute Engine permission to terminate it if its resources are needed elsewhere. You can save a lot of money with preemptible VMs, although be sure to make your job able to be stopped and restarted.

You don't have to select a particular option or machine type to get high throughput

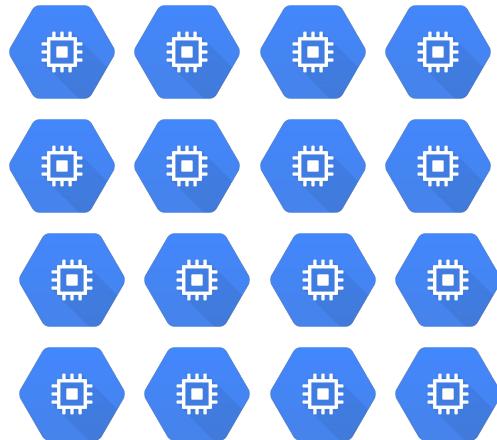
between your processing and your persistent disks. That's the default.

You can choose the machine properties of your instances, such as the number of virtual CPUs and the amount of memory, by using a set of predefined machine types or by creating your own custom machine types.

## Scale up or scale out with Compute Engine



*Use big VMs for memory- and compute-intensive applications*



*Use Autoscaling for resilient, scalable applications*



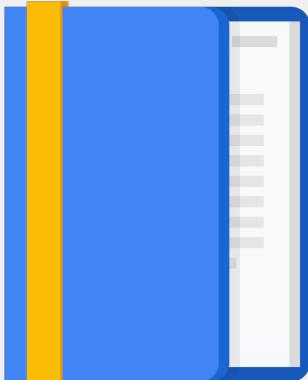
You can make very large VMs in Compute Engine. At the time this deck was produced, the maximum number of virtual CPUs in a VM was zone-dependent and at 96, and the maximum memory size was at 624 GB (6.5 GB per CPU).

You can also use a mega-memory machine type that scales to 1.4 TB memory.

Check the GCP website to see where these maximums are today.

These huge VMs are great for workloads like in-memory databases and CPU-intensive analytics. But most GCP customers start off with scaling out, not up. Compute Engine has a feature called Autoscaling that lets you add and take away VMs from your application based on load metrics. The other part of making that work is balancing the incoming traffic among the VMs. And Google VPC supports several different kinds of load balancing! We'll consider those in the next section.

## Agenda



- Virtual Private Cloud (VPC) Network
- Compute Engine
- **Important VPC capabilities**
- Quiz and lab



## You control the topology of your VPC network

- Use its **route table** to forward traffic within the network, even across subnets.
- Use its **firewall** to control what network traffic is allowed.
- Use **Shared VPC** to share a network, or individual subnets, with other GCP projects.
- Use **VPC Peering** to interconnect networks in GCP projects.

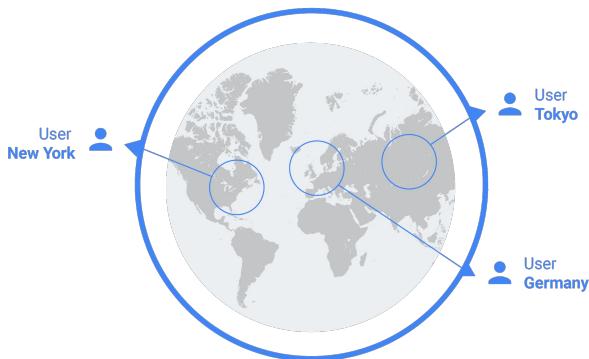


Much like physical networks, VPCs have routing tables. These are used to forward traffic from one instance to another instance within the same network, even across subnetworks and even between GCP zones, without requiring an external IP address. VPCs' routing tables are built in; you don't have to provision or manage a router.

Another thing you don't have to provision or manage for GCP: a firewall. VPCs give you a global distributed firewall you can control to restrict access to instances, both incoming and outgoing traffic. You can define firewall rules in terms of metadata tags on Compute Engine instances, which is really convenient. For example, you can tag all your web servers with, say, "WEB," and write a firewall rule saying that traffic on ports 80 or 443 is allowed into all VMs with the "WEB" tag, no matter what their IP address happens to be.

Recall that VPCs belong to GCP projects. But what if your company has several GCP projects, and the VPCs need to talk to each other? If you simply want to establish a peering relationship between two VPCs, so that they can exchange traffic, configure VPC Peering does. On the other hand, if you want to use the full power of IAM to control who and what in one project can interact with a VPC in another, configure Shared VPC.

With **global Cloud Load Balancing**, your application presents a single front-end to the world



- Users get a single, global anycast IP address.
- Traffic goes over the Google backbone from the closest point-of-presence to the user.
- Backends are selected based on load.
- Only healthy backends receive traffic.
- No pre-warming is required.



A few slides back, we talked about how virtual machines can autoscale to respond to changing load. But how do your customers get to your application when it might be provided by four VMs one moment and forty VMs at another? Cloud Load Balancing is the answer.

Cloud Load Balancing is a fully distributed, software-defined, managed service for all your traffic. And because the load balancers don't run in VMs you have to manage, you don't have to worry about scaling or managing them. You can put Cloud Load Balancing in front of all of your traffic: HTTP(S), other TCP and SSL traffic, and UDP traffic too.

With Cloud Load Balancing, a single anycast IP front-ends all your backend instances in regions around the world. It provides cross-region load balancing, including automatic multi-region failover, which gently moves traffic in fractions if backends become unhealthy. Cloud Load Balancing reacts quickly to changes in users, traffic, network, backend health, and other related conditions.

And what if you anticipate a huge spike in demand? Say, your online game is already a hit; do you need to file a support ticket to warn Google of the incoming load? No. No so-called "pre-warming" is required.



## Google VPC offers a suite of load-balancing options

Global HTTP(S)	Global SSL Proxy	Global TCP Proxy	Regional	Regional internal
Layer 7 load balancing based on load	Layer 4 load balancing of non-HTTPS SSL traffic based on load	Layer 4 load balancing of non-SSL TCP traffic	Load balancing of any traffic (TCP, UDP)	Load balancing of traffic inside a VPC
Can route different URLs to different back ends	Supported on specific port numbers	Supported on specific port numbers	Supported on any port number	Use for the internal tiers of multi-tier applications



If you need cross-regional load balancing for a Web application, use HTTP(S) load balancing. For Secure Sockets Layer traffic that is not HTTP, use the Global SSL Proxy load balancer. If it's other TCP traffic that does not use Secure Sockets Layer, use the Global TCP Proxy load balancer.

Those two proxy services only work for specific port numbers, and they only work for TCP. If you want to load balance UDP traffic, or traffic on any port number, you can still load balance across a GCP region with the Regional load balancer.

Finally, what all those services have in common is that they're intended for traffic coming into the Google network from the Internet. But what if you want to load balance traffic inside your project, say, between the presentation layer and the business layer of your application? For that, use the Internal load balancer. It accepts traffic on a GCP internal IP address and load balances it across Compute Engine VMs.

## Cloud DNS is highly available and scalable

- Create managed zones, then add, edit, delete DNS records
- Programmatically manage zones and records using RESTful API or command-line interface



One of the most famous Google services that people don't pay for is 8.8.8.8, which provides a public Domain Name Service to the world. DNS is what translates Internet hostnames to addresses, and as you would imagine, Google has a highly developed DNS infrastructure. It makes 8.8.8.8 available so that everybody can take advantage of it.

But what about the Internet hostnames and addresses of applications you build in GCP?

GCP offers Cloud DNS to help the world find them. It's a managed DNS service running on the same infrastructure as Google. It has low latency and high availability, and it's a cost-effective way to make your applications and services available to your users. The DNS information you publish is served from redundant locations around the world.

Cloud DNS is also programmable. You can publish and manage millions of DNS zones and records using the GCP Console, the command-line interface, or the API.

## Cloud CDN (Content Delivery Network)

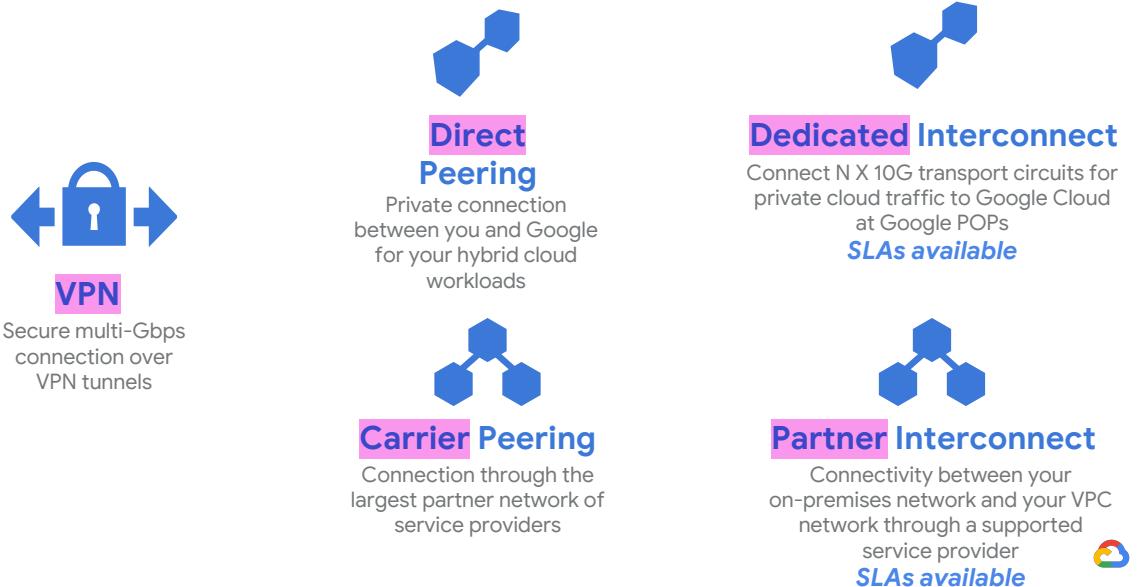
- Use Google's globally distributed edge caches to cache content close to your users
- Or use CDN Interconnect if you'd prefer to use a different CDN



Google has a global system of edge caches. You can use this system to accelerate content delivery in your application using Google Cloud CDN. Your customers will experience lower network latency, the origins of your content will experience reduced load, and you can save money too. Once you've set up HTTP(S) Load Balancing, simply enable Cloud CDN with a single checkbox.

There are lots of other CDNs out there, of course. If you are already using one, chances are, it is a part of GCP's CDN Interconnect partner program, and you can continue to use it.

## Google Cloud Platform offers many interconnect options



Lots of GCP customers want to interconnect their other networks to their Google VPCs. such as on-premises networks or their networks in other clouds. There are many good choices.

Many customers start with a Virtual Private Network connection over the Internet, using the IPsec protocol. To make that dynamic, they use a GCP feature called Cloud Router. Cloud Router lets your other networks and your Google VPC exchange route information over the VPN using the Border Gateway Protocol. For instance, if you add a new subnet to your Google VPC, your on-premises network will automatically get routes to it.

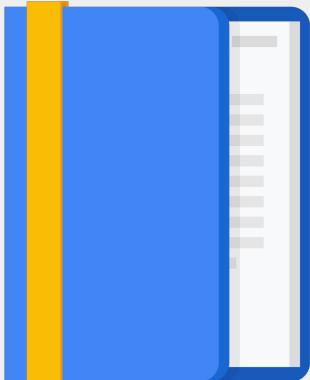
But some customers don't want to use the Internet, either because of security concerns or because they need more reliable bandwidth. They can consider peering with Google using Direct Peering. Peering means putting a router in the same public datacenter as a Google point of presence and exchanging traffic. Google has more than 100 points of presence around the world. Customers who aren't already in a point of presence can contract with a partner in the Carrier Peering program to get connected.

One downside of peering, though, is that it isn't covered by a Google Service Level Agreement. Customers who want the highest uptimes for their interconnection with

Google should use Dedicated Interconnect, in which customers get one or more direct, private connections to Google. If these connections have topologies that meet Google's specifications, they can be covered by up to a 99.99% SLA. These connections can be backed up by a VPN for even greater reliability.

Partner Interconnect provides connectivity between your on-premises network and your VPC network through a supported service provider. A Partner Interconnect connection is useful if your data center is in a physical location that can't reach a Dedicated Interconnect colocation facility or if your data needs don't warrant an entire 10 Gbps connection. Depending on your availability needs, you can configure Partner Interconnect to support mission-critical services or applications that can tolerate some downtime. As with Dedicated Interconnect, if these connections have topologies that meet Google's specifications, they can be covered by up to a 99.99% SLA, but note that Google is not responsible for any aspects of Partner Interconnect provided by the third party service provider nor any issues outside of Google's network.

## Agenda



- Virtual Private Cloud (VPC) Network
- Compute Engine
- Important VPC capabilities
- **Quiz and lab**



# Question #1

---

Name 3 robust networking services available to your applications on Google Cloud Platform.



# Question #1

---

Name 3 robust networking services available to your applications on Google Cloud Platform.

Cloud Virtual Network, Cloud Interconnect, Cloud DNS, Cloud Load Balancing, and Cloud CDN.



# Question #2

---

Name 3 Compute Engine pricing features.



# Question #2

---

Name 3 Compute Engine pricing features.

Per-second billing, custom machine types, preemptible instances.



# Question #3

---

True or False: Google Cloud Load Balancing lets you balance HTTP traffic across multiple Compute Engine regions.



# Question #3

---

True or False: Google Cloud Load Balancing lets you balance HTTP traffic across **multiple** Compute Engine regions.

True



# Lab

---

In this lab, you will create virtual machine (VM) instances and connect to them. You will also connect between both instances.

## Objectives

- Create a Compute Engine virtual machine using the Google Cloud Platform Console
- Create a Compute Engine virtual machine using the gcloud command-line interface
- Connect between the two instances



# More resources

---

Google Compute Engine <https://cloud.google.com/compute/docs/>

Google Cloud Platform VPC <https://cloud.google.com/compute/docs/vpc/>

Google Cloud Stackdriver <https://cloud.google.com/stackdriver/docs/>

gcloud tool guide <https://cloud.google.com/source-repositories/docs/>





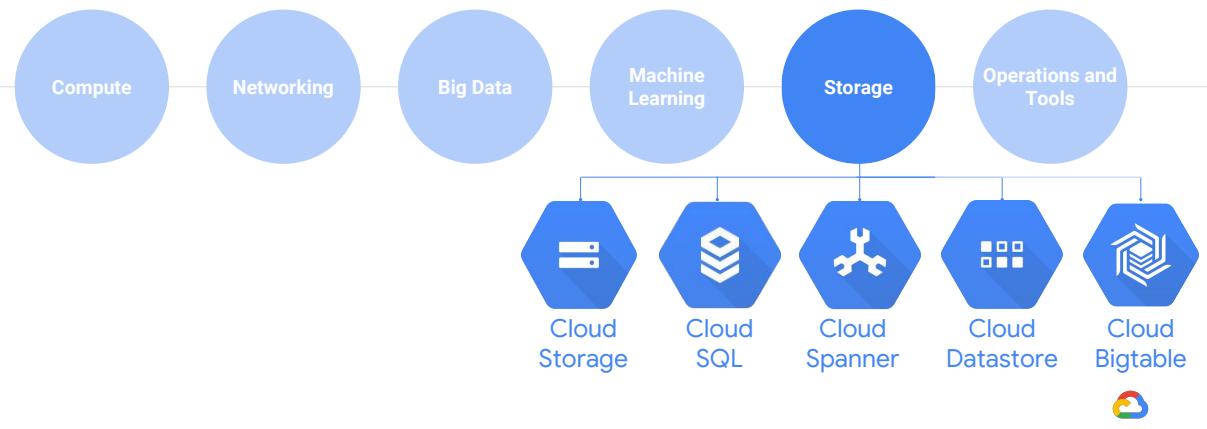
**Google Cloud**

---

GCP Fundamentals: Core Infrastructure

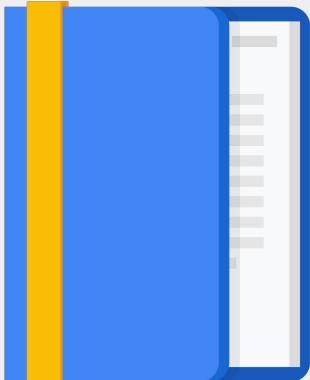
## **Storage in the Cloud**

## Google Cloud Platform



Google Cloud Platform has many storage options that satisfy nearly every customer use case. In this module, we turn our attention to the core storage options: Google Cloud Storage, Google Cloud SQL, Google Cloud Spanner, Cloud Datastore, and Google Cloud Bigtable.

## Agenda



- **Cloud Storage**
- Cloud Bigtable
- Cloud SQL and Cloud Spanner
- Cloud Datastore
- Comparing storage options
- Quiz and Lab



## Cloud Storage is binary large-object storage

- High performance, internet-scale
  - Simple administration
- Does not require capacity management
- Data encryption at rest
- Data encryption in transit by default from Google to endpoint
- Online and offline import services are available



Google Cloud Storage offers developers and IT organizations durable and highly available object storage. It assesses no minimum fee; you pay only for what you use. Prior provisioning of capacity isn't necessary.

What's object storage? It's not the same as file storage, in which you manage your data as a hierarchy of folders. It's not the same as block storage, in which your operating system manages your data as chunks of disk. Instead, object storage means this: you say to your storage, "Here, keep this arbitrary sequence of bytes," and the storage lets you address it with a unique key. In Google Cloud Storage and in other systems, these unique keys are in the form of URLs, which means object storage interacts well with web technologies.

Google Cloud Storage always encrypts your data on the server side, before it is written to disk, at no additional charge. Data traveling between a customer's device and Google is encrypted by default using HTTPS/TLS (Transport Layer Security). In fact, Google was the first major cloud provider to enable HTTPS/TLS by default.

Google Cloud Storage is not a file system, although it can be accessed as one via third-party tools such as Cloud Storage FUSE. The storage objects offered by Google Cloud Storage are "immutable," which means that you do not edit them in place, but instead create a new version. Google Cloud Storage's primary use is whenever binary

large-object storage is needed: online content, backup and archiving, storage of intermediate results in processing workflows, and more.

Offline Media Import/Export is a third-party solution that allows you to load data into Google Cloud Storage by sending your physical media, such as hard disk drives (HDDs), tapes, and USB flash drives, to a third-party service provider who uploads data on your behalf. Offline Media Import/Export is helpful if you're limited to a slow, unreliable, or expensive internet connection.

Offline import is available through third-party providers:

<https://cloud.google.com/storage/docs/offline-media-import-export>

Cloud Storage Transfer Service enables you to import large amounts of online data into Google Cloud Storage quickly and cost-effectively. To use Cloud Storage Transfer Service, you set up a transfer from a data source to data sink. Data sources can be an Amazon Simple Storage Service (Amazon S3) bucket, an HTTP/HTTPS location, or another Google Cloud Storage bucket. Data sinks are always a Google Cloud Storage bucket.

Example uses of Cloud Storage Transfer Service include:

Backing up data to a Google Cloud Storage bucket from other storage providers.

Moving data from a Standard Storage bucket to a Nearline Storage bucket to lower your storage costs.

Your Cloud Storage files are organized into **buckets**

Bucket attributes	Bucket contents
Globally unique name	Files (in a flat namespace)
Storage class	
Location (region or multi-region)	
IAM policies or Access Control Lists	Access Control Lists
Object versioning setting	
Object lifecycle management rules	



Your Cloud Storage files are organized into buckets. When you create a bucket: you give it a globally-unique name; you specify a geographic location where the bucket and its contents are stored; and you choose a default storage class. Pick a location that minimizes latency for your users. For example, if most of your users are in Europe, you probably want to pick a European location: a GCP region in Europe, or else the EU multi-region.

