Using Constraints

This guide explains how to create an organization policy with a particular constraint. The constraints used in these examples will not be actual constraints, but generalized samples for educational purposes.

For more information on constraints and the problems they solve, review the list of all Organization Policy Service constraints (https://cloud.google.com/resource-manager/docs/organization-policy/org-policy-constraints).

Before you begin

- Read the Introduction to the Organization Policy Service (https://cloud.google.com/resource-manager/docs/organization-policy/overview) page to learn how organization policy works.
- Read the Understanding constraints (https://cloud.google.com/resource-manager/docs/organization-policy/understanding-constraints) page to learn how constraints are constructed.
- Read the Understanding hierarchy evaluation (https://cloud.google.com/resource-manager/docs/organization-policy/understanding-hierarchy) page to learn about policy inheritance.

Add an organization policy administrator

To add a user as an **Organization Policy Administrator**, you must have the **Organization Administrator** (https://cloud.google.com/resource-manager/docs/creating-managing-organization#adding_an_organization_admin) role. This role can only be granted at the Organization level. You must have the **Organization Policy Administrator** role to set or change organization policies.

CONSOLE GCLOUD

To add an organization policy administrator:

1. Sign in to the Google Cloud Console as a G Suite or Cloud Identity super administrator and go to the **Manage resources** page:

   GO TO THE MANAGE RESOURCES PAGE (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/CLOUD-RESOU

2. On the **Organization** drop-down list, select your organization.
3. In the list of resources that appears, select the check box next to the Organization resource.
4. On the right side **Info Panel**, under **Permissions**, click **Add member**.
5. Enter the email address of the member you want to add.
6. In the **Select a role** drop-down list, select **Organization Policy > Organization Policy Administrator**.
7. Click **Save**. A dialog will appear to confirm the addition or update of the member's new role.

Using list constraints in organization policy

Set up enforcement on the organization resource

You can set an organization policy on your organization resource that uses a list constraint to deny access to a particular service. The following process describes how to set an organization policy using the gcloud command-line tool. For instructions on how to view and set organization policies using the Cloud Console, see Creating and Managing Policies (https://cloud.google.com/resource-manager/docs/organization-policy/creating-managing-policies).

1. Get the current policy on the organization resource using the `describe` command:

   ```
   gcloud beta resource-manager org-policies describe \
     LIST_CONSTRAINT --organization ORGANIZATION_ID
   ```

   Where:

   - ***ORGANIZATION_ID*** (https://cloud.google.com/resource-manager/docs/creating-managing-organization#retrieving_your_organization_id) is a unique identifier for the organization resource.
   - ***LIST_CONSTRAINT*** is the list constraint for the service that you want to enforce.

You can also apply the organization policy to a folder or a project with the `--folder` or the `--project` flags, and the folder ID (https://cloud.google.com/resource-manager/docs/creating-managing-folders#configuring_access_to_folders) and project ID (https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects), respectively.

Because a policy isn't set, an incomplete policy is returned, like the following example:

```
constraint: "constraints/LIST_CONSTRAINT"
etag: BwVJi0OOESU=
```

2. Use the `deny` command to add the denied value for the service to which you want to restrict access.

```
gcloud beta resource-manager org-policies deny \
  LIST_CONSTRAINT VALUE_A \
  --organization ORGANIZATION_ID
```

The output of the command will be:

```
constraint: constraints/LIST_CONSTRAINT
etag: BwVJi0OOESU=
listPolicy:
  deniedValues:
    - VALUE_A
updateTime: CURRENT_TIME
```

3. View the current effective policy using `describe --effective`.

```
gcloud beta resource-manager org-policies describe \
  LIST_CONSTRAINT --effective \
  --organization ORGANIZATION_ID
```

The output of the command will be:

```
constraint: constraints/LIST_CONSTRAINT
listPolicy:
  deniedValues:
    - VALUE_A
```

Because this organization policy was set at the organization level, it will be inherited (https://cloud.google.com/resource-manager/docs/organization-policy/understanding-hierarchy) by all child resources that allow inheritance.

Set up enforcement against a hierarchy subtree

List constraints take explicitly defined values to determine which resources should be allowed or denied. Some constraints can also accept values that use the prefix `under:`, which specifies a subtree with that resource as the root. Using the `under:` prefix on an allowed or denied value causes the organization policy to act on that resource and all of its children. For information about the constraints that allow using the `under:` prefix, see the Organization policy constraints (https://cloud.google.com/resource-manager/docs/organization-policy/org-policy-constraints) page.

