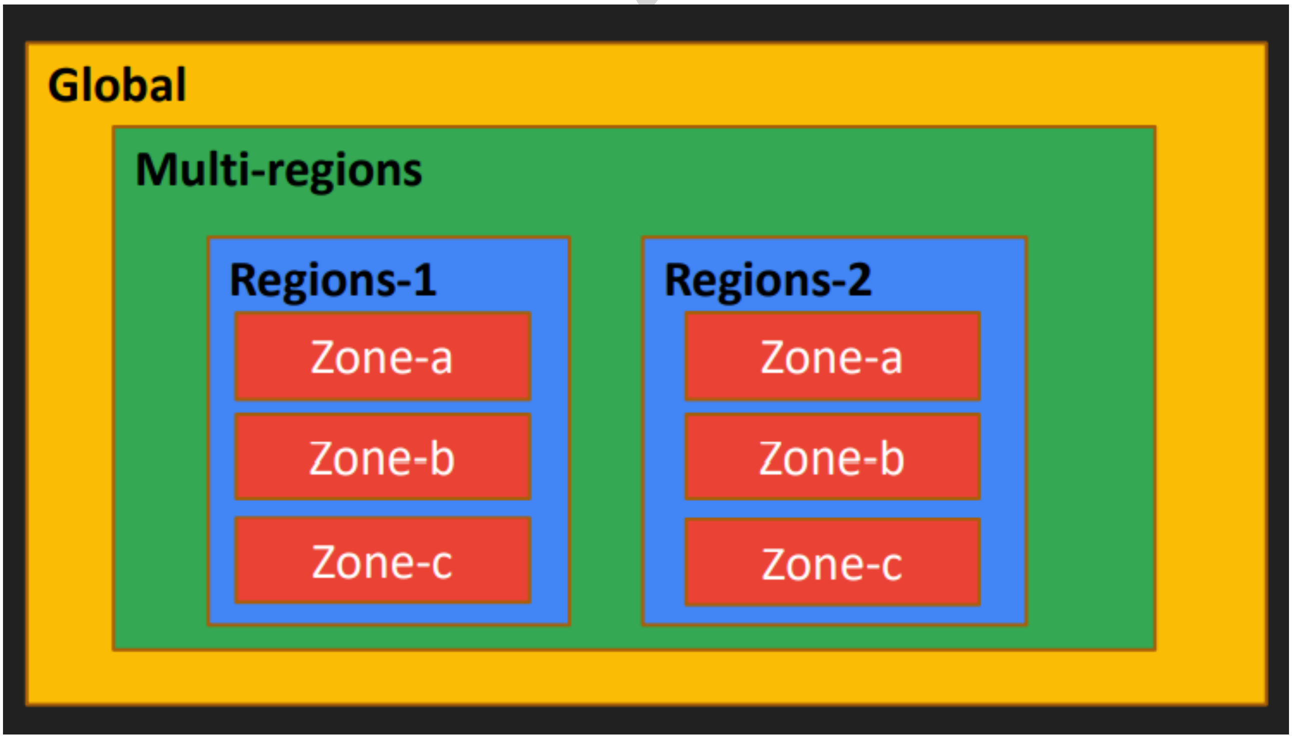
GCP Fundamentals - Getting Started With Google Cloud Platform

