Google Cloud Platform

Getting Started with Google Cloud Platform

Google Cloud Platform Fundamentals

V2.1

Timing: Approximately 30 minutes

Agenda

1. Google Cloud Platform Projects
2. Identity and Access Management (IAM)
3. Interacting with Google Cloud Platform
4. Quiz & Lab

Projects (1 of 2)

- All Google Cloud Platform services are associated with a project that is used to:
  - Track resource and quota usage
  - Enable billing
  - Manage permissions and credentials
  - Enable services and APIs

Notes:
All Google Cloud Platform resources belong to a Google Cloud Platform Console project. Projects form the basis for enabling and using the Google Cloud Platform services, including managing APIs, enabling billing, adding and removing collaborators and enabling other Google services. Each project is a separate compartment and each resource belongs to exactly one such compartment. Projects can have different owners and users, are billed separately, and are managed separately.

Projects (2 of 2)

- Projects use three identifying attributes:
  - Project Name
  - Project Number
  - Project ID
    - Also known as Application ID
- Interact with projects using the Cloud Console or the Cloud Resource Manager API

Notes:
Google Cloud Platform provides container resources such as Organizations and Projects, that allow you to group and hierarchically organize other Cloud Platform resources. This hierarchical organization lets you easily manage common aspects of your resources such as access control and configuration settings. The Google Cloud Resource Manager API enables you to programmatically manage these container resources.

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

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

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

- Through the RPC API
- Through the REST API

Project Permissions - Primitive Roles

Owner

Invite members
Remove members
Can delete project
Includes Editor rights

Editor

Deploy applications
Modify code
Configure services
Includes Viewer rights

Viewer

Read-only access

Billing administrator

Manage billing
Add administrators
Remove administrators

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

Notes:
There are two kinds of roles in Cloud IAM:

- *Primitive roles*: The roles historically available in the Google Cloud Platform Console. These are the Owner, Editor, and Viewer roles.
- *Curated roles*: Curated roles are the new IAM roles that give finer-grained access control than the primitive roles (discussed in the next section).

Team members for a Project are assigned and managed in the console Permissions pane for the project, where an owner can add new team members by entering their email address, which is associated with a Google account. They will need to sign in and accept the invitation before becoming a team member. If you are using a Google Apps domain, the administrator for your domain should first create the Google account from within the Users panel of your Admin Console.

Team members may be authorized to have one of three levels of access:

- “can View” allows read-only access.
- “can Edit” allows modify and delete access. This allows a developer to deploy the application and modify or configure its resources.
- “is Owner” allows full administrative access. This includes the ability to add members and set the authorization level of team members.

You are advised to give “is Owner” permission to multiple team members for continuity in case a single Owner leaves the organization or has their account deleted for any other reason. If the sole owner’s account is deleted, then the entire project would also be deleted. The recommended practice is to use Google accounts for your project’s team members. When someone leaves the company, your Google Apps administrator should use the Google Apps Admin Console to mark the account as suspended (instead of deleting the account.) The domain administrator can then perform various tasks such as reassigning ownership of apps and docs before deleting the team member’s account.

Agenda

1. Google Cloud Platform Projects
2. Identity and Access Management (IAM)
3. Interacting with Google Cloud Platform
4. Quiz & Lab