There are several ways to control users' access to your objects and buckets. For most purposes, Cloud IAM is sufficient. Roles are inherited from project to bucket to object. If you need finer control, you can create access control lists ("ACLs") that offer finer control, ACLs define who has access to your buckets and objects, as well as what level of access they have. Each ACL consists of two pieces of information: A scope, which defines who can perform the specified actions (for example, a specific user or group of users). And a permission, which defines what actions can be performed (for example, read or write).

Remember that Cloud Storage objects are immutable. You can turn on object versioning on your buckets if you want. If you do, Cloud Storage keeps a

history of modifications—that is, overwrites or deletes—of all objects in the bucket. You can list the archived versions of an object, restore an object to an older state, or permanently delete a version, as needed. If you don't turn on object versioning, new always overwrites old.

Cloud Storage also offers lifecycle management policies. For example, you could tell Cloud Storage to delete objects older than 365 days, or to delete objects created before January 1, 2013; or to keep only the 3 most recent versions of each object in a bucket that has versioning enabled.

## Choosing among Cloud Storage classes

	Multi-regional	Regional	Nearline	Coldline
Intended for data that is...	Most frequently accessed	Accessed frequently within a region	Accessed less than once a month	Accessed less than once a year
Availability SLA	99.95%	99.90%	99.00%	99.00%
Access APIs	<b>Consistent APIs</b>			
Access time	<b>Millisecond access</b>			
Storage price	Price per GB stored per month			
Retrieval price	Total price per GB transferred			
Use cases	Content storage and delivery	In-region analytics, transcoding	Long-tail content, backups	Archiving, disaster recovery

Cloud Storage lets you choose among four different types of storage classes: Regional, Multi-regional, Nearline and Coldline. Multi-regional and Regional are high-performance object storage, whereas Nearline and Coldline are backup and archival storage. All of the storage classes are accessed in analogous ways using the Cloud Storage API, and they all offer millisecond access times.

**Regional Storage** lets you store your data in a specific GCP region, us-central1, europe-west1 or asia-east1. It's cheaper than multi-regional storage, but it offers less redundancy.

**Multi-Regional Storage** costs a bit more, but it's geo-redundant. That means you pick a broad geographical location, like United States, the European Union, or Asia, and Cloud Storage stores your data in at least two geographic locations separated by at least 160 kilometers.

Multi-Regional Storage is appropriate for storing frequently accessed storing data: website content, interactive workloads, or data that's part of mobile and gaming applications. People use regional storage, on the other hand, to store data close to their Compute Engine virtual machines or their Kubernetes Engine clusters. That gives better performance for data-intensive computations.

**Nearline storage** is a low-cost, highly durable storage service for storing infrequently accessed data. This storage class is a better choice than Multi-Regional Storage or Regional Storage in scenarios where you plan to read or modify your data on average once a month or less. For example, if you want to continuously add files to Cloud Storage and plan to access those files once a month for analysis, Nearline Storage is a great choice.

**Coldline Storage** is a very-low-cost, highly durable storage service for data archiving, online backup, and disaster recovery. Coldline Storage is the best choice for data that you plan to access at most once a year, due to its slightly lower availability, 90-day minimum storage duration, costs for data access, and higher per-operation costs. For example, if you want to archive data or have access in the event of a disaster recovery event.

The availability of these storage classes varies, with multi-regional having the highest availability of 99.95%, followed by regional with 99.9% and nearline and coldline with 99.0%.

As for pricing, all storage classes incur a cost per gigabyte of data stored per month, with multi-regional having the highest storage price and coldline the lowest storage price. Egress and data transfer charges may also apply.

In addition to those charges, Nearline storage also incurs an access fee per gigabyte of data read, and Coldline storage incurs a higher fee per gigabyte of data read.

There are several ways to bring data into Cloud Storage



### Online transfer

Self-managed copies using command-line tools or drag-and-drop



### Storage Transfer Service

Scheduled, managed batch transfers



### Transfer Appliance Beta

Rackable appliances to securely ship your data



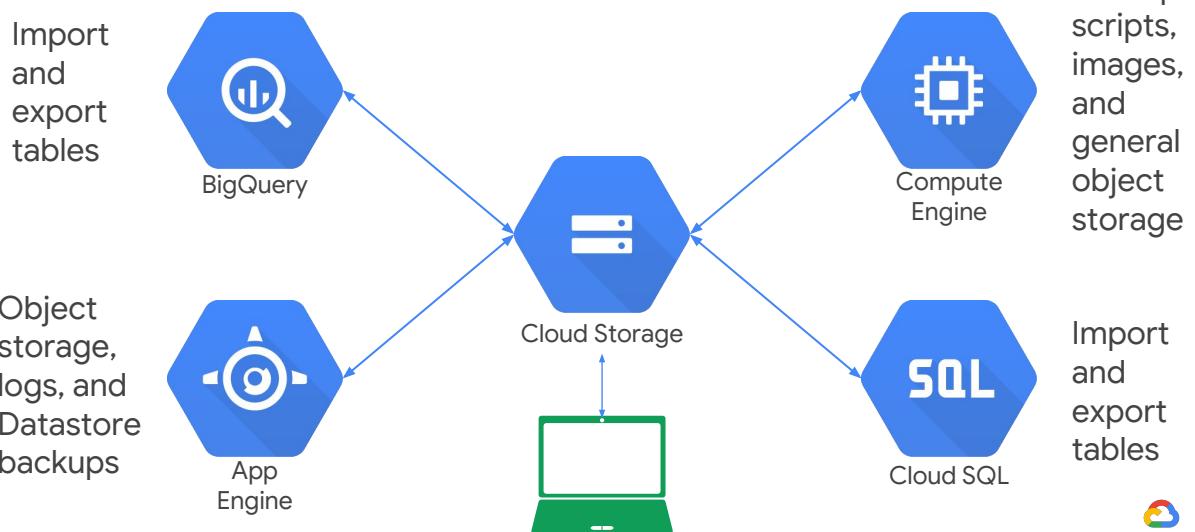
Regardless of which storage class you choose, there are several ways to bring data into Cloud Storage.

Many customers simply use gsutil, which is the Cloud Storage command from the Cloud SDK. You can also move data in with a drag and drop in the GCP Console, if you use the Google Chrome browser. But what if you have to upload terabytes or even petabytes of data? Google Cloud Platform offers the online Storage Transfer Service and the offline Transfer Appliance to help.

The Storage Transfer Service lets you schedule and manage batch transfers to Cloud Storage from another cloud provider, from a different Cloud Storage region, or from an HTTP(S) endpoint.

The Transfer Appliance is a rackable, high-capacity storage server that you lease from Google Cloud. You simply connect it to your network, load it with data, and then ship it to an upload facility where the data is uploaded to Cloud Storage. The service enables you to securely transfer up to a petabyte of data on a single appliance. As of this recording, it's still beta, and it's not available everywhere, so check the website for details.

## Cloud Storage works with other GCP services



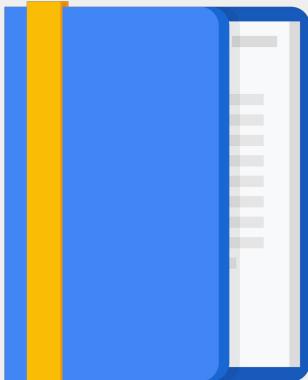
There are other ways of getting your data into Cloud Storage, as this storage option is tightly integrated with many of the Google Cloud Platform products and services.

For example, you can import and export tables from and to BigQuery, as well as Cloud SQL.

You can also store App Engine logs, Cloud Datastore backups, and objects used by App Engine applications like images. Cloud Storage can also store instance startup scripts, Compute Engine images, and objects used by Compute Engine applications.

In short, Cloud Storage is often the ingestion point for data being moved into the cloud, and is frequently the long-term storage location for data.

## Agenda



- Cloud Storage
- **Cloud Bigtable**
- **Cloud SQL** and **Cloud Spanner**
- Cloud Datastore
- Comparing storage options
- Quiz and Lab



## Cloud Bigtable is managed NoSQL

- Fully managed **NoSQL, wide-column database** service for terabyte applications
- Integrated
  - Accessed using **HBase API**
  - Native compatibility with big data, Hadoop ecosystems



Cloud Bigtable is Google's NoSQL big data database service. It's the same database that powers many core Google services, including Search, Analytics, Maps, and Gmail.

## Why choose Cloud Bigtable?

- Replicated storage
- Data encryption in-flight and at rest
- Role-based ACLs
- Drives major applications such as Google Analytics and Gmail



Customers frequently choose Bigtable if the data is:

*Big*

- Large quantities (>1 TB) of semi-structured or structured data

*Fast*

- Data is high throughput or rapidly changing

*NoSQL*

- Transactions, strong relational semantics not required

And especially if it is:

*Time series*

- Data is time-series or has natural semantic ordering

*Big data*

- You run asynchronous batch or real-time processing on the data

*Machine learning*

- You run machine learning algorithms on the data

Bigtable is designed to handle massive workloads at consistent low latency and high throughput, so it's a great choice for both operational and analytical applications, including IoT, user analytics, and financial data analysis.

For more information on Cloud Bigtable, see

<https://www.google.com/url?q=https://cloudplatform.googleblog.com/2015/05/introducing-Google-Cloud-Bigtable.html>

## Bigtable Access Patterns



### Application API

Data can be read from and written to Cloud Bigtable through a data service layer like Managed VMs, the HBase REST Server, or a Java Server using the HBase client. Typically this will be to serve data to applications, dashboards, and data services.

### Streaming

Data can be streamed in (written event by event) through a variety of popular stream processing frameworks like Cloud Dataflow Streaming, Spark Streaming, and Storm.

### Batch Processing

Data can be read from and written to Cloud Bigtable through batch processes like Hadoop MapReduce, Dataflow, or Spark. Often, summarized or newly calculated data is written back to Cloud Bigtable or to a downstream database.



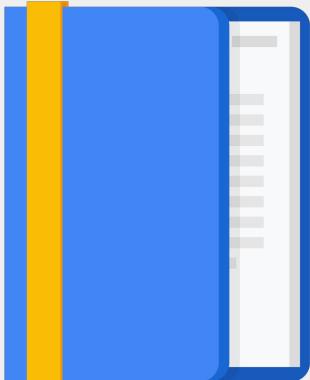
As Cloud Bigtable is part of the GCP ecosystem, it can interact with other GCP services and third-party clients.

From an application API perspective, data can be read from and written to Cloud Bigtable through a data service layer like Managed VMs, the HBase REST Server, or a Java Server using the HBase client. Typically this will be to serve data to applications, dashboards, and data services.

Data can also be streamed in through a variety of popular stream processing frameworks like Cloud Dataflow Streaming, Spark Streaming, and Storm.

If streaming is not an option, data can also be read from and written to Cloud Bigtable through batch processes like Hadoop MapReduce, Dataflow, or Spark. Often, summarized or newly calculated data is written back to Cloud Bigtable or to a downstream database.

## Agenda



- Cloud Storage
- Cloud Bigtable
- **Cloud SQL and Cloud Spanner**
- Cloud Datastore
- Comparing storage options
- Quiz and Lab



## Cloud SQL is a managed RDBMS

- Offers MySQL and PostgreSQL databases as a service
- Automatic replication
- Managed backups
- Vertical scaling (read and write)
- Horizontal scaling (read)
- Google security



Cloud SQL is an easy-to-use service that delivers fully managed relational databases. Cloud SQL lets you hand off to Google the mundane, but necessary and often time-consuming tasks—like applying patches and updates, managing backups, and configuring replications—so you can put your focus on building great applications.

Every Cloud SQL instance includes a network firewall, allowing you to control network access to your database instance by granting access.

Cloud SQL is easy to use: it doesn't require any software installation or maintenance.

Easily scale up to 64 processor cores and more than 100 GB of RAM. Quickly scale out with read replicas.

### *Automatic replication*

Google Cloud SQL supports the following read replica scenarios:

- Cloud SQL instances replicating from a Cloud SQL master instance  
Replicas are other instances in the same project and location as the master instance. This feature is in Beta.
- Cloud SQL instances replicating from an external master instance  
The master instance is external to Google Cloud SQL. For example, it

- can be outside the Google network or in a Google Compute Engine instance. This feature is in Beta.
- External MySQL instances replicating from a Cloud SQL master instance  
External replicas are in hosting environments, outside of Cloud SQL.

#### *Managed backups*

Cloud SQL takes care of securely storing your backed-up data and makes it easy for you to restore from a backup and perform a point-in-time recovery to a specific state of an instance. Cloud SQL retains up to 7 backups for each instance, which is included in the cost of your instance.

Cloud SQL customer data is encrypted when on Google's internal networks and when stored in database tables, temporary files, and backups.

(MySQL instances are available in either First Generation or Second Generation. Google recommends the use of Second Generation instances for most use cases. First Generation instances are recommended primarily when MySQL 5.5 compatibility is required. Also, First Generation instances may be cost-effective for infrequently used or test/dev database instances, because of their available Per-Use billing plan and the available ON DEMAND activation policy, which causes your instance to automatically shut itself off after 15 minutes of inactivity.)

## Cloud SQL can be used with other GCP services



Cloud SQL can be used with App Engine using standard drivers.

You can configure a Cloud SQL instance to follow an App Engine application.



Compute Engine instances can be authorized to access Cloud SQL instances using an external IP address.

Cloud SQL instances can be configured with a preferred zone.



Cloud SQL can be used with external applications and clients.

Standard tools can be used to administer databases.

External read replicas can be configured.



Another benefit of Cloud SQL instances is that they are accessible by other GCP services and even external services. You can use Cloud SQL with App Engine using standard drivers like Connector/J for Java or MySQLdb for Python.

You can authorize Compute Engine instances to access Cloud SQL instances and configure the Cloud SQL instance to be in the same zone as your virtual machine.

Cloud SQL also supports other applications and tools that you might be used to, like SQL Workbench, Toad and other external applications using standard MySQL drivers.

Cloud Spanner is a horizontally scalable RDBMS

Cloud Spanner supports:

- Automatic replication
- Strong global consistency
- Managed instances with high availability
- SQL (ANSI 2011 with extensions)



Cloud Spanner supports strong consistency, including strongly consistent secondary indexes, SQL, and managed instances with high availability through synchronous and built-in data replication. Battle tested by Google's own mission-critical applications and services, Spanner powers Google's \$80 billion business.

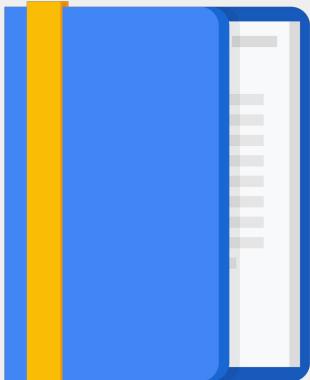
Cloud Spanner is especially suited for applications requiring:

- A SQL RDBMS, with joins and secondary indexes
- Built-in high availability
- Strong global consistency
- Database sizes exceeding ~2 TB
- Many IOPS (Tens of thousands of reads/writes per second or more)

For a technical overview of Cloud Spanner, see

<https://cloudplatform.googleblog.com/2017/02/inside-Cloud-Spanner-and-the-CAP-Theorem.html>.

## Agenda



- Cloud Storage
- **Cloud Bigtable**
- Cloud SQL and Cloud Spanner
- **Cloud Datastore**
- Comparing storage options
- Quiz and Lab



**Cloud Datastore** is a horizontally scalable NoSQL DB

- [NoSQL](#) designed for application backends
- Fully managed
  - Uses a distributed architecture to automatically manage scaling
- Built-in redundancy
- Supports [ACID](#) transactions



Cloud Datastore is a highly-scalable NoSQL database for your applications. Like Cloud Bigtable, there is no need for you to provision database instances. Cloud Datastore uses a distributed architecture to automatically manage scaling. Your queries scale with the size of your result set, not the size of your data set.

Cloud Datastore runs in Google data centers, which use redundancy to minimize impact from points of failure. Your application can still use Cloud Datastore when the service receives a planned upgrade.

The total size of Cloud Datastore databases can grow to terabytes and more.

## Google Cloud Datastore: benefits

- Schemaless access
  - No need to think about underlying data structure
- Local development tools
- Includes a free daily quota
- Access from anywhere through a [RESTful interface](#)

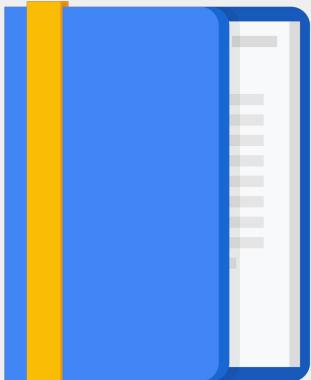


## Cloud Datastore features:

- *Atomic transactions*  
Datastore can execute a set of operations where either all succeed, or none occur.
- *High availability of reads and writes*  
Datastore runs in Google data centers, which use redundancy to minimize impact from points of failure.
- *Massive scalability with high performance*  
Datastore uses a distributed architecture to automatically manage scaling. Datastore uses a mix of indexes and query constraints so your queries scale with the size of your result set, not the size of your data set.
- *Flexible storage and querying of data*  
Datastore maps naturally to object-oriented and scripting languages and is exposed to applications through multiple clients. It also provides a SQL-like [query language](#).
- *Balance of strong and eventual consistency*  
Datastore ensures that entity lookups and ancestor queries always receive strongly consistent data. All other queries are eventually consistent. The consistency models allow your application to deliver a great user experience while handling large amounts of data and users.

- *Encryption at rest*  
Datastore automatically encrypts all data before it is written to disk and automatically decrypts the data when read by an authorized user. For more information, see [Server-Side Encryption](#).
- *Fully managed with no planned downtime*  
Google handles the administration of the Datastore service so you can focus on your application. Your application can still use Datastore when the service receives a planned upgrade.

## Agenda



- Cloud Storage
- Cloud Bigtable
- Cloud SQL and Cloud Spanner
- Cloud Datastore
- **Comparing storage options**
- Quiz and Lab



## Comparing storage options: technical details

	Cloud Datastore	Bigtable	Cloud Storage	Cloud SQL	Cloud Spanner	BigQuery
Type	NoSQL document	NoSQL wide column	Blobstore	Relational SQL for OLTP	Relational SQL for OLTP	Relational SQL for OLAP
Transactions	Yes	Single-row	No	Yes	Yes	No
Complex queries	No	No	No	Yes	Yes	Yes
Capacity	Terabytes+	Petabytes+	Petabytes+	Up to ~10 TB	Petabytes	Petabytes+
Unit size	1 MB/entity	~10 MB/cell ~100 MB/row	5 TB/object	Determined by DB engine	10,240 MiB/row	10 MB/row



Now that we covered GCP's core storage options, let's compare them to help you choose the right service for your application or workflow.

This table focuses on the technical differentiators of the storage services. Each row is a technical specification and each column is a service. Let me cover each service from left to right.

Consider using Cloud Datastore, if you need to store structured objects, or if you require support for transactions and SQL-like queries. This storage service provides terabytes of capacity with a maximum unit size of 1 MB per entity.

Consider using Cloud Bigtable, if you need to store a large amount of structured objects. Cloud Bigtable does not support SQL queries, nor does it support multi-row transactions. This storage service provides petabytes of capacity with a maximum unit size of 10 MB per cell and 100 MB per row.

Consider using Cloud Storage, if you need to store immutable blobs larger than 10 MB, such as large images or movies. This storage service provides petabytes of capacity with a maximum unit size of 5 TB per object.