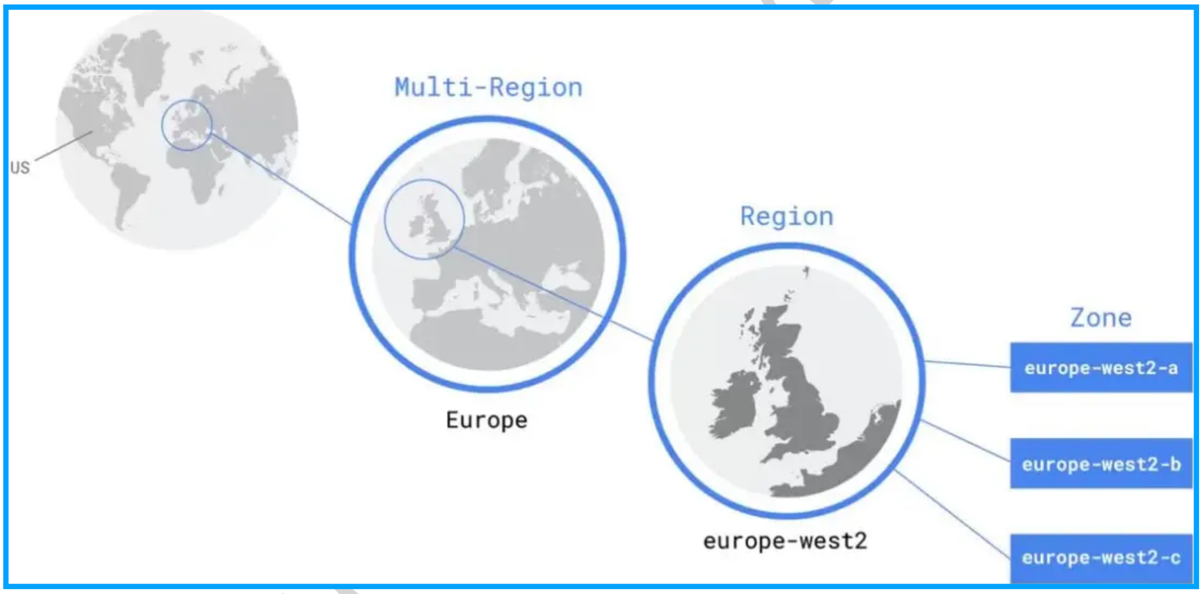
Google Cloud - Regions and Zones

- In Google Cloud **Region** is a geographical region made up of zones where you can host your GCP resources.
- Any Resource in Google Cloud either be **Regional, Zonal or Global.**
- Google Cloud Zone is a GCP resource deployment area (virtual machine, database).

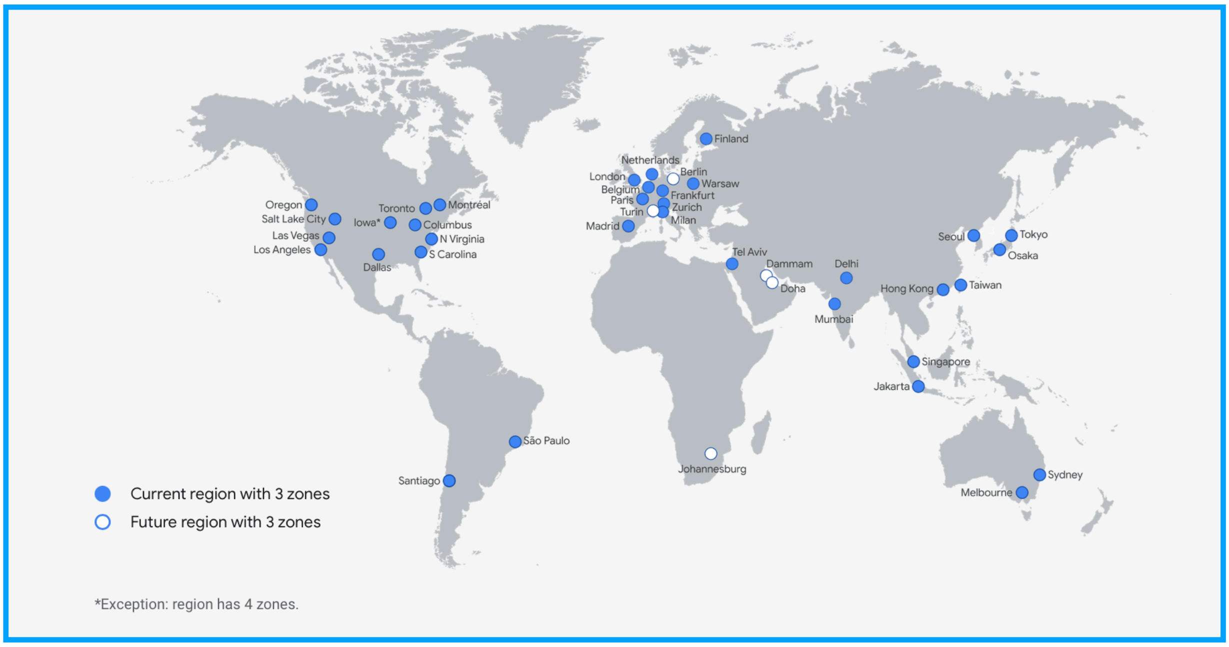


- There are at least three zones in each region. For example, the fully-qualified name for zone `a` in region `us-central1` is `us- central1-a`.
- The main benefit of zones is that they increase availability and fault tolerance within a single region.
- Zones have high-bandwidth, low-latency network connections to other zones in the same region. In order to deploy fault-tolerant applications that have high availability, Google recommends deploying applications across multiple zones in a region.

