Google Cloud

Partner Certification Academy

Associate Cloud Engineer

The information in this presentation is classified:

Google confidential & proprietary

⚠ This presentation is shared with you under NDA.

- Do **not** record or take screenshots of this presentation.
- Do **not** share or otherwise distribute the information in this presentation with anyone **inside** or **outside** of your organization.

Thank you!

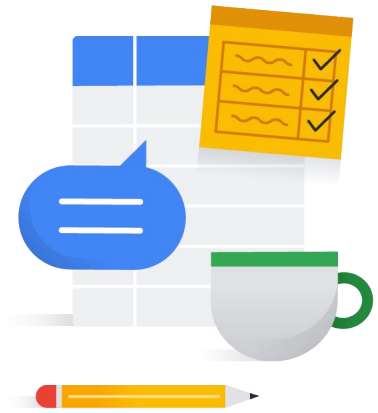
Google Cloud

Session logistics

- When you have a question, please:
  - Click the Raise hand button in Google Meet.
  - Or add your question to the Q&A section of Google Meet.
  - Please note that answers may be deferred until the end of the session.
- These slides are available in the Student Lecture section of your Qwiklabs classroom.
- The session is **not recorded.**
- Google Meet does not have persistent chat.
  - If you get disconnected, you will lose the chat history.
  - Please copy any important URLs to a local text file as they appear in the chat.

Program issues or concerns?

- Problems with **accessing** Cloud Skills Boost for Partners
  - partner-training@google.com
- Problems with **a lab** (locked out, etc.)
  - support@qwiklabs.com
- Problems with accessing Partner Advantage
  - https://support.google.com/googlecloud/topic/9198654



Associate Cloud Engineer

The Google Cloud Certified

Associate Cloud Engineer exam assesses your ability to:

- Setup a cloud solution environment
- Plan and configure a cloud solution
- Deploy and implement a cloud solution
- Ensure successful operation of a cloud solution
- Configure access and security

For more information:

https://cloud.google.com/certification/cloud-engineer

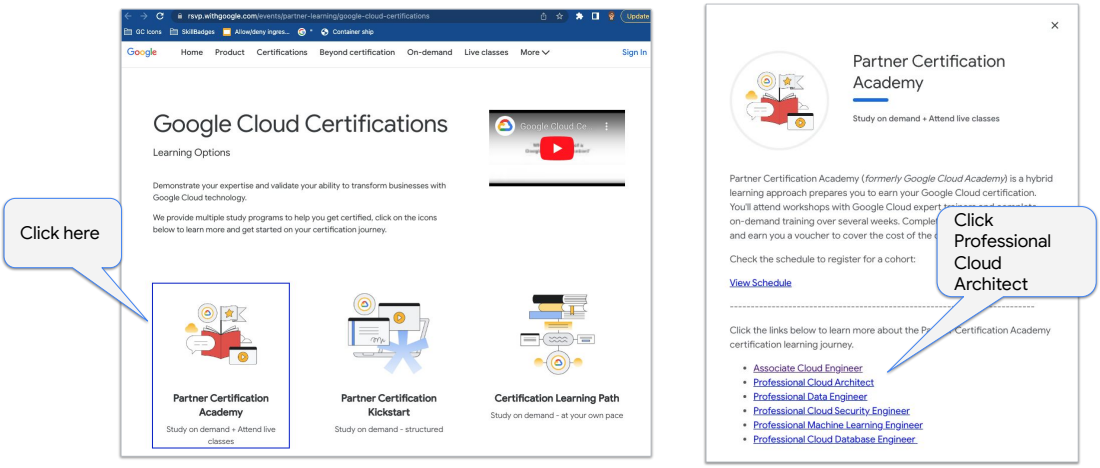
Associate Cloud Engineer
https://cloud.google.com/certification/cloud-engineer

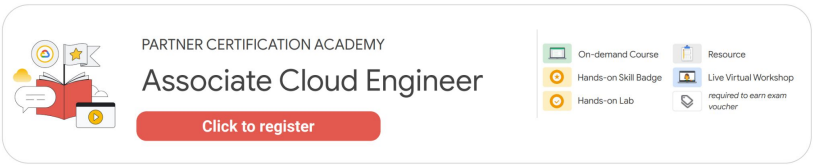
Exam Guide
https://cloud.google.com/certification/guides/cloud-engineer

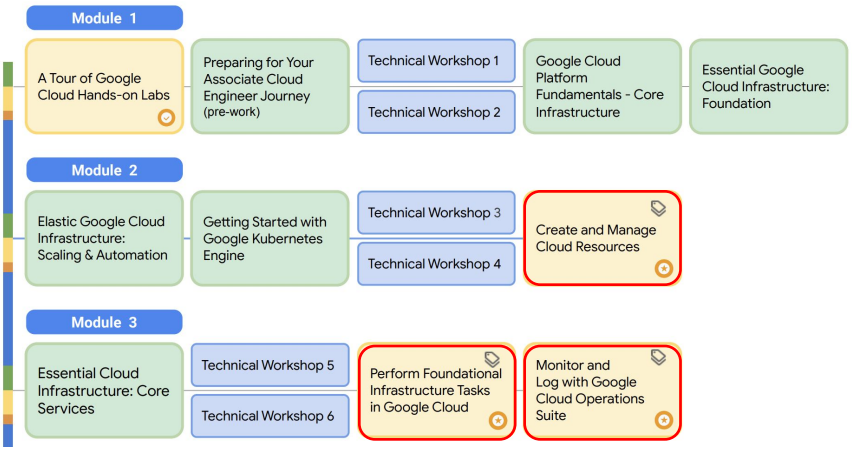
Sample Questions
https://docs.google.com/forms/d/e/1FAIpQLSfexWKtXT2OSFJ-obA4iT3GmzgiOCGvjrT9OfxilWC1yPtmfQ/viewform

Learning Path - Partner Certification Academy Website

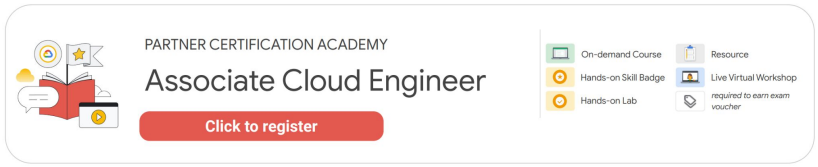
Go to: https://rsvp.withgoogle.com/events/partner-learning/google-cloud-certifications

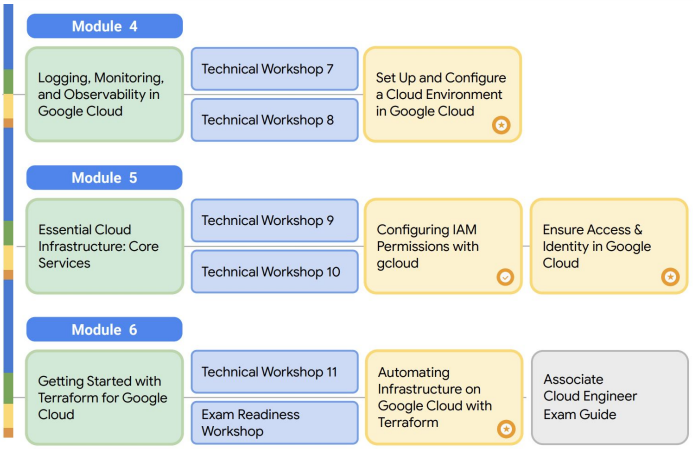






Needed for Exam Voucher





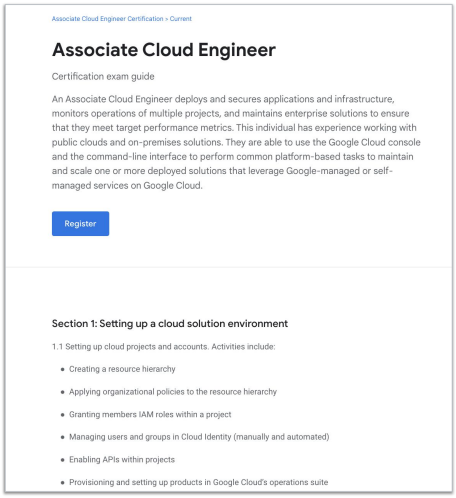
Associate Cloud Engineer (ACE) Exam Guide

Each module of this course covers Google Cloud services based on the topics in the ACE Exam Guide

The primary topics are:

- Compute Engine
- VPC Networks
- Google Kubernetes Engine
- Cloud Run, Cloud Functions and App Engine
- Storage and database options
- Resource Hierarchy/Identity and Access Management (IAM)
- Logging and Monitoring

Next discussion



Associate Cloud Engineer Certification > Current

Associate Cloud Engineer

Certification exam guide

An Associate Cloud Engineer deploys and secures applications and infrastructure, monitors operations of multiple projects, and maintains enterprise solutions to ensure that they meet target performance metrics. This individual has experience working with public clouds and on-premises solutions. They are able to use the Google Cloud console and the command-line interface to perform common platform-based tasks to maintain and scale one or more deployed solutions that leverage Google-managed or self-managed services on Google Cloud.

Register

Section 1: Setting up a cloud solution environment

1.1 Setting up cloud projects and accounts. Activities include:

- Creating a resource hierarchy
- Applying organizational policies to the resource hierarchy
- Granting members IAM roles within a project
- Managing users and groups in Cloud Identity (manually and automated)
- Enabling APIs within projects
- Provisioning and setting up products in Google Cloud's operations suite

https://cloud.google.com/certification/guides/cloud-engineer/

Identity and Access Management (IAM), Operations, IaC, Marketplace

Exam Guide Overview - Resource Hierarchy, Billing and IAM

Cloud Identity

Cloud Resource Manager

Identity & Access Management

1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

5.2 Managing service accounts. Tasks include:

- 5.2.1 Creating service accounts
- 5.2.2 Using service accounts in IAM policies with minimum permissions
- 5.2.3 Assigning service accounts to resources
- 5.2.4 Managing IAM of a service account
- 5.2.5 Managing service account impersonation
- 5.2.6 Creating and managing short-lived service account credentials

1.2 Managing billing configuration. Activities include:

- 1.2.1 Creating one or more billing accounts
- 1.2.2 Linking projects to a billing account
- 1.2.3 Establishing billing budgets and alerts
- 1.2.4 Setting up billing exports

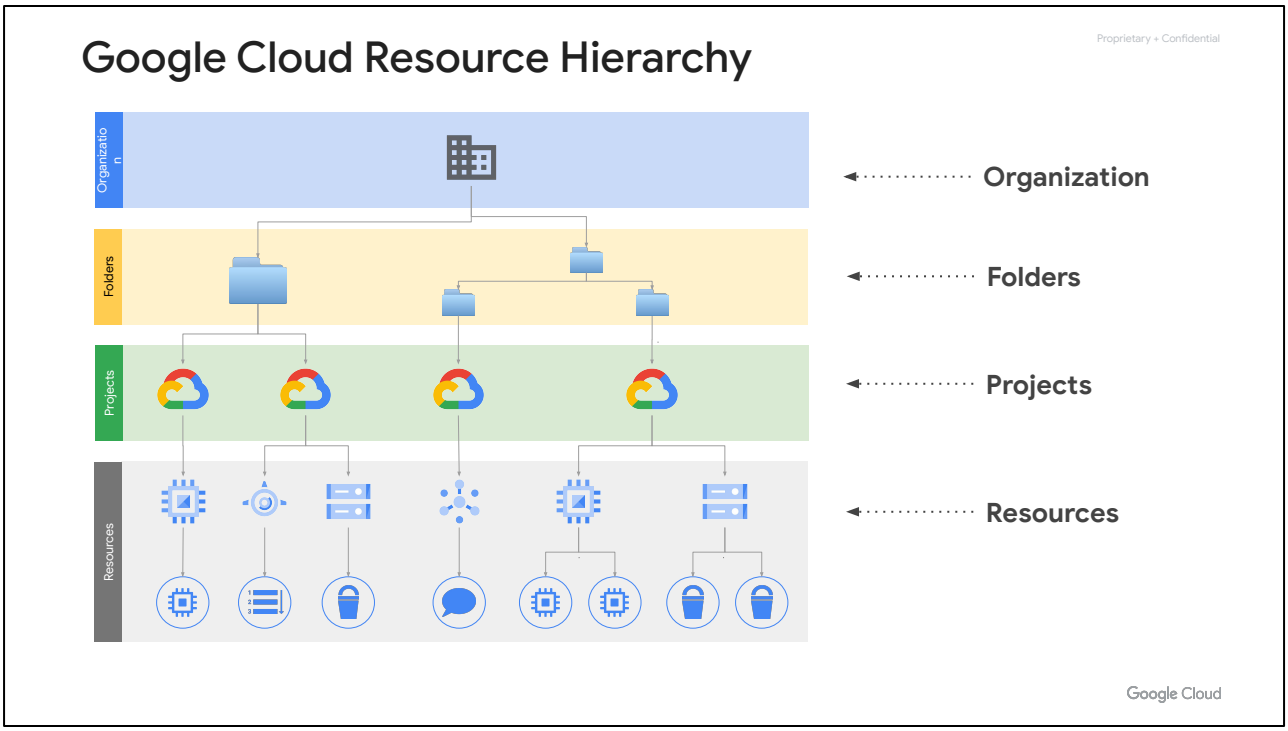
Exam Guide - Resource Hierarchy, Billing and IAM

1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)



Resource Manager
https://cloud.google.com/resource-manager

Resource hierarchy
https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy

Resource hierarchy and billing:
https://cloud.google.com/billing/docs/concepts

Organization and Folders

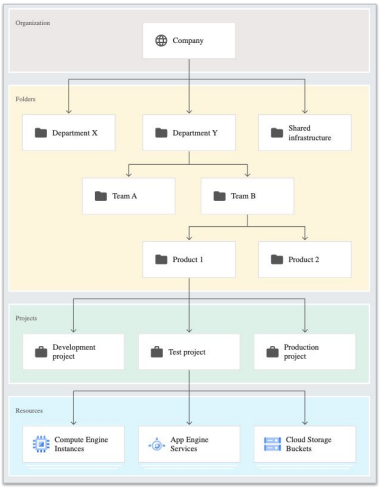
Organization is a registered domain	Folders can be added to the organization	Projects are added to either the organization or to a folder	Resources are added to projects
Rights can be granted at the organization level	Rights can be granted to folders Folders can contain other folders	Rights can be granted at the project level	Rights can be granted at the resource level

The IAM Hierarchy is organized in four distinct levels. The levels are:

- Organization, which is a registered domain. Rights granted at the organization level are applied to the entire infrastructure.
- Folders are an optional level used to organize projects. Rights can be granted to folders and folders can contain other folders.
- Projects are accounts that can be added to an organization or folder. Rights granted at the project level are applied to all resources in a project.
- Resources are the products and services used within a project. Rights can also be granted at the resource level.

Folders provide a logical way to organize teams and projects

- Must have an Organization node in order to create folders
- Folders contain projects, other folders, or both
- Roles and Organization policies can be applied at the folder level (and also at the project/organization level)

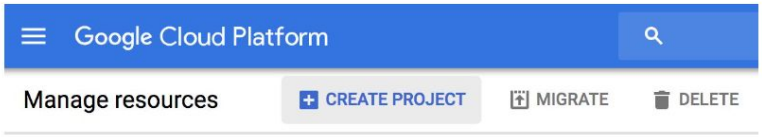


https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy#folders

All products & services are associated with a project

- Enable/disable APIs
- Enabling billing and track resource usage
- Contains resources such as VPC networks, VMs, Databases, etc.
- IAM permissions can be applied at this level

Creating a project



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

Creating and managing projects
https://cloud.google.com/resource-manager/docs/creating-managing-projects

CLI - creating projects:
https://cloud.google.com/sdk/gcloud/reference/projects/create

Resources

- Resources are anything created when using services
- All resources are associated with a project
- Examples of resources are:
  - Virtual machines
  - Persistent disk
  - Cloud Storage buckets
  - BigQuery datasets and tables
  - Spanner databases
  - Kubernetes Engine clusters
- Must enable service-specific APIs before creating resources within a project

All resources must be associated with a project, so proper billing can be implemented.

Examples of resources include:

- Virtual machines
- Disks
- Cloud Storage buckets
- BigQuery datasets and tables
- Spanner databases
- Kubernetes Engine clusters

These are just a small sample.

Exam Guide - Resource Hierarchy, Billing and IAM

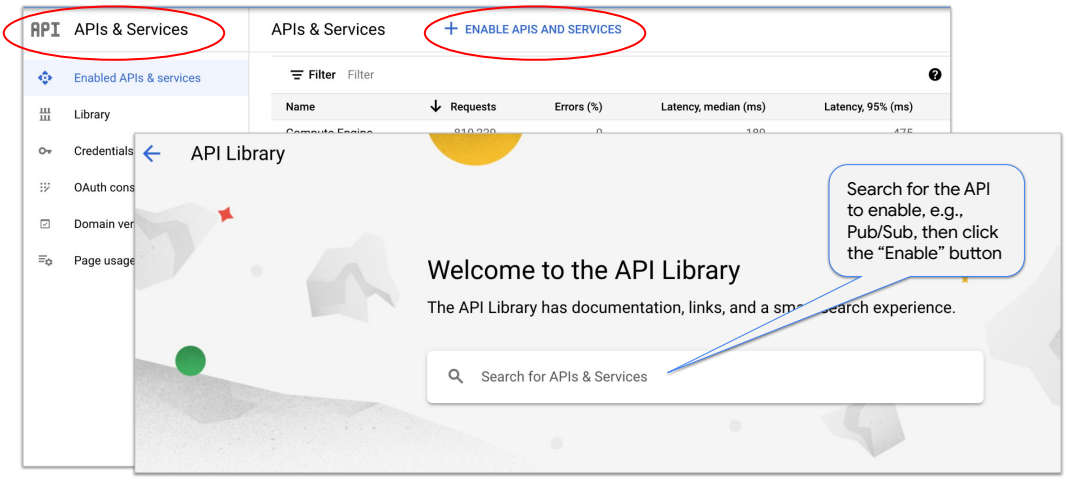
1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

Enabling APIs - Console & Command Line




```
gcloud services enable pubsub.googleapis.com
```


API Explorer:
https://developers.google.com/apis-explorer

Getting started:
https://cloud.google.com/apis/docs/getting-started

Client Libraries
https://cloud.google.com/apis/docs/client-libraries-explained
https://cloud.google.com/apis/docs/cloud-client-libraries

Enabling and Disabling Services:
https://cloud.google.com/service-management/enable-disable

Lists the services that are enabled or available to be enabled by a project:
https://cloud.google.com/sdk/gcloud/reference/services/list

Enabling services:
https://cloud.google.com/sdk/gcloud/reference/services/enable

Disabling services: https://cloud.google.com/sdk/gcloud/reference/services/disable

Exam Guide - Resource Hierarchy, Billing and IAM

1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

Cloud Identity centrally manages users and groups for Google Cloud

- **Google's Identity as a Service (IDaaS) solution**
  - Users and groups that are to be added to Google Cloud need accounts in Cloud Identity
- **Manually creating user accounts**
  - Add users individually using the Google Admin console
  - Add several users at once by uploading their names in a CSV file
- **Options for large organizations**
  - Use Google Cloud Directory Sync to synchronize user data in your existing LDAP directory with your Google account
  - Use the Admin SDK Directory API to provision a large number of users with data from your existing LDAP directory, such as Microsoft® Active Directory®
    - Requires programming

Focus of this module

*Cloud Identity has advanced management features not covered in this module, e.g., mobile app management, 2-Step verification, etc.