Consider using Cloud SQL or Cloud Spanner if you need full SQL support for an online transaction processing system. Cloud SQL provides up to 10,230 GB, depending on machine type, while Cloud Spanner provides petabytes. If Cloud SQL does not fit your requirements because you need horizontal scalability, not just

through read replicas, consider using Cloud Spanner.

We didn't cover BigQuery in this module as it sits on the edge between data storage and data processing, but you will learn more about it in the "Big Data and Machine Learning in the Cloud" module. The usual reason to store data in BigQuery is to use its big data analysis and interactive querying capabilities. You would not want to use BigQuery, for example, as the backing store for an online application.

## Comparing storage options: use cases

	Cloud Datastore	Cloud Bigtable	Cloud Storage	Cloud SQL	Cloud Spanner	BigQuery
Type	NoSQL document	NoSQL wide column	Blobstore	Relational SQL for OLTP	Relational SQL for OLTP	Relational SQL for OLAP
Best for	Semi-structured application data, durable key-value data	"Flat" data, Heavy read/write, events, analytical data	Structured and unstructured binary or object data	Web frameworks, existing applications	Large-scale database applications (> ~2 TB)	Interactive querying, offline analytics
Use cases	Getting started, App Engine applications	AdTech, Financial and IoT data	Images, large media files, backups	User credentials, customer orders	Whenever high I/O, global consistency is needed	Data warehousing



Considering the technical differentiators of the different storage services helps some people decide which storage service to choose, others like to consider use cases. Let me go through each service one more time.

Cloud Datastore is best for semi-structured application data that is used in App Engine applications.

Bigtable is best for analytical data with heavy read and write events, like AdTech, financial or IoT data.

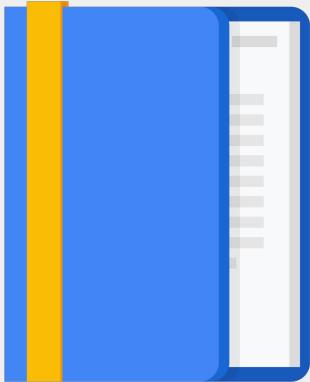
Cloud Storage is best for structured and unstructured binary or object data, like images, large media files and backups.

Cloud SQL is best for web frameworks and existing applications, like storing user credentials and customer orders.

Cloud Spanner is best for large-scale database applications that are larger than 2 TB. For example, for financial trading and e-commerce use cases.

As I mentioned at the beginning of the module, depending on your application you might use one or several of these services to get the job done.

## Agenda



- Cloud Storage
- Cloud Bigtable
- Cloud SQL and Cloud Spanner
- Cloud Datastore
- Comparing storage options
- **Quiz and Lab**



# Question #1

---

Your application transcodes large video files. Which storage service should you consider first?



# Question #1

---

Your application transcodes large video files. Which storage service should you consider first?

Google Cloud Storage



## Question #2

---

You stream huge amounts of data from devices with sensors. Which storage service should you consider first?



# Question #2

---

You stream huge amounts of data from devices with sensors. Which storage service should you consider first?

Google Cloud Bigtable



# Lab

---

In this lab you will store an image in a Cloud Storage bucket and configure GCE to use a Cloud SQL database to reference the image.

## Objectives

- Create a Cloud Storage bucket and place an image into it
- Create a Cloud SQL instance and configure it
- Connect to a Cloud SQL instance from a web server
- Use an image stored in a Cloud Storage bucket in a web page



In this lab you will create a Google Cloud Storage bucket and place an image in it. You'll also configure an application running in Google Compute Engine to use a database managed by Google Cloud SQL and to reference the image in the Cloud Storage bucket.

# More resources

---

Overview of Cloud Storage <https://cloud.google.com/storage/>

Getting started with Google Cloud SQL <https://cloud.google.com/sql/docs/quickstart>

Cloud Bigtable <https://cloud.google.com/stackdriver/docs/>

Cloud Spanner <https://cloud.google.com/spanner/docs/>

Cloud Datastore <https://cloud.google.com/datastore/docs/>





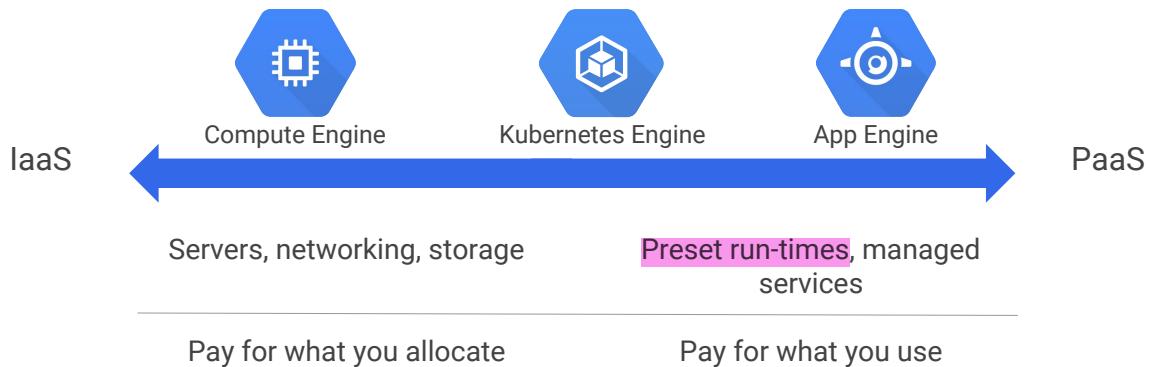
# Google Cloud

---

GCP Fundamentals: Core Infrastructure

**Containers in the Cloud**

## Review: IaaS and PaaS

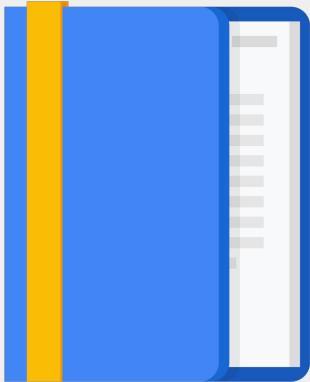


We've already discussed the spectrum between Infrastructure as a Service and Platform as a Service. And We've already discussed Compute Engine, which is GCP's Infrastructure as a Service offering, with access to servers, file systems, and networking.

Now we'll introduce you to containers and Kubernetes Engine which is a hybrid which conceptually sits between the two, offering the managed infrastructure of an IaaS offering with the developer orientation of a PaaS offering.

Another choice for organizing your compute is using containers rather than virtual machines, and using Kubernetes Engine to manage them. Containers are simple and interoperable, and they enable seamless, fine-grained scaling. Google launches billions of containers every week.

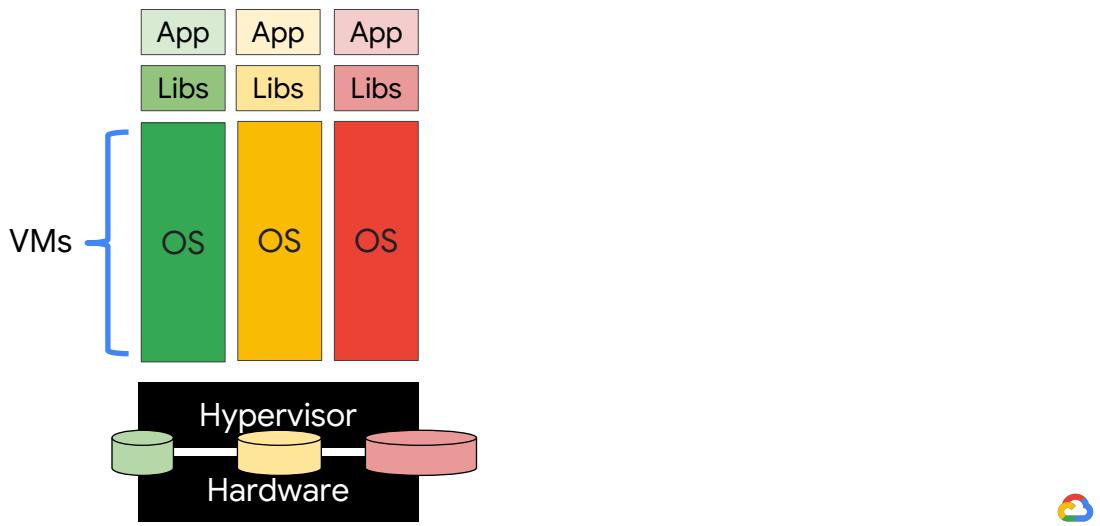
## Agenda



- **Introduction to Containers**
- Kubernetes and Kubernetes Engine
- Quiz and Lab



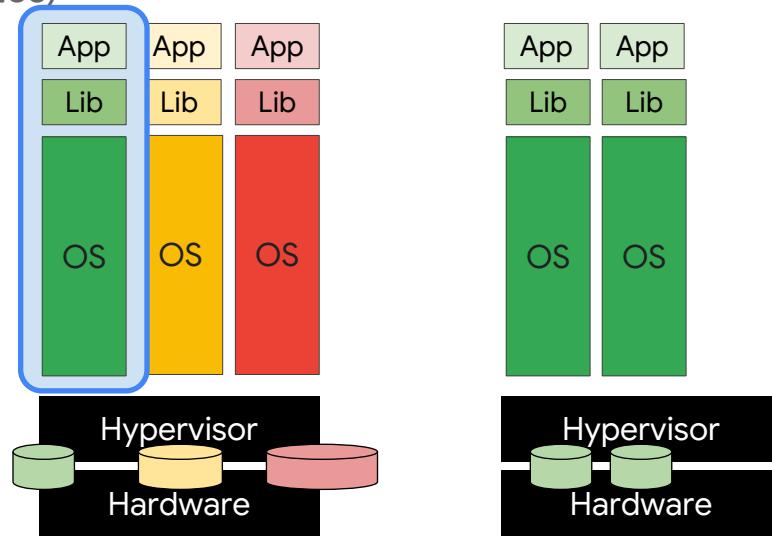
IaaS allows you to share resources by virtualizing the hardware



Let's begin, by remembering that Infrastructure as a Service allows you to share compute resources with other developers by virtualizing the hardware using virtual machines. Each developer can deploy their own operating system, access the hardware, and build their applications in a self-contained environment with access to their own runtimes and libraries as well as their own partitions of RAM, file systems, networking interfaces, and so on.

You have your tools of choice on your own configurable system. So you can install your favorite runtime, web server, database, or middleware, configure the underlying system resources, such as disk space, disk I/O, or networking and build as you like.

But flexibility has costs in boot time (minutes) and resources (Gigabytes)

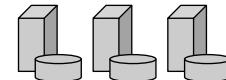


But flexibility comes with a cost. The smallest unit of compute is an app with its VM. The guest OS may be large, even gigabytes in size, and takes minutes to boot.

As demand for your application increases, you have to copy an entire VM and boot the guest OS for each instance of your app, which can be slow and costly.

PaaS provides hosted services and an environment that can scale workloads independently

- All you do is write code in self-contained workloads and include any libraries
- It's easy to decouple code because you're not tied to the underlying hardware, OS, or a lot of the software stacks you used to manage



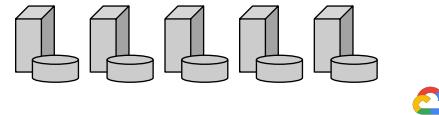
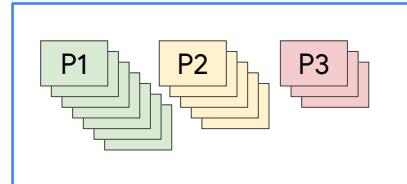
With Platform as a Service you get access to hosted programming services.

All you do is write your code in self-contained workloads that use these services and include any dependent libraries.

Workloads do not need to represent entire applications. They're easier to decouple because they're not tied to the underlying hardware, OS, or a lot of the software stacks that you used to manage.

As demand for your app increases, the platform scales your app rapidly and independently by workload and infrastructure

- This encourages you to build your apps as decoupled microservices
- But you may not be able to fine-tune the underlying architecture to save cost



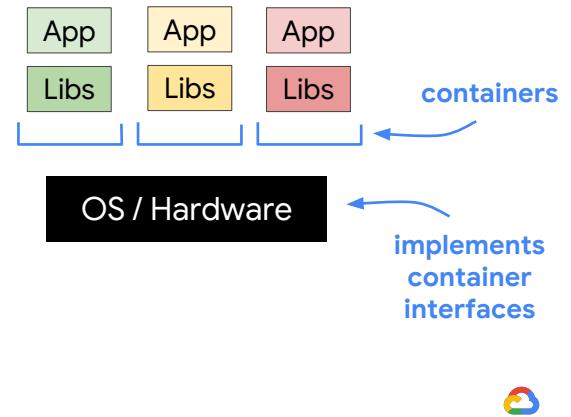
As demand for your app increases, the platform scales your app seamlessly and independently by workload and infrastructure.

This scales rapidly and encourages you to build your applications as decoupled microservices that run more efficiently, but you won't be able to fine-tune the underlying architecture to save cost.

## Containers offer IaaS flexibility and PaaS scalability

Containers provide:

- An abstraction layer of the hardware and OS
- An invisible box with configurable access to isolated partitions of file system, RAM, and networking
- Fast startup (only a few system calls)



That's where containers come in.

The idea of a container is to give you the independent scalability of workloads in PaaS and an abstraction layer of the OS and hardware in IaaS.

What you get is an invisible box around your code and its dependencies, with limited access to its own partition of the file system and hardware.

It only requires a few system calls to create and it starts as quickly as a process.

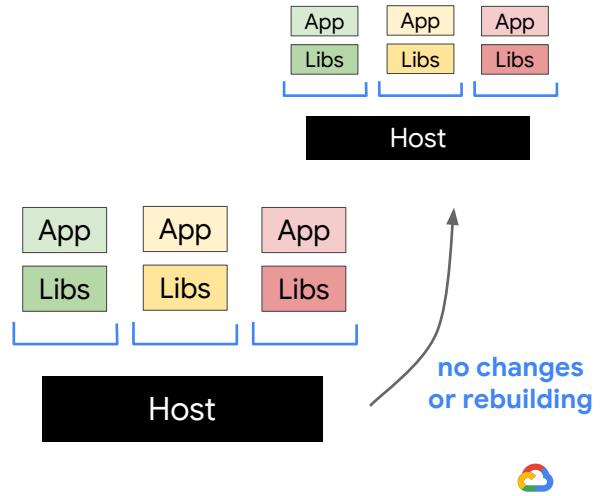
All you need on each host is an OS kernel that supports containers and a container runtime.

In essence, you are virtualizing the OS. It scales like PaaS, but gives you nearly the same flexibility as IaaS.

## Containers are configurable, self-contained, and ultra-portable

With containers, you can:

- Define your own hardware, OS, and software stack configurations
- Bundle any dependencies
- Treat the OS and hardware as a black box and go from dev, to staging, to production, or your laptop to the cloud, without changing or rebuilding anything



Because a container can configure its own isolated partition of file system, RAM, and networking, and specify its own dependencies on top of that, it's ultra portable.

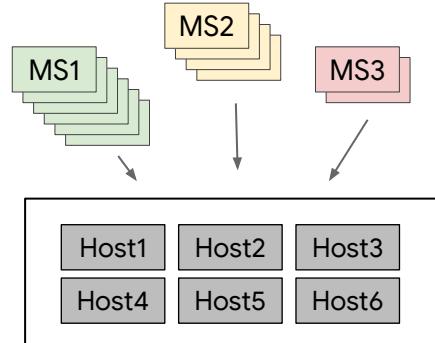
You can effectively treat the OS and hardware as a black box.

So you can go from development, to staging, to production, or from your laptop to the cloud, without changing or rebuilding anything.

With a common host configuration, you can deploy hundreds or thousands of containers on a group of servers called a cluster

With a cluster, you can:

- Connect containers using network connections
- Build code modularly
- Deploy it easily
- Scale containers and hosts independently for maximum efficiency and savings



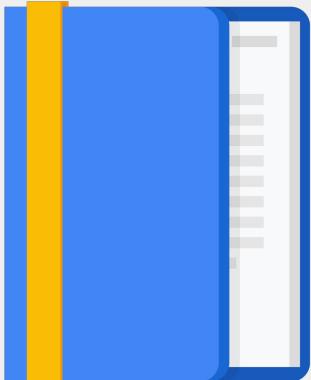
If you want to scale, for example, a web server, you can do so in seconds and deploy dozens or hundreds of them (depending on the size or your workload) on a single host or group of hosts.

You'll likely want to build your applications using lots of containers, each performing their own function like microservices.

If you build them this way, and connect them with network connections, you can make them modular, deploy easily, and scale independently across a group of hosts.

And the hosts can scale up and down and start and stop containers as demand for your app changes or as hosts fail.

## Agenda



- Introduction to Containers
- **Kubernetes and Kubernetes Engine**
- Quiz and Lab



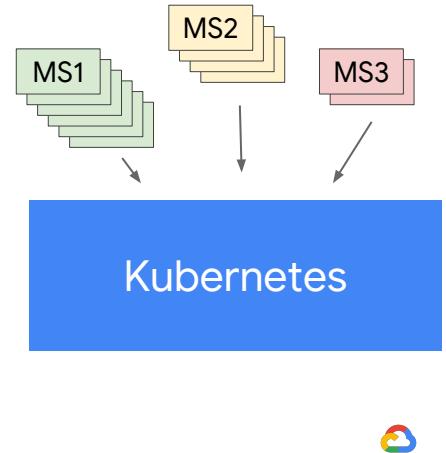
## Kubernetes makes it easy to orchestrate many containers on many hosts

It's an open-source container management platform

- It was built by Google to run applications at scale
- Google starts billions of containers with it every week

Kubernetes lets you:

- Install the system on local servers in the cloud
- Manage container networking and data storage
- Deploy rollouts and rollbacks
- Monitor and manage container and host health



A tool that helps you do this well is Kubernetes.

You can install Kubernetes on a group of your own managed servers or run it as a hosted service in GCP on a cluster of managed GCE instances called Kubernetes Engine.

Kubernetes makes it easy to orchestrate many containers on many hosts, scale them as microservices, and deploy rollouts and rollbacks.

## Let's begin by building and running an app as a container

- We'll use an open-source tool called **Docker** that bundles your app, its dependencies, and any system settings
  - You could use another tool like **Google Cloud Build**
- Here is an example of some code you may have written:
  - It's a Python app that says 'hello world'
  - Or if you hit the second endpoint, it gives you the version

app.py

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!\n"

@app.route("/version")
def version():
    return "Helloworld 1.0\n"

if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

First, I'll show you how you build and run containers.

I'll use an open-source tool called **Docker** that defines a format for bundling your application, its dependencies, and machine-specific settings into a container; you could use a different tool like **Google Cloud Build**. It's up to you.

Here is an example of some code you may have written.

It's a Python app.

It says "Hello World"

Or if you hit the second endpoint '/version', it gives you the version.

## How do you get this app into Kubernetes?

### requirements.txt

```
Flask==0.12
uwsgi==2.0.15
```

You use a Dockerfile to specify such things as:

- A requirements.txt file for Flask dependencies
- Your OS image and version of Python
- How to install Python
- How to run your app

### Dockerfile

```
FROM ubuntu:18.10
RUN apt-get update -y && \
    apt-get install -y python3-pip python3-dev
COPY requirements.txt /app/requirements.txt
WORKDIR /app
RUN pip3 install -r requirements.txt
COPY . /app
ENDPOINT ["python3", "app.py"]
```



So how do you get this app into Kubernetes?

You have to think about your version of Python, what dependency you have on Flask, how to use the requirements.txt file, how to install Python, and so on.

So you use a **Dockerfile** to specify how your code gets packaged into a container.

For example, if you're a developer, and you're used to using Ubuntu with all your tools, you start there.

You can install Python the same way you would on your dev environment.

You can take that requirements file from Python that you know.