A value that uses the `under:` prefix is called a hierarchy subtree string. A hierarchy subtree string specifies the type of resource it applies to. For example, using a subtree string of ***projects/PROJECT_ID*** when setting the `constraints/compute.storageResourceUseRestrictions` constraint will allow or deny the use of Compute Engine storage for ***PROJECT_ID*** and all of its children.

Hierarchy subtree value prefixes are a beta feature, might be changed in backward-incompatible ways, and are not subject to any SLA or deprecation policy.

1. Get the current policy on the organization resource using the `describe` command:

```
gcloud beta resource-manager org-policies describe \
  LIST_CONSTRAINT --organization ORGANIZATION_ID
```

Where:

- ***ORGANIZATION_ID*** (https://cloud.google.com/resource-manager/docs/creating-managing-organization#retrieving_your_organization_id) is a unique identifier for the organization resource.
- ***LIST_CONSTRAINT*** is the list constraint for the service that you want to enforce.

You can also apply the organization policy to a folder or a project with the `--folder` or the `--project` flags, and the folder ID (https://cloud.google.com/resource-manager/docs/creating-managing-folders#configuring_access_to_folders) and project ID (https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects), respectively.

Because a policy isn't set, an incomplete policy is returned, like the following example:

```
constraint: "constraints/LIST_CONSTRAINT"
etag: BwVJi0OOESU=
```

2. Use the `deny` command to add the denied value for the service to which you want to restrict access. The `under:` prefix sets the constraint to deny the named resource and all of its child resources.

```
gcloud beta resource-manager org-policies deny \
  LIST_CONSTRAINT under:folders/VALUE_A \
  --organization ORGANIZATION_ID
```

Where:

- ***under:*** is a prefix that signifies what follows is a subtree string.
- ***folders/VALUE_A*** is the folder ID (https://cloud.google.com/resource-manager/docs/creating-managing-folders#configuring_access_to_folders) of the root resource you want to deny. This resource and all of its children in the resource hierarchy will be denied.
- ***VALUE_B*** and ***VALUE_C*** are projects that exist in the hierarchy with ***VALUE_A*** as their parent.

The output of the deny command will be:

```
constraint: constraints/LIST_CONSTRAINT
etag: BwVJi0OOESU=
listPolicy:
  deniedValues:
    - under:folders/VALUE_A
updateTime: CURRENT_TIME
```

You can also apply the `under:` prefix to organizations and projects, as in the following examples:

- ***under:organizations/VALUE_X***

- ***under:projects/VALUE_Y***

3. View the current effective policy using `describe --effective`.

```
gcloud beta resource-manager org-policies describe \
  LIST_CONSTRAINT --effective \
  --organization ORGANIZATION_ID
```

The output of the command will be:

```
constraint: constraints/LIST_CONSTRAINT
listPolicy:
  deniedValues:
    - under:folders/VALUE_A
```

The policy now evaluates to deny the folder ***VALUE_A*** and all of its child resources, in this case ***VALUE_B*** and ***VALUE_C***.

Merge the organization policy on a project

You can set a custom organization policy on a resource, which will merge with any policy inherited from its parent resource. This merged policy will then be evaluated to create a new effective policy based on the rules of inheritance (https://cloud.google.com/resource-manager/docs/organization-policy/understanding-hierarchy).

1. Get the current policy on the resource using the `describe` command:

```
gcloud beta resource-manager org-policies describe \
  LIST_CONSTRAINT --project PROJECT_ID
```

Where:

- ***PROJECT_ID*** is the unique identifier (https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects) of your project.
- ***LIST_CONSTRAINT*** is the list constraint for the service that you want to enforce.

Because a policy isn't set, an incomplete policy is returned, like the following example:

```
constraint: "constraints/LIST_CONSTRAINT"
etag: BwVJi0OOESU=
```

2. Display the current effective policy using the `describe --effective` command:

   ```
   gcloud beta resource-manager org-policies describe \
     LIST_CONSTRAINT --effective \
      --project PROJECT_ID
   ```

   The output of the command will include a denied value that it inherits from the organization resource:

   ```
   constraint: constraints/LIST_CONSTRAINT
   listPolicy:
     deniedValues:
       - VALUE_A
   ```

3. Set the policy on the project using the `set-policy` command.

   a. Create a temporary file `/tmp/policy.yaml` to store the policy:

      ```
      constraint: constraints/LIST_CONSTRAINT
      listPolicy:
        deniedValues:
          - VALUE_B
          - VALUE_C
        inheritFromParent: true

      ```

   b. Run the `set-policy` command:

      ```
      gcloud beta resource-manager org-policies set-policy

        --project PROJECT_ID /tmp/policy.yaml

      ```

   c. The output of the command will be:

