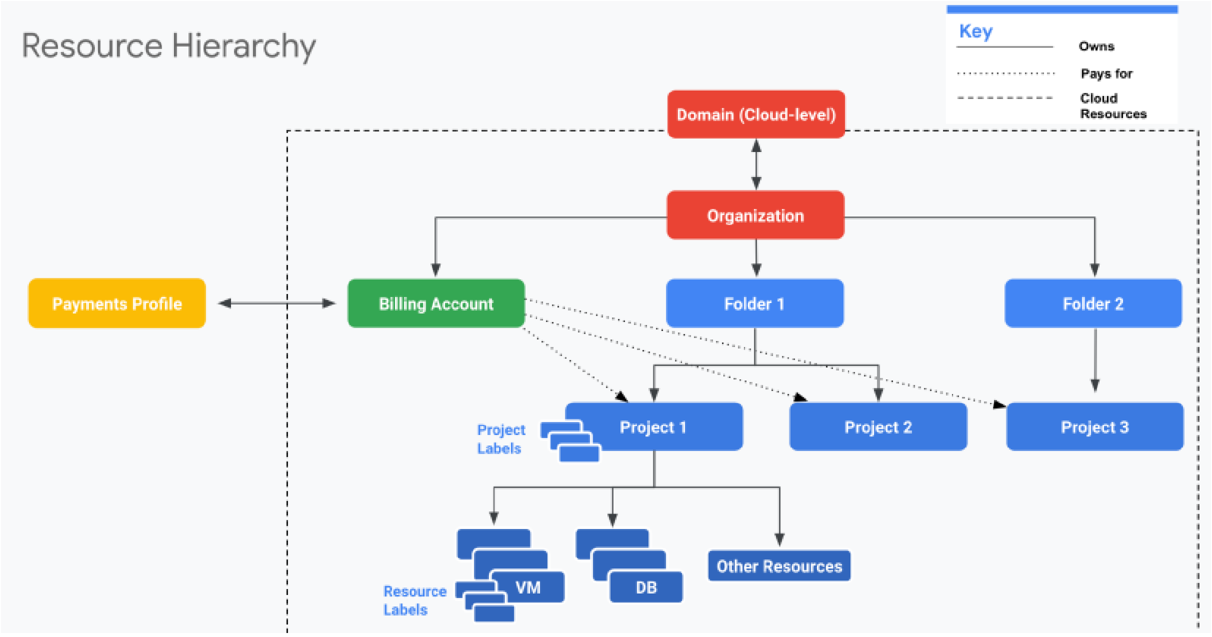
# Organizing GCP Resources

## Resource Hierarchy in GCP



- Well defined hierarchy:
- Organization > Folder > Project > Resources
- Resources are created in projects
- A Folder can contain multiple projects
- Organization can contain multiple Folders

## Resource Hierarchy - Recommendations for Enterprises

- Create separate projects for different environments:
- Complete isolation between test and production environments
- Create separate folders for each department:

- Isolate production applications of one department from another
- We can create a shared folder for shared resources
- One project per application per environment:
- Let's consider two apps: "A1" and "A2"
- Let's assume we need two environments: "DEV" and "PROD"
- In the ideal world you will create four projects: A1-DEV, A1-PROD, A2-DEV, A2-PROD:
- Isolates environments from each other
- DEV changes will NOT break PROD
- Grant all developers complete access (create, delete, deploy) to DEV Projects
- Provide production access to operations teams only!

## Billing Accounts

- Billing Account is mandatory for creating resources in a project:
- Billing Account contains the payment details
- Every Project with active resources should be associated with a Billing Account
- Billing Account can be associated with one or more projects
- You can have multiple billing accounts in an Organization

- (RECOMMENDATION) Create Billing Accounts representing your organization structure:
- A startup can have just one Billing account
- A large enterprise can have a separate billing account for each department
- Two Types:
- Self Serve : Billed directly to Credit Card or Bank Account
- Invoiced : Generate invoices (Used by large enterprises)

## Managing Billing - Budget, Alerts and Exports

- Setup a Cloud Billing Budget to avoid surprises:
- (RECOMMENDED) Configure Alerts
- Default alert thresholds set at 50%, 90% & 100%
- Send alerts to Pub Sub (Optional)
- Billing admins and Billing Account users are alerted by e-mail
- Billing data can be exported (on a schedule) to:
- Big Query (if you want to query information or visualize it)
- Cloud Storage (for history/archiving)

