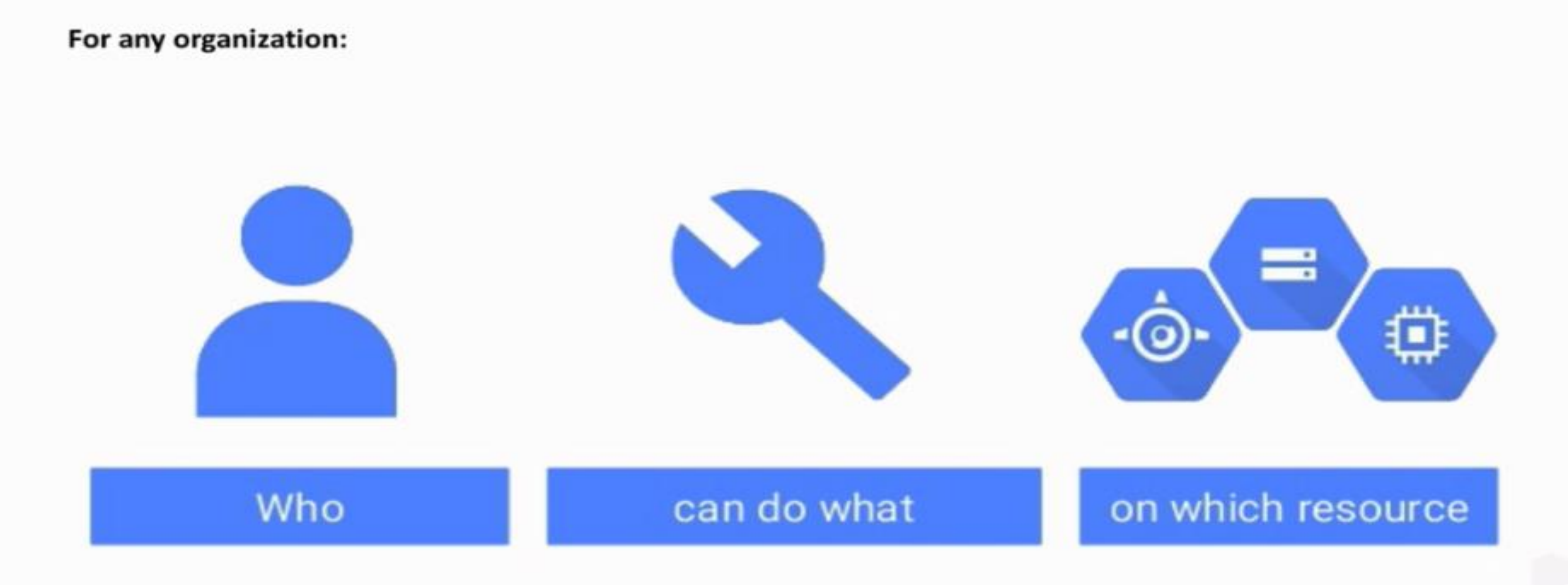
Cloud Computing

Identity and Access Management (IAM)

- Discuss the importance of resource access and management.
- Overview of GCP’s implementation of Identity and Access Management(IAM)
- Members, Roles, Resources
- What are Members?
- Primitive and Predefined Roles
- IAM Policy Hierarchy
- Hands on demo

Identity and Access Management (IAM)



Why is this important?