      ```
      constraint: constraints/LIST_CONSTRAINT
      etag: BwVLO2timxY=
      listPolicy:
        deniedValues:
      ```

```
      - VALUE_B
      - VALUE_C
    inheritFromParent: true
```

4. Use the `describe --effective` command again to display the updated policy:

   ```
   gcloud beta resource-manager org-policies describe \
     LIST_CONSTRAINT --effective \
     --project PROJECT_ID
   ```

   The output of the command will include the effective result of merging the policy from the resource and from the parent:

   ```
   constraint: constraints/LIST_CONSTRAINT
     listPolicy:
       deniedValues:
         - VALUE_A
         - VALUE_B
         - VALUE_C
   ```

Restore default constraint behavior

You can use the `restoreDefault` value in an organization policy to reset the policy to use the constraint's default behavior. The following example assumes that the default constraint behavior is to allow all values.

1. Get the effective policy on the project to show the current merged policy:

   ```
   gcloud beta resource-manager org-policies describe \
     LIST_CONSTRAINT --effective \
     --project PROJECT_ID
   ```

   Where ***PROJECT_ID*** is the unique identifier (https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects) of your project. The output of the command will be:

   ```
   constraint: constraints/LIST_CONSTRAINT
     listPolicy:
       deniedValues:
   ```

```
    - VALUE_A
    - VALUE_B
    - VALUE_C
```

2. Set the policy on the project using the `set-policy` command.

   a. Create a temporary file `/tmp/restore-policy.yaml` to store the policy:

   ```
   restoreDefault: {}
   constraint: constraints/LIST_CONSTRAINT
   ```

   b. Run the `set-policy` command:

   ```
   gcloud beta resource-manager org-policies set-policy

     --project PROJECT_ID /tmp/restore-policy.yaml
   ```

   c. The output of the command will be:

   ```
   constraint: constraints/LIST_CONSTRAINT
   etag: BwVJi9D3VLY=
   restoreDefault: {}
   ```

3. Get the effective policy to verify the default behavior:

   ```
   gcloud beta resource-manager org-policies describe \
     LIST_CONSTRAINT --effective \
     --project PROJECT_ID
   ```

   The output of the command will allow all values:

   ```
   Constraint: constraints/LIST_CONSTRAINT
   listPolicy:
     allValues: ALLOW
   ```

Delete an organization policy

You can delete an organization policy from a resource. A resource without an organization policy set will inherit any policy of its parent resource. If you delete the organization policy on the organization resource, the effective policy will be the constraint's default behavior.

The following steps describe how to delete an organization policy on an organization:

1. Delete the policy on the organization resource using the `delete` command:

   ```
   gcloud beta resource-manager org-policies delete \
     LIST_CONSTRAINT --organization ORGANIZATION_ID
   ```

   Where ***ORGANIZATION_ID*** (https://cloud.google.com/resource-manager/docs/creating-managing-organization#retrieving_your_organization_id) is the unique identifier for the organization resource. The output of the command will be:

   ```
   Deleted [<Empty>].
   ```

2. Get the effective policy on the organization to verify it's not enforced:

   ```
   gcloud beta resource-manager org-policies describe \
     LIST_CONSTRAINT --effective \
     --organization ORGANIZATION_ID
   ```

   The output of the command will be:

   ```
   constraint: constraints/LIST_CONSTRAINT
   listPolicy:
     allValues: ALLOW
   ```

The following steps describe how to delete an organization policy on a project:

1. Delete the policy on a project using the `delete` command:

   ```
   gcloud beta resource-manager org-policies delete \
     LIST_CONSTRAINT --project PROJECT_ID
   ```

   Where ***PROJECT_ID*** is the unique identifier (https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects) of your project. The output of the command will be:

```
Deleted [<Empty>].
```

2. Get the effective policy on the project to verify it's not enforced:

```
gcloud beta resource-manager org-policies describe \
  --effective \
  LIST_CONSTRAINT --project PROJECT_ID
```

The output of the command will be:

```
constraint: constraints/LIST_CONSTRAINT
listPolicy:
  allValues: ALLOW
```

Using boolean constraints in organization policy

Set up enforcement on the organization resource

You can set an organization policy on your organization resource to enforce a boolean constraint. The following process describes how to set an organization policy using the gcloud command-line tool. For instructions on how to view and set organization policies using the Cloud Console, see Creating and Managing Policies (https://cloud.google.com/resource-manager/docs/organization-policy/creating-managing-policies).