- Zones have high-bandwidth, low-latency network connections to other zones in the same region.
- There are one or more distinctive clusters in each zone (distinct physical infrastructure that is housed in a data centre).
  - Cluster : distinct physical infrastructure that is housed in a data center
- One or more data centres can be found in a single zone. Low latency links connect each of the zones in the same region.

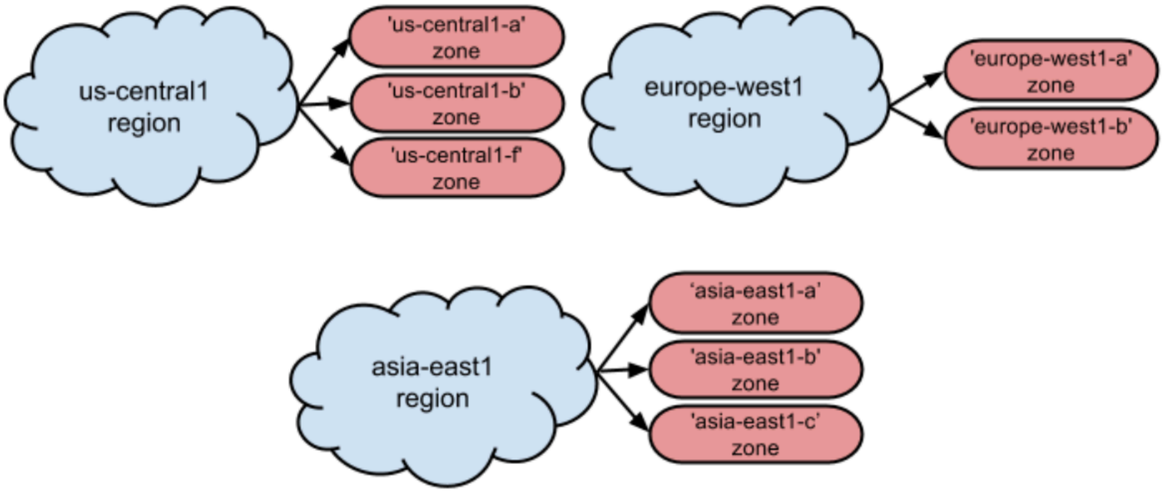


- GCP have 36 Regions and 109 Zones across the Globe.176 network edge locations and present in 200+ countries.



There are three main advantages of GCP regions and zones.

- GCP has a number of locations around the world. It also contributes to the application's excellent availability.
- Assists in achieving low latency and serving users from the closest available site.
- Follow the rules set forth by the government. Data policies varies from country to country.