- Important to provide granular access to resources
- Prevent unwanted access to other resources
- Adopt the security principle of least privileges→

grant only the necessary access to your resources and prevent unwanted access to other resources

Google IAM

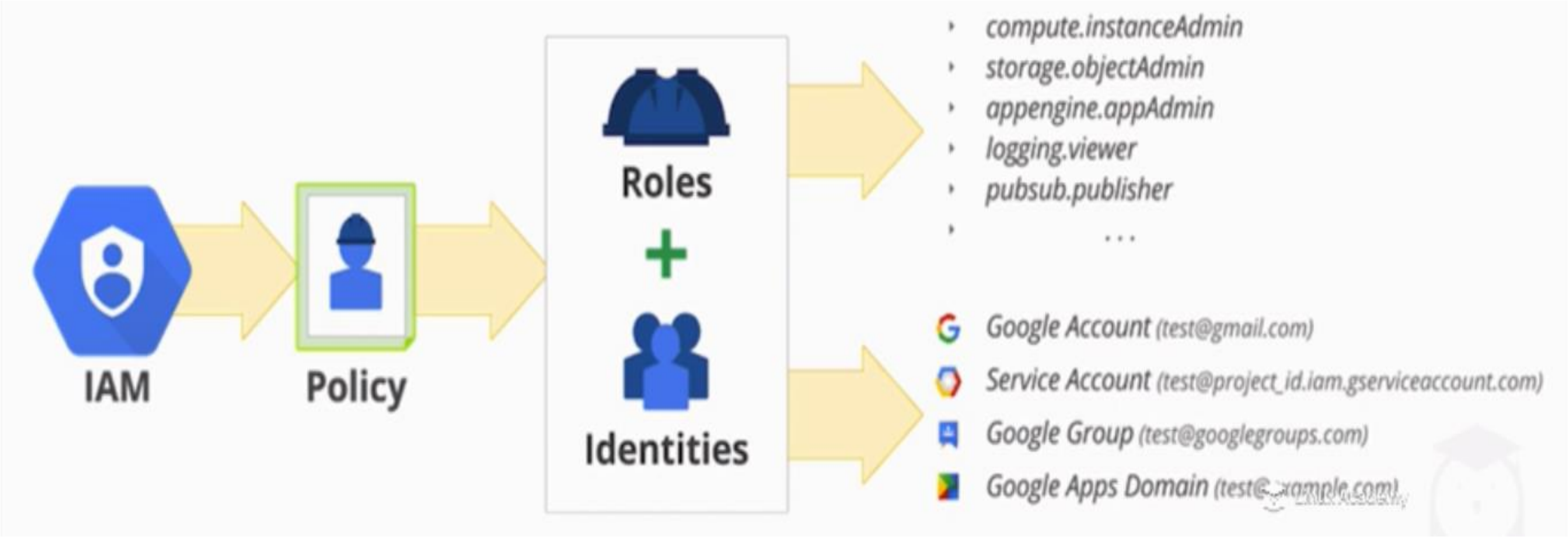
- Important to provide granular access to resources.

- Prevent unwanted access to other resources

- Adopt the security principle of least privilege
  - The concept of grant only the necessary access to your organization resources and nothing beyond that

Google IAM

- Members (who) are granted permissions and roles (what) to GCP services (resource) using the principle of least privilege.


IAM
Policy
Roles
+
Identities
compute.instanceAdmin
storage.objectAdmin
appengine.appAdmin
logging.viewer
pubsub.publisher
...
Google Account (test@gmail.com)
Service Account (test@project_id.iam.gserviceaccount.com)
Google Group (test@googlegroups.com)
Google Apps Domain (test@example.com)

Members (the who)?

Can be either a person or a service account:

- People
  - Google account
    - A Google account represents a developer, an administrator, or any other person who interacts with Google Cloud Platform.
  - Google group – e.g. DevTeam@mycompany.com
    - 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
  - G Suite Domain
    - A G Suite domain represents a virtual group of all the members in an organization. GSuite customers can associate their email accounts with an internet domain name. When you do this, each email account takes the form username@yourdomain.com.
- Service account
  - Application account

Service Account

- A service account is a special type of Google account that belongs to your application or a virtual machine (VM), instead of an individual end user. Your application calls Google APIs assuming the identity of the service account, so that the users aren’t directly involved.