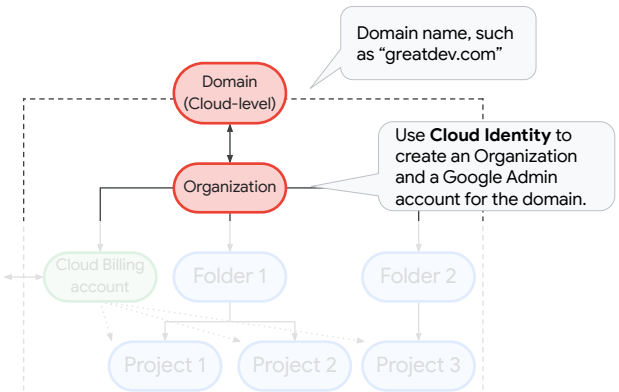
Overview of Cloud Identity
https://cloud.google.com/identity/docs/overview

Features provided by Cloud Identity
https://cloud.google.com/identity/docs/editions

Creating users and groups
https://support.google.com/cloudidentity/answer/7332836?hl=en&ref_topic=7558418

How is an Organization Created?

- **Cloud Identity** manages the users and groups that have access to Google Cloud
  - Federated identities from Google Workspace and other identity providers, such as Active Directory and Azure Active Directory
    - Bring existing users/groups into Cloud Identity
      - Use Identity and Access Management (IAM) to manage access to Google Cloud resources



Creating and managing organizations

Creating and managing organizations:
https://cloud.google.com/resource-manager/docs/creating-managing-organization

Managing multiple organization resources:
https://cloud.google.com/resource-manager/docs/managing-multiple-orgs

Frequently asked question - Setting up reseller accounts:
https://support.google.com/channelservices/answer/11559816?hl=en

Google Cloud Directory Sync (GCDS)

Best practice

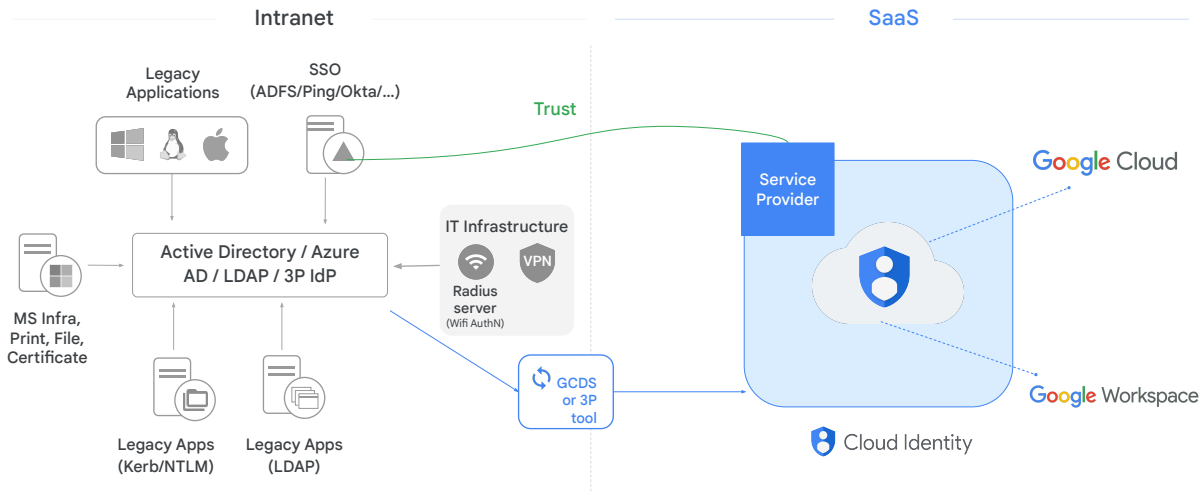
Provision from a single source of truth

- One-way synchronization of corporate data (no writing to LDAP system)
- Only synchronizes deltas for fastest possible provisioning
- Syncs all object types (users, aliases, profiles, groups)
- Utilizes Google APIs to provision all object types, the same APIs available to customers

Google Cloud Directory Sync examples
https://cloud.google.com/community/tutorials/gcds-use-cases-common-and-complex

- Recommendation is to **provision from single source of truth** (i.e. single consolidated LDAP source). Potential multi-directory pitfalls:
  - Involves creating highly complex configuration files
  - Difficult to troubleshoot problems
  - Difficult to maintain configuration files
  - High probability of inadvertently deleting objects
- Can be installed on **Windows or Linux**
- Easy to schedule (**Task Scheduler / cron jobs**) or run manually
- Helps automate onboarding and off-boarding scenarios
- Supports **multiple authentication methods**

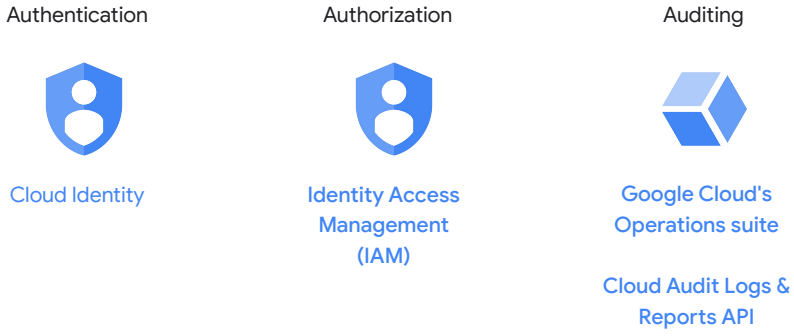
Third-party as an identity provider: Typical architecture



Google Cloud Directory Sync (GCDS) can be used to import users from a 3rd party Identity Provider into Cloud Identity

User will authenticate with the third-party IdP (AD Federation Service in this diagram, but it can also be Azure AD, Ping, OKTA, etc.)

Controlling access



- Authentication: identifying a user typically done through a process in which the user provides a private form of verification (for example, password, a key, etc.). We use Cloud Identity for AuthN
- Authorization: set of permissions that a user is allocated post authentication. A user can be authenticated (thus confirming his identity) but authentication, on its own, provides no set of permissions. We use IAM for AuthZ (and Cloud Identity for assigning Admin roles).
- Auditing: a form of monitoring the resources accessed or modified by a particular identity. We use Google Cloud operations suite (formerly known as Stackdriver), Cloud Audit Logs for auditing in Google Cloud, Reports API for auditing in Cloud Identity operations.

Exam Guide - Resource Hierarchy, Billing and IAM

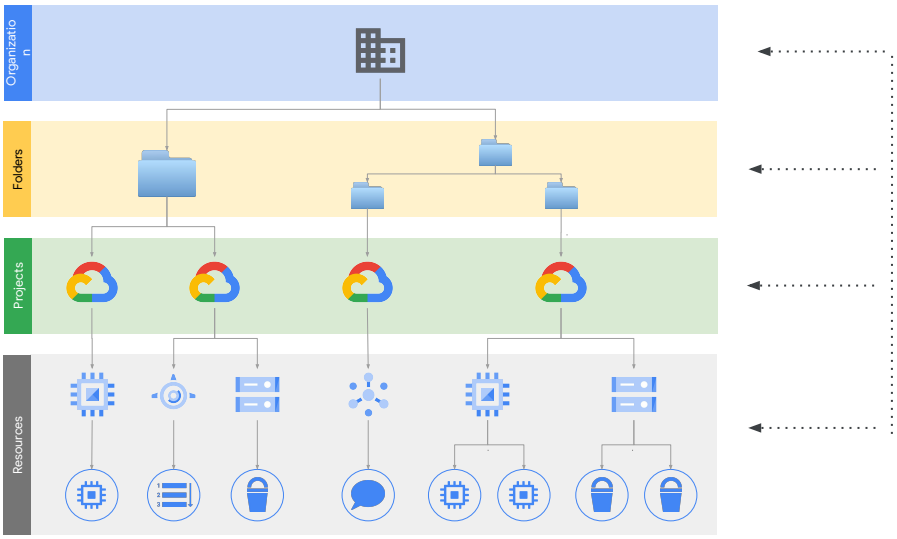
1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

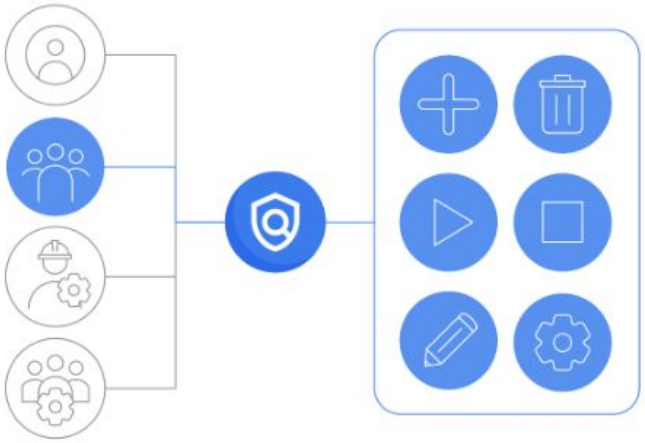
Hierarchy inheritance



Next discussion: Identity and Access Management

Identity and Access Management (IAM) applies policies

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



IAM overview:
https://cloud.google.com/iam/docs/overview

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

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

- The "who" part of an IAM policy can be a Google account, a Google group, a service account, or Cloud Identity domain.
- The "can do what" part of an IAM policy is defined by a role. An IAM role is a collection of permissions. 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.

Who ...?

Service Accounts are discussed later

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

- Google account
- Google group
- Service account
- Google Workspace or Cloud Identity domain

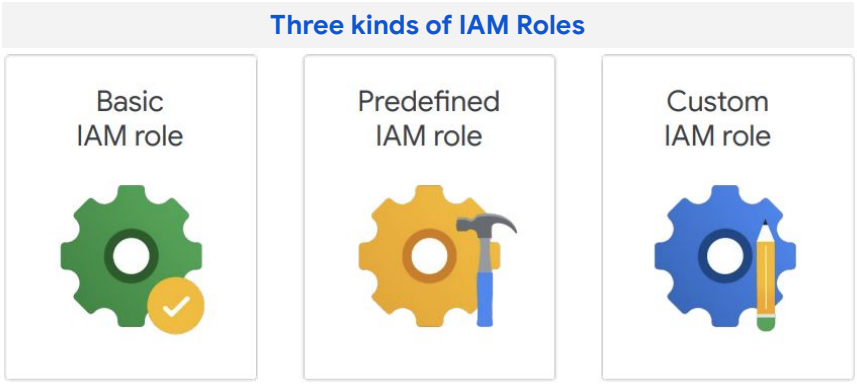
Also

- allAuthenticatedUsers
- allUsers

Types of principals

… can do what …?

- The “can do what” is defined by an IAM role.



Role types
https://cloud.google.com/iam/docs/roles-overview#role-types

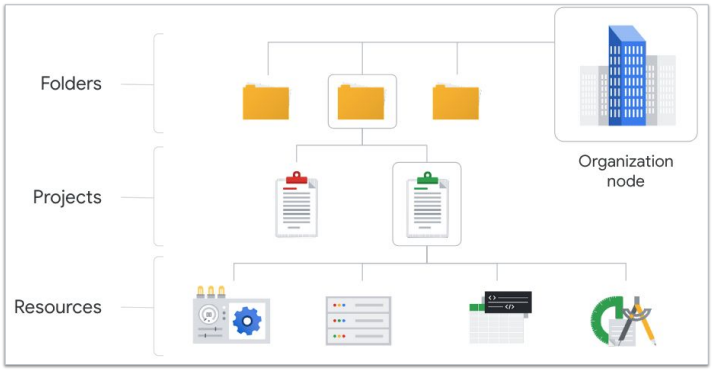
Basic roles, which include the Owner, Editor, and Viewer roles that existed prior to the introduction of IAM.

Predefined roles, which provide granular access for a specific service and are managed by Google Cloud.

Custom roles, which provide granular access according to a user-specified list of permissions.

… on which resource?

- Can manage IAM at the project level, folder level or organization level*



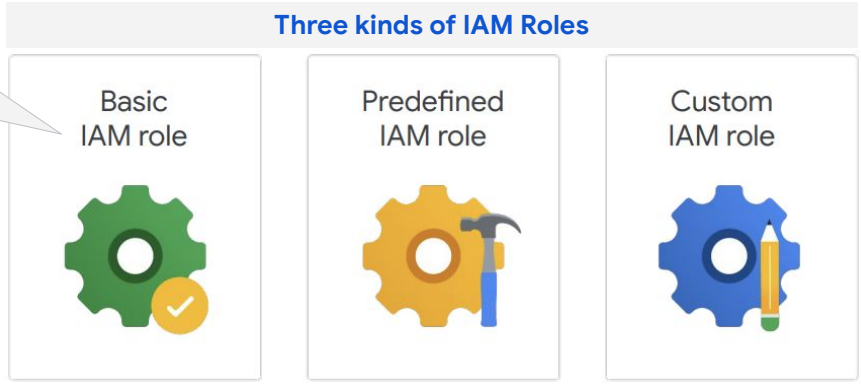
*Note: Some Cloud Storage IAM is applied at the bucket level

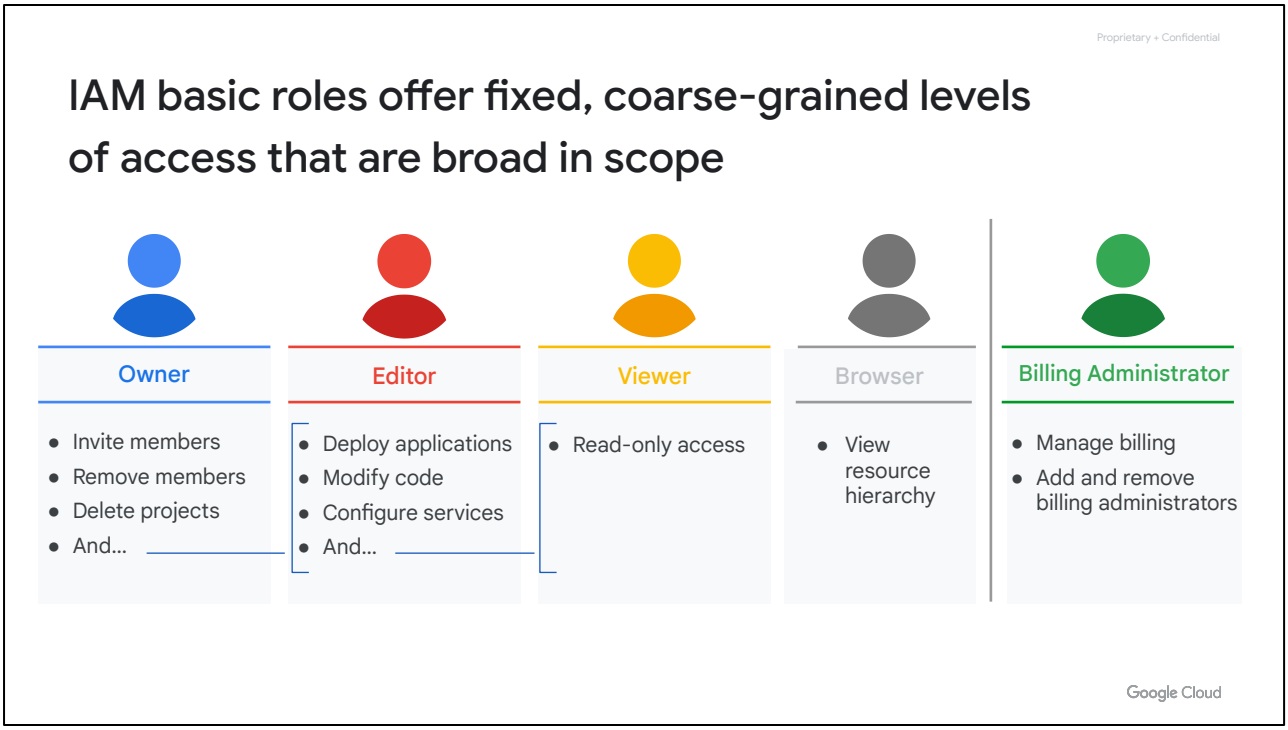
… can do what …?

Revisiting this slide

- The “can do what” is defined by an IAM role.

Basic roles discussed next





IAM basic and predefined roles reference
https://cloud.google.com/iam/docs/understanding-roles

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

The basic roles are the Owner, Editor, Viewer, Browser and Billing Administrator roles.

- The owner has full administrative access. This includes the ability to add and remove members and delete projects.
- The editor role has modify and delete access. This allows a developer to deploy applications and modify or configure its resources.
- The viewer role has read-only access
- The browser role can view only the resource hierarchy

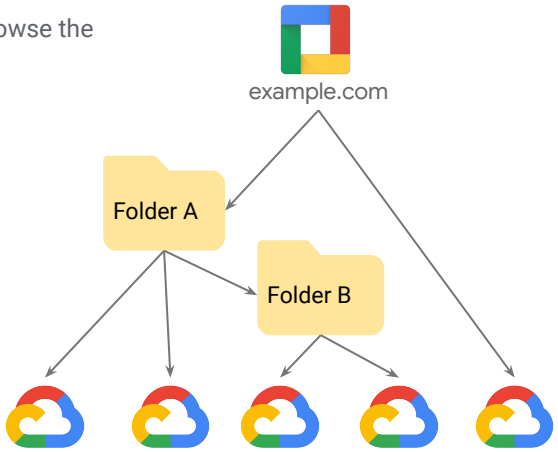
The Owner role includes the permissions of the Editor role, and the Editor role includes the permissions of the Viewer role.

There is also a billing administrator role to manage billing and add or remove administrators without the right to change the resources in the project.

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

The **Browser** basic role

This role provides read access to browse the hierarchy for a project, including the organization and folders.



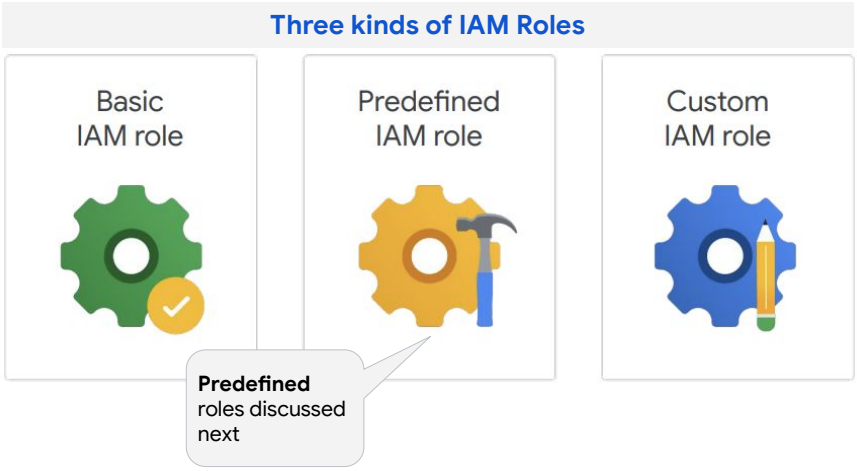
The predefined Browser role provides read-access to browse the **hierarchy** for a project, including the folder, organization, and IAM policy.