1. Get the current policy on the organization resource by using the `describe` command:

```
gcloud beta resource-manager org-policies describe \
  BOOLEAN_CONSTRAINT --organization ORGANIZATION_ID
```

Where ***ORGANIZATION_ID*** (https://cloud.google.com/resource-manager/docs/creating-managing-organization#retrieving_your_organization_id) is the unique identifier for the organization resource. You can also apply the organization policy to a folder or a project with the `--folder` or the `--project` flags, and the folder ID (https://cloud.google.com/resource-manager/docs/creating-managing-folders#configuring_access_to_folders) and project ID

(https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects)
, respectively.

Because a policy isn't set, an incomplete policy is returned, like the following example:

```
booleanPolicy: {}
constraint: "constraints/BOOLEAN_CONSTRAINT"
```

2. Set the policy to enforce on the organization using the `enable-enforce` command:

```
gcloud  resource-manager org-policies enable-enforce \
  BOOLEAN_CONSTRAINT --organization ORGANIZATION_ID
```

The output of the command will be:

```
booleanPolicy:
  enforced: true
constraint: constraints/BOOLEAN_CONSTRAINT
etag: BwVJitxdiwY=
```

3. View the current effective policy using `describe --effective`:

```
gcloud beta resource-manager org-policies describe \
  BOOLEAN_CONSTRAINT --effective \
  --organization ORGANIZATION_ID
```

The output of the command will be:

```
booleanPolicy:
  enforced: true
constraint: constraints/BOOLEAN_CONSTRAINT
```

Override the organization policy for a project

To override the organization policy for a project, set a policy that disables enforcement of the boolean constraint to all resources in the hierarchy below the project.

1. Get the current policy on the resource to show it's empty.

```
gcloud beta resource-manager org-policies describe \
  BOOLEAN_CONSTRAINT --project PROJECT_ID
```

Where ***PROJECT_ID*** is the unique identifier (https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects) of your project. The output of the command will be:

```
booleanPolicy: {}
constraint: "constraints/BOOLEAN_CONSTRAINT"
```

2. Get the effective policy on the project, which confirms that the constraint is being enforced at this project.

```
gcloud beta resource-manager org-policies describe \
  BOOLEAN_CONSTRAINT --effective \
  --project PROJECT_ID
```

The output of the command will be:

```
booleanPolicy:
  enforced: true
constraint: constraints/BOOLEAN_CONSTRAINT
```

3. Set the policy on the project to not enforce the constraint, using the `disable-enforce` command:

```
gcloud beta resource-manager org-policies disable-enforce \
  BOOLEAN_CONSTRAINT --project PROJECT_ID
```

The output of the command will be:

```
booleanPolicy: {}
constraint: constraints/BOOLEAN_CONSTRAINT
etag: BwVJivdnXvM=
```

4. Get the effective policy to show that it is no longer enforced on the project.

```
gcloud beta resource-manager org-policies describe \
  --effective \
  BOOLEAN_CONSTRAINT --project PROJECT_ID
```

The output of the command will be:

```
booleanPolicy: {}
constraint: constraints/BOOLEAN_CONSTRAINT
```

Delete an organization policy

You can delete an organization policy from a resource. A resource without an organization policy set will inherit any policy of its parent resource. If you delete the organization policy on the organization resource, the effective policy will be the constraints' default behavior.

The following steps describe how to delete an organization policy on an organization and a project:

1. Delete the policy from the organization resource using the `delete` command:

   ```
   gcloud beta resource-manager org-policies delete \
     BOOLEAN_CONSTRAINT --organization ORGANIZATION_ID
   ```

   Where ***ORGANIZATION_ID*** (https://cloud.google.com/resource-manager/docs/creating-managing-organization#retrieving_your_organization_id) is a unique identifier for the organization resource. The output of the command will be:

   ```
   Deleted [<Empty>].
   ```

2. Get the effective policy on the organization to verify it's not enforced:

   ```
   gcloud beta resource-manager org-policies describe \
     --effective \
     BOOLEAN_CONSTRAINT --organization ORGANIZATION_ID
   ```

   The output of the command will be:

   ```
   booleanPolicy: {}
   constraint: constraints/BOOLEAN_CONSTRAINT
   ```

3. Delete the organization policy from the project using the `delete` command:

```
gcloud beta resource-manager org-policies delete \
  BOOLEAN_CONSTRAINT --project PROJECT_ID
```

The output of the command will be:

```
Deleted [<Empty>].
```

4. Get the effective policy on the project to verify it's not enforced:

```
gcloud beta resource-manager org-policies describe \
  BOOLEAN_CONSTRAINT --effective \
  --project PROJECT_ID
```

Where the **PROJECT_ID** (https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects) is the unique identifier of your project. The output of the command will be:

```
booleanPolicy: {}
constraint: constraints/BOOLEAN_CONSTRAINT
```

Last updated December 4, 2019.