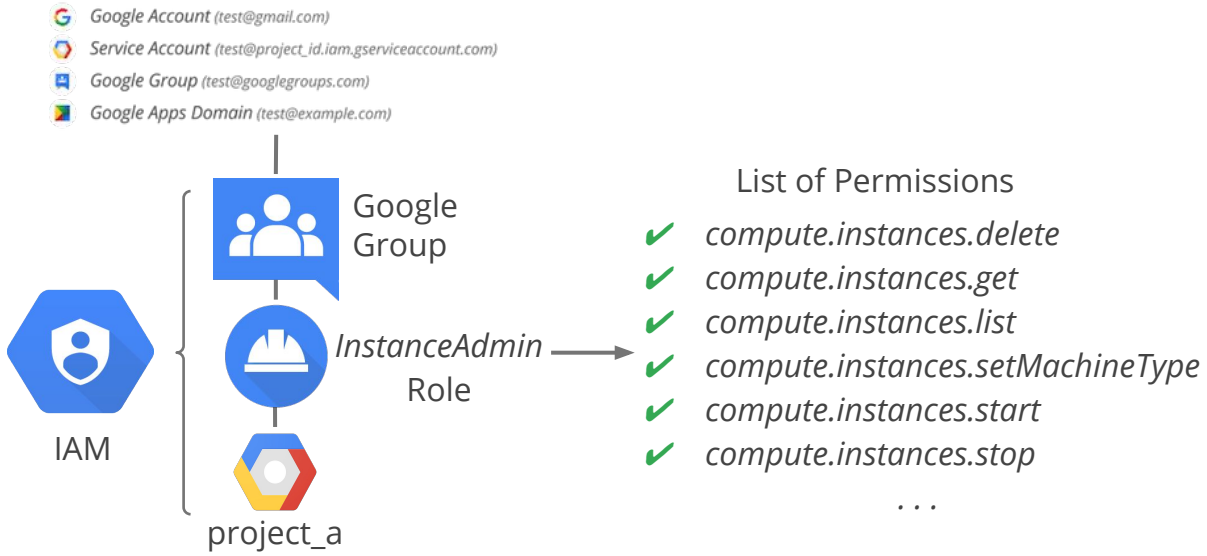
Identity and Access Management

Who

can do what

on which resource

IAM Roles - Curated Roles



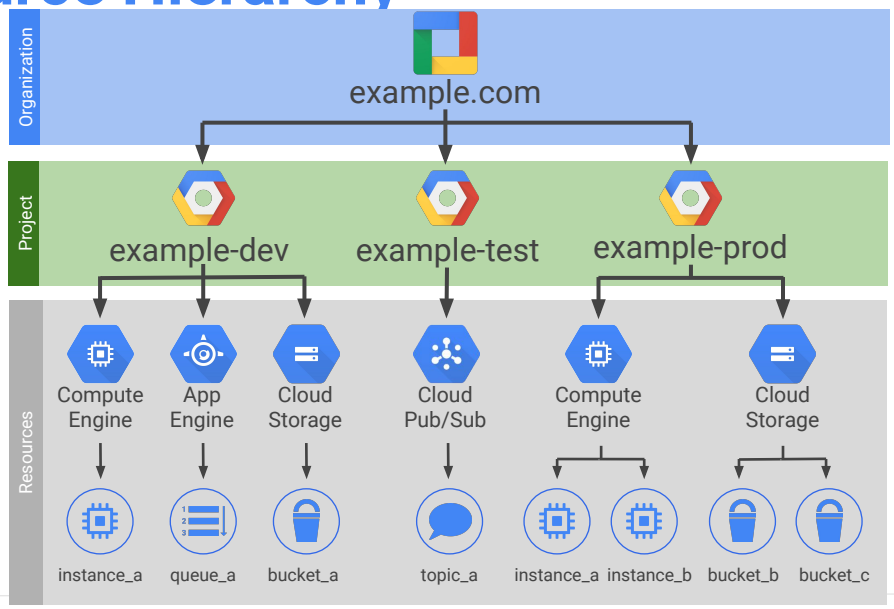
Notes:
The “can do what” part is defined by an IAM role. An IAM role is a collection of permissions. Most of the time to do any meaningful operations you need more than 1 permission. For example to manage instances in a project, you need to create, delete, start, stop and change an instance. So the permissions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

To give a user the desired permissions, you grant a role to the user on a resource. In this example we are granting a group of users the InstanceAdmin role on project a so the user can manage instances in the project. Whenever possible, it is a best practice to use groups. You should also strictly control the ability to change policies and group memberships which will allow additional users to gain access to resources.

For a complete list of roles by product, see:
https://cloud.google.com/iam/docs/#supported_cloud_platform_services

IAM Resource Hierarchy

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



Notes:
Cloud Platform resources are organized hierarchically, where the Organization node is the root node in the hierarchy, the projects are the children of the Organization, and the other resources are the children of projects. Each resource has exactly one parent.