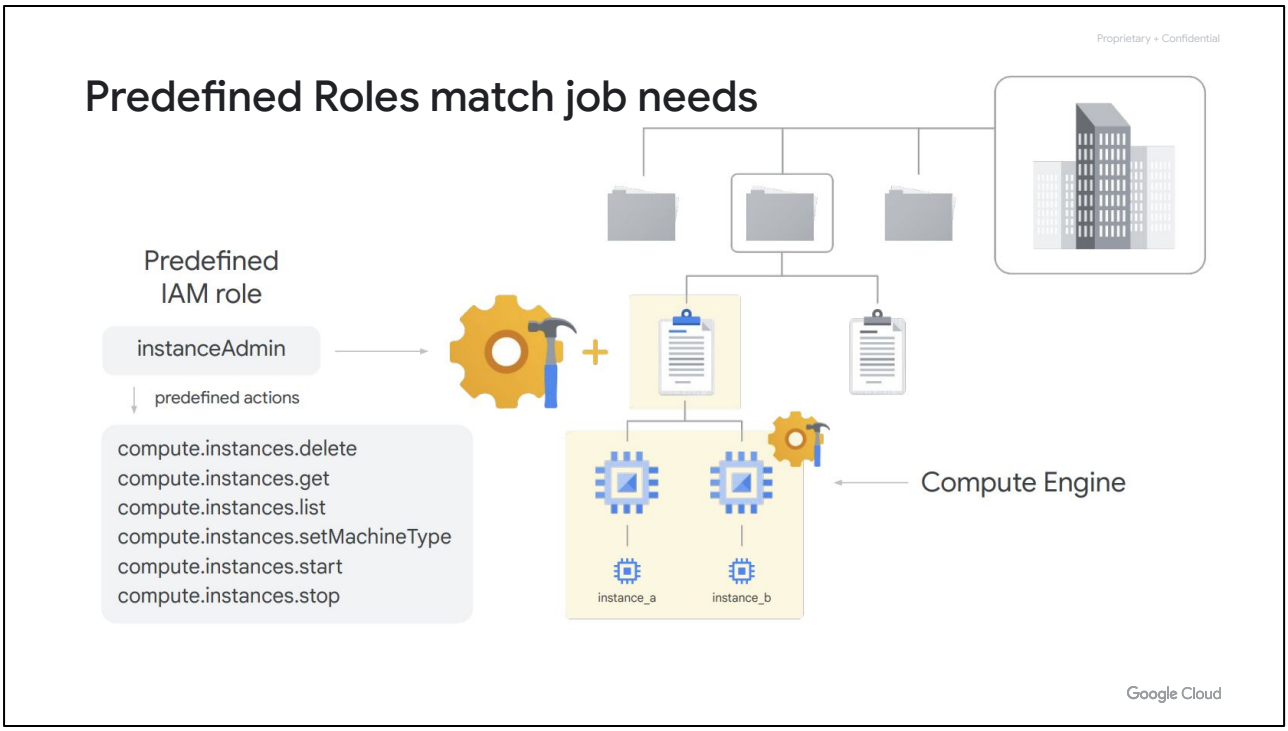
The Browser role does not include permission to view resources in the project.

… can do what …?

Revisiting this slide

- The “can do what” is defined by an IAM role.





Choosing the correct predefined role:
https://cloud.google.com/iam/docs/choose-predefined-roles

Google Cloud uses roles as a method to distribute permissions.

Members are assigned one or more roles. The roles a member is assigned will determine the permissions of a member.

A role is a list of permissions organized by a service.

This list of permissions correlate with Google Cloud API calls and are aligned based on job functions.

The example shown are some of the roles for Compute Engine Instance Admin

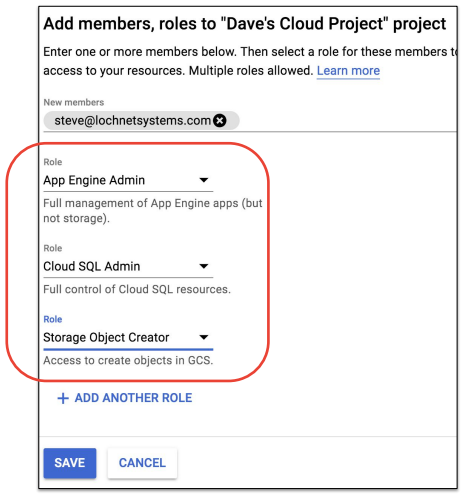
Using predefined roles

Google-created roles for each service

- Permissions defined for different job roles working with each service
- Permissions maintained by Google

Best practice is to avoid the basic roles and use predefined roles when adding members or service accounts

- Principle of least privilege



Google provides roles for each service. There are hundreds of predefined roles in Google Cloud.

Predefined roles are based on different job roles when working with different services.

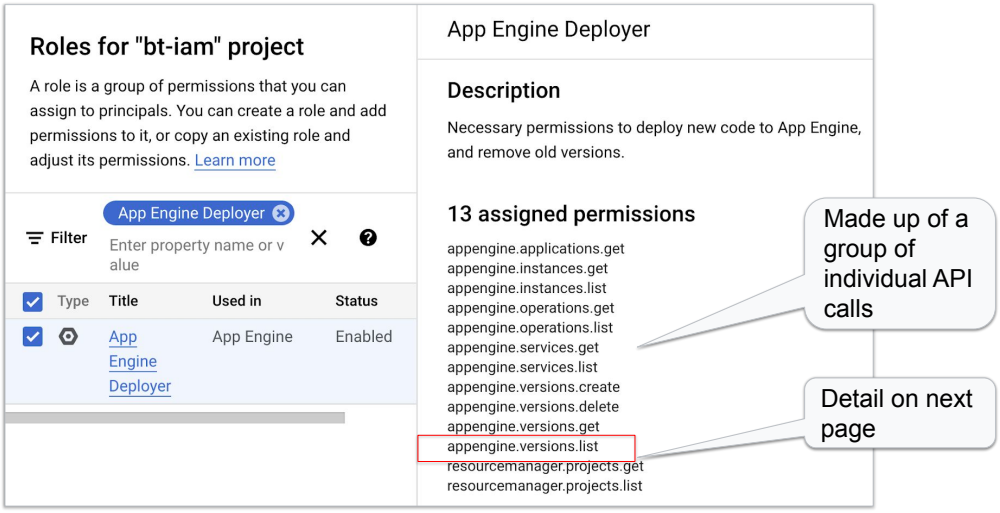
And are maintained by Google.

A Google best practice is to avoid using basic roles. Basic roles may not be specific enough.

Always follow the principle of least privilege.

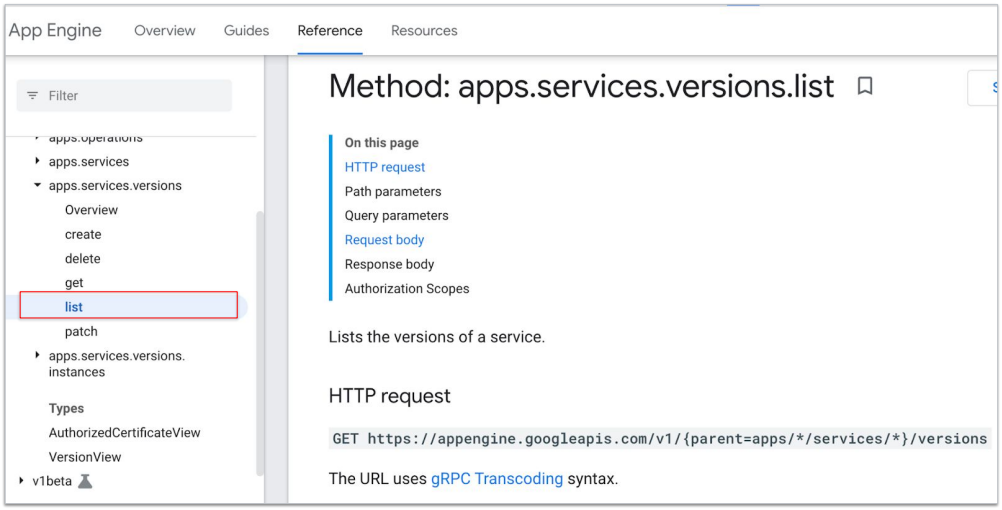
Role permissions

- Cloud Console > IAM & Admin > Roles



IAM basic and predefined roles reference
https://cloud.google.com/iam/docs/understanding-roles

App Engine API Documentation



https://cloud.google.com/appengine/docs/admin-api/reference/rest/v1/apps.services.versions/list

… can do what …?

Revisiting this slide

- The “can do what” is defined by an IAM role.

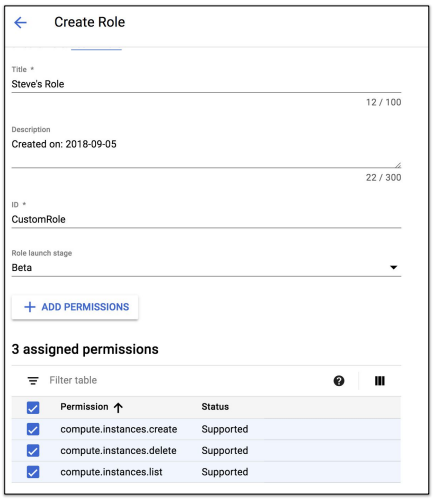
Three kinds of IAM Roles

Basic IAM role	Predefined IAM role	Custom IAM role
		

Custom roles discussed next

Custom Roles

- Roles that you create
  - Fine-grained control over permissions
- Can add any permissions you like
- Can create custom roles based on predefined roles and add /remove permissions
- Custom roles add operational overhead
  - You must maintain the permissions
- Applied at the project or organization level



Understanding custom roles:
https://cloud.google.com/iam/docs/understanding-custom-roles

Custom roles are an option when there is not a predefined role available that provides the permissions you would like to group.

Custom roles give you fine-grained control over permissions by allowing you to add any permission you like to a role you create.

You can create custom roles by copying and modifying a predefined role or you can create one from scratch.

Custom roles add operational overhead, as you are responsible for maintaining the permissions of a custom role.

Custom roles can only be applied either to project level or organization level. They can't be applied to the folder level.

Exam Guide - Resource Hierarchy, Billing and IAM

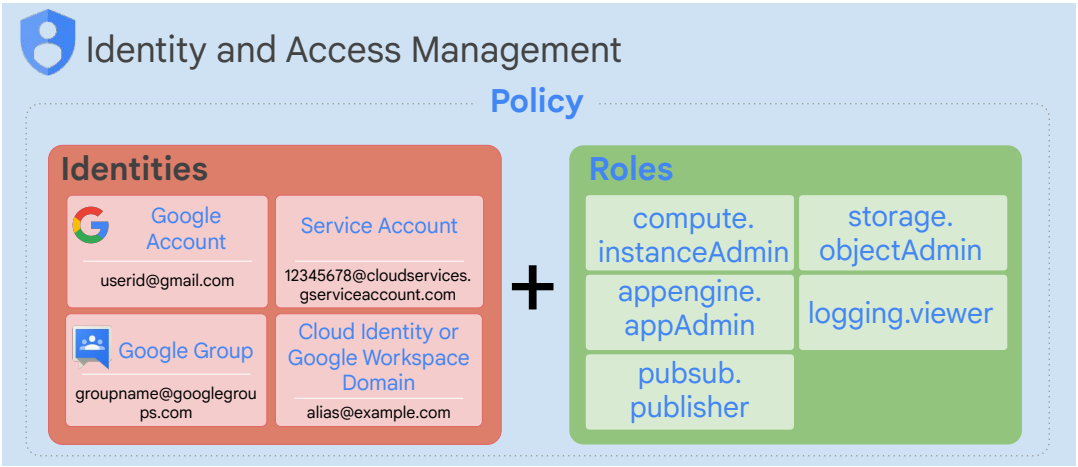
1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

A policy is a combination of principals + assigned roles



Also allUsers and allAuthenticatedUsers

Note: You *don't* use IAM to create or manage your users or groups. IAM roles provide permissions to do certain things

Policy
https://cloud.google.com/iam/docs/reference/rest/v1/Policy

There are five different types of members: Google Accounts, Service Accounts, Google Groups, Google Workspace domains, and Cloud Identity domains.

A Google account represents a developer, an administrator, or any other person who interacts with Google Cloud. Any email address that is associated with a Google account can be an identity, including gmail.com or other domains. New users can sign up for a Google account by going to the Google account signup page, without receiving mail through Gmail.

A service account is an account that belongs to your application instead of to an individual end user. When you run code that is hosted on Google Cloud, you specify the account that the code should run as. You can create as many service accounts as needed to represent the different logical components of your application.

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

A Workspace domain represents a virtual group of all the Google accounts that have

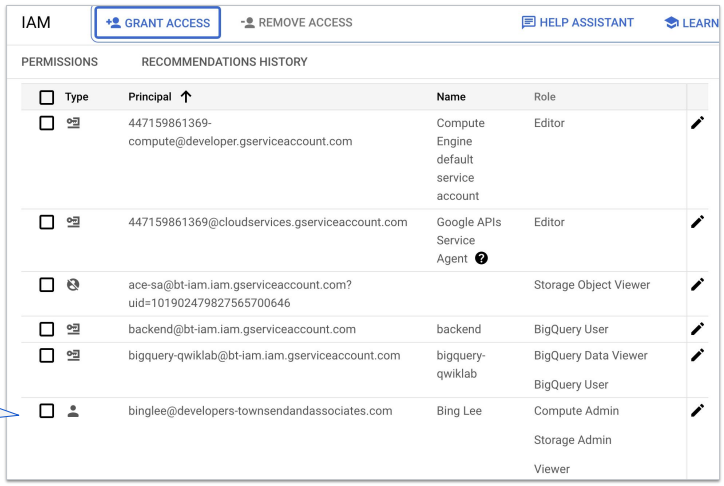
been created in an organization's Workspace account. Workspace domains represent your organization's internet domain name, such as example.com, and when you add a user to your Workspace domain, a new Google account is created for the user inside this virtual group, such as username@example.com.

Google Cloud customers who are not Workspace customers can get these same capabilities through Cloud Identity. Cloud Identity lets you manage users and groups using the Google Admin Console, but you do not pay for or receive Workspace's collaboration products such as Gmail, Docs, Drive, and Calendar. Refer to the documentation for more information on Cloud Identity Editions.

Now it's important to note that you cannot use IAM to create or manage your users or groups. Instead, you can use Cloud Identity or Workspace to create and manage users.

Viewing IAM policies in the Console

- **Cloud Console** > IAM & Admin > IAM



User Bing has 3 roles - Compute Admin, Storage Admin and Viewer

Viewer is a basic role; the other two are predefined roles

Viewing IAM policies in the CLI

- Policies consist of a collection of bindings
- One or more principles can be “bound” a single role
- Principals can be user accounts, groups, service accounts and domains (such as Google Workspace or [name here].com)

```
gcloud projects get-iam-policy [project-id]

gcloud projects get-iam-policy [project-id] \
--format json >out.text
```

Example output from the `gcloud get-iam-policy` command


```
{
  "bindings": [
    {
      "members": [
        "serviceAccount:bigquery-qwiklab@bt-iam.iam.gserviceaccount.com"
      ],
      "role": "roles/bigquery.dataViewer"
    },
    {
      "members": [
        "serviceAccount:backend@bt-iam.iam.gserviceaccount.com",
        "serviceAccount:bigquery-qwiklab@bt-iam.iam.gserviceaccount.com"
      ],
      "role": "roles/bigquery.user"
    },
    {
      "members": [
        "user:binglee@developers-townsendandassociates.com",
        "user:leevalley@developers-townsendandassociates.com"
      ],
      "role": "roles/compute.admin"
    },
    {
      "members": [
        "user:caliagusta@developers-townsendandassociates.com"
      ],
      "role": "roles/compute.instanceAdmin.v1"
    },
```

Exam Guide - Resource Hierarchy, Billing and IAM

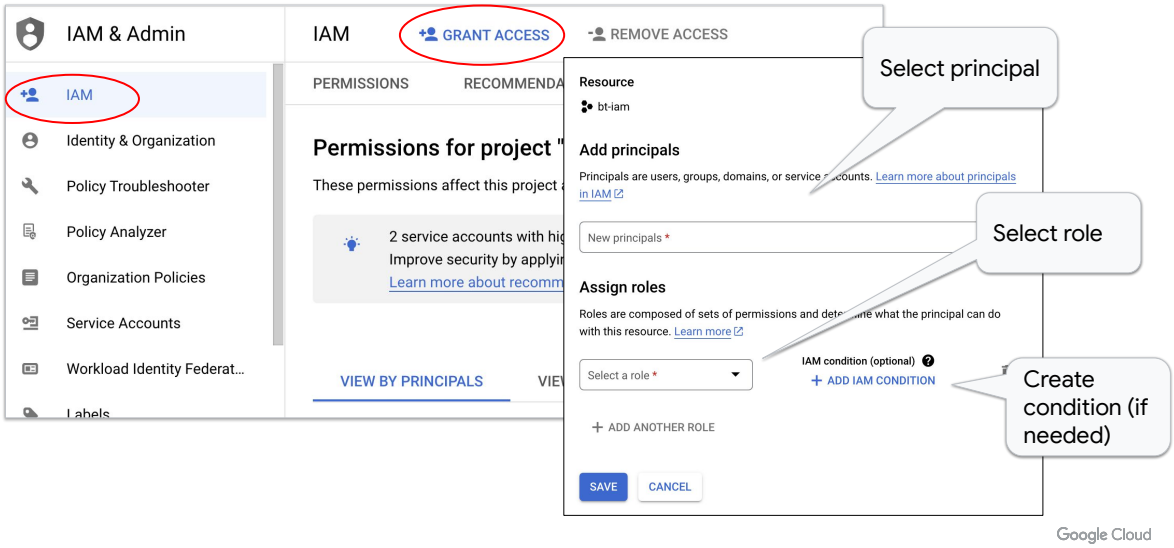
1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