And you can use tools inside **Docker** or **Cloud Build** to install your dependencies the way you want.

Eventually, it produces an app, and you run it with the ENDPOINT command.

Then you build and run the container as an image

```
$> docker build -t py-server .
$> docker run -d py-server
```

- **docker build** builds a container and stores it locally as a runnable image
- You can upload images to a registry service (like [Google Container Registry](#)) for sharing
- **docker run** starts the container image



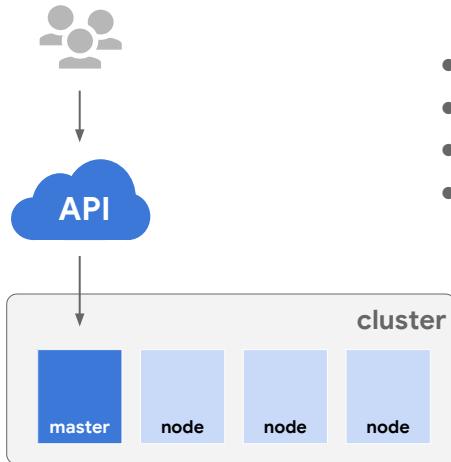
Then you use the "[docker build](#)" command to build the container.

This builds the container and stores it locally as a runnable **image**. You can save and upload the image to a container registry service and share or download it from there.

Then you use the "[docker run](#)" command to run the image.

As it turns out, packaging applications is only about 5% of the issue. The rest has to do with: application configuration, service discovery, managing updates, and monitoring. These are the components of a reliable, scalable, distributed system.

You use Kubernetes APIs to deploy containers on a set of nodes called a cluster



- Masters run the control plane
- Nodes run containers
- Nodes are VMs (in GKE they're GCE instances)
- You describe the apps, Kubernetes figures out how to make that happen



Now, I'll show you where **Kubernetes** comes in.

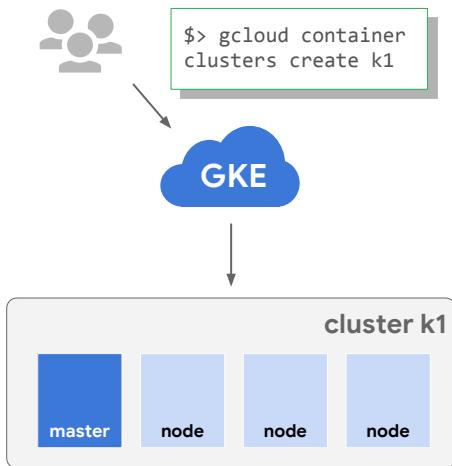
Kubernetes is an open-source **orchestrator** that abstracts containers at a higher level so you can better manage and scale your applications.

At the highest level, Kubernetes is a set of APIs that you can use to deploy containers on a set of **nodes** called a **cluster**.

The system is divided into a set of **master** components that run as the control plane and a set of **nodes** that run containers. In Kubernetes, a node represents a computing instance, like a machine. In Google Cloud, nodes are virtual machines running in Compute Engine.

You can describe a set of applications and how they should interact with each other and Kubernetes figures how to make that happen

## Here's how to bootstrap Kubernetes Engine



In a GKE cluster, you can specify:

- Machine types
- Number of nodes
- Network settings and so on



Now that you've built a container, you'll want to deploy one into a **cluster**.

Kubernetes can be configured with many options and add-ons, but can be time consuming to bootstrap from the ground up. Instead, you can bootstrap Kubernetes using **Kubernetes Engine** or (GKE).

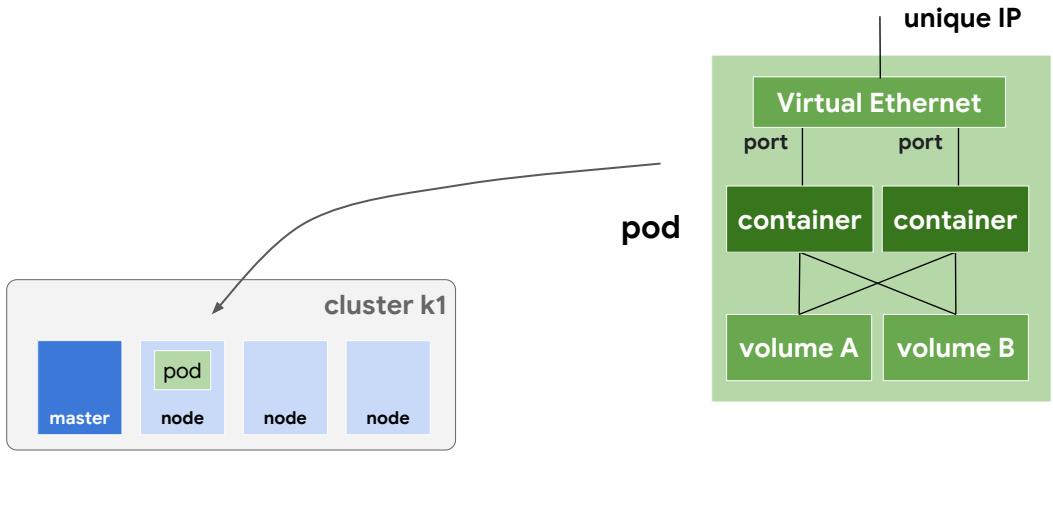
**GKE** is a hosted Kubernetes by Google. GKE clusters can be customized and they support different machine types, number of nodes, and network settings.

To start up Kubernetes on a **cluster** in GKE, all you do is run this command:

At this point, you should have a cluster called 'k1' configured and ready to go.

You can check its status in admin console.

When you deploy containers on nodes you use a wrapper called a **Pod**



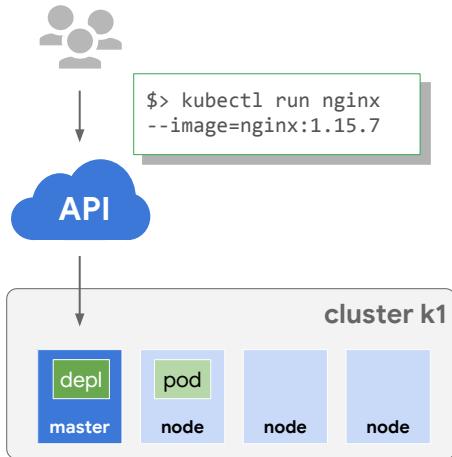
Then you deploy **containers** on nodes using a wrapper around one or more containers called a Pod.

A **Pod** is the smallest unit in Kubernetes that you create or deploy. A Pod represents a running process on your cluster as either a component of your application or an entire app.

Generally, you only have one container per pod, but if you have multiple containers with a hard dependency, you can package them into a single pod and share networking and storage. The Pod provides a unique network IP and set of ports for your containers, and options that govern how containers should run.

Containers inside a Pod can communicate with one another using localhost and ports that remain fixed as they're started and stopped on different nodes.

You can run a container in a Pod using **kubectl run**



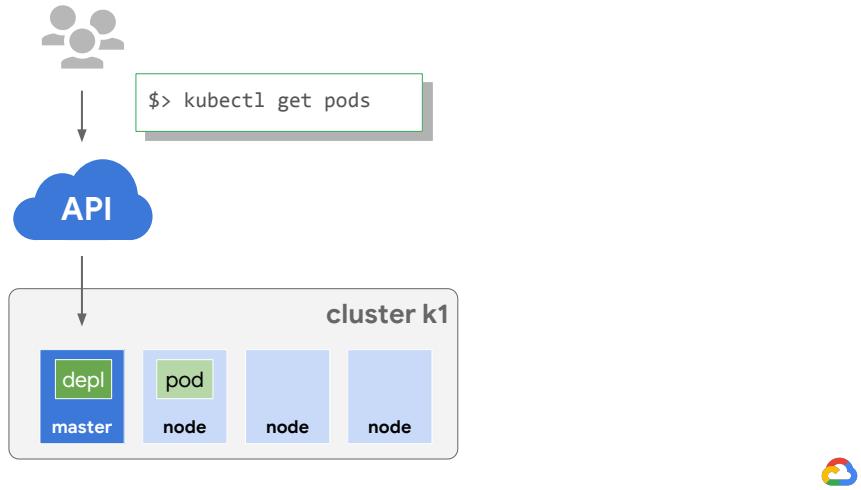
- **kubectl** is the command-line client to the Kubernetes API
- This command starts a **Deployment** with a container running in a Pod
- In this case, the container is an image of the NGINX server



One way to run a container in a Pod in Kubernetes is to use the **kubectl run** command. We'll learn a better way later in this module, but this gets you started quickly.

This starts a **Deployment** with a container running in a **Pod** and in this case, the container inside the Pod is an image of the nginx server.

You use a **Deployment** to manage a set of replica Pods for an app or workload and make sure the desired number is running and healthy

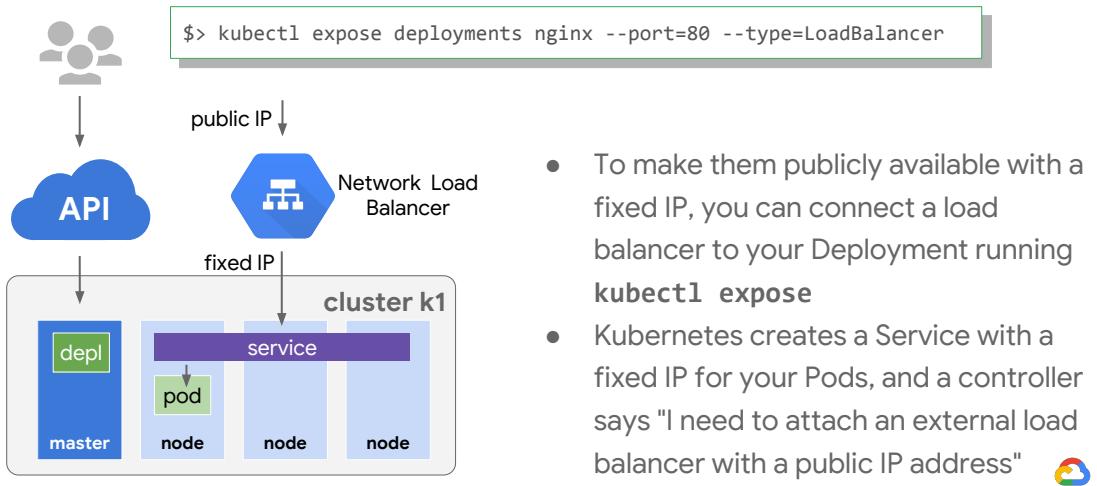


A **Deployment** manages a set of replica Pods representing an app or workload and makes sure the number of desired is running and healthy. It keeps your Pods running even when nodes they run on fail. It could represent a component of an application or an entire app. In this case, it's the nginx web server.

To see the running nginx Pods, run the command:

```
$ kubectl get pods
```

By default, Pods are only available inside a cluster and they get ephemeral IPs



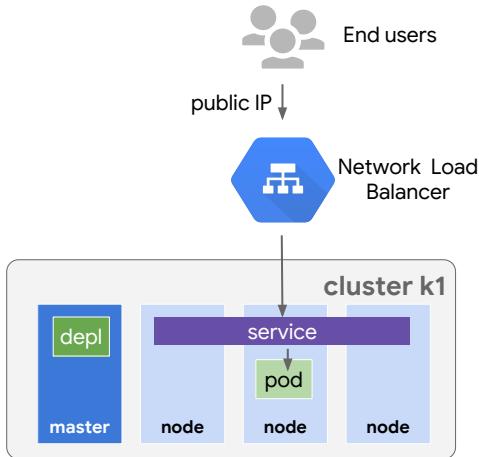
By default, Pods in a Deployment are only accessible by default in the cluster and have ephemeral IP addresses.

To make them publicly available with a fixed IP, you can connect a load balancer to your Deployment by running the **kubectl expose** command:

Kubernetes creates a **Service** with a fixed IP for your Pods, and a controller says "I need to attach an external **load balancer** with a public IP address to that **Service** so others outside the cluster can access it".

In **GKE**, the load balancer is created as a **Network Load Balancer**.

Any client hitting that IP will be routed to a Pod behind the Service



- For example, if you create two sets of Pods called frontend and backend, and put them behind their own Services, backend Pods may change, but frontend Pods are not aware of this. They simply refer to the backend Service.



Any client that hits that IP address will be routed to a Pod behind the Service, in this case there is only one--your simple nginx Pod.

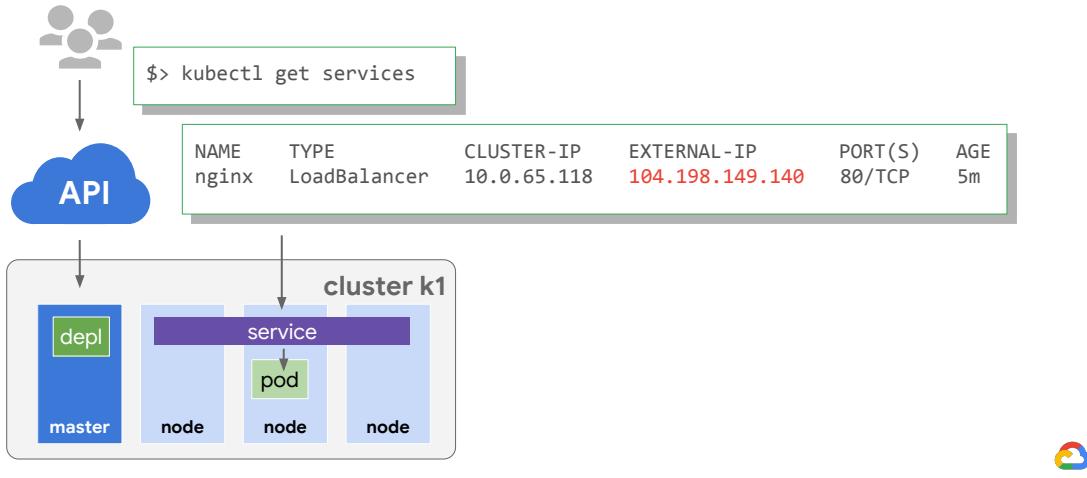
A **Service** is an abstraction which defines a logical set of Pods and a policy by which to access them.

As Deployments create and destroy Pods, Pods get their own IP address. But those addresses don't remain stable over time.

A Service groups a set of Pods and provides a stable endpoint (or fixed IP) for them.

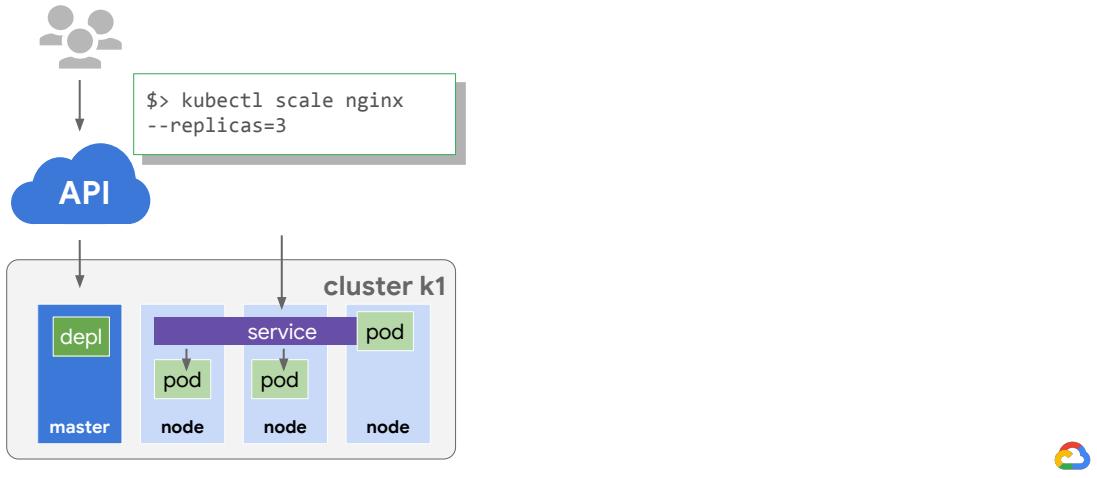
For example, if you create two sets of Pods called frontend and backend, and put them behind their own Services, backend Pods may change, but frontend Pods are not aware of this. They simply refer to the backend Service.

To get the Service's public IP run **kubectl get services**



You can run the **kubectl get services** command to get the public IP to hit the nginx container remotely.

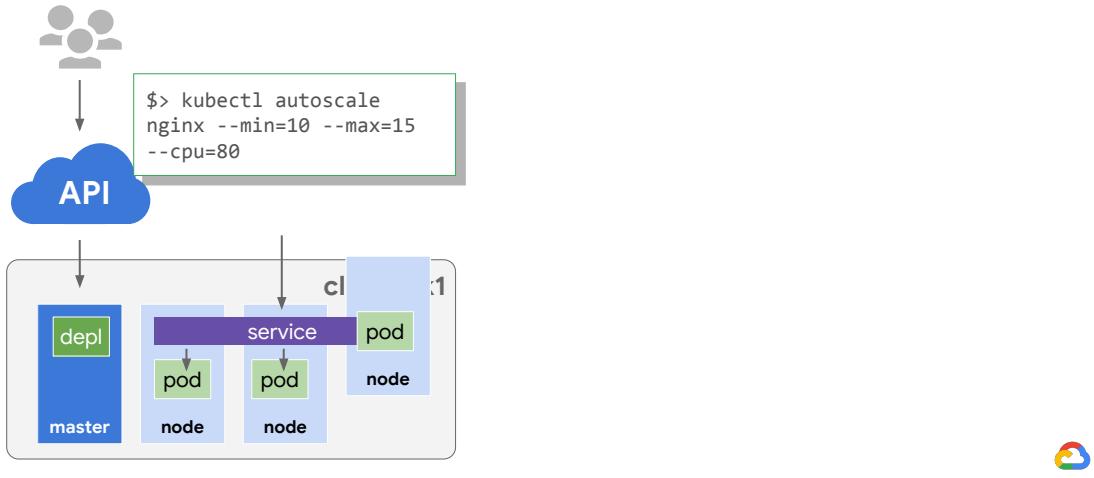
To scale a Deployment, run **kubectl scale**



To scale a Deployment, run the **kubectl scale** command.

In this case, three Pods are created in your Deployment and they're placed behind the Service and share one fixed IP.

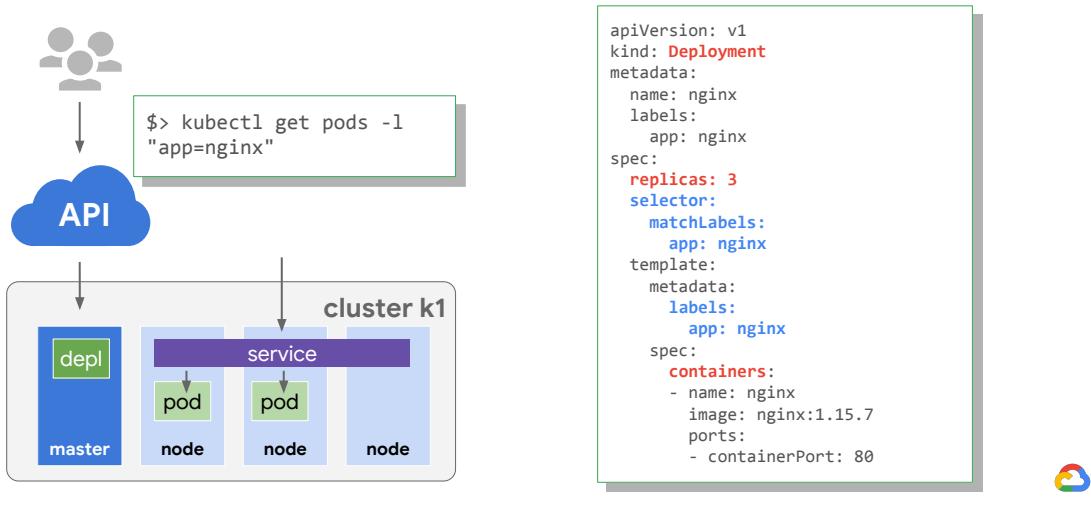
You can also run autoscaling with all kinds of parameters or put it behind programming logic for intelligent management



You could also use **autoscaling** with all kinds of parameters.