IAM allows you to set policies at the following levels of the resource hierarchy:

- Organization level. The Organization resource represents your company. IAM roles granted at this level are inherited by all resources under the organization.
- Project level. Projects represent a trust boundary within your company. Services within the same project have a default level of trust. For example, App Engine instances can access Cloud storage buckets within the same project. IAM roles granted at the project level are inherited by resources within that project. When setting policies at the project level, be sure to use audit logs to track project level permission changes.
- Resource level. In addition to the existing Cloud Storage and BigQuery ACL systems, additional resources such as Genomics Datasets and Pub/Sub topics support resource-level roles so that you can grant certain users permission to a single resource.

Resources inherit the policies of the parent resource. If you set a policy at the organization level, it is automatically inherited by all its children projects, and if you set a policy at the project level, it is inherited by all its children resources. The effective policy for a resource is the union of the policy set at that resource and the policy inherited from its parent. This policy inheritance is transitive; in other words, resources inherit policies from the project, which inherit policies from the organization. Therefore, the organization-level policies also apply at the resource level.

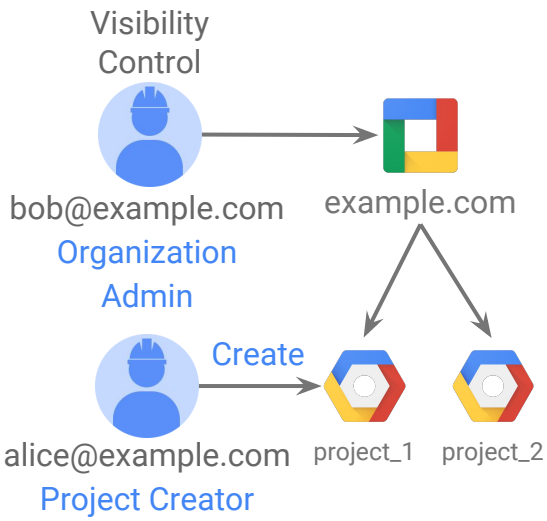
The IAM policy hierarchy follows the same path as the Cloud Platform resource hierarchy. If you change the resource hierarchy, the policy hierarchy changes as well. For example, moving a project into an organization will update the project's IAM policy to inherit from the organization's IAM policy.

Child policies cannot restrict access granted at the parent. For example, if you grant editor role to a user for a project, and grant viewer role to the same user for a child resource, then the user still has editor role for the child resource. When using IAM, a best practice is to follow the principle of least privilege. The principle applies to identities, roles, and resources. Always select the smallest scope that’s necessary to reduce your exposure to risk. You wouldn’t want to grant everybody the owner role on your entire organization - intentional hacks or accidental mistakes can bring down your apps. You want to be specific and deliberate. Assign a specific security admin group the security admin role to manage SSL and firewall rules on specific projects.

For more information, see: Identity and Access Management Overview

Organization Node Beta

- Organization node is root node for Google Cloud resources
  - Can be managed via Cloud Resource Manager API
- 2 organization roles:
  - *Organization Admin* - Control over all cloud resources
  - *Project Creator* - Controls project creation



Notes:
A large number of projects can become unwieldy to manage at scale. This is why IAM includes the concept of an Organization Node. The Organization Node sits above Projects and is your company's root node for Google Cloud resources. If you have a Google for Work account, when you enable the Organization Node, any project created by users in your domain will automatically belong to your Organization Node - no more shadow projects and no more rogue admins.

The Organization Admin role gives your admin visibility and control over all of your company's resources on Google Cloud Platform. Using the Project Creator role, you can restrict who can create projects regardless of whether they have policies on individual projects. The project roles can also be applied at the organization level and can be inherited by all the projects in your company. For example, you can assign your networking team the Network Admin role at the organization level so they have permissions to manage all the networks in all the projects in your company.

As of this writing, the Organization Node is in Alpha. If you have a Google for Work account, you can work with your account manager or support partner to request to create an Organization Node for your company.