## IAM Best Practices

- Principle of Least Privilege - Give least possible privilege needed for a role!
- Basic Roles are NOT recommended
- Prefer predefined roles when possible

- Use Service Accounts with minimum privileges
- Use different Service Accounts for different apps/purposes
- Separation of Duties - Involve atleast 2 people in sensitive tasks:
- Example: Have separate deployer and traffic migrator roles
- AppEngine provides App Engine Deployer and App Engine Service Admin roles
- App Engine Deployer can deploy new version but cannot shift traffic
- App Engine Service Admin can shift traffic but cannot deploy new version!
- Constant Monitoring: Review Cloud Audit Logs to audit changes to IAM policies and access to Service Account keys
- Archive Cloud Audit Logs in Cloud Storage buckets for long term retention
- Use Groups when possible
- Makes it easy to manage users and permissions

## User Identity Management in Google Cloud

- Email used to create free trial account => "Super Admin"
- Access to everything in your GCP organization, folders and projects
- Manage access to other users using their Gmail accounts
- However, this is NOT recommended for enterprises
- Option 1: Your Enterprise is using Google Workspace

- Use Google Workspace to manage users (groups etc)
- Link Google Cloud Organization with Google Workspace
- Option 2: Your Enterprise uses an Identity Provider of its own
- Federate Google Cloud with your Identity Provider

## Corporate Directory Federation

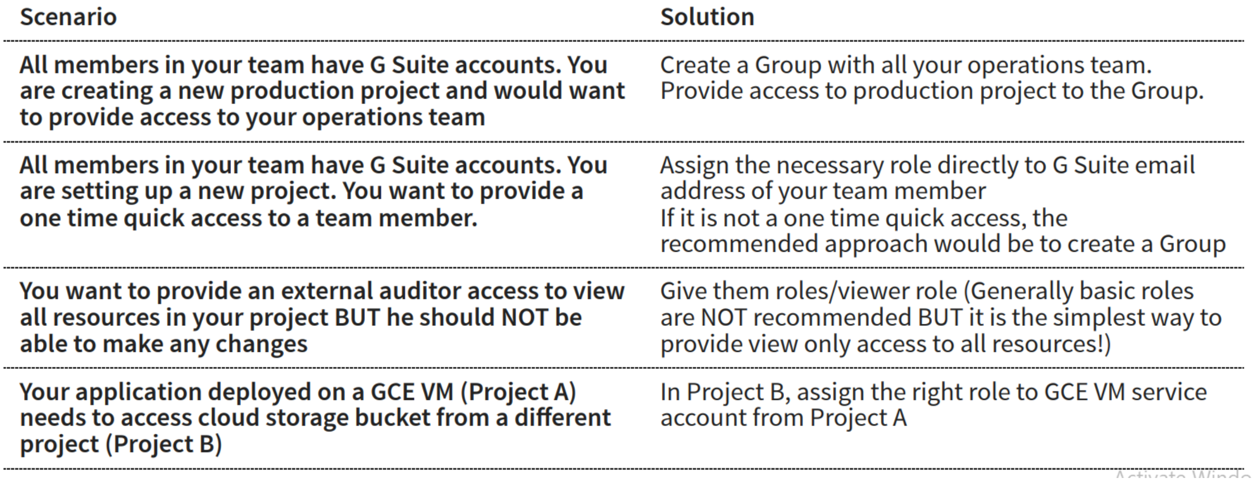
- Federate Cloud Identity or Google Workspace with your external identity provider (IdP) such as Active Directory or Azure Active Directory.
- Enable Single Sign On:
- 1: Users are redirected to an external IdP to authenticate
- 2: When users are authenticated, SAML assertion is sent to Google Sign-In
- Examples:
- Federate Active Directory with Cloud Identity by using Google Cloud Directory Sync (GCDS) and Active Directory Federation Services (AD FS)
- Federating Azure AD with Cloud Identity

## IAM Members/Identities

- Google Account - Represents a person (an email address)
- Service account - Represents an application account (Not person)
- Google group - Collection - Google & Service Accounts

- Has an unique email address
- Helps to apply access policy to a group
- Google Workspace domain: Google Workspace (formerly G Suite) provides collaboration services for enterprises:
- Tools like Gmail, Calendar, Meet, Chat, Drive, Docs etc are included
- If your enterprise is using Google Workspace, you can manage permissions using your Google Workspace domain
- Cloud Identity domain - Cloud Identity is an Identity as a Service (IDaaS) solution that centrally manages users and groups.
- You can use IAM to manage access to resources for each Cloud Identity account.