Here's an example of how to autoscale the Deployment to between 10 and 15 Pods when CPU utilization reaches 80 percent.

The real strength of Kubernetes comes when you work in a declarative way (here's how to get a config file)



So far, I've shown you how to run **imperative** commands like **expose** and **scale**. This works well to learn and test Kubernetes step-by-step.

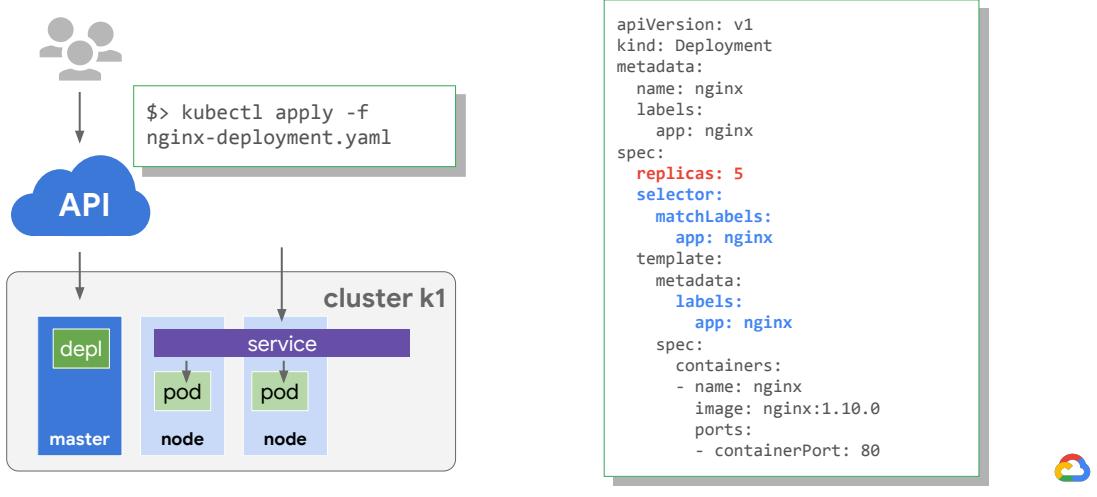
But the real strength of Kubernetes comes when you work in a **declarative** way.

Instead of issuing commands, you provide a configuration file that tells Kubernetes what you want your desired state to look like, and Kubernetes figures out how to do it.

Let me show you how to scale your Deployment using an existing Deployment config file.

To get the file, you can run a kubectl get pods command like the following.

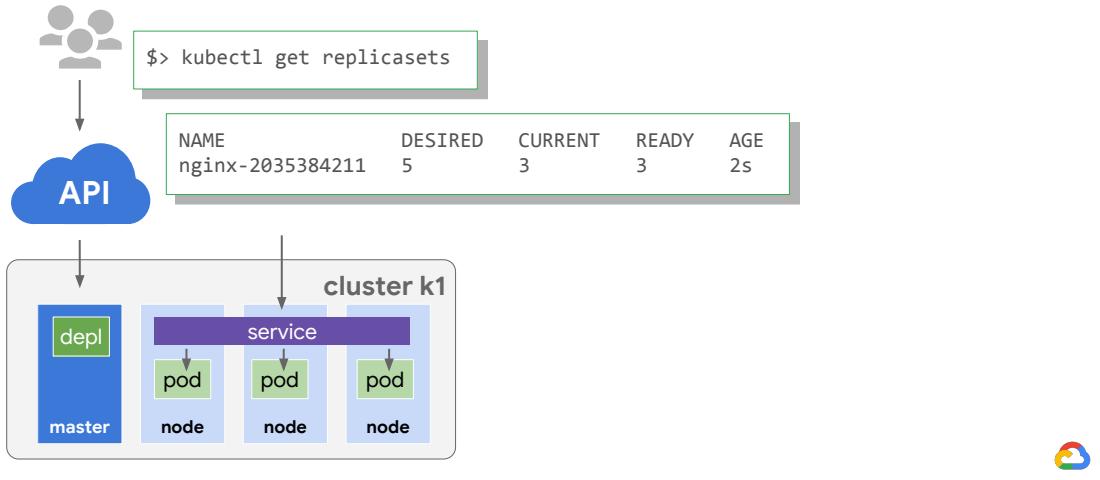
To declaratively apply changes run **kubectl apply -f**



To run five replicas instead of three, all you do is update the Deployment config file.

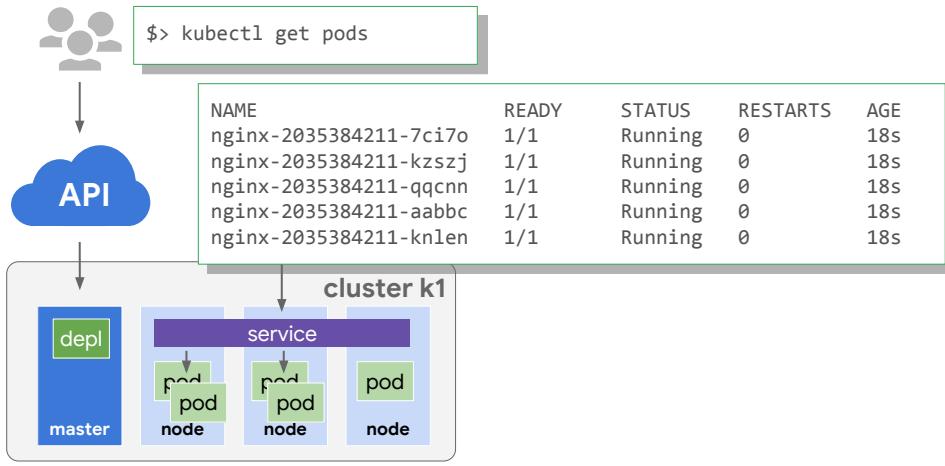
And run the **kubectl apply** command to use the config file.

Run `kubectl get replicsets` to see the updated state



Now look at your replicas to see their updated state.

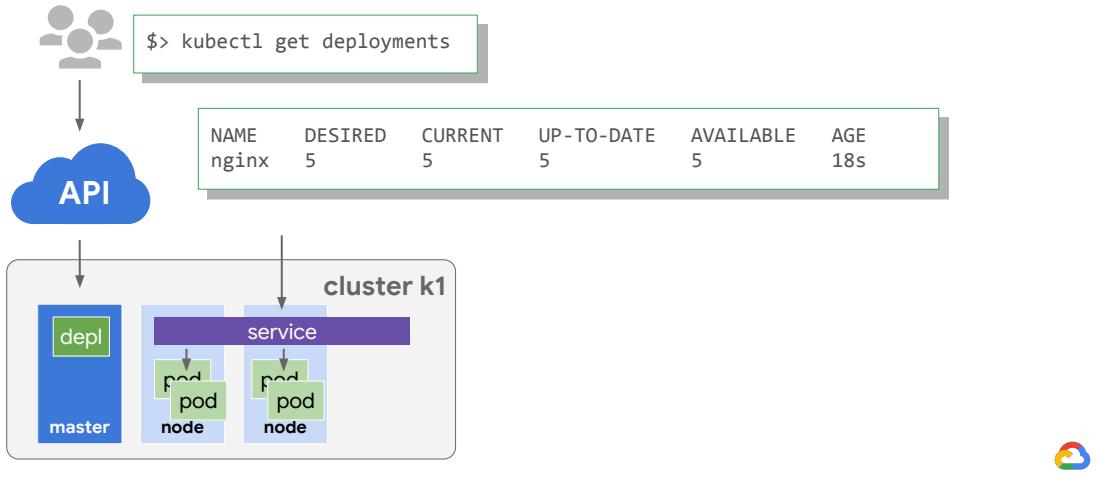
Run **kubectl get pods** to watch the Pods come on line



Then use the **kubectl get pods** command to watch the pods come on line.

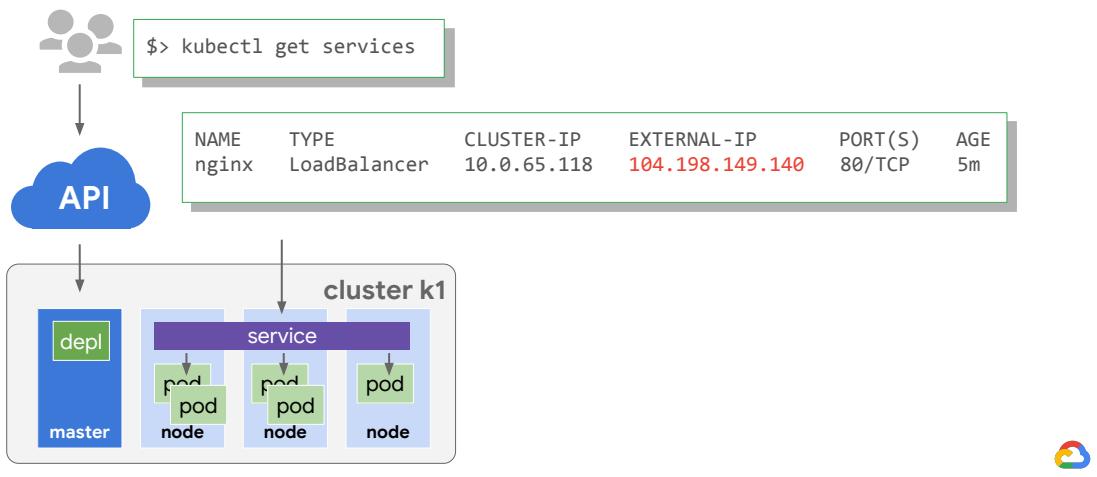
In this case, all five are **READY** and **RUNNING**.

Run `kubectl get deployments` or `describe deployments` to ensure the proper number of replicas are running



And check the Deployment to make sure the proper number of replicas are running using either `$ kubectl get deployments` or `$ kubectl describe deployments`. In this case, all five Pod replicas are **AVAILABLE**.

Then you build the container and run its image



And you can still hit your endpoint like before using `$ kubectl get services` to get the external IP of the Service, and hit the public IP from a client.

At this point, you have five copies of your nginx Pod running in GKE, and you have a single Service that's proxying the traffic to all five Pods. This allows you to share the load and scale your Service in Kubernetes.

To update to a new version of your app, you can use a variety of rollout strategies

```
spec:  
  # ...  
  replicas: 5  
  strategy:  
    rollingUpdate:  
      maxSurge: 1  
      maxUnavailable: 0  
    type: RollingUpdate  
  # ...
```



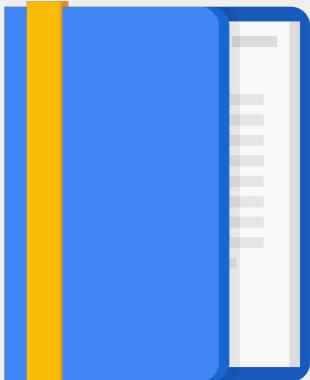
The last question is what happens when you want to update a new version of your app?

You want to update your container to get new code out in front of users, but it would be risky to roll out all those changes at once.

So you use **kubectl rollout** or change your deployment configuration file and apply the change using **kubectl apply**.

New Pods will be created according to your update strategy. Here is an example configuration that will create new version Pods one by one, and wait for a new Pod to be available before destroying one of the old Pods.

## Agenda



- Introduction to Containers
- Kubernetes and Kubernetes Engine
- **Quiz and Lab**



# Question #1

---

Name two reasons for deploying applications using containers.



# Question #1

---

Name two reasons for deploying applications using containers.

1. Portability across development, testing, and production environments
2. Simpler to migrate workloads
3. Loose coupling
4. Agility



## Question #2

---

True or False: Kubernetes lets you manage container clusters in multiple cloud providers.



# Question #2

---

True or False: Kubernetes lets you manage container clusters in multiple cloud providers.

True: But Kubernetes Engine only runs in GCP



## Question #3

---

True or False: GCP provides a private, high-speed container image storage service for use with Kubernetes Engine.



# Question #3

---

True or False: GCP provides a private, high-speed container image storage service for use with Kubernetes Engine.

True: It's called Google Container Registry



# Lab

---

In this lab you will create a Kubernetes Engine cluster containing several containers, each containing a web server. You'll place a load balancer in front of the cluster and view its contents.

## Objectives

- Provision a Kubernetes cluster using Kubernetes Engine
- Deploy and manage Docker containers using kubectl



# More resources

---

Kubernetes Engine <https://cloud.google.com/kubernetes-engine/docs/>

Kubernetes <http://kubernetes.io/>

Google Cloud Build <https://cloud.google.com/cloud-build/docs/>

Google Container Registry <https://cloud.google.com/container-registry/docs/>





# Google Cloud

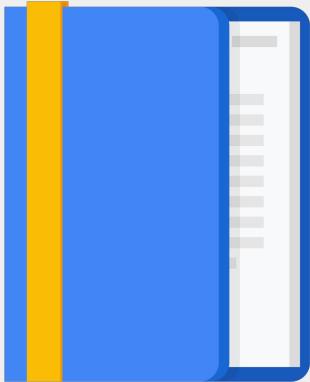
---

GCP Fundamentals: Core Infrastructure

## **Applications in the Cloud**

Cloud Identity and Access Management (Cloud IAM) and firewall rules are some of the ways to control access to your VPC network.

## Agenda



- **Google App Engine**
- Google App Engine Standard Environment
- Google App Engine Flexible Environment
- Google Cloud Endpoints and Apigee Edge
- Quiz and Lab



## App Engine is a PaaS for building scalable applications

- App Engine makes deployment, maintenance, and scalability easy so you can focus on innovation
- Especially suited for building scalable web applications and mobile backends



App Engine is a platform for building scalable web applications and mobile backends. It allows you to concentrate on innovating your applications by managing the application infrastructure for you. For example, App Engine manages the hardware and networking infrastructure required to run your code.

App Engine provides you with built-in services and APIs such as NoSQL datastores, memcache, load balancing, health checks, application logging, and a user authentication API, common to most applications.

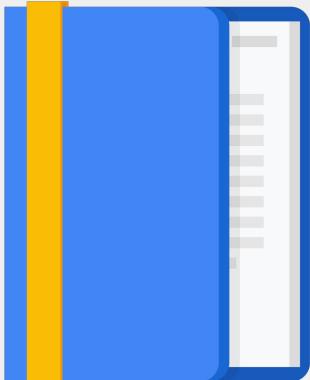
App Engine will scale your application automatically in response to the amount of traffic it receives so you only pay for the resources you use. Just upload your code and Google will manage your app's availability. There are no servers for you to provision or maintain.

Security Scanner automatically scans and detects common web application vulnerabilities. It enables early threat identification and delivers very low false-positive rates. You can easily set up, run, schedule, and manage security scans from the Google Cloud Platform Console.

App Engine works with popular development tools such as Eclipse, IntelliJ, Maven, Git, Jenkins, and PyCharm. You can build your apps with the tools you

love without changing your workflow.

## Agenda



- Google App Engine
- **Google App Engine Standard Environment**
- Google App Engine Flexible Environment
- Google Cloud Endpoints and Apigee Edge
- Quiz and Lab



## App Engine standard environment

- Easily deploy your applications
- Autoscale workloads to meet demand
- Economical
  - Free daily quota
  - Usage based [pricing](#)
- SDKs for development, testing and deployment



The App Engine standard environment is based on container instances running on Google's infrastructure. Containers are preconfigured with one of several available runtimes (Java 7, Python 2.7, Go and PHP). Each runtime also includes libraries that support App Engine standard APIs. For many applications, the standard environment runtimes and libraries may be all you need.

The App Engine standard environment makes it easy to build and deploy an application that runs reliably even under heavy load and with large amounts of data. It includes the following features:

Persistent storage with queries, sorting, and transactions

Automatic scaling and load balancing

Asynchronous task queues for performing work outside the scope of a request

Scheduled tasks for triggering events at specified times or regular intervals

Integration with other Google cloud services and APIs

## App Engine standard environment: Requirements

- Specific versions of Java, Python, PHP, and Go are supported
- Your application must conform to sandbox constraints:
  - No writing to local file system
  - All requests time out at 60 seconds
  - Third-party software installations are limited



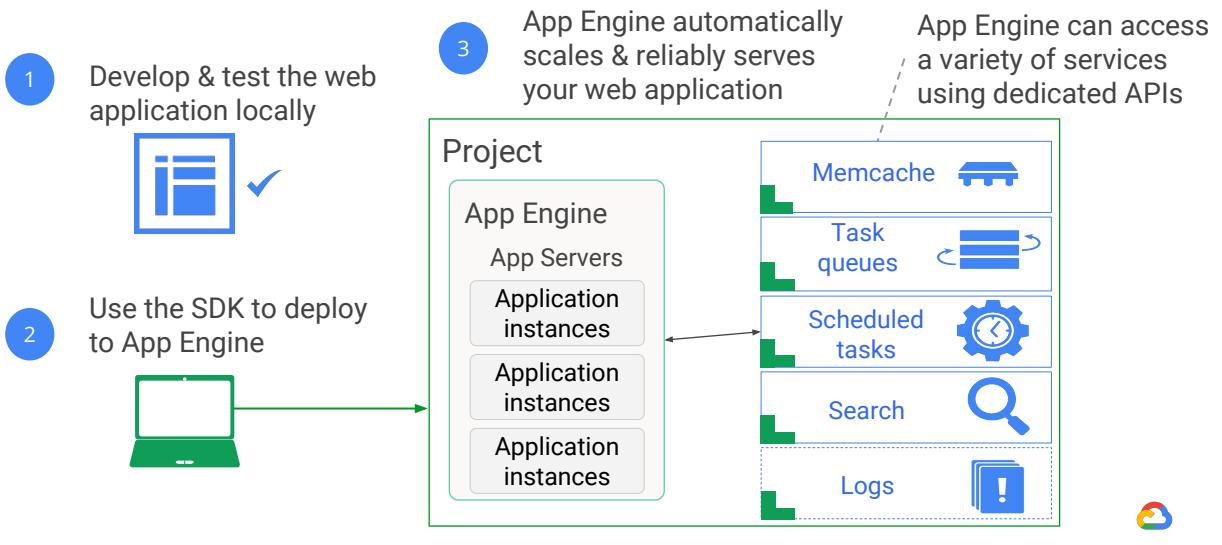
Software Development Kits (SDKs) for App Engine are available in all supported languages. Each SDK includes:

- All of the APIs and libraries available to App Engine
- A simulated, secure sandbox environment that emulates all of the App Engine services on your local computer
- Deployment tools that allow you to upload your application to the cloud and manage different versions of your application

The SDK manages your application locally, and the Google Cloud Platform Console manages your application in production. The Google Cloud Platform Console uses a web-based interface to create new applications, configure domain names, change which version of your application is live, examine access and error logs, and much more.

Applications run in a secure, sandboxed environment, allowing the App Engine standard environment to distribute requests across multiple servers, and scaling servers to meet traffic demands. Your application runs within its own secure, reliable environment that is independent of the hardware, operating system, or physical location of the server.