- Provide an identity for carrying out server-to-server interactions in a project

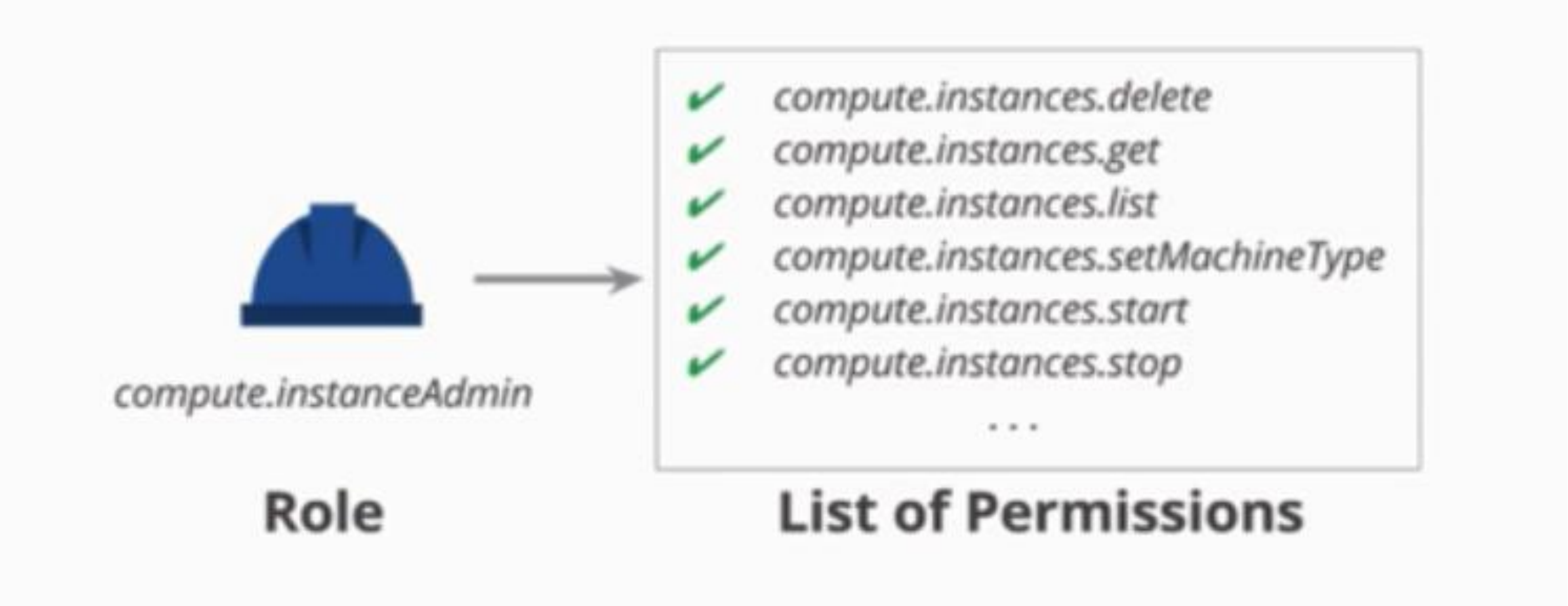
- Used to authenticate from one service to another

- Can be used with primitive and curated roles

- Identified with an email address:
  - ‘@developer.gserviceaccount.com’ ‘@developer.gserviceaccount.com’

Roles (the what)?

- What is a Role?
  - Collection of permissions to give access to given resource
  - Permissions represented in form <service>.<resource>.<verb>.
  - E.g.: compute.instances.delete
- Permission vs. Role
  - A users are not directly assigned permissions, but are assigned roles which contain a collection of permissions



Roles (the what)?

- Primitive vs. Predefined Roles:
  - Primitive Roles: The roles historically available in the Google Cloud Platform Console will continue to work. These are the Owner, Editor, and Viewer roles.
    - Viewer: read only actions (i.e. can’t make changes)
    - Editor: same as above + can modify state
    - Owner: same as above + manage access to project and all project resources. It can also set up project billing.