## IAM Members/Identities - Use Cases

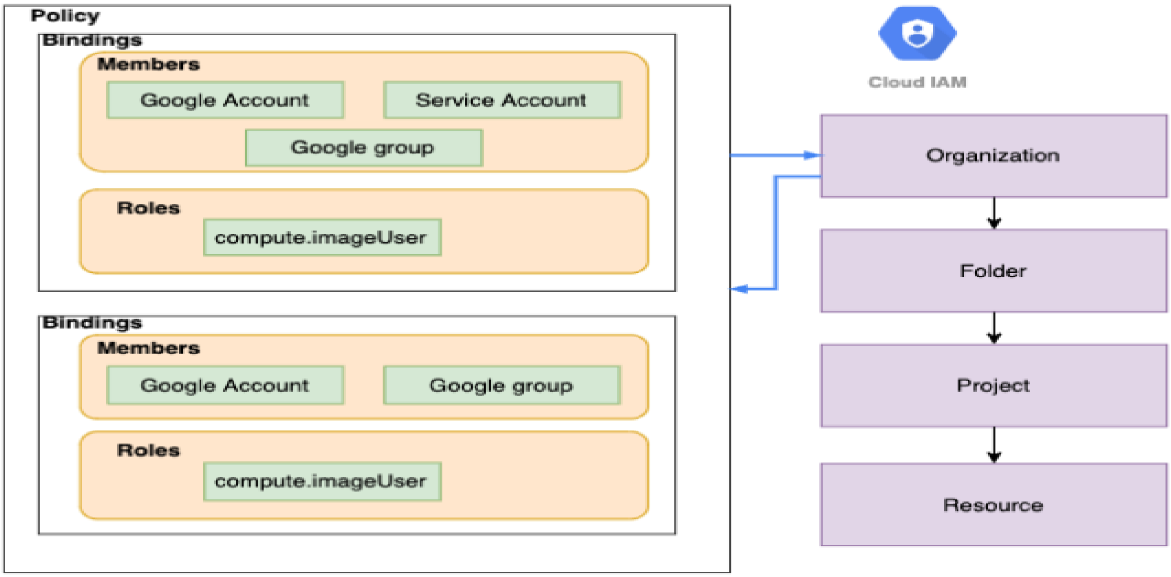


| Scenario | Solution |
|---|---|
| **All members in your team have G Suite accounts. You are creating a new production project and would want to provide access to your operations team** | Create a Group with all your operations team.<br>Provide access to production project to the Group. |
| **All members in your team have G Suite accounts. You are setting up a new project. You want to provide a one time quick access to a team member.** | Assign the necessary role directly to G Suite email address of your team member<br>If it is not a one time quick access, the recommended approach would be to create a Group |
| **You want to provide an external auditor access to view all resources in your project BUT he should NOT be able to make any changes** | Give them roles/viewer role (Generally basic roles are NOT recommended BUT it is the simplest way to provide view only access to all resources!) |
| **Your application deployed on a GCE VM (Project A) needs to access cloud storage bucket from a different project (Project B)** | In Project B, assign the right role to GCE VM service account from Project A |

## Organization Policy Service

- How to enable centralized constraints on all resources created in an Organization?
- Configure Organization Policy
- Example: Disable creation of Service Accounts
- Example: Allow/Deny creation of resources in specific regions
- Needs a Role - Organization Policy Administrator
- (Remember) IAM focuses on Who
- Who can take specific actions on resources?
- (Remember) Organization Policy focuses on What
- What can be done on specific resources?