Google Cloud - Resource Hierarchy

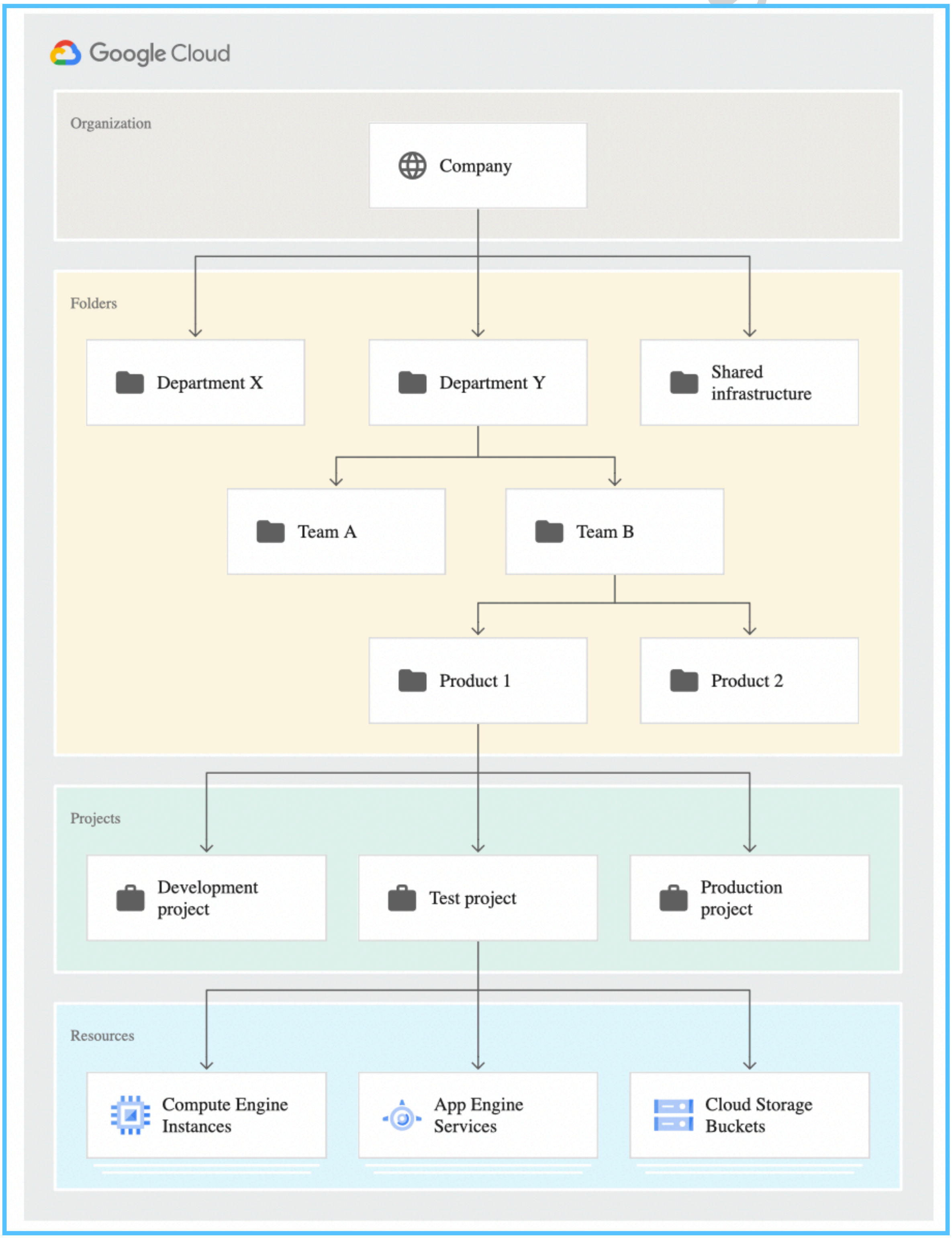
Google Cloud resources are organized hierarchically:

- The ***organization*** is the root node in the hierarchy. **Organization** can contain multiple Folders
- ***Folders*** are children of the organization.A **Folder** can contain multiple projects
- ***Projects*** are children of the organization, or of a folder.
- ***Resources*** for each service are descendants of projects.**Resources** are created in projects

The purpose of the Google Cloud resource hierarchy is two-fold:

- **Provide a hierarchy of ownership**, which binds the lifecycle of a resource to its immediate parent in the hierarchy.
- Provide **attach points and inheritance** for access control and organization policies.

- Google Cloud resources are organized hierarchically.
  - **Well defined hierarchy**:
    - Organization > Folder > Project > Resources



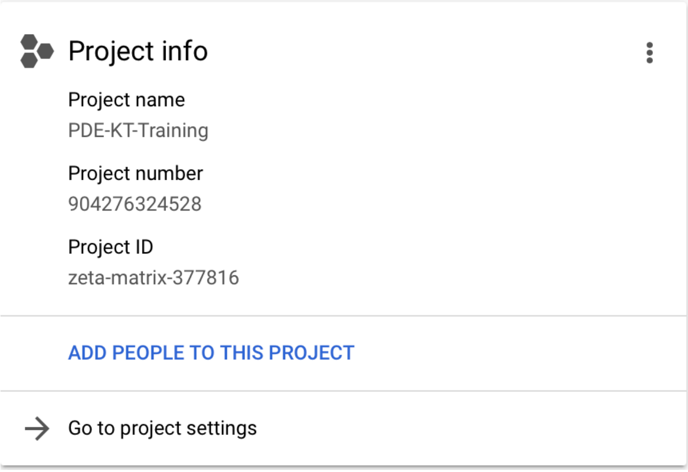
- All the resources except for the highest resource in a hierarchy have exactly one parent.
- At the lowest level, service resources are the fundamental components that make up all Google Cloud services.
- Examples of service resources include Compute Engine Virtual Machines (VMs), Pub/Sub topics, Cloud Storage buckets, App Engine instances.
- All these lower level resources have project resources as their parents, which represent the first grouping mechanism of the Google Cloud resource hierarchy.
- The Google Cloud resource hierarchy, especially in its most complete form which includes an organization resource and folder resources, allows companies to map their organization onto Google Cloud and provides logical attachment points for access management policies (IAM) and Organization policies.