- Note

  Prior to Cloud IAM, you could only grant Owner, Editor, or Viewer roles. These roles give very broad access on a **project** and did not allow separation of duties. Cloud Platform services now offer additional roles that give finer-grained access control than the Owner, Editor, and Viewer roles. For example, Compute Engine offers roles such as Instance Admin and Network Admin, while App Engine offers roles such as App Admin and Service Admin. These predefined roles are available in addition to the legacy Owner, Editor, and Viewer roles.

When would I use primitive roles?

- When the Cloud Platform *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 need to allow a member to modify permissions for a project, you’ll want to grant them the owner role because only owners have the permission to grant access to other users for projects.

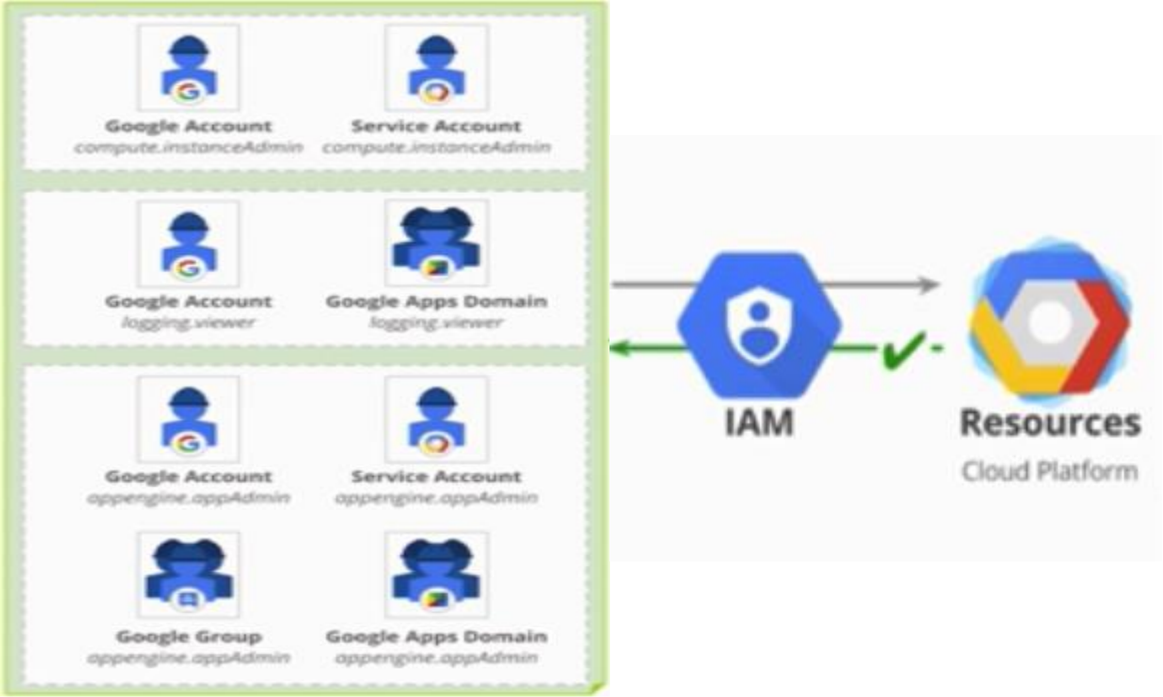
- When you work in a small team where the team members don’t need granular permissions.