## Resource Hierarchy & IAM Policy



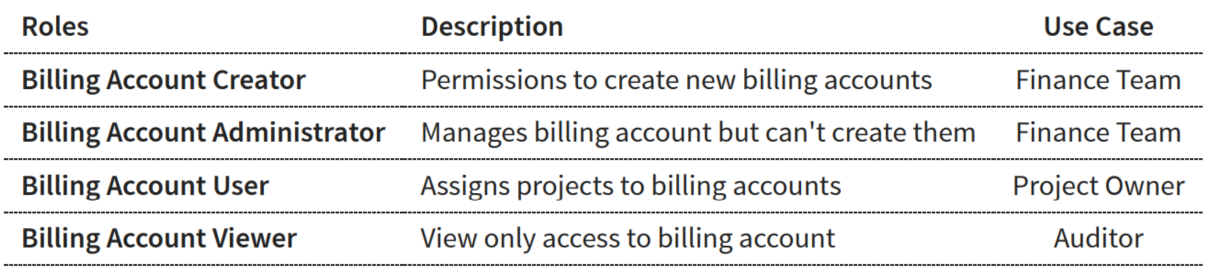
- IAM Policy can be set at any level of the hierarchy
- Resources inherit the policies of All parents
- The effective policy for a resource is the union of the policy on that resource and its parents
- Policy inheritance is transitive:
- For example:
- Organization policies are applied at resource level
- You can't restrict policy at lower level if permission is given at an higher level

## Organization, Billing and Project Roles

- Organization Administrator
- Define Resource Hierarchy
- Define Access Management Policies
- Manage other users and roles
- Billing Account Creator - Create Billing Accounts Billing
- Account Administrator - Manage Billing Accounts (payment instruments, billing exports, link and unlink projects, manage roles on billing account)
- CANNOT create a Billing Account
- Billing Account User - Associate Projects with Billing Accounts
- Typically used in combination with Project Creator
- These two roles allow user to create new project and link it with billing account

- Billing Account Viewer - See all Billing Account details

## Billing Roles - Quick Review



| Roles | Description | Use Case |
|---|---|---|
| **Billing Account Creator** | Permissions to create new billing accounts | Finance Team |
| **Billing Account Administrator** | Manages billing account but can't create them | Finance Team |
| **Billing Account User** | Assigns projects to billing accounts | Project Owner |
| **Billing Account Viewer** | View only access to billing account | Auditor |

## Organization, Billing and Project Roles – Scenarios

- Scenario 1: I'm creating a project and I want to associate an existing billing account with the project
- Roles needed : Project Creator and Billing Account User (link project to billing account)
- Scenario 2: I'm a billing auditor
- Roles needed : Billing Account Viewer role

## Compute Engine Roles

- Compute Engine IAM Roles
- Compute Engine Admin - Complete control of compute - Instances, Images, Load Balancers, Network, Firewalls etc...
- Compute Instance Admin - Create, modify, and delete virtual machine instances and disks
- Compute Engine Network Admin - Complete access to networking resources (routes, networks, health checks,

VPN, Gateways etc) and READ ONLY access to (firewall rules and SSL certificates)

- Compute Engine Security Admin - Complete access to firewall rules and SSL certificates
- Compute Storage Admin - Complete access to disks, images, snapshots
- Compute Engine Viewer - Read ONLY access to everything in compute
- Compute OS Admin Login - Log in to a Compute Engine instance as an administrator user
- Compute OS Login - Log in to a Compute Engine instance as a standard user.

## App Engine Roles

- App Engine Roles (CRUD - Create, Read (get/list), Update, Delete)
- App Engine Creator - applications(CD) (Responsible for creating an application)
- App Engine Admin - applications(RU), services/instances/versions(CRUD), operations
- App Engine Viewer - applications/services/instances/versions(R), operations
- App Engine Code Viewer - appengine.versions.getFileContents (ONLY role that can view code)

- App Engine Deployer - versions(CRD), applications/services/versions(R)
- Deploy a new version of an app (if you also grant the Service Account User role)
- App Engine Service Admin - versions(RUD), applications(R), services/instances(CRUD), operations: Split or migrate traffic, Start and stop a version
- App Engine Roles DO NOT allow you to
- View and download application logs
- View Monitoring charts in the Cloud Console
- Enable and Disable billing
- Access configuration or data stored in other services