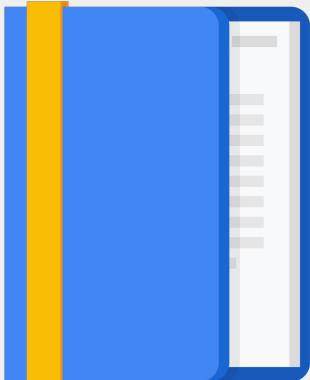
## Example App Engine standard workflow: Web applications



In this diagram we see App Engine Standard Environment in practice. You'll develop your application and run a test version of it locally using the App Engine SDK. Then, when you're ready, you'll use the SDK to deploy it.

Each App Engine application runs in a GCP project. App Engine automatically provisions server instances and scales and load-balances them. Meanwhile, your application can make calls to a variety of services using dedicated APIs. For example, a NoSQL datastore to make data persistent; caching of that data using memcache; searching; logging; user login; and the ability to launch actions not triggered by direct user requests, like task queues and a task scheduler.

## Agenda



- Google App Engine
- Google App Engine Standard Environment
- **Google App Engine Flexible Environment**
- Google Cloud Endpoints and Apigee Edge
- Quiz and Lab



## App Engine flexible environment

- Build and deploy containerized apps with a click
- No sandbox constraints
- Can access App Engine resources
- Standard runtimes: Python, Java, Go, Node.js
- Custom runtime support: Any language that supports HTTP requests
- Package your runtime as a Dockerfile



If the restrictions of App Engine Standard Environment's sandbox model don't work for you, but you still want to take advantage of the benefits of App Engine (like automatic scaling up and down), consider App Engine Flexible Environment. Instead of the sandbox, App Engine Flexible Environment lets you specify the container your application runs in.

Your application runs inside Docker containers on Google Compute Engine virtual machines (VMs). App Engine manages these Compute Engine machines for you. They're health-checked, healed as necessary, and you get to choose what geographical region they run in. And critical, backward-compatible updates to their operating systems are automatically applied. All this so that you can just focus on your code.

Microservices, authorization, SQL and noSQL databases, traffic splitting, logging, search, versioning, security scanning, memcache, and content delivery networks are all supported natively. In addition, the App Engine flexible environment allows you to customize your runtime and even the operating system of your virtual machine using Dockerfiles.

- *Runtimes:* The flexible environment includes native support for Java 8/Servlet 3.1/Jetty 9, Python 2.7 and Python 3.4, Node.js, and Go. Developers can customize these runtimes or provide their own runtime, such as Ruby or PHP, by supplying a custom Docker image or

- Dockerfile from the open source community.
- *Infrastructure customization*: Because VM instances in the flexible environment are Compute Engine virtual machines, you can use SSH to connect to every single VM and Docker container for debugging purposes and further customization.
- *Performance*: Take advantage of a wide array of CPU and memory configurations. You can specify how much CPU and memory each instance of your application needs, and the flexible environment will provision the necessary infrastructure for you.

App Engine manages your virtual machines, ensuring that:

- Instances are health-checked, healed as necessary, and co-located with other module instances within the project.
- Critical, backward-compatible updates are automatically applied to the underlying operating system.
- VM instances are automatically located by geographical region according to the settings in your project. Google's management services ensure that all of a project's VM instances are co-located for optimal performance.
- VM instances are restarted on a weekly basis. During restarts, Google's management services will apply any necessary operating system and security updates.

App Engine flexible environment apps that use standard runtimes can access App Engine services: Datastore, Memcache, task queues, logging, users, and so on.

## Comparing the App Engine environments

	Standard Environment	Flexible Environment
Instance startup	Milliseconds	Minutes
SSH access	No	Yes (although not by default)
Write to local disk	No	Yes (but writes are ephemeral)
Support for 3rd-party binaries	No	Yes
Network access	Via App Engine services	Yes
Pricing model	After free daily use, pay per instance class, with automatic shutdown	Pay for resource allocation per hour; no automatic shutdown



Here's a side-by-side comparison of Standard and Flexible. Notice that Standard Environment starts up instances of your application faster, but that you get less access to the infrastructure in which your application runs. For example, Flexible Environment lets you ssh into the virtual machines on which your application runs; it lets you use local disk for scratch space; it lets you install third-party software; and it lets your application make calls to the network without going through App Engine. On the other hand, Standard Environment's billing can drop to zero for a completely idle application.

## Deploying Apps: Kubernetes Engine vs App Engine

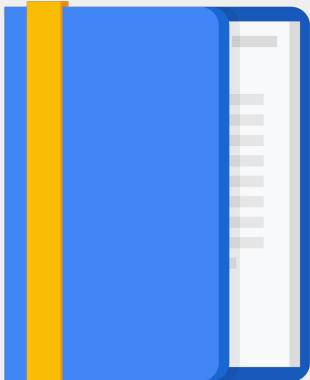
	Kubernetes Engine	App Engine Flexible	App Engine Standard
Language support	Any	Any	Java, Python, Go, PHP
Service model	Hybrid	PaaS	PaaS
Primary use case	Container-based workloads	Web and mobile applications, container-based workloads	Web and mobile applications

 Toward managed infrastructure      Toward dynamic infrastructure 

Because we mentioned App Engine's use of Docker containers, you may be wondering how App Engine compares to Kubernetes Engine. Here's a side-by-side comparison of App Engine with Kubernetes Engine. App Engine Standard Environment is for people who want the service to take maximum control of their application's deployment and scaling. Kubernetes Engine gives the application owner the full flexibility of Kubernetes. App Engine Flexible Edition is in between.

Also, App Engine Environment treats containers as a means to an end. But for Kubernetes Engine, containers are a fundamental organizing principle.

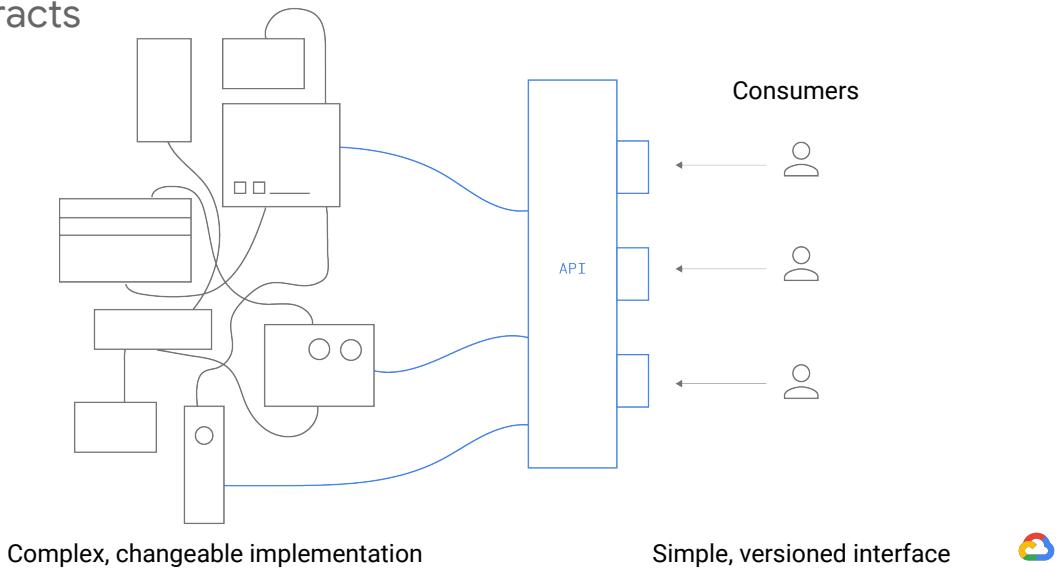
## Agenda



- Google App Engine
- Google App Engine Standard Environment
- Google App Engine Flexible Environment
- **Google Cloud Endpoints and Apigee Edge**
- Quiz and Lab



**Application Programming Interfaces** hide detail, enforce contracts



Let's be precise about what an API is. A software service's implementation can be complex and changeable. If other software services had to be explicitly coded all that detail in order to use that service, the result would be brittle and error-prone. So instead, application developers structure the software they write so that it presents a clean, well-defined interface that abstracts away needless detail, and then they document that interface. That's an Application Programming Interface. The underlying implementation can change, as long as the interface doesn't, and other pieces of software that use the API don't have to know or care.

Sometimes you do have to change an API, such as to add or deprecate a feature. To make this kind of API change cleanly, developers version their APIs. Version 2 of an API might contain calls that version 1 does not; programs that consume the API can specify the API version they want to use in their calls.

Supporting an API is a very important task, and Google Cloud Platform provides two API management tools. They approach related problems in a different way, and each has a particular forte.

Cloud Endpoints helps you create and maintain APIs

- Distributed API management through an API console
- Expose your API using a RESTful interface
- Control access and validate calls with JSON Web Tokens and Google API keys
  - Identify web, mobile users with Auth0 and Firebase Authentication
- Generate client libraries



Cloud Endpoints is a distributed API management system. It provides an API console, hosting, logging, monitoring, and other features to help you create, share, maintain, and secure your APIs. You can use Cloud Endpoints with any APIs that support the OpenAPI Specification, formerly known as the Swagger spec.

Cloud Endpoints uses the distributed Extensible Service Proxy to provide low latency and high performance for serving even the most demanding APIs. Extensible Service Proxy is a service proxy based on NGINX. It runs in its own Docker container for better isolation and scalability. The proxy is containerized and distributed in the Container Registry and Docker registry, and can be used with App Engine, Kubernetes Engine, Compute Engine or Kubernetes.

## Cloud Endpoints features

### User authentication

- JSON Web Token validation and a streamlined developer experience for Firebase Auth, Google Auth and Auth0.

### Automated deployment

- With App Engine, the proxy is deployed automatically with your application. On Kubernetes Engine or Compute Engine, use Google's containerized ESP for simple deployment.

### Logging and monitoring

- Monitor traffic, error rates and latency, and review logs in Cloud Logging. Use Cloud Trace to dive into performance and BigQuery for analysis.

#### *API keys*

- Generate API keys in Google Cloud Platform Console and validate on every API call. Share your API with other developers to allow them to generate their own keys.

#### *Easy integration*

- Get started quickly by using one of Google's Cloud Endpoints Frameworks or by simply adding an Open API specification to your deployment.

## Cloud Endpoints: Supported platforms

Runtime environment
App Engine Flexible Environment
Kubernetes Engine
Compute Engine

Clients
Android
iOS
Javascript



Cloud Endpoints supports applications running in GCP's compute platforms, in your choice of languages, and your choice of client technologies.

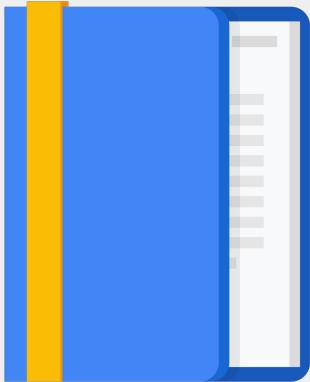
## Apigee Edge helps you secure and monetize APIs

- A platform for making APIs available to your customers and partners
- Contains analytics, monetization, and a developer portal



Apigee Edge is also a platform for developing and managing API proxies. It has a different orientation, though: it has a focus on business problems like rate limiting, quotas, and analytics. Many users of Apigee Edge are providing a software service to other companies, and those features come in handy. Because the backend services for Apigee Edge need not be in GCP, engineers also often use it when they are working to take a legacy application apart. Instead of replacing a monolithic application in one risky move, they can instead use Apigee Edge to peel off its services one by one, standing up microservices to implement each in turn, until the legacy application can finally be retired.

## Agenda



- Google App Engine
- Google App Engine Standard Environment
- Google App Engine Flexible Environment
- Google Cloud Endpoints and Apigee Edge
- **Quiz and Lab**



# Question #1

---

Name 3 advantages of using the App Engine flexible environment over App Engine standard.



# Question #1

---

Name 3 advantages of using the App Engine flexible environment over App Engine standard.

The flexible environment allows SSH access, allows disk writes, and supports third-party binaries (also allows stack customization and background processes)



# Question #2

---

What is the difference between Cloud Endpoints and Apigee Edge?



# Question #2

---

What is the difference between Cloud Endpoints and Apigee Edge?

Cloud Endpoints helps you create and maintain APIs; Apigee Edge helps you secure and monetize APIs



# Lab

---

In this lab you will create a simple App Engine application using the Cloud Shell local development environment and then deploy it to App Engine.

## Objectives

- Preview an App Engine application using Cloud Shell
- Launch an App Engine application
- Disable an App Engine application



# More resources

---

Google App Engine <https://cloud.google.com/appengine/docs/>

Google App Engine Flexible Environment <https://cloud.google.com/appengine/docs/flexible/>

Google App Engine Standard Environment

<https://cloud.google.com/appengine/docs/standard/>

Google Cloud Endpoints <https://cloud.google.com/endpoints/docs/>

Apigee Edge <http://docs.apigee.com/api-services/content/what-apigee-edge>





# Google Cloud

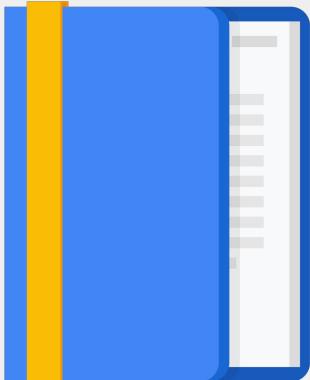
---

GCP Fundamentals: Core Infrastructure

## **Developing, Deploying, and Monitoring in the Cloud**

Cloud Identity and Access Management (Cloud IAM) and firewall rules are some of the ways to control access to your VPC network.

## Agenda



- **Development in the cloud**
- Deployment: Infrastructure as code
- Monitoring: Proactive instrumentation
- Lab



## Cloud Source Repositories

- Fully featured Git repositories hosted on Google Cloud Platform
- Supports collaborative development of cloud apps
- Includes integration with Stackdriver Debugger



Cloud Source Repositories provides Git version control to support collaborative development of any application or service, including those that run on App Engine and Compute Engine. If you are using the Stackdriver Debugger, you can use Cloud Source Repositories and related tools to view debugging information alongside your code during application runtime. Cloud Source Repositories also provides a source viewer that you can use to browse and view repository files from within the Google Cloud Platform Console.

With Cloud Source Repositories, you can have any number of private Git repositories, which allows you to organize the code associated with your cloud project in whatever way works best for you. Google Cloud diagnostics tools like the Debugger and Error Reporting can use the code from your Git repositories to let you track down issues to specific errors in your deployed code without slowing down your users. If you've already got your code in GitHub or BitBucket repositories, you can bring that into your cloud project and use it just like any other repository, including browsing and diagnostics.

## Cloud Functions

- Create single-purpose functions that respond to events without a server or runtime
  - Event examples: New instance created, file added to Cloud Storage
- Written in Javascript; execute in managed Node.js environment on Google Cloud Platform



Cloud Functions is a lightweight, event-based, asynchronous compute solution that allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment. You can use these functions to construct applications from bite-sized business logic. You can also use Cloud Functions to connect and extend cloud services.

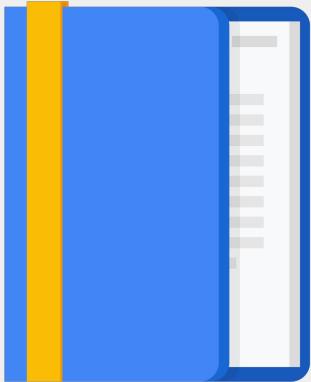
You are billed, to the nearest 100 milliseconds, only while your code is running.

Cloud Functions are written in Javascript and execute in a managed Node.js environment on Google Cloud Platform. Events from Cloud Storage and Cloud Pub/Sub can trigger Cloud Functions asynchronously, or you can use HTTP invocation for synchronous execution.

Cloud Events are things that happen in your cloud environment. These might be things like changes to data in a database, files added to a storage system, or a new virtual machine instance being created.

Events occur whether or not you choose to respond to them. Creating a response to an event is done with a trigger. A trigger is a declaration that you are interested in a certain event or set of events. You create triggers to capture events and act on them.

## Agenda



- Development in the cloud
- **Deployment: Infrastructure as code**
- Monitoring: Proactive instrumentation
- Lab



## Deployment Manager

- Infrastructure management service
- Create a .yaml template describing your environment and use Deployment Manager to create resources
- Provides repeatable deployments



Deployment Manager is an infrastructure management service that automates the creation and management of your Google Cloud Platform resources for you.

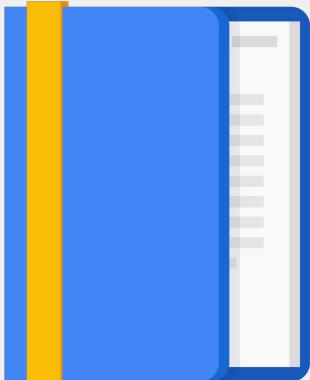
Setting up your environment in GCP can entail many steps: setting up compute, network, and storage resources and keeping track of their configurations. You can do it all by hand if you want to, taking an imperative approach. But it is more efficient to use a template. That means a specification of what the environment should look like, declarative rather than imperative.

GCP provides Deployment Manager to let do just that. It's an infrastructure management service that automates the creation and management of your Google Cloud Platform resources for you.

To use Deployment Manager, you create a template file, using either the YAML markup language or Python, that describes what you want the components of your environment to look like. Then you give the template to Deployment Manager, which figures out and does the actions needed to create the environment your template describes. If you need to change your environment, edit your template, and then tell Deployment Manager to update the environment to match the change.

Here's a tip: you can store and version-control your Deployment Manager templates in Cloud Source Repositories.

## Agenda



- Development in the cloud
- Deployment: Infrastructure as code
- **Monitoring: Proactive instrumentation**
- Lab





# Stackdriver



Monitoring



Logging



Debug



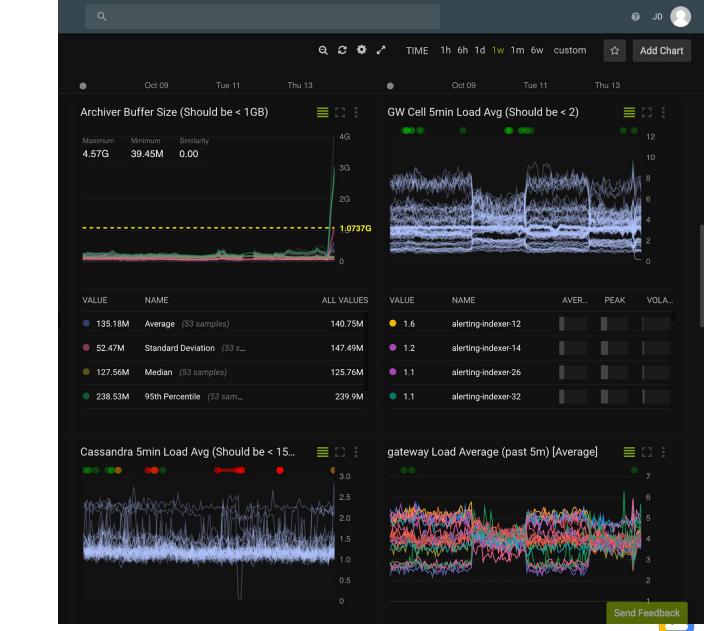
Error Reporting



Trace



Profiler



You can't run an application stably without monitoring. Monitoring lets you figure out whether the changes you made were good or bad. It lets you respond with information rather than with panic when one of your end users complains that your application is down.

Stackdriver is GCP's tool for monitoring, logging, and diagnostics. Stackdriver gives you access to many different kinds of signals from your infrastructure platforms, virtual machines, containers, middleware, and application tier: logs, metrics, traces. It gives you insight into your application's health, performance, and availability, so if issues occur you can fix them faster.