Granting roles to identities at the project level - console



Granting roles to identities at the project level - CLI

- Syntax:
  - **`gcloud projects add-iam-policy-binding`** `[PROJECTID] --member user:[USER-EMAIL] --role [ROLE-ID]`
- Examples
  - `gcloud projects add-iam-policy-binding bt-iam --member='user:binglee@developers.com' --role='roles/editor'`
  - `gcloud projects add-iam-policy-binding bt-iam --member 'group:developers@test.com' --role 'roles/appengine.deployer'`
- Valid values for `member`
  - `user:bing@test.com`
  - `group: developers@test.com`
  - `serviceAccount:test-555@my-projectid.iam.gserviceaccount.com`
  - `domain:test.com`

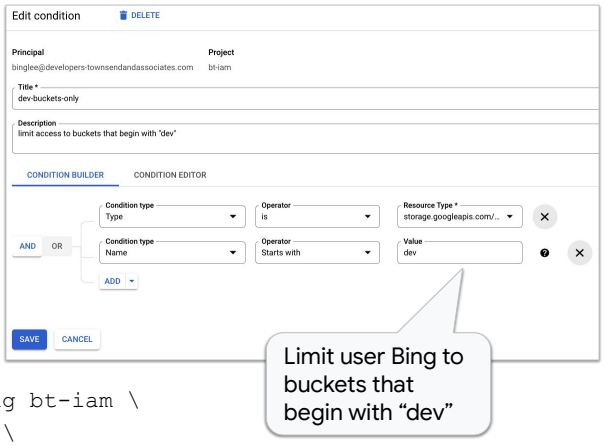
IAM Policies

Policy:
https://cloud.google.com/iam/docs/reference/rest/v1/Policy

Assigning users to pre-defined IAM roles within a project:
https://cloud.google.com/sdk/gcloud/reference/projects/add-iam-policy-binding

IAM Conditions

- Conditions can be applied to role assignments
- Can be based on
  - A time range
  - Specific resource types and names
  - And more




```
gcloud projects add-iam-policy-binding bt-iam \
--member 'group:developers@test.com' \
--role 'roles/storage.bucketadmin' \
--condition="expression=resource.name.startsWith("projects/bt-iam/buckets/dev\"),
title=dev-buckets-only"
```


Overview of IAM Conditions
https://cloud.google.com/iam/docs/conditions-overview

Configure resource-based access
https://cloud.google.com/iam/docs/configuring-resource-based-access

Configure temporary access:
https://cloud.google.com/iam/docs/configuring-temporary-access

Manage conditional role bindings:
https://cloud.google.com/iam/docs/managing-conditional-role-bindings

Other IAM condition examples

- Allow access to Compute Engine VM instances, but no other type of resource
  - `resource.type == "compute.googleapis.com/Instance"`
- Allow access to Cloud Storage resources, but no other resources
  - `resource.service == "storage.googleapis.com"`
- Allow access to Cloud Storage objects inside a specific bucket
  - `resource.type == "storage.googleapis.com/Object" && resource.name.startsWith("projects/_/buckets/exampleco-site-assets/")`
- Allow access only for specific months and year, based on the time zone for Berlin, Germany
  - `request.time.getFullYear("Europe/Berlin") == 2020 && request.time.getMonth("Europe/Berlin") < 6`

Condition attributes
https://cloud.google.com/iam/docs/conditions-overview#attributes

Exam Guide - Resource Hierarchy, Billing and IAM

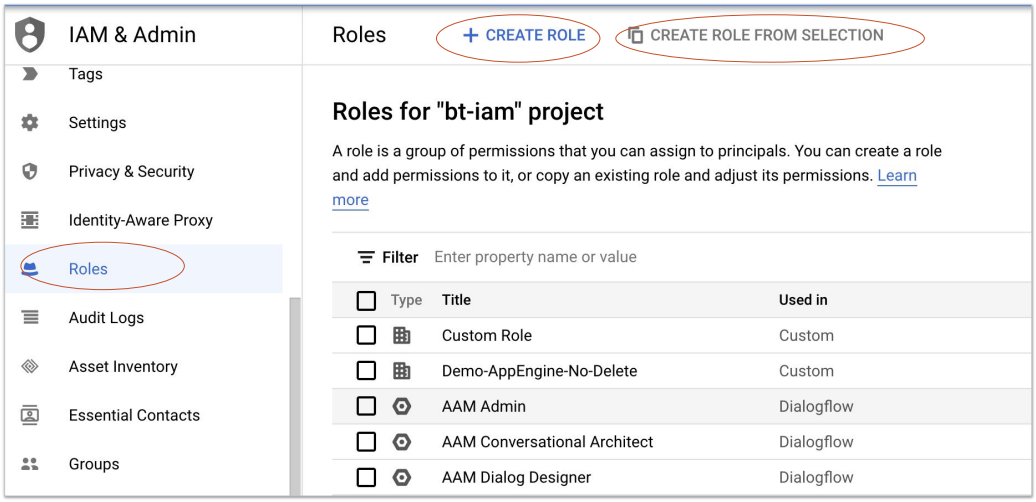
1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

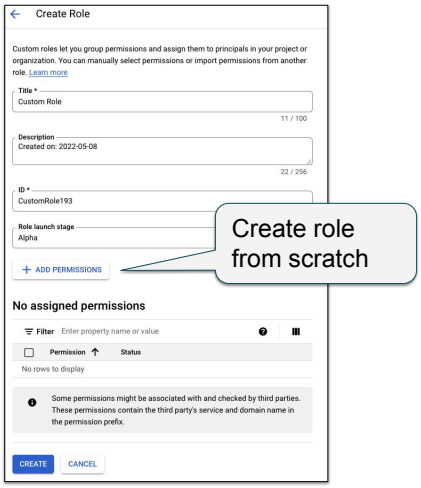
5.1 Managing Identity and Access Management (IAM). Tasks include:

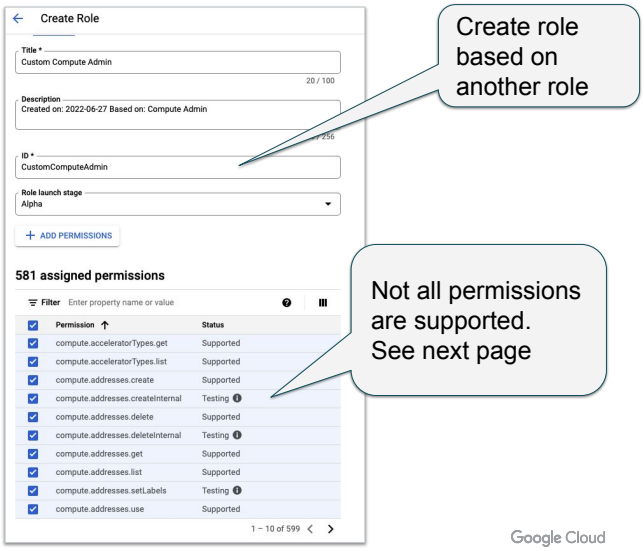
- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

Creating a Custom Role in the Console



Creating a Custom Role in the Console (continued)





Permission support levels

- Each permission has one of the following support levels for use in custom roles:

Support level	Description
SUPPORTED	The permission is fully supported in custom roles.
TESTING	Google is testing the permission to check its compatibility with custom roles. You can include the permission in custom roles, but you might see unexpected behavior. Not recommended for production use.
NOT_SUPPORTED	The permission is not supported in custom roles.

- For more information
  - https://cloud.google.com/iam/docs/creating-custom-roles#viewing-resource-permissions

Creating custom roles
https://cloud.google.com/iam/docs/creating-custom-roles
https://cloud.google.com/sdk/gcloud/reference/iam/roles/create

Available permissions for custom roles
https://cloud.google.com/iam/docs/creating-custom-roles#viewing-resource-permissions

Creating a Custom Role in the CLI

- Create a custom role using `gcloud`
  - Two options are available
    - Use flags

```
gcloud iam roles create AppEngineVersionUpdater --project=bt-iam
--title=AppEngineVersionMaint \
--description="Can create, delete, update and list versions" \
--permissions=appengine.versions.delete, appengine.versions.list,
appengine.versions.get, appengine.versions.create
```

    - Use YAML

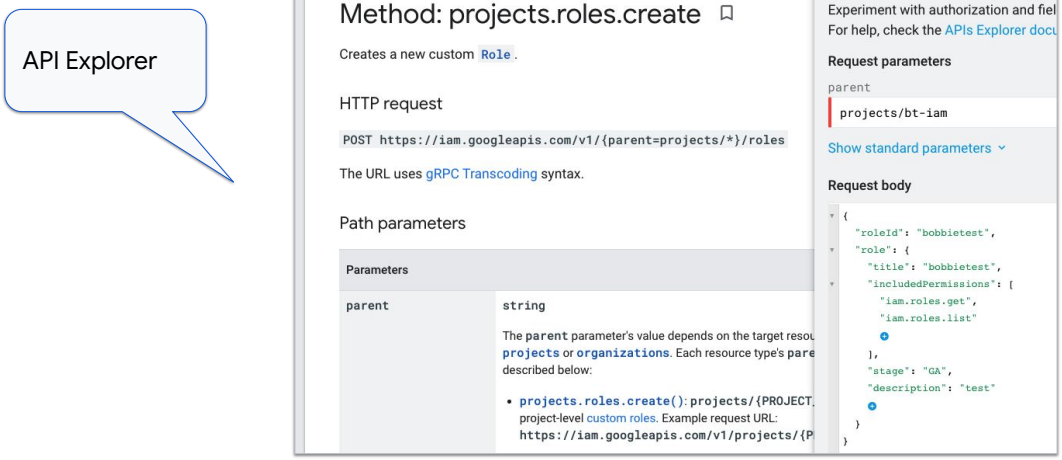
```
gcloud iam roles create
AppEngineVersionViewer --project=bt-iam
-file=path-to-yaml-file
```

```
title: AppEngineVersionViewer
description: View-only
stage: GA
includedPermissions:
- appengine.versions.list
- appengine.versions.get
```

Example YAML file

Using gcloud:
https://cloud.google.com/iam/docs/creating-custom-roles#iam-custom-roles-create-gcloud

Creating a Custom Role with the API



API Explorer:
https://cloud.google.com/iam/docs/reference/rest/v1/projects.roles/create

This is just a reminder that anything that can be done with the Console or the CLI can also be done via an API call

Creating a Custom Role using a client library

PHP example

```php
$client = new Google_Client();
$client->setApplicationName('Google-iamSample/0.1');
$client->useApplicationDefaultCredentials();
$client->addScope('https://www.googleapis.com/auth/cloud-platform');

$service = new Google_Service_Iam($client);

// The resource name of the parent resource in one of the following formats:
// `organizations/{ORGANIZATION_ID}`
// `projects/{PROJECT_ID}`
$parent = 'projects/my-project';  // TODO: Update placeholder value.

// TODO: Assign values to desired properties of `requestBody`:
$requestBody = new Google_Service_Iam_CreateRoleRequest();

$response = $service->projects_roles->create($parent, $requestBody);

// TODO: Change code below to process the `response` object:
echo '<pre>', var_export($response, true), '</pre>', "\n";
?>
```


Using Client Libraries:
https://cloud.google.com/apis/docs/cloud-client-libraries

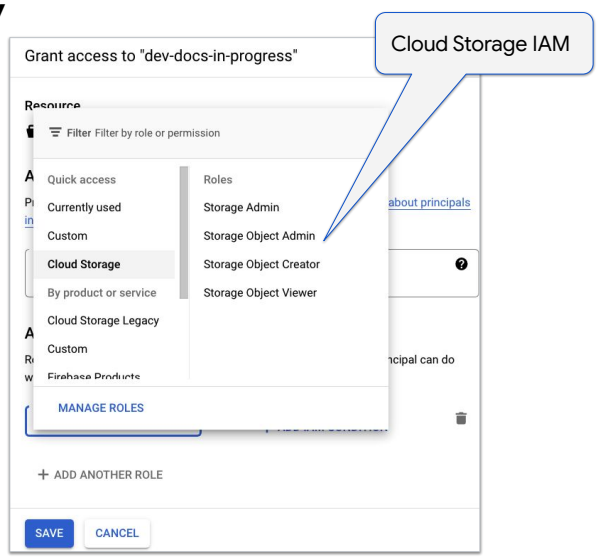
Google's client libraries also support IAM activities, so making API calls is typically not necessary if you use a client library

Cloud Storage Bucket Security

Permissions can be added to control access to buckets and objects

- Use IAM users and groups

4 best practices for ensuring privacy and security of your data in Cloud Storage



4 best practices for ensuring privacy and security of your data in Cloud Storage
https://cloud.google.com/blog/products/storage-data-transfer/google-cloud-storage-best-practices-to-help-ensure-data-privacy-and-security

Overview of access control:
https://cloud.google.com/storage/docs/access-control

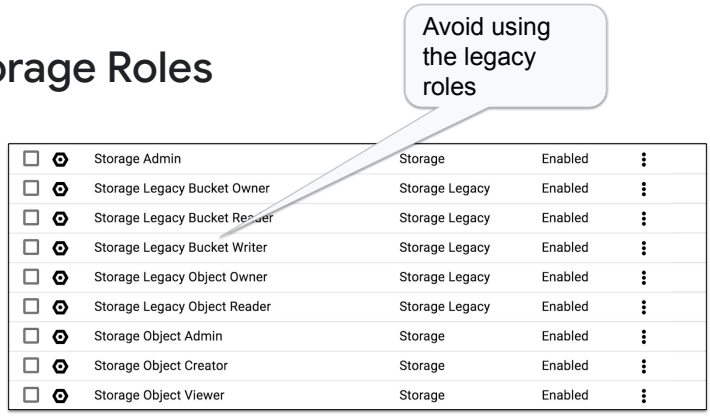
Identity and Access Management for Cloud Storage:
https://cloud.google.com/storage/docs/access-control/iam

As with everything in Google Cloud, Cloud Storage security is managed using IAM roles and members.

You can add IAM users and groups to buckets or objects and assign roles to those members. The roles determine what permissions they have. Note that there are a couple built-in groups for assigning public access to Storage—these are allUsers and allAuthenticatedUsers.

Predefined IAM Storage Roles

- Roles can be added to a principle or service account at the organization, folder, project or bucket level
  - Bucket level - applies to that specific bucket only



IAM permissions can be granted at the organization, folder, project or bucket level

Storage Role Permissions

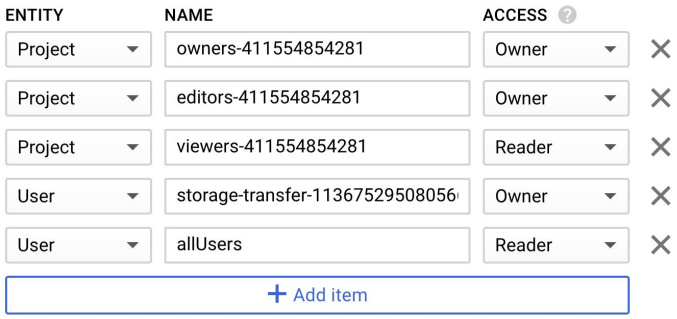
Storage Object Admin	Storage Object Creator	Storage Object Viewer
Description Full control of storage objects. 9 assigned permissions: • resourcemanager.projects.get • resourcemanager.projects.list • storage.objects.create • storage.objects.delete • storage.objects.get • storage.objects.getIamPolicy • storage.objects.list • storage.objects.setIamPolicy • storage.objects.update	Description Access to create objects in storage. 3 assigned permissions: • resourcemanager.projects.get • resourcemanager.projects.list • storage.objects.create	Description Read access to storage objects. 4 assigned permissions: • resourcemanager.projects.get • resourcemanager.projects.list • storage.objects.get • storage.objects.list

The Storage Object Viewer role grants read-only permissions to Storage. Storage Object Creator grants write permissions and Storage Object Admin grants full control over Cloud Storage Buckets and Objects.

Storage Object ACLs

Access Control Lists (ACLs) can be used to grant access to objects in buckets

Not best practice



Access control lists (ACLs):
https://cloud.google.com/storage/docs/access-control/lists

You can also grant read or read/write access to individual objects within a bucket using Access Control Lists. Google recommends to use IAM at the bucket level instead, for each of administration and security

Signed URLs

- Provide temporary access to buckets
  - Create a service account with rights to storage
  - Create a service account key
  - Use signurl command to create a URL that allows access to the resource
    - -d parameter is used to specify duration

```
gcloud iam service-accounts keys create ~/key.json --iam-account
storage-admin-sa@doug-demo-project.iam.gserviceaccount.com


gsutil signurl -d 10m ~/key.json gs://super-secure-bucket/noir.jpg
```

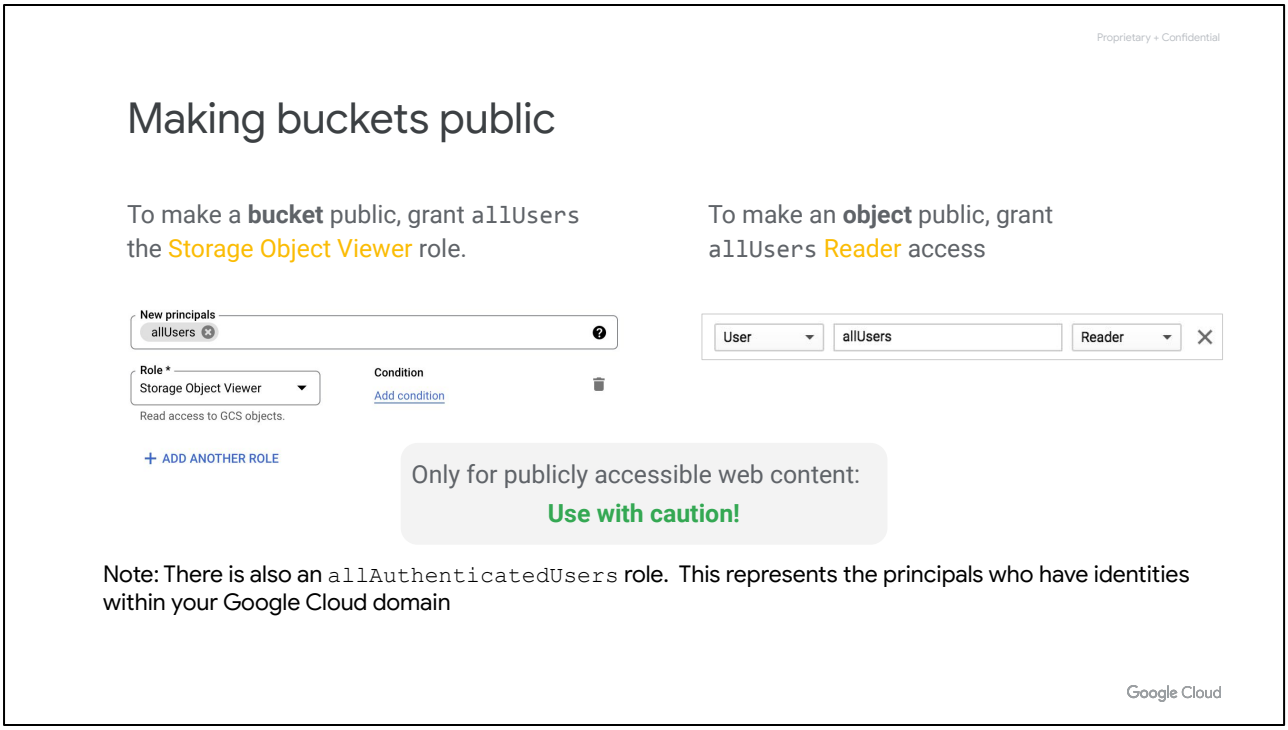

Signed URLs
https://cloud.google.com/storage/docs/access-control/signed-urls

Sometimes, you want to programatically grant temporary access to an object in a bucket. You can do this with a signed URL. Use the gsutil signurl command as shown on the screen. The -d parameter determines how long the signed URL works.