## Compute Engine and App Engine Roles - Few Scenarios

- Scenario 1: What is the difference between Compute Engine Admin vs Compute Instance Admin?
- Compute Instance Admin can do everything with instances and disks ONLY. Compute Engine Admin is admin for everything in compute - instances, disks, images, network, firewalls etc.
- Scenario 2: What is a secure way of setting up application deployment?
- Application Deployer - Roles: App Engine Deployer + Service Account User

- Limited to deploying new versions and deleting old versions that are not serving traffic
- Will NOT be able to configure traffic
- Operations - Role: App Engine Service Admin
- CANNOT deploy a new version of an app
- Change traffic between versions

## Google Kubernetes Engine (GKE) IAM Roles

- Kubernetes Engine Admin (roles/container.admin) - Complete Access to Clusters and Kubernetes API objects
- Kubernetes Engine Cluster Admin - Provides access to management of clusters (Cannot access Kubernetes API objects - Deployments, Pods etc)
- Kubernetes Engine Developer - Manage Kubernetes API objects (and read cluster info)
- Kubernetes Engine Viewer - get/list cluster and kubernetes api objects

## Cloud Storage – Roles

- Storage Admin - storage.buckets.*, storage.objects.*
  Storage Object Admin - storage.objects.* (DOES NOT HAVE storage.buckets.*)
- Storage Object Creator - storage.objects.create
- Storage Object Viewer - storage.objects.get, storage.objects.list

- (REMEMBER) Container Registry stores container images in Cloud Storage buckets
- Control access to images in Container Registry using Cloud Storage permissions!
- (REMEMBER) Storage Admin vs Storage Object Admin
- Storage Admin can create buckets and play with objects
- Storage Object Admin CANNOT create buckets but can play with objects in a bucket!

## Cloud BigQuery Roles

- Cloud BigQuery IAM Roles
- BigQuery Admin - bigquery.*
- BigQuery Data Owner - bigquery.datasets.* , bigquery.models.* , bigquery.routines.* , bigquery.tables.* (Does NOT have access to Jobs!)
- BigQuery Data Editor - bigquery.tables.(create/delete/export/get/getData/getIamPolicy/ list/update/updateData/updateTag), bigquery.models.* , bigquery.routines.* , bigquery.datasets.(create/get/getIamPolicy/updateTag)
- BigQuery Data Viewer - get/list bigquery.(datasets/models/routines/tables)
- BigQuery Job User - bigquery.jobs.create
- BigQuery User - BigQuery Data Viewer + get/list (jobs, capacityCommitments, reservations etc)

- To see data, you need either BigQuery User or BigQuery Data Viewer roles
- You CANNOT see data with BigQuery Job User roles
- BigQuery Data Owner or Data Viewer roles do NOT have access to jobs!

## Logging IAM Roles and Service Account Roles

- Logging and Audit Logging:
- roles/logging.viewer (Logs Viewer): Read all Logs except Access Transparency logs and Data Access audit logs.
- roles/logging.privateLogViewer (Private Logs Viewer): Logs Viewer + Read Access Transparency logs and Data Access audit logs
- roles/logging.admin (Logging Admin): All permissions related to Logging
- Service Accounts:
- roles/iam.serviceAccountAdmin: Create and manage service accounts
- roles/iam.serviceAccountUser: Run operations as the service account
- roles/iam.serviceAccountUser => create and manage instances that use a service account. This needs to be added to Admin roles if you want them to attach service accounts with instances.

- roles/iam.serviceAccountTokenCreator - Impersonate service accounts (create OAuth2 access tokens, sign blobs or JWTs, etc).
- roles/iam.serviceAccountKeyAdmin - Create and manage (and rotate) service account keys.

## Other Important IAM Roles

- roles/iam.securityAdmin - Get and set any IAM policy
- roles/iam.securityReviewer - List all resources & IAM policies
- roles/iam.organizationRoleAdmin - Administer all custom roles in the organization and the projects below it
- roles/iam.organizationRoleViewer - Read all custom roles in the organization and the projects below it
- roles/iam.roleAdmin - Provides access to all custom roles in the project
- roles/iam.roleViewer - Provides read access to all custom roles in the project
- roles/browser - Read access to browse the hierarchy for a project, including the folder, organization, and IAM policy
- This role doesn't include permission to view resources in the project.