## Stackdriver offers capabilities in six areas

### Monitoring



Platform, system, and application metrics

Uptime/health checks

Dashboards and alerts

### Error Reporting



Error notifications

Error dashboard

### Logging

Platform, system, and application logs

Log search, view, filter, and export

Log-based metrics

### Debugger

Debug applications



### Trace

Latency reporting and sampling

Per-URL latency and statistics



### Profiler Beta



Continuous profiling of CPU and memory consumption



Here are the core components of Google Stackdriver: monitoring, logging, trace, error reporting, and debugging.

Stackdriver Monitoring checks the endpoints of web applications and other internet-accessible services running on your cloud environment. You can configure uptime checks associated with URLs, groups, or resources, such as instances and load balancers. You can set up alerts on interesting criteria, like when health check results or uptimes fall into levels that need action. You can use Monitoring with a lot of popular notification tools. And you can create dashboards to help you visualize the state of your application.

Stackdriver Logging lets you view logs from your applications, and filter and search on them. Logging also lets you define metrics based on log contents that are incorporated into dashboards and alerts. You can also export logs to BigQuery, Cloud Storage, and Cloud Pub/Sub.

Stackdriver Error Reporting tracks and groups the errors in your cloud applications , and it notifies you when new errors are detected.

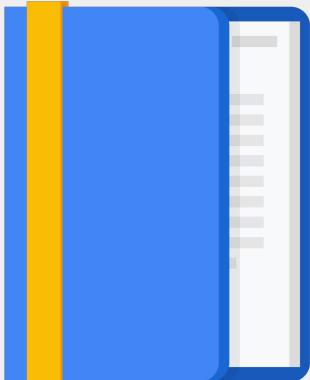
With Stackdriver Trace, you can sample the latency of App Engine applications and report per-URL statistics.

A painful way to debug an existing application is to go back into it and add lots of logging statements. Stackdriver Debugger offers a different way. It connects your

application's production data to your source code, so you can inspect its state of your application at any code location in production. That means you can view the application state without adding logging statements. Stackdriver Debugger works best when your application's source code is available, such as in Cloud Source Repositories, although it can be in other repositories too.

To profile an application is to look at its execution of a program and observe the call patterns between functions, how much CPU time each function consumes, and how much memory is allocated for each. Stackdriver Profiler is a statistical, low-overhead profiler that continuously gathers CPU usage and memory-allocation information from your production applications. It attributes that information to the application's source code, helping you identify the parts of the application consuming the most resources, and otherwise illuminating the performance characteristics of the code. At the time of this writing, Stackdriver Profiler is in beta. This means that it is not covered by any SLA or deprecation policy and may be subject to backward-incompatible changes.

## Agenda



- Development in the cloud
- Deployment: Infrastructure as code
- Monitoring: Proactive instrumentation
- **Lab**



# Lab

---

In this lab you will create a deployment using Deployment Manager and use it to maintain a consistent state of your deployment. You will also view resource usage in a VM instance using Stackdriver.

## Objectives

- Create a Deployment Manager deployment.
- Update a Deployment Manager deployment.
- View the load on a VM instance using Google Stackdriver.



# More resources

---

Cloud Source Repositories <https://cloud.google.com/source-repositories/docs/>

Deployment Manager <https://cloud.google.com/deployment-manager/docs/>

Google Stackdriver <https://cloud.google.com/stackdriver/docs/>





# Google Cloud

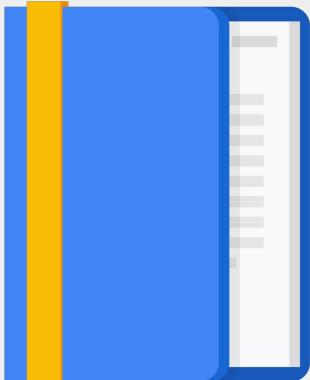
---

GCP Fundamentals: Core Infrastructure

## **Big Data and Machine Learning in the Cloud**

Cloud Identity and Access Management (Cloud IAM) and firewall rules are some of the ways to control access to your VPC network.

## Agenda



- **Google Cloud Big Data Platform**
- Google Cloud Machine Learning Platform
- Quiz and Lab



## Google Cloud's big data services are fully managed and scalable



Cloud  
Dataproc

Managed  
Hadoop  
MapReduce,  
Spark, Pig, and  
Hive service



Cloud  
Dataflow

Stream and  
batch  
processing;  
unified and  
simplified  
pipelines



BigQuery

Analytics  
database;  
stream data at  
100,000  
rows per second



Cloud  
Pub/Sub

Scalable and  
flexible  
enterprise  
messaging



Cloud  
Datalab

Interactive data  
exploration



Google Cloud Big Data solutions are designed to help you transform your business and user experiences with meaningful data insights. It is an integrated, serverless platform. “Serverless” means you don’t have to provision compute instances to run your jobs. The services are fully managed, and you pay only for the resources you consume. The platform is “integrated” so GCP data services work together to help you create custom solutions.

## Cloud Dataproc is managed Hadoop

- Fast, easy, managed way to run Hadoop and Spark/Hive/Pig on GCP
- Create clusters in 90 seconds or less on average.
- Scale clusters up and down even when jobs are running.



Apache Hadoop is an open-source framework for big data. It is based on the MapReduce programming model, which Google invented and published. The MapReduce model, at its simplest, means that one function -- traditionally called the “map” function -- runs in parallel across a massive dataset to produce intermediate results; and another function -- traditionally called the “reduce” function -- builds a final result set based on all those intermediate results. The term “Hadoop” is often used informally to encompass Apache Hadoop itself and related projects, such as Apache Spark, Apache Pig, and Apache Hive.

Cloud Dataproc is a fast, easy, managed way to run Hadoop, Spark, Hive, and Pig on Google Cloud Platform. All you have to do is to request a Hadoop cluster. It will be built for you in 90 seconds or less, on top of Compute Engine virtual machines whose number and type you can control. If you need more or less processing power while your cluster’s running, you can scale it up or down. You can use the default configuration for the Hadoop software in your cluster, or you can customize it. And you can monitor your cluster using Stackdriver.

## Why use Cloud Dataproc?

- Easily migrate on-premises Hadoop jobs to the cloud.
- Quickly analyze data (like log data) stored in Cloud Storage; create a cluster in 90 seconds or less on average, and then delete it immediately.
- Use Spark/Spark SQL to quickly perform data mining and analysis.
- Use Spark Machine Learning Libraries (MLlib) to run classification algorithms.



Running on-premises Hadoop jobs requires a hardware investment. On the other hand, running these jobs in Cloud Dataproc allows you to pay only for hardware resources during the life of the ephemeral customer you create. You can further save money using preemptible instances for batch processing.

You can also save money by telling Cloud Dataproc to use preemptible Compute Engine instances for your batch processing. You have to make sure that your jobs can be restarted cleanly if they're terminated and you get a significant break in the cost of the instances. At the time this video was made, preemptible instances were around 80% cheaper. Be aware that the cost of the Compute Engine instances isn't the only component of the cost of a Dataproc cluster, but it's a significant one.

Once your data is in a cluster, you can use Spark and Spark SQL to do data mining, and you can use MLlib, which is Apache Spark's Machine Learning Libraries, to discover patterns through machine learning.

Cloud **Dataflow** offers managed data pipelines

- Processes data using Compute Engine instances.
  - Clusters are sized for you
  - Automated scaling, no instance provisioning required
- Write code once and get batch and streaming.
  - Transform-based programming model



Cloud Dataproc is great when you have a dataset of known size, or when you want to manage your cluster size yourself. But what if your data shows up in realtime? Or it's of unpredictable size or rate? That's where Cloud Dataflow is a particularly good choice. It's both a unified programming model and a managed service, and it lets you develop and execute a big range of data processing patterns: extract-transform-and-load, batch computation, and continuous computation. You use Dataflow to build data pipelines, and the same pipelines work for both batch and streaming data.

Dataflow is a unified programming model and a managed service for developing and executing a wide range of data processing patterns including ETL, batch computation, and continuous computation. Cloud Dataflow frees you from operational tasks like resource management and performance optimization.

Cloud Dataflow features:

#### *Resource Management*

Cloud Dataflow fully automates management of required processing resources. No more spinning up instances by hand.

#### *On Demand*

All resources are provided on demand, enabling you to scale to meet your business needs. No need to buy reserved compute instances.

#### *Intelligent Work Scheduling*

Automated and optimized work partitioning which can dynamically rebalance lagging work. No more chasing down “hot keys” or pre-processing your input data.

#### *Auto Scaling*

Horizontal auto scaling of worker resources to meet optimum throughput requirements results in better overall price-to-performance.

#### *Unified Programming Model*

The Dataflow API enables you to express MapReduce like operations, powerful data windowing, and fine grained correctness control regardless of data source.

#### *Open Source*

Developers wishing to extend the Dataflow programming model can fork and or submit pull requests on the Java-based Cloud Dataflow SDK. Dataflow pipelines can also run on alternate runtimes like Spark and Flink.

#### *Monitoring*

Integrated into the Google Cloud Platform Console, Cloud Dataflow provides statistics such as pipeline throughput and lag, as well as consolidated worker log inspection—all in near-real time.

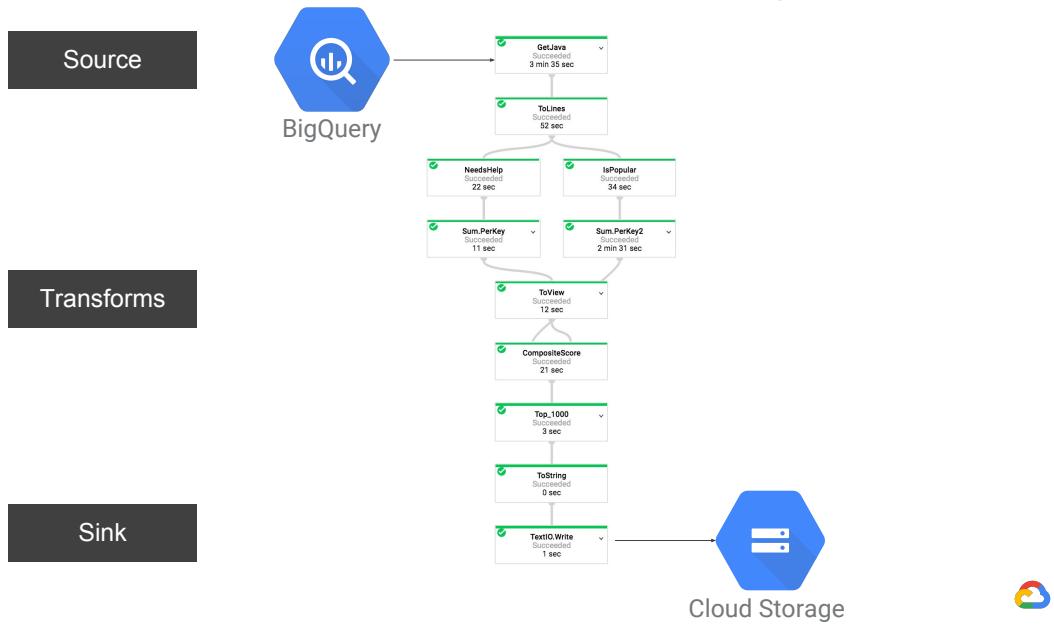
#### *Integrated*

Integrates with Cloud Storage, Cloud Pub/Sub, Cloud Datastore, Cloud Bigtable, and BigQuery for seamless data processing. And can be extended to interact with others sources and sinks like Apache Kafka and HDFS.

#### *Reliable & Consistent Processing*

Cloud Dataflow provides built-in support for fault-tolerant execution that is consistent and correct regardless of data size, cluster size, processing pattern or pipeline complexity.

## Dataflow pipelines flow data from a source through transforms



This example Dataflow pipeline reads data from a BigQuery table (the “source”), processes it in various ways (the “transforms”), and writes its output to Cloud Storage (the “sink”). Some of those transforms you see here are map operations, and some are reduce operations. You can build really expressive pipelines.

Each step in the pipeline is elastically scaled. There is no need to launch and manage a cluster. Instead, the service provides all resources on demand. It has automated and optimized work partitioning built in, which can dynamically rebalance lagging work. That reduces the need to worry about “hot keys” -- that is, situations where disproportionately large chunks of your input get mapped to the same cluster.

## Why use Cloud Dataflow?

- **ETL** (extract/transform/load) pipelines to move, filter, enrich, shape data
- **Data analysis**: batch computation or continuous computation using streaming
- **Orchestration**: create pipelines that coordinate services, including external services
- **Integrates with GCP services** like Cloud Storage, Cloud Pub/Sub, BigQuery, and Bigtable
  - Open source Java and Python SDKs



People use Dataflow in a variety of use cases. For one, it serves well as a general-purpose ETL tool.

And its use case as a data analysis engine comes in handy in things like these: fraud detection in financial services; IoT analytics in manufacturing, healthcare, and logistics; and clickstream, Point-of-Sale, and segmentation analysis in retail.

And, because those pipelines we saw can orchestrate multiple services, even external services, it can be used in realtime applications such as personalizing gaming user experiences.

**BigQuery** is a fully managed data warehouse

- Provides near real-time interactive analysis of massive datasets (hundreds of TBs)
- Query using SQL syntax (SQL 2011)
- No cluster maintenance is required.



If, instead of a dynamic pipeline, you want to do ad-hoc SQL queries on a massive dataset, that is what BigQuery is for. BigQuery is Google's fully managed, petabyte scale, low cost analytics data warehouse.

BigQuery is Google's fully managed, petabyte scale, low cost analytics data warehouse. BigQuery is NoOps: there is no infrastructure to manage and you don't need a database administrator, so you can focus on analyzing data to find meaningful insights, use familiar SQL, and take advantage of our pay-as-you-go model. BigQuery is a powerful big data analytics platform used by all types of organizations, from startups to Fortune 500 companies.

BigQuery's features:

#### *Flexible Data Ingestion*

Load your data from Cloud Storage or Cloud Datastore, or stream it into BigQuery at 100,000 rows per second to enable real-time analysis of your data.

#### *Global Availability*

You have the option to store your BigQuery data in European locations while continuing to benefit from a fully managed service, now with the option of geographic data control, without low-level cluster maintenance.

### *Security and Permissions*

You have full control over who has access to the data stored in BigQuery. If you share datasets, doing so will not impact your cost or performance; those you share with pay for their own queries.

### *Cost Controls*

BigQuery provides cost control mechanisms that enable you to cap your daily costs at an amount that you choose. For more information, see [Cost Controls](#).

### *Highly Available*

Transparent data replication in multiple geographies means that your data is available and durable even in the case of extreme failure modes.

### *Super Fast Performance*

Run super-fast SQL queries against multiple terabytes of data in seconds, using the processing power of Google's infrastructure.

### *Fully Integrated*

In addition to SQL queries, you can easily read and write data in BigQuery via Cloud Dataflow, Spark, and Hadoop.

### *Connect with Google Products*

You can automatically export your data from Google Analytics Premium into BigQuery and analyze datasets stored in Google Cloud Storage, Google Drive, and Google Sheets.

BigQuery can make Create, Replace, Update, and Delete changes to databases, subject to [some limitations](#) and with certain [known issues](#).

**BigQuery** runs on Google's high-performance infrastructure

- Compute and storage are separated with a terabit network in between
- You only pay for storage and processing used
- Automatic discount for long-term data storage



It's easy to get data into BigQuery. You can load from Cloud Storage or Cloud Datastore, or stream it into BigQuery at up to 100,000 rows per second.

BigQuery is used by all types of organizations, from startups to Fortune 500 companies. Smaller organizations like BigQuery's free monthly quotas. Bigger organizations like its seamless scale and its available 99.9% service level agreement.

Long term storage pricing is an automatic discount for data residing in BigQuery for extended periods of time. When the age of your data reaches 90 days in BigQuery, Google will automatically drop the price of storage from \$0.02 per GB per month down to \$0.01 per GB per month.

For more information on the architecture of BigQuery, see:  
<https://cloud.google.com/blog/big-data/2016/01/bigquery-under-the-hood>

## Cloud Pub/Sub is scalable, reliable messaging

- Supports many-to-many asynchronous messaging
  - Application components make push/pull subscriptions to topics
- Includes support for offline consumers
- Based on proven Google technologies
- Integrates with Cloud Dataflow for data processing pipelines



Cloud Pub/Sub is a fully managed real-time messaging service that allows you to send and receive messages between independent applications. You can leverage Cloud Pub/Sub's flexibility to decouple systems and components hosted on Google Cloud Platform or elsewhere on the internet. By building on the same technology Google uses, Cloud Pub/Sub is designed to provide "at least once" delivery at low latency with on-demand scalability to 1 million messages per second (and beyond).

Cloud Pub/Sub features:

### *Highly Scalable*

Any customer can send up to 10,000 messages per second, by default—and millions per second and beyond, upon request.

### *Push and Pull Delivery*

Subscribers have flexible delivery options, whether they are accessible from the internet or behind a firewall.

### *Encryption*

Encryption of all message data on the wire and at rest provides data security and protection.

### *Replicated Storage*

Designed to provide “at least once” message delivery by storing every message on multiple servers in multiple zones.

### *Message Queue*

Build a highly scalable queue of messages using a single topic and subscription to support a one-to-one communication pattern.

### *End-to-End Acknowledgement*

Building reliable applications is easier with explicit application-level acknowledgements.

### *Fan-out*

Publish messages to a topic once, and multiple subscribers receive copies to support one-to-many or many-to-many communication patterns.

### *REST API*

Simple, stateless interface using JSON messages with API libraries in many programming languages.

## Why use Cloud Pub/Sub?

- Building block for data ingestion in Dataflow, Internet of Things (IoT), Marketing Analytics
- Foundation for Dataflow streaming
- Push notifications for cloud-based applications
- Connect applications across Google Cloud Platform (push/pull between Compute Engine and App Engine)



Cloud Pub/Sub builds on the same technology Google uses internally. It's an important building block for applications where data arrives at high and unpredictable rates, like Internet of Things systems. If you're analyzing streaming data, Cloud Dataflow is a natural pairing with Pub/Sub.

Cloud Datalab offers interactive data exploration

- Interactive tool for large-scale data exploration, transformation, analysis, and visualization
- Integrated, open source
  - Built on Jupyter (formerly IPython)



For data science, an online lab notebook metaphor is a useful environment, because it feels natural to intersperse data analyses with comments about their results. A popular open-source system for hosting those is Project Jupyter. It lets you create and maintain web-based notebooks containing Python code, and you can run that code interactively and view the results.

Cloud Datalab lets you use Jupyter notebooks to explore, analyze, and visualize data on the Google Cloud Platform. It runs in a Compute Engine virtual machine. To get started, you specify the virtual machine type you want and what GCP region it should run in. When it launches, it presents an interactive Python environment that's ready to use. And it orchestrates multiple GCP services automatically, so you can focus on exploring your data. You only pay for the resources you use; there's no additional charge for Datalab itself.

Cloud Datalab features:

#### *Integrated*

Cloud Datalab handles authentication and cloud computation out of the box and is integrated with BigQuery, Compute Engine, and Cloud Storage.

#### *Multi-Language Support*

Cloud Datalab currently supports Python, SQL, and JavaScript (for BigQuery user-defined functions).

#### *Notebook Format*

Cloud Datalab combines code, documentation, results, and visualizations together in an intuitive notebook format.

#### *Pay-per-use Pricing*

Only pay for the cloud resources you use: the App Engine application, BigQuery, and any additional resources you decide to use, such as Cloud Storage.

#### *Interactive Data Visualization*

Use Google Charts or matplotlib for easy visualizations.

#### *Collaborative*

Git-based source control of notebooks with the option to sync with non-Google source code repositories like GitHub and Bitbucket.

#### *Open Source*

Developers who want to extend Cloud Datalab can fork and/or submit pull requests on the [GitHub hosted project](#).

#### *Custom Deployment*

Specify your minimum VM requirements, the network host, and more.