Service Accounts

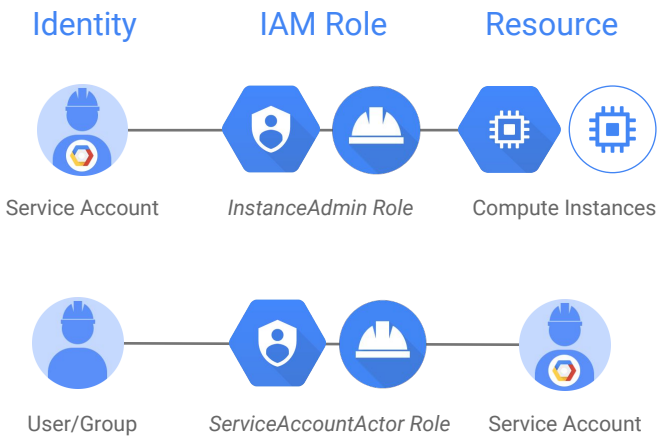
- Provide an identity for carrying out **server-to-server** interactions in a project
- Used to **authenticate** from one service to another
- Can be used with primitive and curated roles
- Identified with an **email** address:

```
<project_number>@developer.gserviceaccount.com
<project_id>@developer.gserviceaccount.com
```


Notes:
A service account is an identity for your programs to use to authenticate and gain access to Google Cloud APIs.

Service Accounts and IAM

- Service accounts authenticate with keys
  - Google manages keys, key rotation for Compute Engine and App Engine
- Can assign an IAM role to the service account
- Can also assign `ServiceAccountActor` role to users/groups

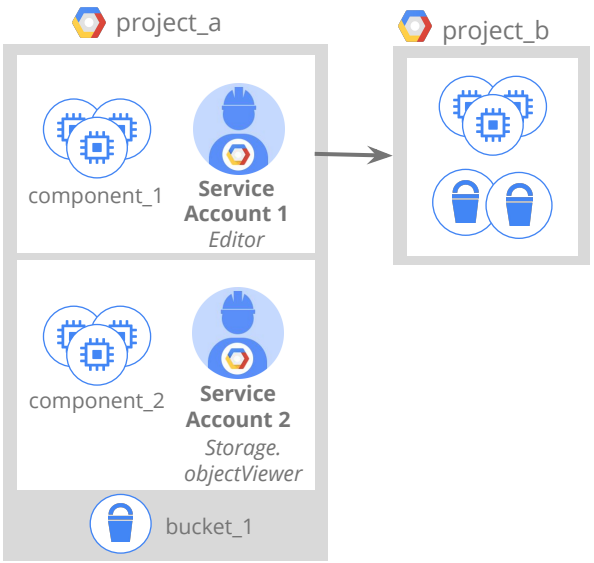


Notes:
Users require a username and password to authenticate. Apps use a key. One or more keys can be generated for each IAM service account. Keys are sensitive and need to be carefully managed because they give you access to resources. When you run applications on Compute Engine or App Engine, Google manages the keys for you and automatically rotates them. You never have the risk of losing/exposing your key. When you run apps elsewhere, you can generate and download the keys to use in your code. Keep them safe and rotate them.

A service account is both an identity and a resource. A service account is used as an identity for your application to authenticate; for example, a Compute Engine VM running as a service account. To give the VM access to the necessary resources, you need to grant the relevant IAM roles to the service account. At the same time, you need to control who can create VMs with the service account so random VMs cannot assume the identity. Here, the service account is the resource to be permissioned. You assign the ServiceAccountActor role to the users you trust to use the service account.

Example: Service Accounts and IAM

- VMs running component_1 are granted `Editor` access to project_b using *Service Account 1*
- VMs running component_2 are granted `objectViewer` access to `bucket_1` using *Service Account 2*
- Service account permissions can be changed without recreating VMs



