Understanding Constraints

A constraint is a type of restriction against a Google Cloud service (https://cloud.google.com/docs/overview/cloud-platform-services) or a list of Google Cloud services. Think of the constraint as a blueprint that defines what behaviors are controlled. This blueprint is then applied to a resource hierarchy (https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy) node as an organization policy, which implements the rules defined in the constraint. The Google Cloud service mapped to that constraint and associated with that resource hierarchy node will then enforce the restrictions configured within the organization policy.

A constraint has a type, which determines the organization policy values that can be entered and used for checking enforcement. The enforcing Google Cloud service will evaluate the constraint type and value to determine restriction.

During hierarchy evaluation (https://cloud.google.com/resource-manager/docs/organization-policy/understanding-hierarchy), the organization policy set on the current resource hierarchy node takes effect. If `inheritFromParent` is set to `TRUE`, then inheritance merging takes effect.

Constraint attributes

Every constraint is defined by these attributes:

- Name: the unique name of the constraint.
  - For example, `constraints/compute.disableSerialPortAccess`
- Display name: the human-friendly name of the constraint.
- Description: details on what enforcements are put in place and by which Google Cloud services.
- Default behavior: behavior in the absence of user-defined configuration within the policy.

Types of constraints

List constraint

A list constraint (https://cloud.google.com/resource-manager/docs/organization-policy/using-constraints#list-constraint) allows or disallows a list of values that is defined in an organization policy. This list of values is expressed as a hierarchy subtree string. The subtree string specifies the type of resource it applies to. For example, a list of project IDs in the form of `projects/`***`PROJECT_ID`*** for `constraints/compute.trustedImageProjects`.

The following table describes some common constraint configurations for policy enforcement:

Policy	Constraint configuration
Allow a specific set of values	Set the allowed values field (`ListPolicy.allowed_values`) to a list of strings Set `ListPolicy.all_values` to `ALL_VALUES_UNSPECIFIED`
Deny a specific set of values	Set the denied values field (`ListPolicy.denied_values`) to a list of strings Set `ListPolicy.all_values` to `ALL_VALUES_UNSPECIFIED`
Deny a value and all of its child values	Set the denied values field (`ListPolicy.denied_values`) to a subtree string, such as `organizations/1234` Set `ListPolicy.all_values` to `ALL_VALUES_UNSPECIFIED`
Allow all valid values	Set `ListPolicy.all_values` to `ALLOW` Do not set `ListPolicy.allowed_values` or `ListPolicy.denied_values`
Deny all values	Set `ListPolicy.all_values` to `DENY` Do not set `ListPolicy.allowed_values` or `ListPolicy.denied_values`

Values can also be given a prefix in the form “prefix:value”, which then gives the value additional meaning:

- `is:` - Applies a comparison against the exact value. This is the same behavior as not having a prefix, and is required when the value includes a colon.
- `under:` - Applies a comparison to the value and all of its child values. If a resource is allowed or denied with this prefix, its child resources are also denied. The value provided must be a hierarchy subtree string, as in the following examples:
    - `organizations/`***`ORGANIZATION_ID`***

- `folders/FOLDER_ID`
- `projects/PROJECT_ID`

Some constraints aren't compatible with using hierarchy subtree strings as values. For information about the constraints that support using hierarchy subtree value prefixes, see Organization policy constraints (https://cloud.google.com/resource-manager/docs/organization-policy/org-policy-constraints).

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. For more information about using prefixed values in constraints, see Set up enforcement against a hierarchy subtree (https://cloud.google.com/resource-manager/docs/organization-policy/using-constraints#hierarchy_subtree)
.

If no list of values is provided, then the default that takes effect, depending on the specific constraint, may be:

- `ALLOW` - Any valid value is allowed.
- `DENY` - No value is allowed.

Boolean constraint

A boolean constraint (https://cloud.google.com/resource-manager/docs/organization-policy/using-constraints#boolean-constraint)
is either in enforcement or not. The policy is enforced by setting `Policy.enforced`to `True`.

For example, `constraints/compute.disableSerialPortAccess` will have two possible states:

- TRUE - the `disableSerialPortAccess` constraint is enforced, and serial port access is not allowed.
- FALSE - the `disableSerialPortAccess` constraint is not enforced or checked, so serial port access is allowed.

If no policy is set, or the policy is set to `RestoreDefault`, then serial port access is allowed because the constraint default is to allow.

Last updated December 4, 2019.