IAM Roles - Predefined Roles

- Much more granular access, prevent unwanted access to other resources
- Granted at resource level
- Example: App Engine Admin – Full access to only App Engine resources
- Multiple predefined roles can be given to individual users
- All current Predefined Roles -
  - https://cloud.google.com/iam/docs/understanding-roles#predefined_roles

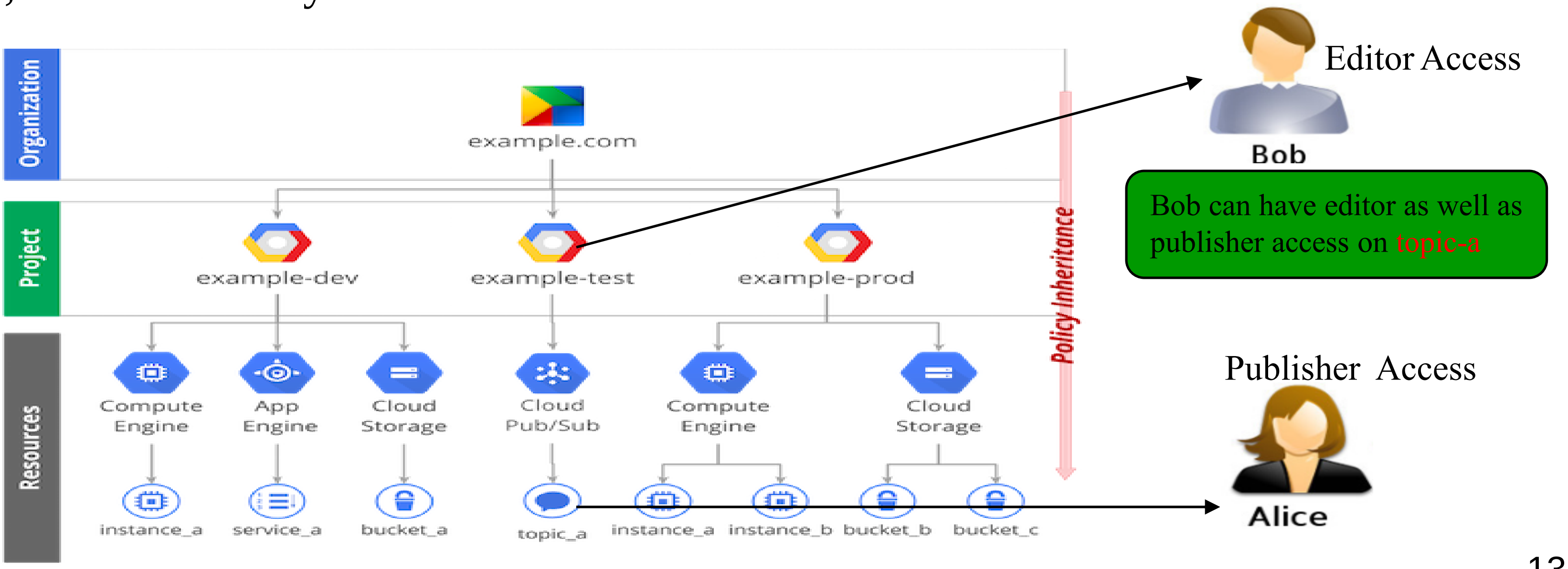
IAM Policy

- Collection of statements that define who has what type of access
- Full list of roles granted to a member to a resource



IAM Policy Hierarchy

- 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.
  - Organization→ Project→ Resources (parent/child format)
- You can set an IAM access control policy at any level in the resource hierarchy: The Organization level, the project level, or the resource level. 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.