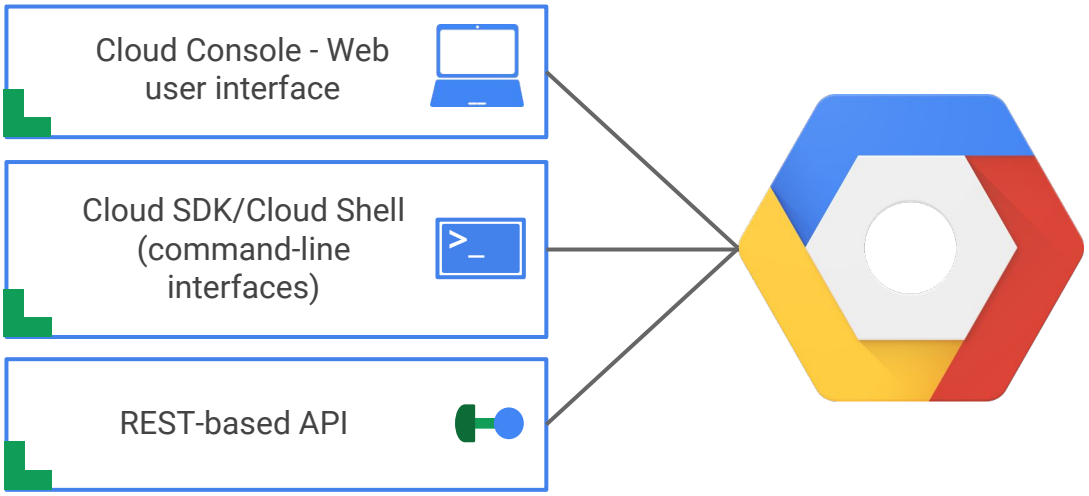
Notes:
You can grant different groups of VMs in your project different identities. This makes it easier to manage different permissions for each group. For example, if one component in your app needs to have the Editor role on another project, you can have a service account with this permission that is used only by the VMs running the component. Other VMs can be assigned the permissions required by their functionalities. This way you can scope permissions for VMs. You also can change the permissions of the service accounts without having to recreate the VMs.

IAM also lets you slice a project into different microservices, each with access to different resources, by creating service accounts to represent each one. You assign the service accounts to the VMs when they are created, and as a security engineer, you don't have to ensure that credentials are being managed correctly. Google Cloud Platform manages security for you.

Agenda

1. Google Cloud Platform Projects
2. Identity and Access Management (IAM)
3. Interacting with Google Cloud Platform
4. Quiz & Lab

Interacting with Google Cloud Platform



Google Cloud Platform Console

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

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

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

Google Cloud SDK

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

Notes:
The Google Cloud SDK is a set of tools that you can use to manage resources and applications hosted on Google Cloud Platform. These include the gcloud tool, which provides the main command-line interface for Cloud Platform products and services, as well as gsutil and bq. All of the tools are located under the bin directory.

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

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

Cloud Shell provides the following:

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

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

You can use Cloud Shell to:

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

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

RESTful APIs

- Programmatic access to products and services
  - Typically use JSON as an interchange format
  - Use OAuth 2.0 for authentication and authorization
- Enabled through the Google Cloud Platform Console
- Most APIs include daily quotas and rates (limits) that can be raised by request
  - Important to **plan ahead** to manage your required capacity
- Experiment with APIs Explorer

Notes:
A typical OAuth 2.0 flow looks like the following:

1. Get the user's credentials.
   - Run OAuth2 "flow" - Interaction between your application, end user, and Google authorization servers.
   - The "flow" may ask for the user's login and authorization.
   - Try it in the OAuth 2.0 Playground.
2. Authorize an HTTP object with credentials.
   - Internally, this sets the HTTP header.
3. Create a service API object with the authorized HTTP object from step 2.

For more information on Using OAuth 2.0 to Access Google APIs, see:
https://developers.google.com/accounts/docs/OAuth2

APIs Explorer

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

Notes:
As time permits, your instructor may conduct a demo using the APIs Explorer.

Authorized Access
Certain API requests access your private data, for example, a request to list your followers on Google Plus. These requests require that you authorize the APIs Explorer to access your data, and it must then send authentication credentials that verify that it is authorized to access data on your behalf. This is known as authenticated access. The first time you try to make an authenticated request, the Explorer presents a dialog asking you to grant it access to a limited set of your private data.

On the other hand, many API methods only access public data. These methods don't require you to send authentication credentials, and requests can be made using unauthenticated access. Some methods support both public and private access, and they may behave differently depending on whether authentication credentials are provided. The Explorer allows you to switch between authenticated and unauthenticated requests, so you can see how the API behaves under different scenarios.

Credentials to access an API (called API keys) are not needed to use the APIs Explorer. The APIs Explorer uses its own API key whenever it makes a request.

What does the "scope" mean when using authorized access?

