04 IAM Policy -KirkYagami👨‍💻🕵️

Policy

[Policy | IAM Documentation | Google Cloud]
An Identity and Access Management (IAM) policy, which specifies access controls for Google Cloud resources.

A `Policy` is a collection of `bindings`. A `binding` binds one or more `members`, or principals, to a single `role`. Principals can be user accounts, service accounts, Google groups, and domains (such as G Suite). A `role` is a named list of permissions; each `role` can be an IAM predefined role or a user-created custom role.

For some types of Google Cloud resources, a `binding` can also specify a `condition`, which is a logical expression that allows access to a resource only if the expression evaluates to `true`. A condition can add constraints based on attributes of the request, the resource, or both.

Setting IAM Policy

We can set IAM Policy at:

- Organization level
- Folder level
- Project level
- Resource level (in some cases)

Inheritance of IAM Policies

- IAM Policy set at the organization level is inherited by all its child folders, projects, and resources.
- IAM Policy set at the project level is inherited by all the child resources (Cloud resources like Compute Engine, Cloud Run, etc.).

The effective policy on a resource is the union of the policy set at that resource and the policy inherited from its ancestors.

Example

- Development Project IAM Policy:

- IAM Policy (COMPANY + Department B + Team B + Product 1 + Development Project)

IAM Role Binding

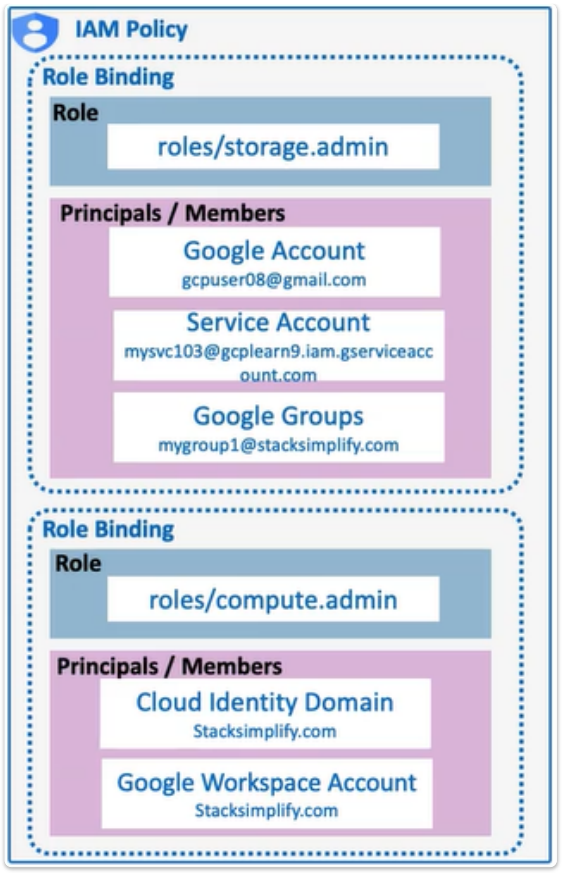
- Bind one or more principals to an individual IAM Role.
- Principals or Members + IAM Role.
- IAM Policy (Default: Allow Policy).
- Collection of role bindings that bind one or more principals to an individual role.
- IAM Policy can have one or more role bindings.
- An allow policy is attached to a resource.

Example

- Organization, Folder, Project, or Cloud Resource (Storage Bucket, VM Instance).
- An allow policy will enforce access control whenever that resource is accessed.

Policy Inheritance

- Policy applied at organization, folder, or project level will be inherited to cloud resource level (Storage Bucket or VM Instance).



IAM Role Binding Commands

- **add-iam-policy-binding:** Add IAM policy binding for a resource.
- **get-iam-policy:** Get IAM policy for a resource.
- **remove-iam-policy-binding:** Remove IAM policy binding for a resource.
- **set-iam-policy:** Set IAM policy for a resource.

Resource

- **Project:** bigdata3844

Example Commands

Free Tier - only project level is available

- **ADD:**

```
gcloud projects add-iam-policy-binding bigdata3844 --member
user:yagamikirk@gmail.com --role=roles/storage.admin
```

- **GET:**

```
gcloud projects get-iam-policy bigdata3844
```

- **REMOVE:**

```
gcloud projects remove-iam-policy-binding bigdata3844 --member
user:yagamikirk@gmail.com --role=roles/storage.admin
```

JSON example:

```
{
  "bindings": [
    {
      "role": "roles/resourcemanager.organizationAdmin",
      "members": [
        "user:mike@example.com",
        "group:admins@example.com",
        "domain:google.com",
        "serviceAccount:my-project-id@appspot.gserviceaccount.com"
      ]
    },
    {
      "role": "roles/resourcemanager.organizationViewer",
      "members": [
        "user:eve@example.com"
      ],
      "condition": {
        "title": "expirable access",
        "description": "Does not grant access after Sep 2020",
        "expression": "request.time < timestamp('2020-10-01T00:00:00.000Z')",
      }
    }
  ],
  "etag": "BwWWja0YfJA=",
  "version": 3
}
```

YAML example:

```
bindings:
- members:
  - user:mike@example.com
  - group:admins@example.com
```

```
    - domain:google.com
    - serviceAccount:my-project-id@appspot.gserviceaccount.com
    role: roles/resourcemanager.organizationAdmin
  - members:
    - user:eve@example.com
    role: roles/resourcemanager.organizationViewer
    condition:
      title: expirable access
      description: Does not grant access after Sep 2020
      expression: request.time < timestamp('2020-10-01T00:00:00.000Z')
  etag: BwWWja0YfJA=
  version: 3
```