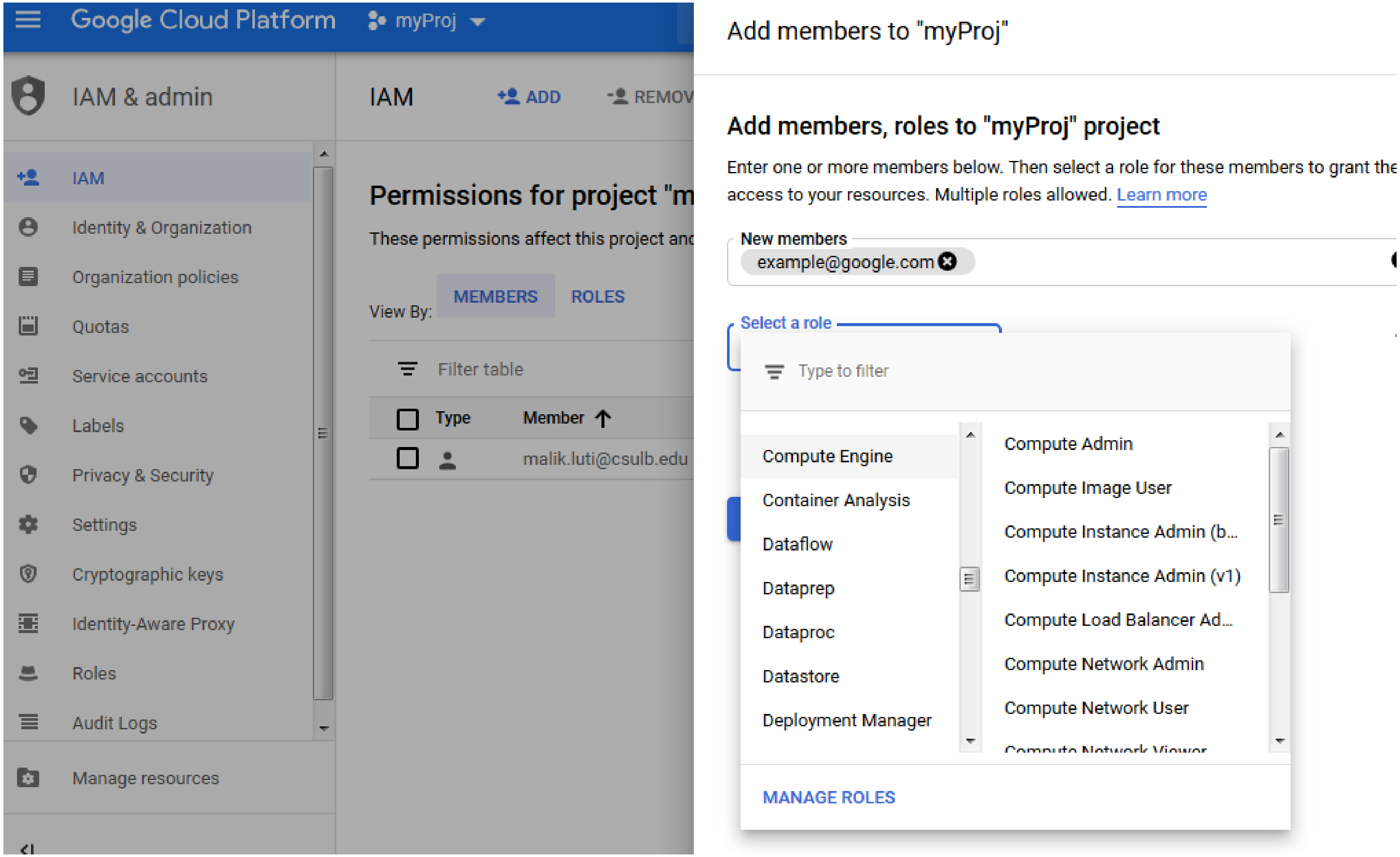


Bob can have editor as well as publisher access on topic-a

Summary

- Grants members (users, groups, organizations, service accounts)
- Roles are a collection of individual permissions
- Roles are assigned to members, not permission directly
- Roles include Primitive (broad) and Predefined (granular)
- IAM Policy is a full list of roles granted to members on a resource
- IAM Policies are hierarchically defined, with parent overruling child policy

Demo



Permissions for project "m

These permissions affect this project and

malik.luti@csulb.edu

Add members to "myProj"

Add members, roles to "myProj" project

Enter one or more members below. Then select a role for these members to grant the access to your resources. Multiple roles allowed. Learn more

New members: example@google.com

Select a role