When you make an authenticated request to an API, the APIs Explorer asks you to grant it permission to access data on your behalf. The specific data that the Explorer can access is limited by the “scope,” which is particular to an API. In other words, the scope limits the power of the access you grant. For example, the Calendar API allows you to grant access to the APIs Explorer using a read-only scope or a regular scope. Before making the first authenticated call, the Explorer asks you to choose which scope you would like to use when granting it access to your data.

Note that different methods might require different scopes. For example, one particular method in an API might require at least a read-only scope, while other methods might require a read-write scope. If you are making an authenticated call and selecting a scope, the APIs Explorer will tell you which scopes are required for calling a method in the selected API; making a request with OAuth credentials obtained using the wrong scope might result in a failed request. If this occurs, you can reset the switch to Authorize your API calls and use a different scope.

Client Libraries

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

Notes:
When you use a client library to access Google Cloud Platform services, your applications benefit from language integration, improved security, and support for easily making authenticated calls. You can use the Google Cloud Client Libraries to access Google Cloud Platform services such as Cloud Storage, Cloud Datastore, Cloud Pub/Sub, and BigQuery.

The Cloud Client libraries have the following advantages:

- Provide an optimized developer experience for the supported languages by using each language's natural conventions and styles.
- Reduce the boilerplate code you have to write because they designed to enable you to work with service metaphors in mind, rather than implementation details or service API concepts.
- Simplify authentication by providing helpers for authorizing to Google Cloud services.
- Community-owned: developers can contribute to the code and documentation.

Use Google Cloud Client Libraries for creating applications built on Google Cloud Platform services. If a Google Cloud Client Libraries for a specific language doesn't yet support a Cloud Platform service you want to work with, use the Google API Client Library for that language.

For more information on Google API Client Libraries, see:
https://developers.google.com/discovery/libraries

Agenda

1. Google Cloud Platform Projects
2. Identity and Access Management (IAM)
3. Interacting with Google Cloud Platform
4. Quiz & Lab

Quiz (1 of 2)

1. *True or False*: In Google Cloud IAM, if a policy gives you Owner permissions at the project level, your access to an individual resource in the project may be restricted to Viewer by applying a more restrictive policy to that resource.
2. *True or False*: All Google Cloud Platform resources are associated with a project.

Quiz (2 of 2)

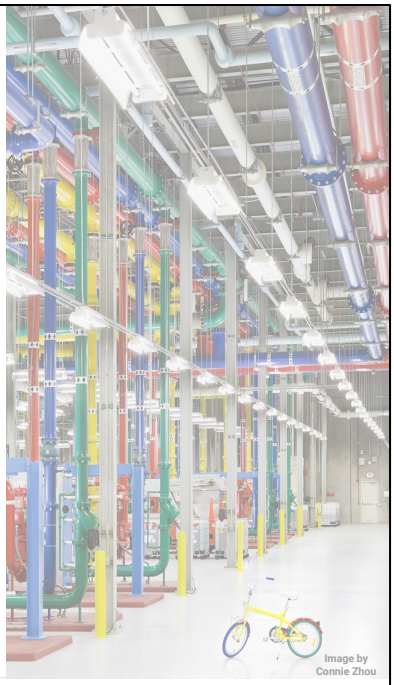
Service accounts are used to provide which of the following?

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

Lab

Deploy a virtual development environment using Google Cloud Launcher.

1. Deploy a Bitnami LAMP stack to Compute Engine using Cloud Launcher
2. Verify the deployment



Resources

- Cloud SDK installation and quick start
  https://cloud.google.com/sdk/#Quick_Start
- ‘gcloud’ tool guide
  https://cloud.google.com/sdk/gcloud/
- IAM
  https://cloud.google.com/iam/
- Configuring permissions on Google Cloud Platform
  https://cloud.google.com/docs/permissions-overview
- Google Cloud Platform security
  https://cloud.google.com/security/

Quiz Answers (1 of 2)

1. *False*: Policies are a union of the parent and the resource. If a parent policy is less restrictive, it overrides a more restrictive resource policy.
2. *True*: All Google Cloud Platform resources are associated with a project.

Quiz Answers (2 of 2)

Service accounts are used to provide which of the following?

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


cloud.google.com