Google Cloud Project ID & Project Name

- A Google Cloud project is an organizing entity for your Google Cloud resources.
- It often contains resources and services; for example, it may hold a pool of virtual machines, a set of databases, and a network that connects them together.
- Projects also contain settings and permissions, which specify security rules and who has access to what resources.

The following are used to identify your project:

- **Project name**: A human-readable name for your project.
  The project name isn't used by any Google APIs. You can edit the project name at any time during or after project creation. Project names do not need to be unique.
- **Project ID**: A globally unique identifier for your project.
  A project ID is a unique string used to differentiate your project from all others in Google Cloud. You can use the Google Cloud console to generate a project ID, or you can choose your own. You can only modify the project ID when you're creating the project.
  Project ID requirements:
  - Must be 6 to 30 characters in length.
  - Can only contain lowercase letters, numbers, and hyphens.
  - Must start with a letter.



Project info

Project name
PDE-KT-Training

Project number
904276324528

Project ID
zeta-matrix-377816

•Cannot end with a hyphen.

•Cannot be in use or previously used; this includes deleted projects.

•Cannot contain restricted strings, such as google and ssl.

•**Project number**: An automatically generated unique identifier for your project.

The project ID is used in the name of many other Google Cloud resources, and any reference to the project or related resources exposes the project ID and resource name.

Google Cloud - Cloud Shell & Cloud Editor

- Cloud Shell is an interactive shell environment for Google Cloud that lets you learn and experiment with Google Cloud and manage your projects and resources from your web browser.

- With Cloud Shell, the Google Cloud CLI and other utilities you need are pre-installed, fully authenticated, up-to-date, and always available when you need them.

- Cloud Shell comes with a built-in code editor with an integrated Cloud Code experience, allowing you to develop, build, debug, and deploy your cloud-based apps entirely in the cloud.

Features of Google Cloud - Cloud Shell & Cloud Editor

1. **Full power access from anywhere** - Manage your Google Cloud resources with the flexibility of a Linux shell.
2. **Persistent storage** - Get 5 GB of persistent storage.
3. **Secure admin** - Up-to-date, pre-authorized admin tools ready to use.
4. **Development tools and Online code editor** - Development and deployment tools for all the popular programming languages. Easy web preview as well.
5. **Source control via Git** - Clone or pull remote repositories or commit your code changes back to your repo

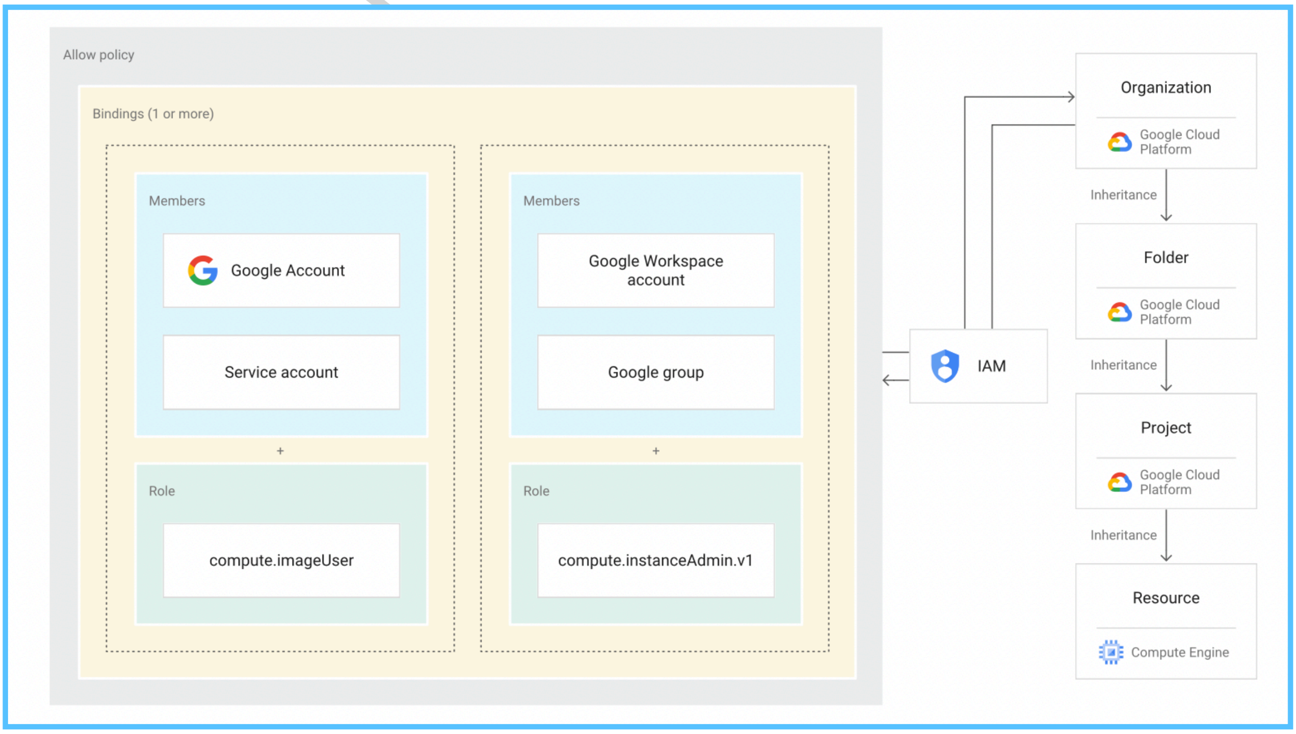
6. **Cloud Code extensions** - Simplified Kubernetes and Cloud Run development with the integrated tools provided by Cloud Code (minikube, Skaffold).