Signed URL Example Output

```
me@doug-demo-project:~$ gsutil signurl -d 10m ~/key.json gs://super-secure-bucket/noir.jpg
URL     HTTP Method     Expiration      Signed URL
gs://super-secure-bucket/noir.jpg       GET     2018-08-31 16:29:25     https://storage.google
apis.com/super-secure-bucket/noir.jpg?x-goog-signature=107d26e38f5c962296c26f4153a1cbeb61a84ac
a905009752e849f8f890de1f9a80e482da3bae562c7796389e12a8657a70c87860700149c4b2218c81ad3d57730cd3
5ced850b266cdffd84de01898ee8c807d742a85136e56f46d83c29ceb792bdd3a22adbe2e540ba27b0f565bbf8f31a
ee6ae61d6ae20968021d5a47c8d0aada43f2d32407f2977a4c7b4c66ef64ddd68bd6f6135936f847ace3530a968d72
63ff5e70f9fc39bf16fabbd472f63584a8d8c6b24b1f81859f1c5176b8e97580a6b4a7613ad76bfcdd403e6afc9a70
90a3e1b4cf95c7fb68142416af86ef5ef6bfab93c00492b307233180df9b3dfeefe1bb9a5bf81cb441f879ecc2e57c
def&x-goog-algorithm=GOOG4-RSA-SHA256&x-goog-credential=storage-admin-sa%40doug-demo-project.i
am.gserviceaccount.com%2F20180831%2Fus%2Fstorage%2Fgoog4_request&x-goog-date=20180831T201925Z&
x-goog-expires=600&x-goog-signedheaders=host
me@doug-demo-project:~$
```


A long URL will be generated. This URL provides access to the file in storage for the duration you specified in the command.



Make data public
https://cloud.google.com/storage/docs/access-control/making-data-public

In Cloud Storage it is possible to make entire buckets or individual objects public. An entire bucket can be made public by granting the **Storage Object Viewer** role to the **allUsers** group on the bucket.

To make individual objects in a bucket public, grant the **Reader** access to **allUsers** on the individual objects.

Be very careful on which objects or buckets you make public - use this option with **extreme caution**. Making buckets or objects public should only be done for publicly accessible web content.

Deny Policies

- Introduced in 2022
- Inherited through the resource hierarchy just like IAM allow policies
- Attached to project, folder or organization
- Denies override grants further down the hierarchy
- Currently, must be created via command line

First create a deny policy and store it in a file

People in the dev@example.com group are not allowed to create or delete service account keys

```
{
  "deniedPrincipals": [
    "principalSet://goog/group/dev@example.com"
  ],
  "deniedPermissions": [
    "iam.googleapis.com/serviceAccountKeys.create",
    "Iam.googleapis.com/serviceAccountKeys.delete"
  ]
}
```

Next apply the deny policy

```
gcloud iam policies create POLICY_ID \
    --attachment-point=[proj-id|folder-id|org-id] \
    --kind=denypolicies \
    --policy-file=POLICY_FILE
```


Deny policies
https://cloud.google.com/iam/docs/deny-overview

Use cases
https://cloud.google.com/iam/docs/deny-overview#use-cases

Exam Guide - Resource Hierarchy, Billing and IAM

1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

Exam Guide - Resource Hierarchy, Billing and IAM

5.2 Managing service accounts. Tasks include:

- 5.2.1 Creating service accounts
- 5.2.2 Using service accounts in IAM policies with minimum permissions
- 5.2.3 Assigning service accounts to resources
- 5.2.4 Managing IAM of a service account
- 5.2.5 Managing service account impersonation
- 5.2.6 Creating and managing short-lived service account credentials

1.2 Managing billing configuration. Activities include:

- 1.2.1 Creating one or more billing accounts
- 1.2.2 Linking projects to a billing account
- 1.2.3 Establishing billing budgets and alerts
- 1.2.4 Setting up billing exports

Who ...?

Saw this slide earlier

Next topic: Service accounts

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

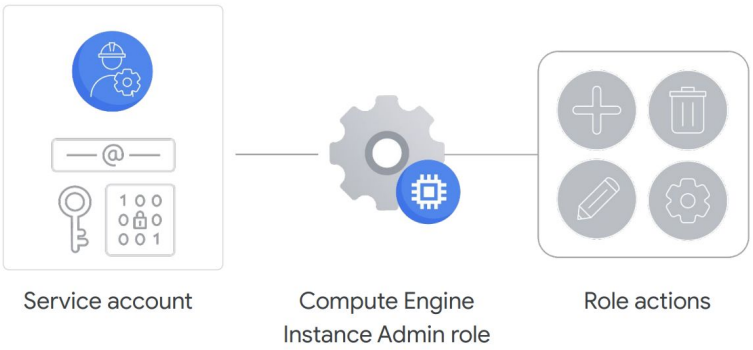
- Google account
- Google group
- **Service account**
- Google Workspace or Cloud Identity domain

Also

- allAuthenticatedUsers
- allUsers

Service accounts are the *only* identities created internally within Google Cloud

- Created in the form of an email address
- Assigned IAM roles
- "Attached" to a resource



Nice set of YouTube videos explaining service accounts:
What are service accounts:
https://www.youtube.com/watch?v=xXk1YlkKW_k&list=PLIivdWyY5sqIlPnZ7cvkg2Ck-8ZZ8TA5t&index=1

Creating, managing and retiring Service accounts:
https://www.youtube.com/watch?v=2TdLmI3G5Rc&list=PLIivdWyY5sqIlPnZ7cvkg2Ck-8ZZ8TA5t&index=2

How to secure your service accounts:
https://www.youtube.com/watch?v=CTcQNPWNRkE&list=PLIivdWyY5sqIlPnZ7cvkg2Ck-8ZZ8TA5t&index=3

Service account keys and impersonation:
https://www.youtube.com/watch?v=SDhMwyyd9_0&list=PLIivdWyY5sqIlPnZ7cvkg2Ck-8ZZ8TA5t&index=4

Service accounts in action:
https://www.youtube.com/watch?v=UhYqag7Xjhw&list=PLIivdWyY5sqIlPnZ7cvkg2Ck-8ZZ8TA5t&index=5

There are two types of Service Accounts

Google-managed service accounts

- Created and managed by Google
- Used by Google to perform operations on your behalf, for example
  - Moving a VM during a maintenance event
- These activities are logged to Cloud Logging for full transparency
- Leave these alone and do not modify their roles
  - Some are not visible in the Console

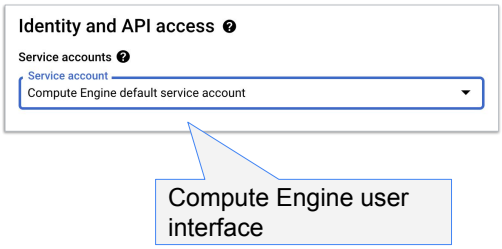
User-managed service accounts

- Created and managed by someone with **Service Account Admin** or **Owner** IAM
  - Are given various IAM roles depending on the use case
- Attached to services, which limits those services to the actions allowed by the roles assigned to the service account

Types of service accounts:
https://cloud.google.com/iam/docs/service-accounts#type s

Service accounts are identities for Google Cloud services

- Special type of account intended to represent a non-human user that needs to authenticate and be authorized to access data
  - Are created *within* Google Cloud, unlike other Principals
- Attached to VMs or other services
  - Applications will only be allowed to perform actions allowed by the roles given to the service account



Compute Engine user interface

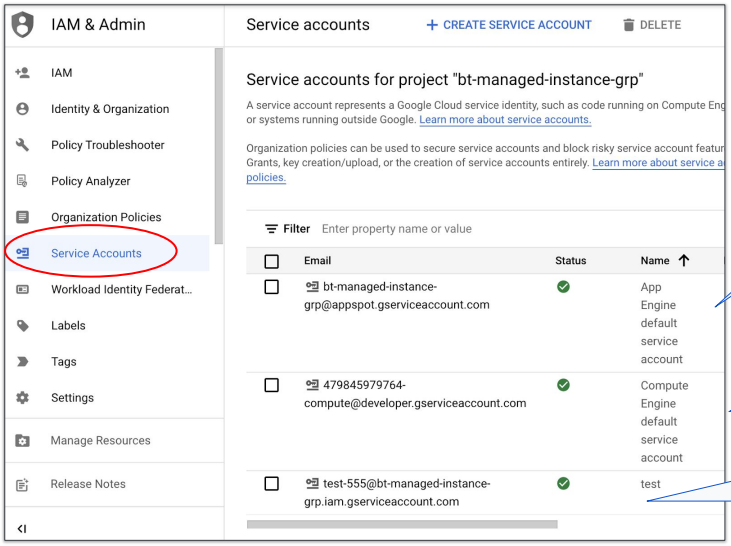
Understanding Service Accounts

Understanding service accounts:
https://cloud.google.com/iam/docs/service-accounts

Service account use cases

- Service accounts can be used in scenarios such as:
  - Running workloads on virtual machines (VMs)
    - E.g., Create a service account with permissions to query BigQuery
      - Attach it to a VM
      - Deploy an application onto the VM that submits SQL commands to BigQuery
  - Running workloads on on-premises workstations or data centers that call Google APIs.
    - E.g., Same example as above, but now the application is running on-premise
  - Running workloads which are not tied to the lifecycle of a human user.
    - E.g., Batch jobs that are scheduled to run periodically

Viewing Service Accounts in the Console



Created by Google when App Engine API is enabled

Created by Google when Compute Engine API is enabled

Custom service account. Individuals need Service Account Admin or Editor role to create them.

Viewing Service Accounts with the CLI


```
gcloud iam service-accounts list
```

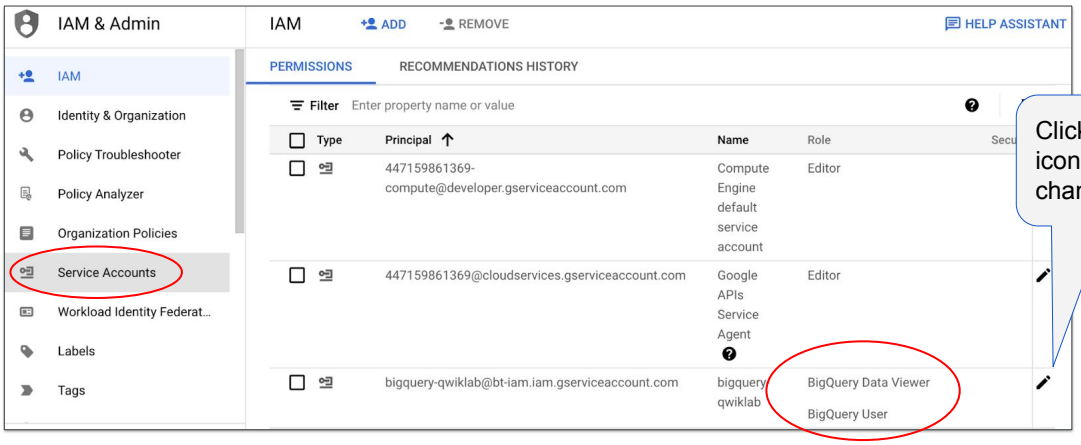
Output:

```
DISPLAY NAME: Compute Engine default service account
EMAIL: 479845979764-compute@developer.gserviceaccount.com
DISABLED: False

DISPLAY NAME: test
EMAIL: test-555@bt-managed-instance-grp.iam.gserviceaccount.com
DISABLED: False

DISPLAY NAME: App Engine default service account
EMAIL: bt-managed-instance-grp@appspot.gserviceaccount.com
DISABLED: False
```

Viewing Service Account IAM in the Console



Viewing Service Accounts IAM with the CLI

Command:

```
gcloud projects get-iam-policy [project-id]
gcloud projects get-iam-policy [project-id] --format json > mypolicy.json &&
cat mypolicy.json
```

Same command that is used for other types of principals.

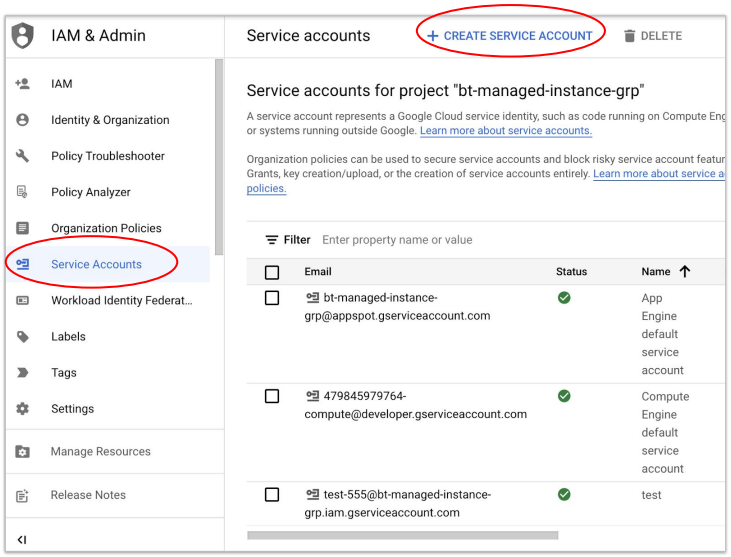
Output*:

```
bindings:
- members:
  - serviceAccount:bigquery-qwiklab@bt-iam.iam.gserviceaccount.com
  role: roles/bigquery.dataViewer
- members:
  - serviceAccount:bigquery-qwiklab@bt-iam.iam.gserviceaccount.com
  role: roles/bigquery.user
- members:
  - user:binglee@developers-townsendandassociates.com
  role: roles/compute.admin
```

*Complete output not shown

Creating Service Accounts in the Console

IAM & Admin > Service Accounts> Create Service account

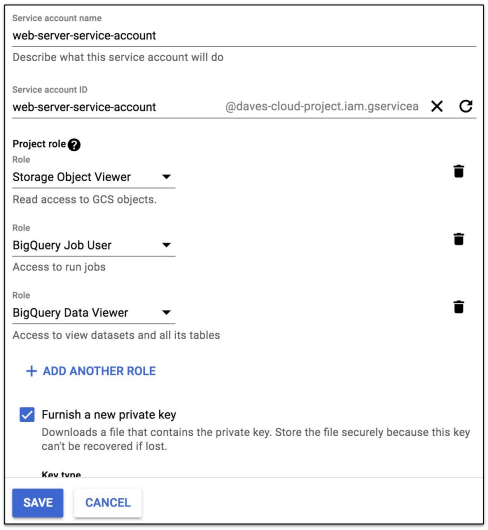


Creating service accounts
https://cloud.google.com/iam/docs/creating-managing-service-accounts

Creating Service Accounts (continued)

Service account name →

Add one or more roles →



Here is an example of creating a service account.

First, give your service account a name. A best practice is to give the account a name that can be used to easily determine the purpose of the service account.

Next, you add one or more roles to associate with the service account. The roles will determine the permissions associated with the account.

CLI - Creating Service Accounts at the command line

Syntax:

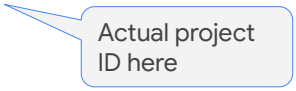
```
gcloud iam service-accounts create NAME
[--description=DESCRIPTION] [--display-name=DISPLAY_NAME]
```

Example:

```
gcloud iam service-accounts create myserviceacct
--display-name="My Service Account"
```

Resulting account:

```
myserviceacct@[project-id].iam.gserviceaccount.com
```



Creating service accounts
https://cloud.google.com/iam/docs/creating-managing-service-accounts

CLI - Assigning Roles to Service Accounts

Same command as used with other types of principals

Syntax:

```
gcloud projects add-iam-policy-binding [PROJECTID]
--member=serviceAccount:[serviceaccount here] --role [ROLE-ID]
```

Example:

```
gcloud projects add-iam-policy-binding my-project \
--member=serviceAccount:myserviceacct@[projid].iam.gserviceaccount.com
--role=roles/storage.objectViewer
```


List and edit service accounts
https://cloud.google.com/iam/docs/service-accounts-list-edit

Granting minimum permissions to service accounts:
https://cloud.google.com/iam/docs/understanding-service-accounts#granting_minimum

Service account permissions for common scenarios:
https://cloud.google.com/iam/docs/understanding-service-accounts#sa_common

Assigning service accounts to resources

- Service accounts can be attached to multiple services in Google Cloud, including
  - Compute Engine (and Kubernetes Engine)
    - Default service account is created by Google when API is enabled
  - App Engine
    - Default service account is created by Google when API is enabled
  - Cloud Run
    - Uses the Compute Engine default service account
  - Cloud Functions
    - Uses the App Engine default service account
- **Best practice: Create custom service accounts for all services**

How compute engine uses service accounts:
https://cloud.google.com/compute/docs/access/service-accounts

Creating and enabling service accounts for instances:
https://cloud.google.com/compute/docs/access/create-enable-service-accounts-for-instances

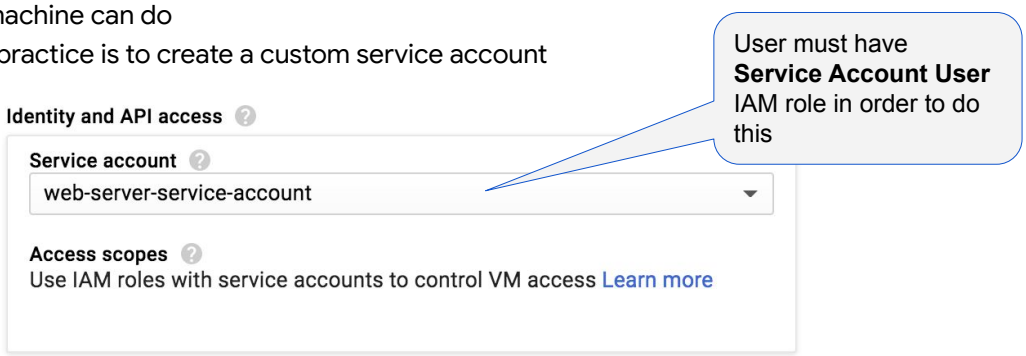
How service accounts work with Compute Engine:
https://cloud.google.com/compute/docs/access/service-accounts

Attaching a service account to a CE instance:
https://cloud.google.com/compute/docs/access/service-accounts#associating_a_service_account_to_an_instance

Authenticating Applications using service account credentials:
https://cloud.google.com/compute/docs/access/create-enable-service-accounts-for-instances#authenticating_applications_using_service_account_credentials

Example: Assigning Service Accounts to a VM

- Default service account is a project editor which requires scopes to control what the machine can do
- Best practice is to create a custom service account



Service accounts can be used to control the level of access a virtual machine can have to other Google Cloud services.