#### *IPython Support*

Cloud Datalab is based on Jupyter (formerly IPython) so you can use a large number of existing packages for statistics, machine learning, etc. Learn from published notebooks and swap tips with a vibrant IPython community.

## Why use Cloud Datalab?

- Create and manage code, documentation, results, and visualizations in intuitive notebook format.
  - Use Google Charts or matplotlib for easy visualizations.
- Analyze data in BigQuery, Compute Engine, and Cloud Storage using Python, SQL, and JavaScript.
- Easily deploy models to BigQuery.

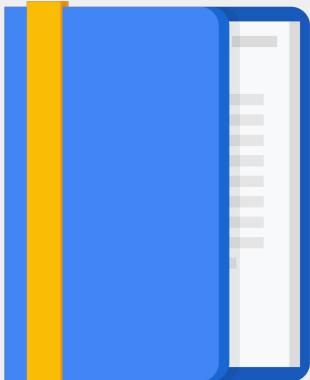


Cloud Datalab is integrated with BigQuery, Compute Engine, and Cloud Storage, so accessing your data doesn't run into authentication hassles.

When you're up and running, you can visualize your data with Google Charts or matplotlib. And, because there's a vibrant interactive Python community, you can learn from published notebooks. There are many existing packages for statistics, machine learning, and so on.

You can [attach a GPU to a Cloud Datalab instance](#) for faster processing. At the time of this writing, this feature was in beta, which means that no SLA is available and that the feature could change in backwards-incompatible ways.

## Agenda



- Google Cloud Big Data Platform
- **Google Cloud Machine Learning Platform**
- Quiz and Lab



## Machine Learning APIs enable apps that see, hear, and understand



Machine learning is one branch of the field of artificial intelligence. It's a way of solving problems without explicitly coding the solution. Instead, human coders build systems that improve themselves over time, through repeated exposure to sample data, which we call "training data."

Major Google applications use Machine Learning, like YouTube, Photos, the Google mobile app, and Google Translate. The Google machine learning platform is now available as a cloud service, so that you can add innovative capabilities to your own applications.

## Cloud Machine Learning Platform



Cloud ML



Machine Learning APIs



Open source tool to build and run neural network models

- Wide platform support: CPU or GPU; mobile, server, or cloud

Fully managed machine learning service

- Familiar notebook-based developer experience
- Optimized for Google infrastructure; integrates with BigQuery and Cloud Storage

Pre-trained machine learning models built by Google

- Speech: Stream results in real time, detects 80 languages
- Vision: Identify objects, landmarks, text, and content
- Translate: Language translation including detection
- Natural language: Structure, meaning of text

Cloud Machine Learning Platform provides modern machine learning services, with pre-trained models and a platform to generate your own tailored models. As with other GCP products, there's a range of services that stretches from the highly general to the pre-customized.

TensorFlow is an open-source software library that's exceptionally well suited for machine learning applications like neural networks. It was developed by Google Brain for Google's internal use and then open-sourced so that the world could benefit. You can run TensorFlow wherever you like, but GCP is an ideal place for it, because machine learning models need lots of on-demand compute resources and lots of training data. TensorFlow can also take advantage of Tensor Processing Units, which are hardware devices designed to accelerate machine learning workloads with TensorFlow. GCP makes them available in the cloud with Compute Engine virtual machines. Each Cloud TPU provides up to 180 teraflops of performance, and, because you pay only for what you use, there's no up-front capital investment required.

Suppose you want a more managed service. Google Cloud Machine Learning Engine lets you easily build machine learning models that work on any type of

data, of any size. It can take any TensorFlow model and perform large scale training on a managed cluster.

Finally, suppose you just want to add various machine-learning capabilities to your applications, without having to worry about the details of how they are provided. Google Cloud also offers a range of machine-learning APIs suited for specific purposes, and I'll discuss them in a moment.

## Why use the Cloud Machine Learning platform?

### For structured data



Classification and regression



Recommendation



Anomaly detection

### For unstructured data



Image and video analytics



Text analytics



People use the Cloud Machine Learning platform for lots of applications. Generally, they fall into two categories, depending on whether the data they work on is structured or unstructured.

Based on structured data, you can use ML for various kinds of classification and regression tasks, like customer churn analysis, product diagnostics, and forecasting. It can be the heart of a recommendation engine, for content personalization and cross-sells and up-sells. You can use ML to detect anomalies, as in fraud detection, sensor diagnostics, or log metrics.

Based on unstructured data, you can use ML for image analytics, such as identifying damaged shipment, identifying styles, and flagging content. You can do text analytics too, like call center log analysis, language identification, topic classification, and sentiment analysis.

In many of the most innovative applications for machine learning, several of these kinds of applications are combined. What if, whenever one of your customers posted praise for one of your products on social media, your application could automatically reach out to them with a customized discount

on another product they'll probably like? The Google Cloud Machine Learning platform makes that kind of interactivity well within your grasp.

## Cloud Vision API

- Analyze images with a simple REST API
  - Logo detection, label detection, etc
- With the Cloud Vision API, you can:
  - Gain insight from images
  - Detect inappropriate content
  - Analyze sentiment
  - Extract text



Cloud Vision API enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy to use REST API. It quickly classifies images into thousands of categories ("sailboat," "lion," "Eiffel Tower"), detects individual objects within images, and finds and reads printed words contained within images. You can build metadata on your image catalog, moderate offensive content, or enable new marketing scenarios through image sentiment analysis. Analyze images uploaded in the request or integrate with your image storage on Cloud Storage.

## Cloud Speech API

- Recognizes over 80 languages and variants
- Can return text in real time
- Highly accurate, even in noisy environments
- Access from any device
- Powered by Google's machine learning



The Cloud Speech API enables developers to convert audio to text. Because you have an increasingly global user base, the API recognizes over 80 languages and variants. You can transcribe the text of users dictating to an application's microphone, enable command-and-control through voice, or transcribe audio files.

## Cloud Natural Language API

- Uses machine learning models to reveal structure and meaning of text.
- Extract information about items mentioned in text documents, news articles, and blog posts.
- Analyze text uploaded in request or integrate with Cloud Storage.



The Cloud Natural Language API offers a variety of natural language understanding technologies to developers.

It can do syntax analysis, breaking down sentences supplied by your users into tokens, identify the nouns, verbs, adjectives, and other parts of speech, and figure out the relationships among the words.

It can do entity recognition: in other words, it can parse text and flag mentions of people, organizations, locations, events, products and media.

It can understand the overall sentiment expressed in a block of text.

And it has these capabilities in multiple languages including English, Spanish, and Japanese.

### Cloud Natural Language API features

#### Syntax Analysis

- Extract tokens and sentences, identify parts of speech (PoS), and create dependency parse trees for each sentence.

### *Entity Recognition*

- Identify entities and label by types such as person, organization, location, events, products and media.

### *Sentiment Analysis*

- Understand the overall sentiment expressed in a block of text.

### *Multi-Language*

- Enables you to easily analyze text in multiple languages including English, Spanish, and Japanese.

### *Integrated REST API*

- Access via REST API. Text can be uploaded in the request or integrated with Cloud Storage.

For more information on the Natural Language API, see:

<https://cloud.google.com/natural-language/docs/>.

## Cloud Translation API

- Translate arbitrary strings between thousands of language pairs
- Programmatically detect a document's language
- Support for dozens of languages



Cloud Translation API provides a simple programmatic interface for translating an arbitrary string into any supported language. Translation API is highly responsive, so websites and applications can integrate with Translation API for fast, dynamic translation of source text from the source language to a target language (e.g., French to English). Language detection is also available in cases where the source language is unknown.

The Translation API supports the standard Google API Client Libraries in Python, Java, Ruby, Objective-C, and other languages.

You can try it in your browser:

<https://developers.google.com/apis-explorer/#p/translate/v2/>

## Cloud Video Intelligence API

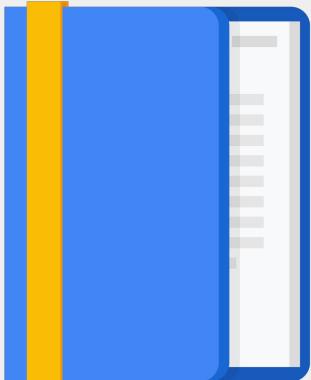
- Annotate the contents of videos
- Detect scene changes
- Flag inappropriate content
- Support for a variety of video formats



The Google Cloud Video Intelligence API allows developers to use Google video analysis technology as part of their applications. The REST API enables users to annotate videos stored in Google Cloud Storage with video and frame-level (1 fps) contextual information. It helps you identify key entities – that is, nouns – within your video, and when they occur. You can use it to make video content searchable and discoverable.

The API supports the annotation of common video formats, including .MOV, .MPEG4, .MP4, and .AVI.

## Agenda



- Google Cloud Big Data Platform
- Google Cloud Machine Learning Platform
- **Quiz and Lab**



# Question #1

---

When would you use Cloud Dataproc?



# Question #1

---

When would you use Cloud Dataproc?

You can use it to migrate on-premises Hadoop jobs to the cloud. You can also use it for data mining and analysis of cloud-based data.



## Question #2

---

Name two use cases for Cloud Dataflow.



# Question #2

---

Name two use cases for Cloud Dataflow.

1. ETL
2. orchestration



# Question #3

---

Name three use cases for the Google machine learning platform.



# Question #3

---

Name three use cases for the Google machine learning platform.

1. Fraud detection
2. Sentiment analysis
3. Content personalization



# Lab

---

In this lab, you will load server log data into BigQuery and perform a SQL query on it.

## Objectives

- Load data from Cloud Storage into BigQuery.
- Perform a query on the data in BigQuery.



# More resources

---

Google Big Data Platform <https://cloud.google.com/products/big-data/>

Google Machine Learning Platform <https://cloud.google.com/products/machine-learning/>



Please take 5 minutes to  
give us feedback

[g.co/CloudTrainEva](https://g.co/CloudTrainEva)





# Google Cloud

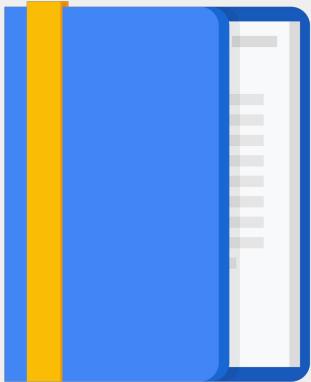
---

GCP Fundamentals: Core Infrastructure

## **Summary and Review**

Cloud Identity and Access Management (Cloud IAM) and firewall rules are some of the ways to control access to your VPC network.

## Agenda



- Course review
- Next steps



## Comparing compute options

	Compute Engine	Kubernetes Engine	App Engine Flex	App Engine Standard	Cloud Functions
Service model	IaaS	Hybrid	PaaS	PaaS	Serverless
Use cases	General computing workloads	Container-based workloads	Web and mobile applications; container-based workloads	Web and mobile applications	Ephemeral functions responding to events

*Toward managed infrastructure*

*Toward dynamic infrastructure*



Remember the continuum that this course discussed at the very beginning: the continuum between managed infrastructure and dynamic infrastructure. GCP's compute services are arranged along this continuum, and you can choose where you want to be on it.

Choose Compute Engine if you want to deploy your application in virtual machines that run on Google's infrastructure. Choose Kubernetes Engine if you want instead to deploy your application in containers that run on Google's infrastructure, in a Kubernetes cluster you define and control.

Choose App Engine instead if you want to just focus on your code, leaving most infrastructure and provisioning to Google. App Engine Flexible Environment lets you use any runtime you want, and gives you full control of the environment in which your application runs; App Engine Standard Environment lets you choose from a set of standard runtimes and offers finer-grained scaling and scale-to-zero.

To completely relieve yourself from the chore of managing infrastructure, build or extend your application using Cloud Functions. You supply chunks of code for business logic, and your code gets spun up on-demand in response to events.

## Comparing load-balancing options

Global HTTP(S)	Global SSL Proxy	Global TCP Proxy	Regional	Regional internal
Layer 7 load balancing based on load	Layer 4 load balancing of non-HTTPS SSL traffic based on load	Layer 4 load balancing of non-SSL TCP traffic	Load balancing of any traffic (TCP, UDP)	Load balancing of traffic inside a VPC
Can route different URLs to different back ends	Supported on specific port numbers	Supported on specific port numbers	Supported on any port number	Use for the internal tiers of multi-tier applications



GCP offers a variety of ways to load-balance inbound traffic. Use Global HTTP(S) Load Balancing to put your web application behind a single anycast IP to the entire Internet; it load-balances traffic among all your backend instances in regions around the world, and it's integrated with GCP's Content Delivery Network.

## Comparing interconnect options



### VPN

Secure multi-Gbps connection over VPN tunnels



### Direct Peering

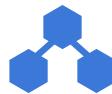
Private connection between you and Google for your hybrid cloud workloads



### Dedicated Interconnect

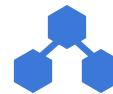
Connect N X 10G transport circuits for private cloud traffic to Google Cloud at Google POPs

**SLAs available**



### Carrier Peering

Connection through the largest partner network of service providers



### Partner Interconnect

Connectivity between your on-premises network and your VPC network through a supported service provider

**SLAs available**



GCP also offers a variety of ways for you to interconnect your on-premises or other-cloud networks with your Google VPC. It's simple to set up a VPN, and you can use Cloud Router to make it dynamic. You can peer with Google at its many worldwide points of presence, either directly or through a carrier partner. Or, if you need a Service Level Agreement and can adopt one of the required network topologies, use Dedicated Interconnect. A Partner Interconnect connection is useful if your data center is in a physical location that can't reach a Dedicated Interconnect colocation facility or if your data needs don't warrant an entire 10 Gbps connection.

## Comparing storage options

	Cloud Datastore	Cloud Bigtable	Cloud Storage	Cloud SQL	Cloud Spanner	BigQuery
Type	NoSQL document	NoSQL wide column	Blobstore	Relational SQL for OLTP	Relational SQL for OLTP	Relational SQL for OLAP
Best for	Getting started, App Engine applications	"Flat" data, Heavy read/write, events, analytical data	Structured and unstructured binary or object data	Web frameworks, existing applications	Large-scale database applications (> ~2 TB)	Interactive querying, offline analytics
Use cases	Getting started, App Engine applications	AdTech, Financial and IoT data	Images, large media files, backups	User credentials, customer orders	Whenever high I/O, global consistency is needed	Data warehousing



Consider using Cloud Datastore if you need to store structured objects, or if you require support for transactions and SQL-like queries.

Consider using Cloud Bigtable if you need to store a large amount of single-keyed data, especially structured objects.

Consider using Cloud Storage if you need to store immutable binary objects.

Consider using Cloud SQL or Cloud Spanner if you need full SQL support for an online transaction processing system. Cloud SQL provides terabytes of capacity, while Cloud Spanner provides petabytes and horizontal scalability.

Consider BigQuery if you need interactive querying in an online analytical processing system with petabytes of scale.

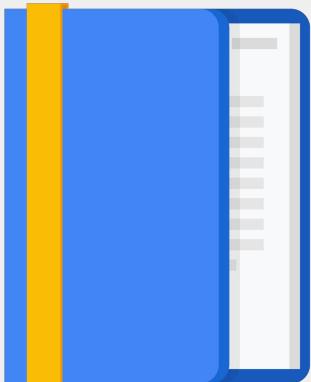
## Choosing among Google Cloud Storage classes

	Multi-regional	Regional	Nearline	Coldline
Intended for data that is...	Most frequently accessed	Accessed frequently within a region	Accessed less than once a month	Accessed less than once a year
Availability SLA	99.95%	99.90%	99.00%	99.00%
Access APIs	<i>Consistent APIs</i>			
Access time	<i>Millisecond access</i>			
<u>Storage price</u>	Price per GB stored per month			
<u>Retrieval price</u>	Total price per GB transferred			
Use cases	Content storage and delivery	In-region analytics, transcoding	Long-tail content, backups	Archiving, disaster recovery



I'd like to zoom into one of those services we just discussed, Cloud Storage, and remind you of its four storage classes. Multi-regional and regional are the classes for warm and hot data. Use multi-regional especially for content that's served to a global web audience, and use regional for working storage for compute operations. Nearline and coldline are the classes for, as you'd guess, cooler data. Use nearline for backups and for infrequently accessed content, and use coldline for archiving and disaster recovery.

## Agenda



- Course review
- **Next steps**



Please take 5 minutes to  
give us feedback

[g.co/CloudTrainEva](https://g.co/CloudTrainEva)



## What's next in the Cloud Infrastructure track?



**Cloud Infrastructure**

This track is designed for IT professionals who are responsible for implementing, deploying, migrating, and maintaining applications in the cloud.



1 Google Cloud Platform Fundamentals: Core Infrastructure



2 Architecting with Google Cloud Platform: Infrastructure



3 Architecting with Google Cloud Platform: Design and Process





If you're a cloud architect, a DevOps person, or any other kind of IT professional who deploys, migrates and maintains applications in the cloud, continue with the course: [Architecting with Google Cloud Platform: Infrastructure](#).

## What's next in the Application Development track?



**Application Development**

This track is designed for application programmers and software engineers who develop software programs in the cloud.

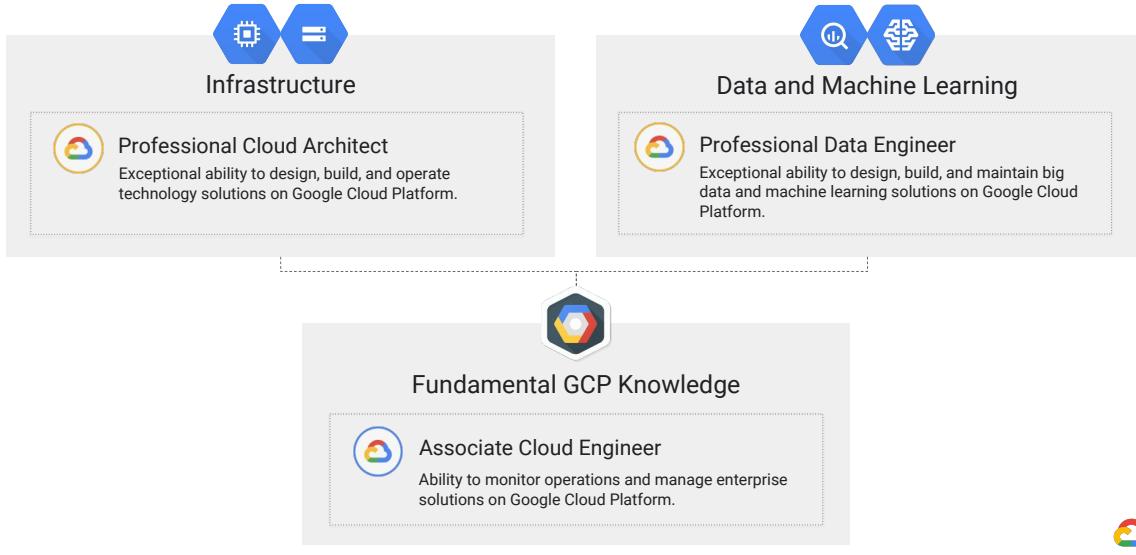
1 Google Cloud Platform Fundamentals: Core Infrastructure

2 Developing Applications with Google Cloud Platform



If you're an application programmer or any other kind of software engineer who writes code for the cloud, continue to the course [Developing Applications with Google Cloud Platform](#).

## Firewall rule parameters: Target and Source



Become Google Cloud Certified and show the world that you can design, develop, manage and administer application infrastructure and data solutions on Google Cloud technology. The Google Cloud Certified designation means you've demonstrated the necessary skills to leverage Google Cloud technology in a way that transforms businesses and meaningfully impacts the people and customers they serve.