Pricing

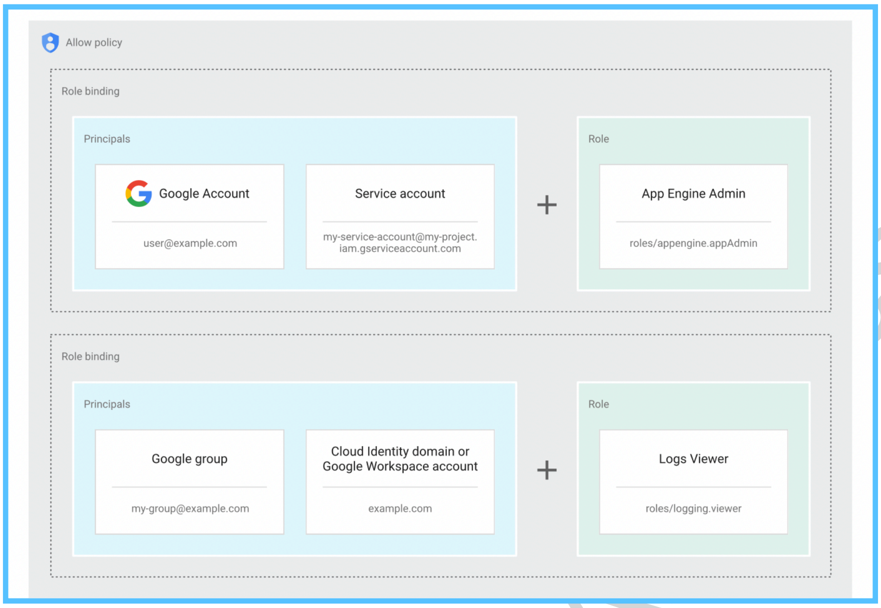
- Cloud Shell is available at no additional cost for Google Cloud customers.

Google Cloud - Identity & Access Management (IAM)

- Identity and Access Management (IAM) lets administrators authorize who can take action on specific resources, giving you full control and visibility to manage Google Cloud resources centrally. In other words, Who can do What on Which resources.
  - Who - Identity
  - What - Action : Create, Update, Delete



  - Which – Resources, Compute Engine, App Engine, Cloud Storage
- IAM lets you adopt the security principle of least privilege, which states that nobody should have more permissions than they actually need.



•In IAM, permission to access a resource isn't granted directly to the end user. Instead, permissions are grouped into roles, and roles are granted to authenticated principals.

•IAM lets you grant granular access to specific Google Cloud resources and helps prevent access to other resources.

1. **Principal -** A principal can be a Google Account (for end users), a service account (for applications and compute workloads), a Google group, or a Google Workspace account or Cloud Identity domain that can access a resource. Each principal has its own identifier, which is typically an email address.

2. **Role -** A role is a collection of permissions. Permissions determine what operations are allowed on a resource. When you grant a role to a principal, you grant all the permissions that the role contains.

3. **Policy -** The allow policy is a collection of role bindings that bind one or more principals to individual roles. When you want to define who (principal) has what type of access (role) on a resource, you create an allow policy and attach it to the resource.