The default service account for a virtual machine gives it project editor privileges. However, the default service account can be overridden by using scopes to set the permissions per service. (Note: Access scopes are a legacy method of specifying authorization for your instance. As such, they will not be further discussed. https://cloud.google.com/compute/docs/access/service-accounts#accesscopesiam)

A better alternative is be to create a new service account, grant it the appropriate IAM roles, and select it when creating a virtual machine.

Exam Guide - Resource Hierarchy, Billing and IAM

5.2 Managing service accounts. Tasks include:

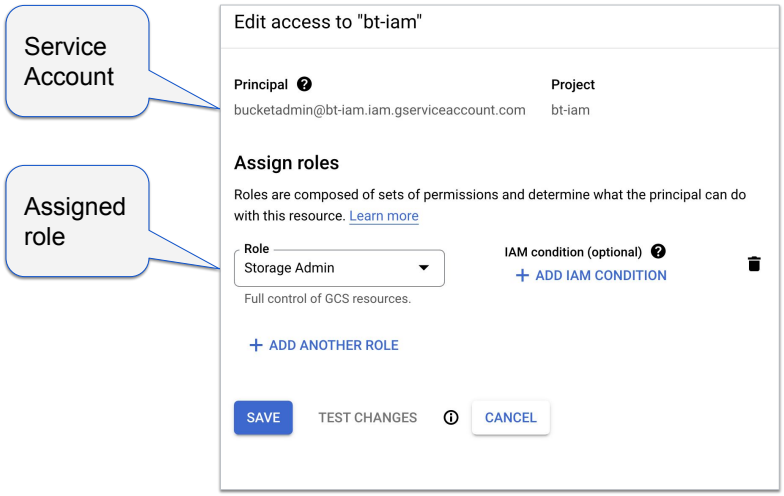
- 5.2.1 Creating service accounts
- 5.2.2 Using service accounts in IAM policies with minimum permissions
- 5.2.3 Assigning service accounts to resources
- 5.2.4 Managing IAM of a service account
- 5.2.5 Managing service account impersonation
- 5.2.6 Creating and managing short-lived service account credentials

1.2 Managing billing configuration. Activities include:

- 1.2.1 Creating one or more billing accounts
- 1.2.2 Linking projects to a billing account
- 1.2.3 Establishing billing budgets and alerts
- 1.2.4 Setting up billing exports

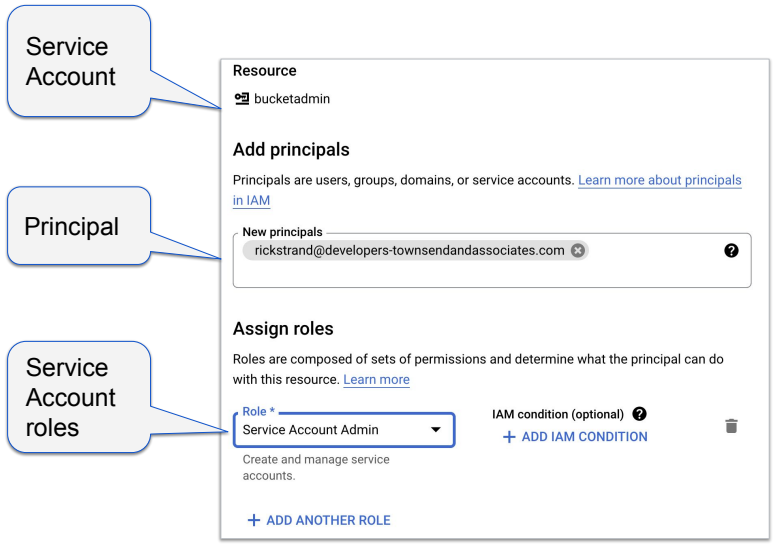
Service accounts are both principals and resources

Is a principal when roles are assigned to it



Service accounts are both principals and resources

Is a resource when users (principals) are given Service Account IAM roles to manage the service account in some way



Service Account predefined roles (not a complete list)

- Assigned to principles
  - Service Account Admin
    - Create and manage service accounts
  - Service Account User
    - Can attach service account to resources (e.g., Compute Engine)
    - Can “impersonate” the service account and perform the tasks allowed by IAM given to the service account
  - Service Account Key Admin
    - Create and manage (and rotate) service account keys
    - Keys are used by applications external to Google Cloud
  - Service Account Token Creator
    - Short lived credentials represented as OAuth 2.0 access tokens, OpenID Connect ID tokens, self-signed JSON Web Tokens (JWTs), and self-signed binary objects (blobs)

All of these are discussed on the next few slides

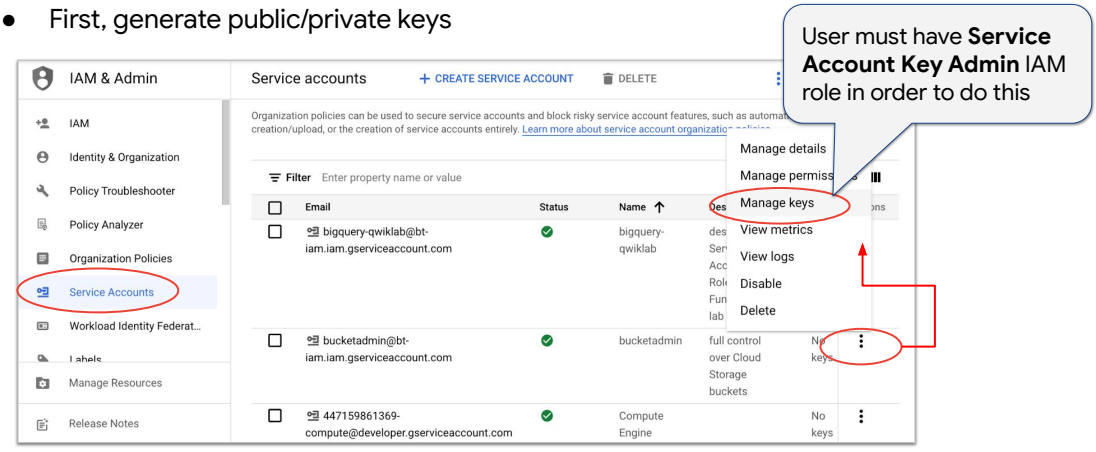
Accessing Google Cloud resources from an external application

- Any user, group, application, etc. must authenticate prior to accessing Google Cloud services*
  - This also applies to external (e.g., on-premise) applications
    - They use service accounts for this purpose
- To authenticate as a service account, applications must use either use
  - Service Account Keys
  - Service Account Tokens
- Both are discussed next

*Exception: When authentication is not required, e.g., a public website hosted in Google Cloud

Service Account keys can be used by external applications when authenticating to Google Cloud

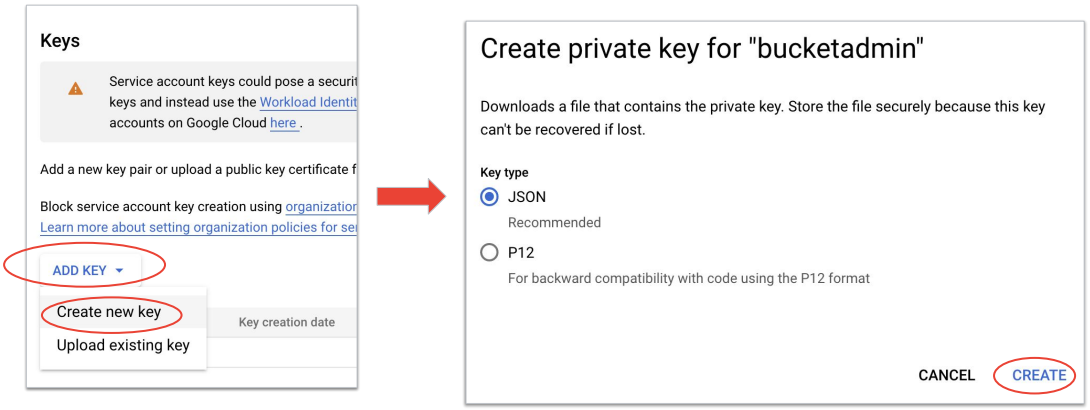
- First, generate public/private keys



Create and manage service account keys
https://cloud.google.com/iam/docs/creating-managing-service-account-keys

Service Account keys stored locally must be secured

- Next, download the private key
  - Public key is kept in Google Cloud
  - **Customer is responsible for storing the private key securely**



Create and manage service account keys
https://cloud.google.com/iam/docs/creating-managing-service-account-keys

Using a Service Account key with code

- To configure gcloud to use a service account:

```
$ gcloud auth activate-service-account \
test-service-account@google.com \
        --key-file=/path/key.json --project=testproject
```

Path where the service account key is stored

- To use credentials in your code (*Python is this example*):
  - The SDK will automatically look for the environment variable value

```
import os
from google.cloud.bigquery.client import Client

os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'path_to_json_file'
bq_client = Client()
```


The top example shows how to configure gcloud to use a service account by linking gcloud to a key that was created.

The bottom example shows how to use your keys via code. In this example, Python.

The SDK will automatically look for the value in the GOOGLE_APPLICATION_CREDENTIALS environment variable.

Using short-lived credentials with a Service Account

- While Service Account keys work for application authentication, they are not the preferred method
  - The private Service Account **key must be secured by the customer** after downloading
    - An unauthorized person with access to the key can authenticate as the Service Account, and do whatever the Service Account's IAM roles allows
  - Service account **keys have an unlimited lifetime**
- The **better alternative is to create short-lived credentials via Service Account tokens**
  - Principals (including Service Accounts) must have **Service Account Token Creator** role in order to create tokens
  - Different types of token are supported
    - OAuth 2.0 access tokens, OpenID Connect ID tokens, self-signed JSON Web Tokens (JWTs), and self-signed binary objects (blobs)

Best practices for managing service account keys
https://cloud.google.com/iam/docs/best-practices-for-managing-service-account-key

Create short-lived credentials for a service account
https://cloud.google.com/iam/docs/create-short-lived-credentials-direct

Types of tokens

- OAuth 2.0 access tokens
  - Used by an application to authenticate to Google APIs
- OpenID Connect (OIDC) tokens
  - Used when
    - Accessing a Cloud Run service
    - Invoking a Cloud Function
    - Authenticating a user to an application secured by Identity-Aware Proxy (IAP)
    - Making a request to an API deployed with API Gateway or Cloud Endpoints
- Self-signed JSON Web Tokens (JWTs)
  - Used to authenticate communication between services in a microservices architecture
  - Used by an application to authenticate to Google APIs (e.g. FireStore)
- Self-signed binary objects (blobs)
  - Used to securely identify the issuer of a request to a Google Cloud Storage bucket

Typical token lifespan is 1 hour, but can vary by type

This is a Developer topic, not an Engineer

Shown for completeness

Further discussion of creating tokens is outside of the scope of the exam

Token types
https://cloud.google.com/docs/authentication/token-types

Using OAuth 2.0 to Access Google APIs
https://developers.google.com/identity/protocols/oauth2

Generate an ID token by impersonating a service account
https://cloud.google.com/docs/authentication/get-id-token#impersonation

Allowing principals to impersonate service accounts

- Impersonation allows principles to indirectly access all the resources that the service account can access.
- **The roles that allow impersonation are**
  - **Service Account User**
    - This role also allows a principal to attach the service account to a resource, e.g., a VM
  - **Service Account Token Creator**
    - Can impersonate the service account and create tokens
  - **Workload identity User**
    - Can impersonate service accounts from GKE workloads
      - Kubernetes service accounts are not the same as IAM service accounts
      - Create IAM service accounts with roles to access Google Cloud resources
      - Map the Kubernetes service account to the IAM service account

Managing service account impersonation
https://cloud.google.com/iam/docs/impersonating-service-accounts

Manage access to service accounts:
https://cloud.google.com/iam/docs/manage-access-service-accounts

Attaching a service account to a resource:
https://cloud.google.com/iam/docs/impersonating-service-accounts#attaching-to-resources

Best practices for using and managing service accounts:
https://cloud.google.com/iam/docs/best-practices-for-using-and-managing-service-accounts

Impersonation use case

- Service account impersonation is useful for delegating permissions to perform certain tasks to service accounts, rather than granting them to individual users
  - For example, a developer needs to troubleshoot a Cloud Run deployment
    - The developer has no IAM roles for Cloud Run
    - A service account exists with the Cloud Run Admin role
    - Allow the developer to “act-as” the service account by granting the **Service Account Token Creator** role on that service account
      - Add a condition to limit to the grant to a certain time period if desired
    - The developer can impersonate the service account for troubleshooting
    - When done, the developer reverts back to their normal IAM roles

Managing service account impersonation
https://cloud.google.com/iam/docs/impersonating-service-accounts

Example: Using Service Account Token Creator role

- Create a Service Account with Cloud Storage Admin role

```
gcloud iam service-accounts create storage-admin \
    --display-name="Storge Admin"
gcloud projects add-iam-policy-binding bt-iam \
    --member=serviceAccount:storage-admin@bt-iam.iam.gserviceaccount.com \
    --role=roles/storage.admin
```

- Give the Service Account Token Creator to a user

```
gcloud beta iam service-accounts add-iam-policy-binding \
   storage-admin@bt-iam.iam.gserviceaccount.com \
   --member=user:caliagusta@developers-townsendandassociates.com \
   --role=roles/iam.serviceAccountTokenCreator
```

- Sign into the command line as the user. Can they create a bucket?
  - No, unless IAM exists elsewhere which allows them to do so

```
gcloud storage buckets create gs://cali-bucket2
```

This should fail

Example: Using Service Account Token Creator role - continued

- Impersonate the Service Account and create a bucket

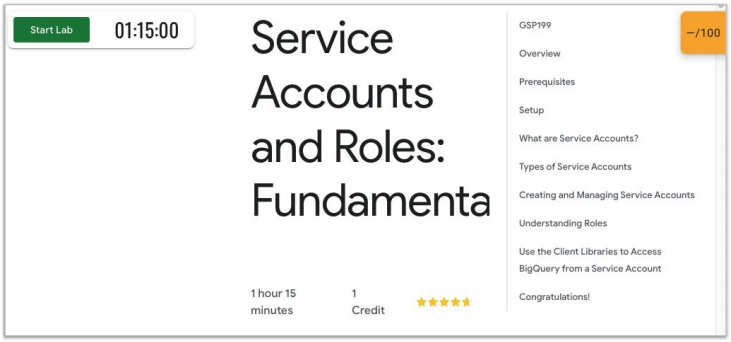
```
gcloud storage buckets create gs://cali-bucket4 \
    --impersonate-service-account=storage-admin@bt-iam.iam.gserviceaccount.com
```

No service account key needed

Suggested lab: Service Accounts and Roles: Fundamentals

In this lab you will

- Create a service account
  - IAM - BigQuery Data Viewer & BigQueryUser
- Attach it to a VM
- Create and run Python app to run a SQL query in BigQuery



https://partner.cloudskillsboost.google/catalog_lab/956

Exam Guide - Resource Hierarchy, Billing and IAM

5.2 Managing service accounts. Tasks include:

- 5.2.1 Creating service accounts
- 5.2.2 Using service accounts in IAM policies with minimum permissions
- 5.2.3 Assigning service accounts to resources
- 5.2.4 Managing IAM of a service account
- 5.2.5 Managing service account impersonation
- 5.2.6 Creating and managing short-lived service account credentials

1.2 Managing billing configuration. Activities include:

- 1.2.1 Creating one or more billing accounts
- 1.2.2 Linking projects to a billing account
- 1.2.3 Establishing billing budgets and alerts
- 1.2.4 Setting up billing exports

Exam Guide - Resource Hierarchy, Billing and IAM

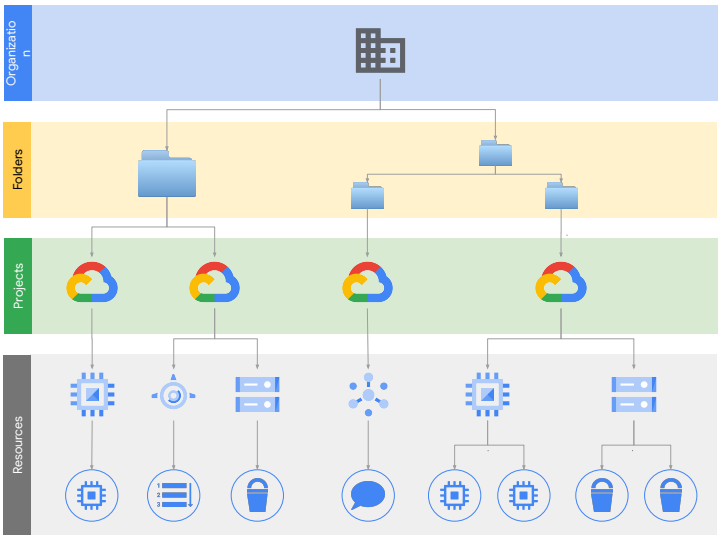
1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

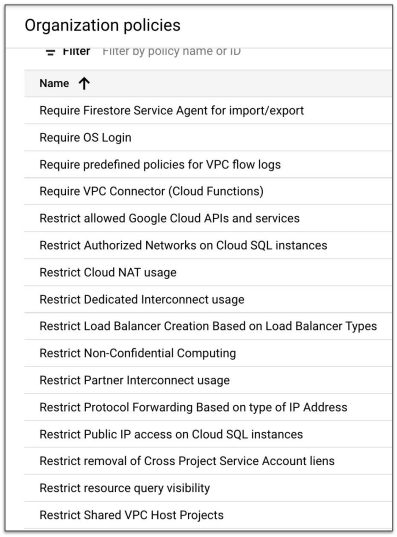
Resource Hierarchy



Next discussion:
Organization
Policies

Organizational policies

- An organization policy is a **restriction** or constraint that you can set over the use of a service
- For example
  - Restrict what type of VMs developers can create
  - Restrict what Google regions resources can be created in
  - Restrict the use of public IPs to some VMs (or none)
- Are set at the organization level, folder level or project level
- Define and establish guardrails for your development teams to stay within compliance boundaries.



Introduction to the Organization Policy Service

Introduction to the Organization Policy Service
https://cloud.google.com/resource-manager/docs/organization-policy/overview

Organization policy constraints - list :
https://cloud.google.com/resource-manager/docs/organization-policy/org-policy-constraints

Restricting resource usage:
https://cloud.google.com/resource-manager/docs/organization-policy/restricting-resources

Organization policy constraints

- Configured with constraints
  - Particular type of restriction against either a Google Cloud service or a group of Google Cloud services
- Descendants of the targeted resource hierarchy node inherit the organization policy.

```
resource: "organizations/[ORG_ID]"
policy: {
  constraint: “constraints/compute.skipDefaultNetworkCreation”
  booleanPolicy: {
    enforced: true
  }
}
```


In order to define an organization policy, you choose a constraint, which is a particular type of restriction against either a Google Cloud service or a group of Google Cloud services. You configure that constraint with your desired restrictions.