## SSHing into Linux VMs – Options

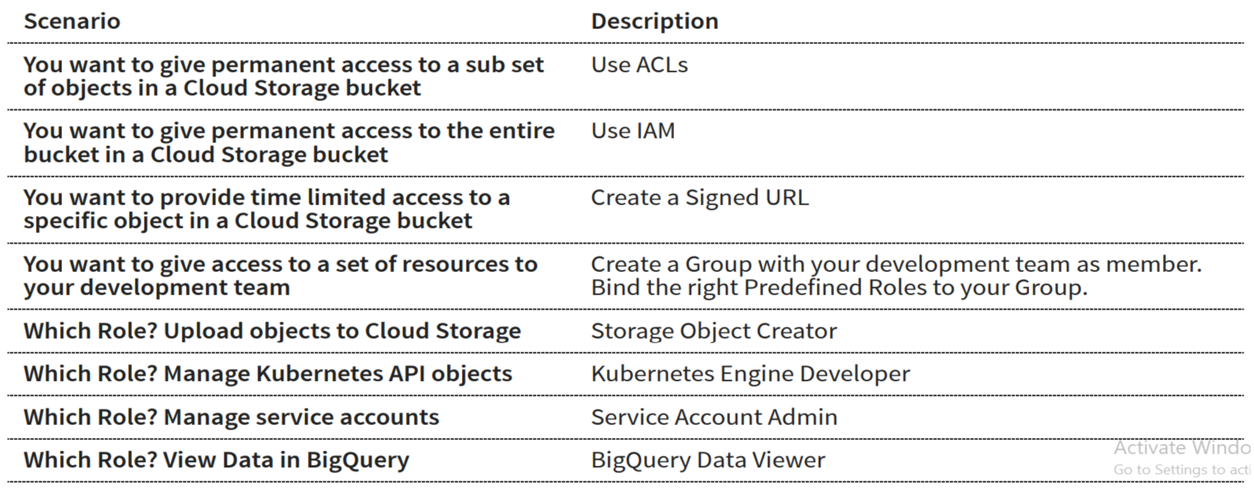
- Compute Engine Linux VMs uses key-based SSH authentication
- Two Options:
- Metadata managed: Manually create and configure individual SSH keys
- OS Login: Manage SSH access without managing individual SSH keys!
- Recommended for managing multiple users across instances or projects
- Your Linux user account is linked to your Google identity
- To enable: Set enable-oslogin to true in metadata
- gcloud compute project-info/instances add-metadata --metadata enable-oslogin=TRUE
- (Advantage) Ability to import existing Linux accounts from on premises AD and LDAP
- Users need to have roles : roles/compute.osLogin or roles/compute.osAdminLogin
- (Windows) Windows instances use password authentication(username and password)
- Generate using console or gcloud (gcloud compute reset-windows-password)

## SSHing into Linux VMs – Details

- Option 1: Console - SSH Button
- Ephemeral SSH key pair is created by Compute Engine

- Option 2: Gcloud - gcloud compute ssh
- A username and persistent SSH key pair are created by Compute Engine
- SSH key pair reused for future interactions
- Option 3: Use customized SSH keys
- (Metadata managed): Upload the public key to project metadata OR
- (OS Login): Upload your public SSH key to your OS Login profile
- gcloud compute os-login ssh-keys add OR
- Use OS Login API : POST
- You can disable Project wide SSH keys on a specific compute instance
- gcloud compute instances add-metadata [INSTANCE_NAME] --metadata blockproject-ssh-keys=TRUE

## IAM – Scenarios



| Scenario | Description |
|---|---|
| **You want to give permanent access to a sub set of objects in a Cloud Storage bucket** | Use ACLs |
| **You want to give permanent access to the entire bucket in a Cloud Storage bucket** | Use IAM |
| **You want to provide time limited access to a specific object in a Cloud Storage bucket** | Create a Signed URL |
| **You want to give access to a set of resources to your development team** | Create a Group with your development team as member. Bind the right Predefined Roles to your Group. |
| **Which Role? Upload objects to Cloud Storage** | Storage Object Creator |
| **Which Role? Manage Kubernetes API objects** | Kubernetes Engine Developer |
| **Which Role? Manage service accounts** | Service Account Admin |
| **Which Role? View Data in BigQuery** | BigQuery Data Viewer |