Features of Google Cloud - Identity and Access Management (IAM)

1. Single access control interface
2. Fine-grained control
3. Automated access control recommendations
4. Context-aware access
5. Flexible roles
6. Web, programmatic, and command-line access

7. Built-in audit trail
8. Support for Cloud Identity
9. Free of charge

In IAM, you grant access to principals. Principals can be of the following types:

- Google Account
- Service account
- Google group
- Google Workspace account
- Cloud Identity domain
- All authenticated users
- All users

- One can assign Role to identity but Cannot assign permission directly. To search with primitive use, roles/owner that is not recommended by google. Legacy rules.

Google Account - A Google Account represents a developer, an administrator, or any other person who interacts with Google Cloud.

Service account - A service account is an account for an application or compute workload instead of an individual end user.

Google group - A Google group is a named collection of Google Accounts and service accounts. Every Google group has a unique email address that's associated with the group. Google Groups don't have login credentials.

Google Workspace account - A Google Workspace account represents a virtual group of all of the Google Accounts that it contains. Google Workspace accounts are associated with your organization's internet domain name, such as example.com.

Cloud Identity domain - A Cloud Identity domain is like a Google Workspace account, because it represents a virtual group of all Google Accounts in an organization. However, Cloud Identity domain users don't have access to Google Workspace applications and features.

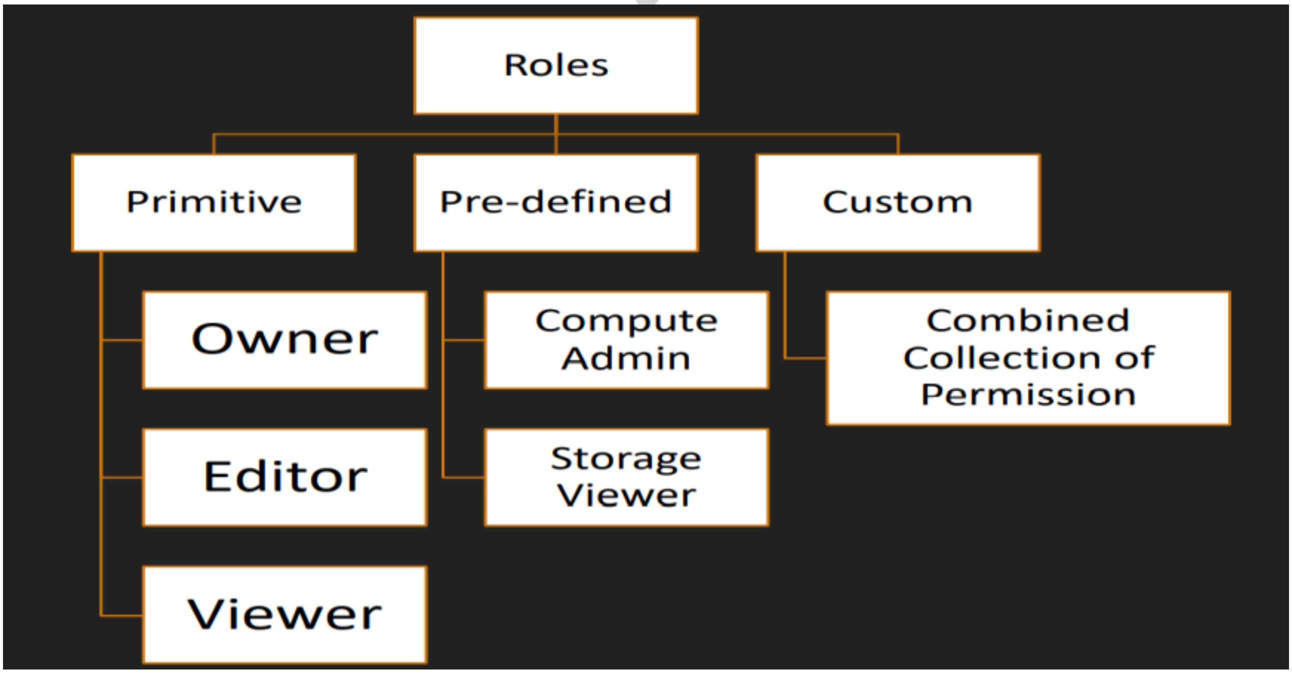
All authenticated users - The value allAuthenticatedUsers is a special identifier that represents all service accounts and all users on the internet who have authenticated with a Google Account.

All users - The value allUsers is a special identifier that represents anyone who is on the internet, including authenticated and unauthenticated users.