Descendants of the targeted resource hierarchy node inherit the organization policy. By applying an organization policy to the root organization node, you are able to effectively drive enforcement of that organization policy and configuration of restrictions across your organization.

The example on the slide shows how an organization policy that’s used to disable the creation of the default network

Exam Guide - Resource Hierarchy, Billing and IAM

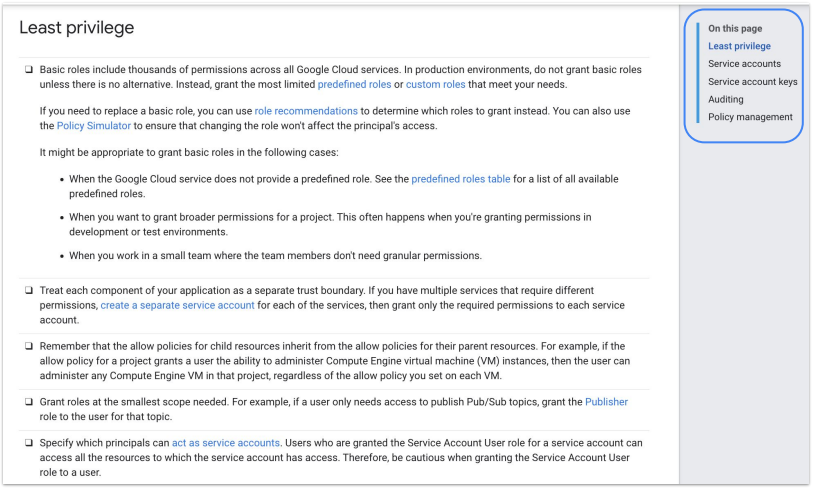
1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

Security best practices when using IAM



Least privilege

- Basic roles include thousands of permissions across all Google Cloud services. In production environments, do not grant basic roles unless there is no alternative. Instead, grant the most limited predefined roles or custom roles that meet your needs.

  If you need to replace a basic role, you can use role recommendations to determine which roles to grant instead. You can also use the Policy Simulator to ensure that changing the role won't affect the principal's access.

  It might be appropriate to grant basic roles in the following cases:

  - When the Google Cloud service does not provide a predefined role. See the predefined roles table for a list of all available predefined roles.
  - When you want to grant broader permissions for a project. This often happens when you're granting permissions in development or test environments.
  - When you work in a small team where the team members don't need granular permissions.

- Treat each component of your application as a separate trust boundary. If you have multiple services that require different permissions, create a separate service account for each of the services, then grant only the required permissions to each service account.
- Remember that the allow policies for child resources inherit from the allow policies for their parent resources. For example, if the allow policy for a project grants a user the ability to administer Compute Engine virtual machine (VM) instances, then the user can administer any Compute Engine VM in that project, regardless of the allow policy you set on each VM.
- Grant roles at the smallest scope needed. For example, if a user only needs access to publish Pub/Sub topics, grant the Publisher role to the user for that topic.
- Specify which principals can act as service accounts. Users who are granted the Service Account User role for a service account can access all the resources to which the service account has access. Therefore, be cautious when granting the Service Account User role to a user.

On this page
Least privilege
Service accounts
Service account keys
Auditing
Policy management

A “must read” for anyone responsible for IAM

https://cloud.google.com/iam/docs/using-iam-securely

Use IAM securely
https://cloud.google.com/iam/docs/using-iam-securely

Exam Guide - Resource Hierarchy, Billing and IAM

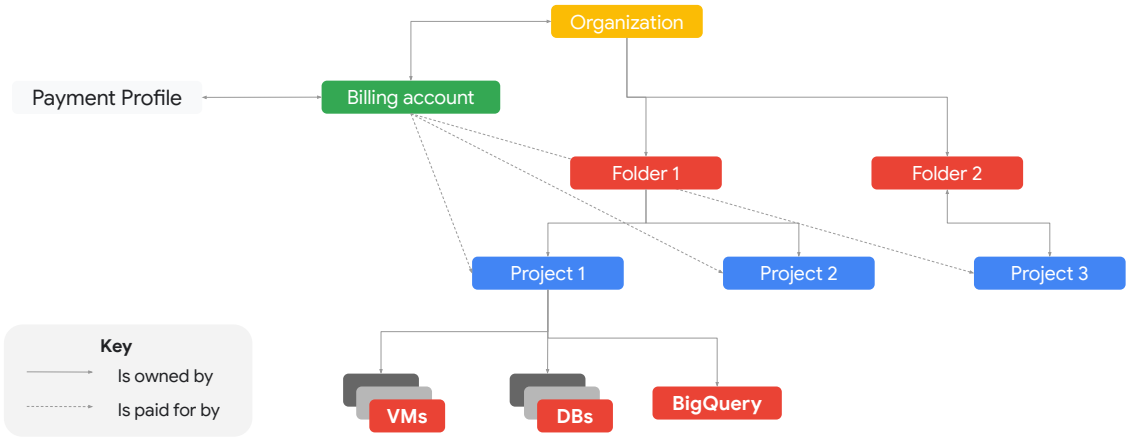
5.2 Managing service accounts. Tasks include:

- 5.2.1 Creating service accounts
- 5.2.2 Using service accounts in IAM policies with minimum permissions
- 5.2.3 Assigning service accounts to resources
- 5.2.4 Managing IAM of a service account
- 5.2.5 Managing service account impersonation
- 5.2.6 Creating and managing short-lived service account credentials

1.2 Managing billing configuration. Activities include:

- 1.2.1 Creating one or more billing accounts
- 1.2.2 Linking projects to a billing account
- 1.2.3 Establishing billing budgets and alerts
- 1.2.4 Setting up billing exports

Cloud Billing resource hierarchy



Overview:
https://cloud.google.com/billing/docs/concepts

Billing documentation:
https://cloud.google.com/billing/docs/

Key points:

- A billing account is therefore **attached to an organization** and **pays for the projects** which are **attached** to it.
- A **single billing account** can **pay for all** your projects.
- **Google recommends using a single billing account as far as possible**.
- Note: It is possible, but not recommended to use multiple billing accounts:
  Does the customer require very strong fiscal isolation for legal or administrative purposes?
  - Can the necessary isolation be achieved in another way?
    - The common use cases for **multiple billing accounts** are:
      - **Re-sellers**
      - Using resources which require **payment in a different currency**
      - Companies with a **legal entity in another country** with **different tax requirements.**
      - **Large multinational organizations** with **multiple orgs**

- and billing accounts (essentially different companies

Understanding billing

- To manage billing accounts and to add projects to them, you must be a billing administrator.
- When you create a new project, you're prompted to choose which of your billing accounts you want to link to the project
  - If you have only one billing account, that account is automatically linked to your project.
- If you don't have a billing account, you must create one and enable billing for your project before you can use many Google Cloud features.

IAM roles for billing-related job functions:
https://cloud.google.com/iam/docs/job-functions/billing

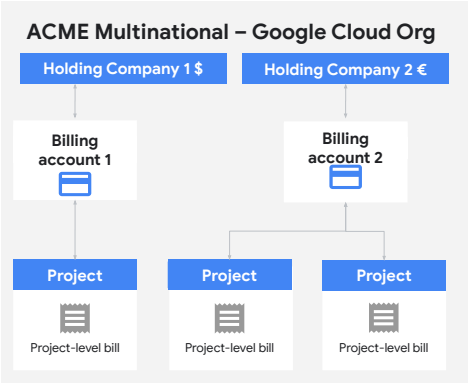
All billing tasks:
https://cloud.google.com/billing/docs/how-to/account-management-overview

Create, modify, or close your self-serve Cloud Billing account:
https://cloud.google.com/billing/docs/how-to/manage-billing-account

Enable, disable, or change billing for a project:
https://cloud.google.com/billing/docs/how-to/modify-project

Linking projects - gcloud command:
https://cloud.google.com/sdk/gcloud/reference/beta/billing/projects/link

Multiple billing accounts



It is possible to use multiple billing accounts in Google Cloud, however this feature is suitable only for specific customers.

1. Customer is a Google Cloud reseller (Cloud Billing account for each customer for separate invoices)
2. Customer's organization is made up of separate operating companies which have strong fiscal and administrative divisions.
3. Customer needs to pay for different parts of the Google Cloud in different currencies (for example, an American division and an European division)

Anti-pattern

Multiple billing accounts for chargeback: Billing accounts are not a good solution for chargeback purposes. Labels are far more flexible, granular, and efficient.

This slide stresses the **difficulties and complexity in using multiple billing accounts**. While there is no technical issue in having multiple billing accounts, **Google recommends using a single billing account as far as possible**.

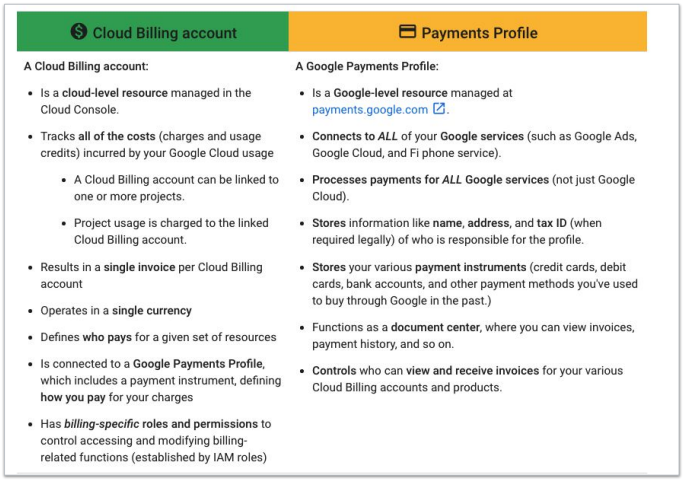
Key points:

- **Google recommends using a single billing account as far as possible**. It is possible to use multiple billing accounts, however this is suitable only for specific customers.
- The common use cases for **multiple billing accounts** are:
  - **Re-sellers - Customers which resell Google Cloud services and need billing accounts for each customer so that they have different invoices.**
  - **Large multinational organizations, with different operating companies which have strong fiscal and administrative divisions** (potentially different tax requirements)
  - **Customer needs to pay for different parts of the Google Cloud in different currencies. Eg. An American division and an European division**
- Important note:
  - Multiple billing accounts come with added overhead and administration. You will have separate invoices and separate billing

    - accounts to manage, including IAM permissions. In addition, customers using tiered pricing products (such as networking products) will pay more for their Google Cloud usage. This is because pricing for these products starts higher and then decreases as usage goes up. With multiple billing accounts, the higher use, cheaper tiers will be reached less quickly. This cost can be significant if usage is large.
- A common customer reason for wanting multiple billing accounts is for chargeback; Charging back Google Cloud usage to internal customers - eg. department x spent y on Google Cloud resources, therefore that amount will be charged back. The customer envisions having separate billing accounts for each department or cost center.
    - This is **bad** solution because:
        - The complexities and higher cost already mentioned
        - It's hard to change in the future. What if two departments merge? What if a department moves cost center etc.? In both cases, either a billing account must be closed or a new billing account set up. While this is technically possible, setting up a new invoiced billing account is not a ‘1 click’ task and can take several days to transition from online to offline.
        - You are limited in the aggregations you can see by the level of the billing account. If your billing accounts are at department level, you may want to aggregate by team
        - Labels are a much more efficient solution. By applying a range of labels to aggregate against, cost can be easily aggregated in any number of ways. Changing a label is simple. From there you can either manually charge back to departments or develop integration with your existing financials tool.

Payment Profile

- Managed outside of Google Cloud



Cloud Billing account	Payments Profile
A Cloud Billing account:	**A Google Payments Profile:**

• Is a **cloud-level resource** managed in the Cloud Console.
• Tracks **all of the costs** (charges and usage credits) incurred by your Google Cloud usage
  ◦ A Cloud Billing account can be linked to one or more projects.
  ◦ Project usage is charged to the linked Cloud Billing account.
• Results in a **single invoice** per Cloud Billing account
• Operates in a **single currency**
• Defines **who pays** for a given set of resources
• Is connected to a **Google Payments Profile**, which includes a payment instrument, defining **how you pay** for your charges
• Has ***billing-specific*** **roles and permissions** to control accessing and modifying billing-related functions (established by IAM roles)
	• Is a **Google-level resource** managed at payments.google.com.
	• **Connects to *ALL*** of your **Google services** (such as Google Ads, Google Cloud, and Fi phone service).
	• **Processes payments for *ALL* Google services** (not just Google Cloud).
	• **Stores** information like **name**, **address**, and **tax ID** (when required legally) of who is responsible for the profile.
	• **Stores** your various **payment instruments** (credit cards, debit cards, bank accounts, and other payment methods you've used to buy through Google in the past.)
	• Functions as a **document center**, where you can view invoices, payment history, and so on.
	• **Controls** who can **view and receive invoices** for your various Cloud Billing accounts and products.

About Cloud Billing accounts and payments profiles
https://cloud.google.com/billing/docs/concepts#billing_account

Exam Guide - Resource Hierarchy, Billing and IAM

5.2 Managing service accounts. Tasks include:

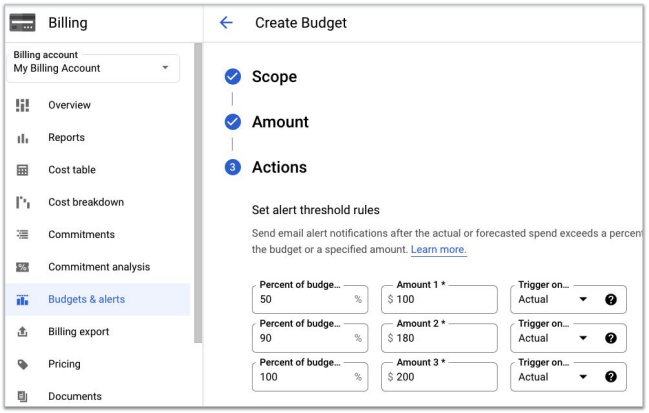
- 5.2.1 Creating service accounts
- 5.2.2 Using service accounts in IAM policies with minimum permissions
- 5.2.3 Assigning service accounts to resources
- 5.2.4 Managing IAM of a service account
- 5.2.5 Managing service account impersonation
- 5.2.6 Creating and managing short-lived service account credentials

1.2 Managing billing configuration. Activities include:

- 1.2.1 Creating one or more billing accounts
- 1.2.2 Linking projects to a billing account
- 1.2.3 Establishing billing budgets and alerts
- 1.2.4 Setting up billing exports

Understanding budgets and alerts

- Monitor all your Google Cloud charges in one place
- Apply budget alerts to either a billing account or one or more projects
- Can specify a budget amount or match it to the previous month's spend.
- Setting a budget does ***not*** cap API usage.
    - Services continue to operate and accrue costs, even if a budget alert has been triggered.



Create, edit, or delete budgets and budget alerts:
https://cloud.google.com/billing/docs/how-to/budgets

Exam Guide - Resource Hierarchy, Billing and IAM

5.2 Managing service accounts. Tasks include:

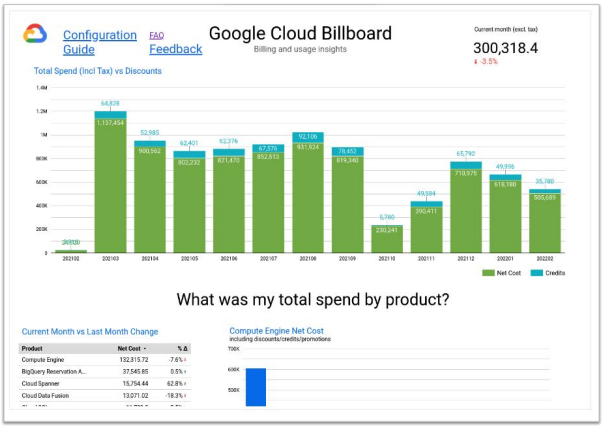
- 5.2.1 Creating service accounts
- 5.2.2 Using service accounts in IAM policies with minimum permissions
- 5.2.3 Assigning service accounts to resources
- 5.2.4 Managing IAM of a service account
- 5.2.5 Managing service account impersonation
- 5.2.6 Creating and managing short-lived service account credentials

1.2 Managing billing configuration. Activities include:

- 1.2.1 Creating one or more billing accounts
- 1.2.2 Linking projects to a billing account
- 1.2.3 Establishing billing budgets and alerts
- 1.2.4 Setting up billing exports

Export Billing Data to BigQuery

- Automatically export detailed billing data (usage, cost estimates, pricing data, etc.) to a BigQuery dataset
  - Data Studio* can be used for visualization
- Setup billing exports in the Billing section of the Cloud Console



*Data Studio and Looker have been consolidated into one product: [Looker Studio]

Visualize your costs with Looker Studio
https://cloud.google.com/billing/docs/how-to/visualize-data

Initial setup: Exporting to BigQuery

Best practice — Set up Cloud Billing export to BigQuery as early as possible for retention, easy analysis, and visualization

Setup

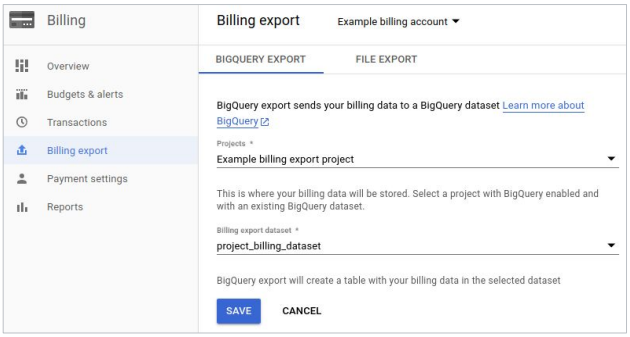
Simple one-time setup exports all billing data attached to the account

Project

Project to host the BigQuery dataset

Billing export dataset

- Dataset where table will be created
- If none exist, you can create one



Export to BigQuery:
https://cloud.google.com/billing/docs/how-to/export-data-bigquery

Example queries for Cloud Billing data export:
https://cloud.google.com/billing/docs/how-to/bq-examples

Different topic, but related to billing - Labeling resources
Creating and managing labels:
https://cloud.google.com/resource-manager/docs/creating-managing-labels

Key points:

- **Billing export** is critical to **capture data on spend**. Setup is simple and the procedure to do so is well-documented.
- **Export to BigQuery** is recommended. Can **leverage analytics and visualization.**
  - Defined on a billing account, exported into a selected project
- ***Set this up as soon as possible****, cost data cannot be retroactively retrieved.*

Example BigQuery query to extract billing data

- BigQuery uses a SQL-like structure
- Multitude of information to view data such as project, label, dates, product, and more

```
SELECT project.labels.key,
project.labels.value,
sum(cost) as cost_total,
sum(usage.amount) as usage_total
FROM BILLINGTABLE
WHERE Start_Time >= '2018-06-01
00:00:00' AND Start_Time <
'2018-07-01 00:00:00'
GROUP BY project.labels.key,
project.labels.value
```