Google Cloud - Types Of Roles

Roles

A role is a collection of permissions. You cannot grant a permission to the user directly. Instead, you grant them a role. When you grant a role to a user, you grant them all the permissions that the role contains.



There are several kinds of roles in IAM:

- **Basic roles**: Roles historically available in the Google Cloud console. These roles are Owner, Editor, and Viewer.
- **Predefined roles**: Roles that give finer-grained access control than the basic roles. For example, the predefined role Pub/Sub Publisher (roles/ pubsub.publisher) provides access to *only* publish messages to a Pub/ Sub topic.
- **Custom roles**: Roles that you create to tailor permissions to the needs of your organization when predefined roles don't meet your needs.

Permissions

Permissions determine what operations are allowed on a resource. In the IAM world, permissions are represented in the form of **service.resource.verb**, for example, pubsub.subscriptions.consume.

For example, if you use Pub/Sub, and you need to call the topics.publish() method, you must have the pubsub.topics.publish permission for that topic.

You don't grant permissions to users directly. Instead, you identify *roles* that contain the appropriate permissions, and then grant those roles to the user.

Allow policy

You can grant roles to users by creating an *allow policy*, which is a collection of statements that define who has what type of access. An allow policy is attached to a resource and is used to enforce access control whenever that resource is accessed.

An allow policy consists of a list of role bindings. A role binding binds a list of principals to a role.

- **role:** The role you want to grant to the principal. role is specified in the form of roles/service.roleName. For example, Cloud Storage provides the roles roles/storage.objectAdmin, roles/storage.objectCreator, and roles/storage.objectViewer, among others.
- **members:** A list of one or more principals as described in the Concepts related to identity section in this document. Each principal type is identified with a prefix, such as a Google Account (user:), service account (serviceAccount:), Google group (group:), or a Google Workspace account or Cloud Identity domain (domain:).
  - In the following example code snippet, the storage.objectAdmin role is granted to the following principals by using the appropriate prefix: user:ali@example.com, serviceAccount:my-other-app@appspot.gserviceaccount.com, group:admins@example.com, and domain:google.com. The objectViewer role is granted to user:maria@example.com.

The following code snippet shows the structure of an allow policy.

```
{
  "bindings": [
    {
      "role": "roles/storage.objectAdmin",
       "members": [
         "user:cloudaianalytics@example.com",
         "serviceAccount:cloud-ai-analytics@appspot.gserviceaccount.com",
         "group:cloudaianalytics@example.com",
         "domain:google.com"
       ]
    },
    {
      "role": "roles/storage.objectViewer",
      "members": [
        "user:cloudaianalytics@example.com"
      ]
    }
  ]
}
```

Google Cloud - Service Account & Types

- A service account is a special Google account that belongs to your application or services like virtual machine (VM) instead of an individual end user. Your application uses the service account to call the Google API of a service, so that the users aren't directly involved.
- A service account is identified by its email address, which is unique to the account.
- Inside a Cloud project, Google Cloud automatically creates one Compute Engine service account and one App Engine service account under that project.
- You can create up to 98 additional service accounts to your project to control access to your resources.
- Applications use service accounts to make authorized API calls by authenticating as either the service account itself, or as Google Workspace or Cloud Identity users through domain-wide delegation.
- When an application authenticates as a service account, it has access to all resources that the service account has permission to access.
- Service account keys can be used for authentication
- Max 10 keys per Service Account
- Max 100 Service Account per project

Types of service accounts

1. User-managed service accounts

a. When you create a new Cloud project using Cloud Console and if Compute Engine API is enabled for your project, a Compute Engine Service account is created for you by default. It is identifiable using the email:

1. PROJECT_NUMBER-compute@developer.gserviceaccount.com
2. PROJECT_ID@appspot.gserviceaccount.com

2. Google-managed service accounts

a. In addition to the user-managed service accounts, you might see some additional service accounts in your project's IAM policy or in the Cloud Console. These service accounts are created and owned by Google. These accounts represent different Google services and each account is automatically granted IAM roles to access your Google Cloud project.

3. Google APIs service account

a. An example of a Google-managed service account is a Google API service account identifiable using the email:

1. PROJECT_NUMBER@cloudservices.gserviceaccount.com

b. This service account is designed specifically to run internal Google processes on your behalf and is not listed in the Service Accounts section of Cloud Console.

c. By default, the account is automatically granted the project editor role on the project and is listed in the IAM section of Cloud Console. This service account is deleted only when the project is deleted.