Results | Explanation | Job Information

Row	project_labels_key	project_labels_value	cost_total	usage_total
1	pubsub	metric	40.88075199999999	3.2135524418585517E17
2	team	sales	2.905382	1.1376485747702972E16
3	gce-enforcer-fw-opt-out	testing-customer-use-cases	187.70291399999994	2.07529806639022797E18
4	testprojectlabel		44.522840999999985	1.0482929879316314E16
5	null	null	5272.94248	2.7390326190994706E19
6	team	marketing	44.522840999999985	1.0482929879316314E16
7	cost_center	34910481	2.905382	1.1376485747702972E16

Tutorial on using labels: Label resources automatically based on Cloud Asset Inventory real-time notifications
https://cloud.google.com/community/tutorials/cloud-asset-inventory-auto-label-resources

Exam Guide - Resource Hierarchy, Billing and IAM

1.1 Setting up cloud projects and accounts. Activities include:

- 1.1.1 Creating a resource hierarchy
- 1.1.2 Applying organizational policies to the resource hierarchy
- 1.1.3 Granting members IAM roles within a project
- 1.1.4 Managing users and groups in Cloud Identity (manually and automated)
- 1.1.5 Enabling APIs within projects
- 1.1.6 Provisioning and setting up products in Google Cloud's operations suite

5.1 Managing Identity and Access Management (IAM). Tasks include:

- 5.1.1 Viewing IAM policies
- 5.1.2 Creating IAM policies
- 5.1.3 Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

Exam Guide - Resource Hierarchy, Billing and IAM

5.2 Managing service accounts. Tasks include:

- 5.2.1 Creating service accounts
- 5.2.2 Using service accounts in IAM policies with minimum permissions
- 5.2.3 Assigning service accounts to resources
- 5.2.4 Managing IAM of a service account
- 5.2.5 Managing service account impersonation
- 5.2.6 Creating and managing short-lived service account credentials

1.2 Managing billing configuration. Activities include:

- 1.2.1 Creating one or more billing accounts
- 1.2.2 Linking projects to a billing account
- 1.2.3 Establishing billing budgets and alerts
- 1.2.4 Setting up billing exports

Exam Guide - Compute Engine

Slide from Module 1

4.1 Managing Compute Engine resources. Tasks include:

- 4.1.1 Managing a single VM instance (e.g., start, stop, edit configuration, or delete an instance)
- 4.1.2 Remotely connecting to the instance
- 4.1.3 Attaching a GPU to a new instance and installing necessary dependencies
- 4.1.4 Viewing current running VM inventory (instance IDs, details)
- 4.1.5 Working with snapshots (e.g., create a snapshot from a VM, view snapshots, delete a snapshot)
- 4.1.6 Working with images (e.g., create an image from a VM or a snapshot, view images, delete an image)
- 4.1.7 Working with instance groups (e.g., set autoscaling parameters, assign instance template, create an instance template, remove instance group)
- 4.1.8 Working with management interfaces (e.g., Google Cloud console, Cloud Shell, Cloud SDK)

One of the options mentioned uses IAM

Connecting to VMs without External IPs

- Use a bastion host
  - Create a machine in the same network that has a public IP
  - Connect to it, then connect to the private machine from there
  - Turn the machine off when you don't need it
- Use Identity Aware Proxy (IAP) for TCP Forwarding
  - Uses IAM to control access to services like SSH and RDP on VM instances
    - Avoids openly exposing these services to the internet
  - Requests must pass authentication and authorization checks
  - Can be used as an alternative to a Bastion host
- Site to site: Use Cloud VPN/Interconnect
  - Connect from your network to the Google Cloud network via a private IP address

Covered in another module

In most cases, you may not want to give an instance an external IP address for security purposes, yet you will still need to connect to them. What are your options?

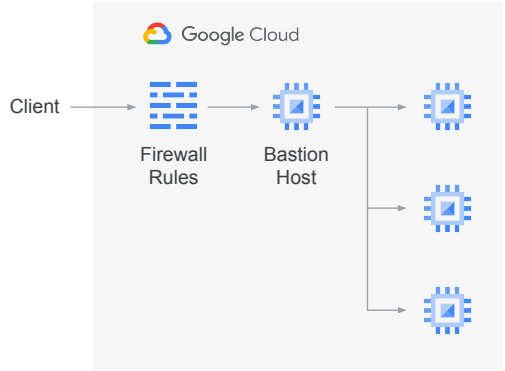
You can also leverage a bastion host, or jumpbox.
This would be one virtual machine with a public IP address that you can connect to, that can then give you access to other instances in the same network via a private IP address.
The bastion host can be turned off when not in use.

You can also leverage a VPN connection.
VPN allows you to connect to your network via an encrypted tunnel over the internet using a private IP address.

Another way is to add an “IAP-secured Tunnel User” role to enable SSH via Identity Away Proxy, or IAP.
This method will work for both the Console and gcloud.
You will have to set up a firewall rule to allow SSH traffic from 35.235.240.0/20.

Bastion Host

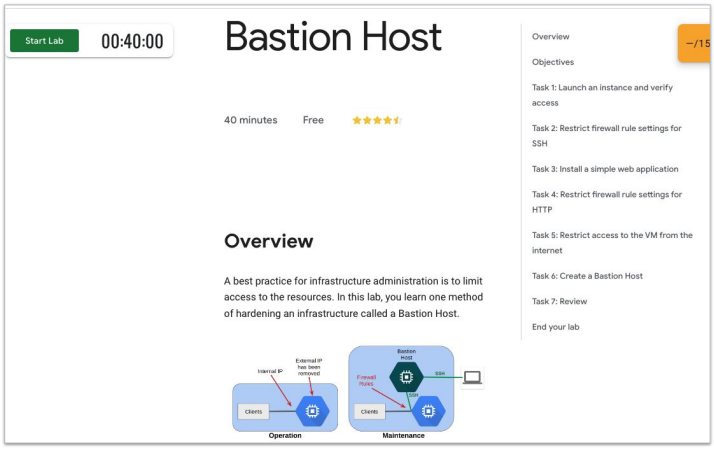
- Create a VM with a public IP
- Connect to it, then connect to the private IPs of instances from there
- Either delete or turn off the VM when done



Bastion host:
https://cloud.google.com/solutions/connecting-securely#bastion

Bastion Host: A bastion host provides an external facing point of entry into a network containing private network instances. This host provides a single point of secure access and can be stopped to disable inbound SSH. This allows the connection to VMs without having to configure firewall rules. Typical hardening initial steps for a bastion host include limiting the CIDR range of source IP addresses that can communicate with the host and configuring firewall rules to only allow SSH to private VM addresses from the bastion host.

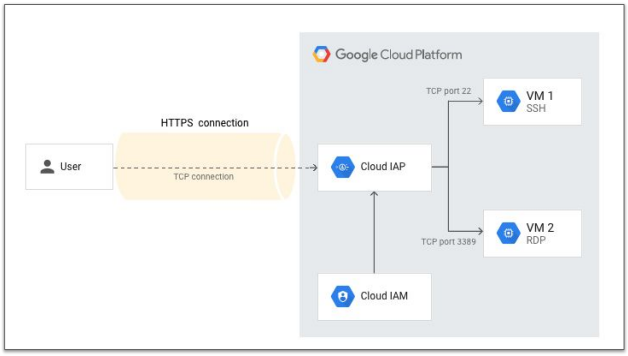
Suggested Lab (if time allows)



https://partner.cloudskillsboost.google/catalog_lab/1389

Identity Aware Proxy (IAP) for TCP Forwarding

- Uses IAM to control access to VMs via SSH and RDP from the public internet
  - Requests must pass authentication and authorization checks
- No public IPs needed on VMs in Google Cloud
- Works for both the Console SSH/RDP and the `gcloud ssh` command



IAP - Overview of TCP forwarding:
https://cloud.google.com/iap/docs/tcp-forwarding-overview

IAP Enabling external identities:
https://cloud.google.com/iap/docs/enable-external-identities

Cloud IAP enables context-aware access to VMs via SSH and RDP without bastion hosts
https://cloud.google.com/blog/products/identity-security/cloud-iap-enables-context-aware-access-to-vms-via-ssh-and-rdp-without-bastion-hosts/

Implementing Identity Aware Proxy (IAP) for TCP Forwarding

- Enable the IAP API
- Add "IAP-secured Tunnel User" role to a user to allow SSH/RDP via IAP
  - Users without the role cannot access instances
- Must allow sources from 35.235.240.0/20 in the SSH/RDP firewall rules


```
gcloud compute firewall-rules create allow-rdp-ingress-from-iap \
  --direction=INGRESS \
  --action=allow \
  --rules=tcp:3389 \
  --source-ranges=35.235.240.0/20
```

Create similar for SSH, port 22

Exam Guide - VPC Network

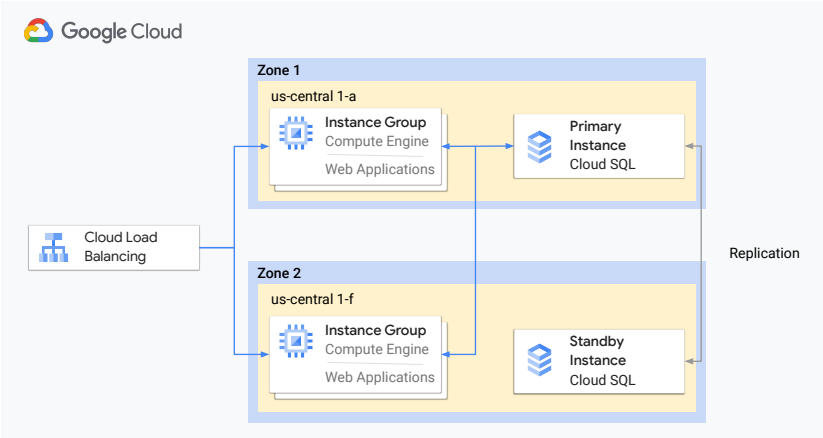
2.4 Planning and configuring network resources. Tasks include:

- 2.4.1 Differentiating load balancing options
- 2.4.2 Identifying resource locations in a network for availability
- 2.4.3 Configuring Cloud DNS

Saved this discussion until the end

When designing for high availability, deploy to multiple zones in a region

- Deploy multiple servers.
- Orchestrate servers with a regional managed instance group.
- Create a failover database in another zone or use a distributed database like Firestore or Spanner.



Google Cloud infrastructure reliability guide
https://cloud.google.com/architecture/infra-reliability-guide

Everything you need to know about architecting reliable infrastructure for Google Cloud workloads
https://cloud.google.com/blog/products/infrastructure-modernization/design-reliable-infrastructure-for-workloads-in-google-cloud

Google Cloud Architecture Framework: Reliability
https://cloud.google.com/architecture/framework/reliability

Design for scale and high availability
https://cloud.google.com/architecture/framework/reliability/design-scale-high-availability

Product reliability guides
https://cloud.google.com/architecture/framework/reliability/product-guides

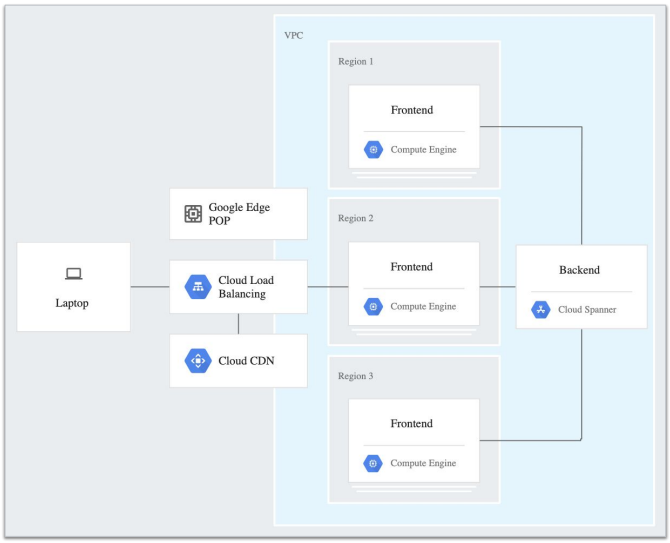
Design for high availability
https://cloud.google.com/architecture/scalable-and-resilient-apps#physically_distribute_resources

When you're using Compute Engine, for higher availability you can use a regional

instance group, which provides built-in functionality to keep instances running. Use auto-healing with an application health check and load balancing to distribute load. For data, the storage solution selected will affect what is needed to achieve high availability. For Cloud SQL, the database can be configured for HA, which provides data redundancy and a standby instance of the database server in another zone.

When designing for globalization, deploy to multiple regions

- Deploy services close to end-users for lowest latency
- Also provides redundancy
  - If one region become unavailable, traffic is automatically routed to the next closest with capacity



Best practices for Compute Engine regions selection
https://cloud.google.com/solutions/best-practices-compute-engine-region-selection

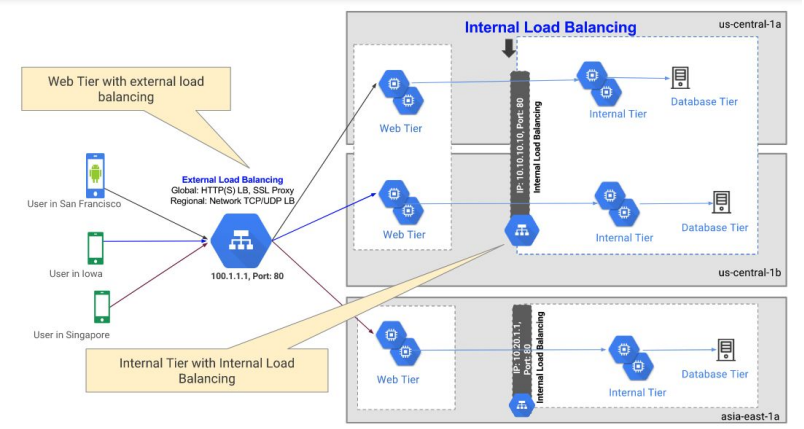
Next, let's look at globalization.

In the previous design we placed resources in different zones in a single region, which provides isolation from many types of infrastructure, hardware, and software failures. Putting resources in different regions as shown on this slide provides an even higher degree of failure independence. This allows you to design robust systems with resources spread across different failure domains.

When using a global load balancer, like the HTTP load balancer, you can route traffic to the region that is closest to the user. This can result in better latency for users and lower network traffic costs for your project.

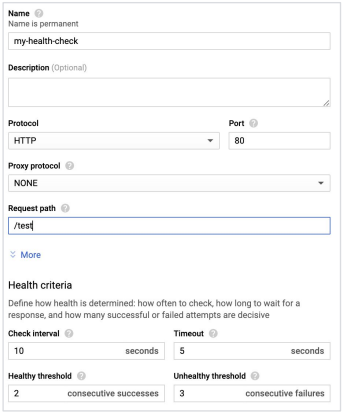
Load balance at each tier

- Load balancing distributes traffic among groups of resources
- This help ensure that individual resources don't become overloaded while others sit idle
- Most load balancers also provide health-checking features to help ensure that traffic isn't routed to unhealthy or unavailable resources



Create a health check when creating instance groups to enable auto healing

- Create a test endpoint in your service.
  - Used to verify that the service is up, and also that it can communicate with dependent backend database and services.
- If health check fails, the instance group will create a new server and delete the broken one.
- Load balancers also use health checks to ensure that they send requests only to healthy instances.



Configure autoscaling

- Configure autoscaling behavior based on key app metrics, on cost profile, and on defined minimum required level of resources
- Many Google Cloud compute products offer autoscaling
  - Serverless managed services like Cloud Run, Cloud Functions, and App Engine are designed to scale quickly and automatically
  - MIGs can scale based on various inputs, including Cloud Monitoring custom metrics and load-balancer serving capacity
    - Can set minimum and maximum limits on the scaling behavior
  - GKE can autoscale to add or remove nodes based on workload compute requirements
  - Spanner/Bigtable can increase/decrease compute capacity w/autoscaling

Patterns for scalable and resilient apps
https://cloud.google.com/architecture/scalable-and-resilient-apps

Autoscaling Cloud Spanner:
https://cloud.google.com/architecture/autoscaling-cloud-spanner

Bigtable scaling:
https://cloud.google.com/bigtable/docs/scaling

Memorystore scaling:
https://cloud.google.com/memorystore/docs/redis/scaling-instances#

Storage buckets offer high availability

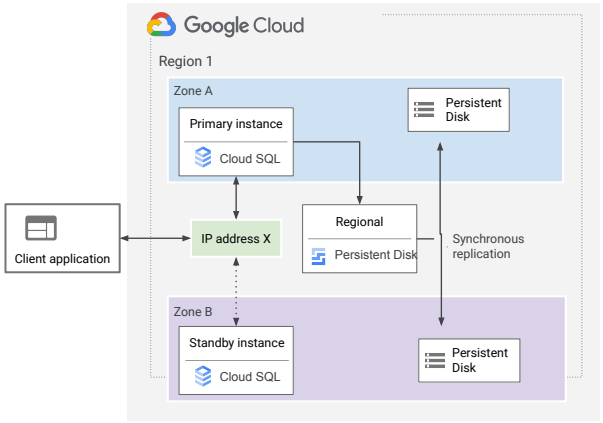
- Regional buckets are redundant across zones
- Multi/dual regions buckets are redundant across regions/zones

Bucket type	Availability
Multi-region	99.95%
Single region	99.90%
Dual-region	99.95%

The multi-region availability benefit is a factor of 2—0.1% to 0.05% unavailability. But in terms of time it is not so much. 99.9% availability means 43 minutes 12 seconds per month unavailability, compared to 99.95%, which means 21 minutes 36 seconds unavailability per month. Multi-region also needs to be carefully considered if data governance is a concern.

Cloud SQL offers failover replica for high availability

- Replica will be created in another zone in the same region as the database
  - Uses Regional persistent disks
- Will automatically switch to the failover if the primary instance is unavailable.
- Doubles the cost of the database.



The HA configuration is available for all three Cloud SQL offerings: MySQL, PostgreSQL, and SQL Server. Full details for the configuration are here: https://cloud.google.com/sql/docs/mysql/high-availability

The primary instance and standby share the same IP address so that in the event of failure, clients will seamlessly be redirected to the standby.

Spanner and Firestore can be deployed in one/multiple regions

Database	Availability SLA
Firestore single region	99.99%
Firestore multi-region	99.999%
Spanner single region	99.99%
Spanner multi-region	99.999%

Both Firestore and Spanner offer single and multi-region deployment. Multi-region locations can withstand the loss of entire regions and maintain availability without losing data. However, multi-region deployments also cost more money. Which you choose ties back to your application requirements and service-level objectives.

Exam Guide - VPC Network

2.4 Planning and configuring network resources. Tasks include:

- 2.4.1 Differentiating load balancing options
- 2.4.2 Identifying resource locations in a network for availability
- 2.4.3 Configuring Cloud DNS


Google Cloud