Restricting Resource Locations

Beta

This product or feature is in a pre-release state and might change or have limited support. For more information, see the product launch stages (https://cloud.google.com/products/#product-launch-stages).

Overview

This guide describes how to set an organization policy (https://cloud.google.com/resource-manager/docs/organization-policy/overview) that includes the resource locations constraint (https://cloud.google.com/resource-manager/docs/organization-policy/understanding-constraints).

You can limit the physical location of a new resource with the Organization Policy Service resource locations constraint. You can use the location property of a resource to identify where it is deployed and maintained by the service. For data-containing resources of some Google Cloud services, this property also reflects the location where data is stored. This constraint allows you to define the allowed Google Cloud locations where the resources for supported services in your hierarchy can be created.

After you define resource locations, this limitation will apply only to newly-created resources. Resources you created before setting the resource locations constraint will continue to exist and perform their function.

A policy that includes this constraint will not be enforced on sub-resource creation for certain services, such as Cloud Storage and Dataproc.

Warning: An organization policy that includes this constraint will prevent certain services from creating new resources that violate the constraint, which can stop those services from working correctly. For details and examples for each of the supported services, see the Resource Locations Supported Services

(https://cloud.google.com/resource-manager/docs/organization-policy/defining-locations-supported-services)
page.

Limitations

The resource locations Organization Policy Service constraint controls the ability to create regional resources. This will not affect where global resources are created. To avoid breaking existing serving infrastructure, you should test any new policy on non-production projects and folders, then apply the policy gradually within your organization.

Warning: The resource locations constraint controls only where resources are created. Some Google Cloud services might not store or process data contained by a resource in the same location where that resource was created.

For data storage commitments, see the Google Cloud Terms of Service (https://cloud.google.com/terms) and the Service Specific Terms (https://cloud.google.com/terms/service-terms). Organization policies that contain the resource resource locations constraint aren't data storage commitments.

This constraint applies to a specific subset of products and resource types. For a list of currently supported services and details on the behavior of each service, see the Resource Locations Supported Services (https://cloud.google.com/resource-manager/docs/organization-policy/defining-locations-supported-services)
page.

Location types

You can deploy Google Cloud resources in location (https://cloud.google.com/about/locations) types that represent different size categories. The largest location type is the `multi-region`, which includes more than one `region`. Each `region` is further subdivided into `zones`. For more information about regions and zones, see the Regions and Zones overview (https://cloud.google.com/compute/docs/regions-zones).

- `Multi-region` locations are backed by physical resources in more than one `region` and are typically only used by storage-based resources. Some examples include `us`, `asia`, `europe`, and `global`.
- `Region` locations are geographically isolated from each other. Some examples include `us-west1` (Oregon), `asia-northeast1` (Tokyo), and `europe-west1` (Belgium).
- `Zone` locations are the most granular and isolated location type used for deploying resources. A `zone` is an independent failure domain within a `region`. Some examples are `us-east1-a`, `us-west1-b`, and `asia-northeast1-a`.

Google recommends that you use Value Groups (#value_groups) instead of individual values while setting up locations. Different resource types support different types of locations, and new locations in your desired geographic location may be added. Using a Value Group curated by Google Cloud allows you to choose geographic location(s), without having to specify current or future Cloud locations.

Setting the organization policy

The resource locations constraint is a type of list constraint (https://cloud.google.com/resource-manager/docs/organization-policy/using-constraints#list-constraint). You can add and remove locations from the `allowed_values` or `denied_values` lists of a resource locations constraint. To prevent organization policies from unexpectedly restricting service behavior as new locations are added to the available list, use a value group (#value_groups), or a list of `allowed_values` that represents the entire geographic boundary you want to define.

To set an organization policy (https://cloud.google.com/resource-manager/reference/rest/v1/Policy) including a resource locations constraint:

CONSOLE GCLOUD API

1. Go to the Organization policies page (https://console.cloud.google.com/iam-admin/orgpolicies) in the Google Cloud Console.

   GO TO THE ORGANIZATION POLICIES PAGE (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/IAM-ADMIN

2. Click **Select**.
3. Select the organization you want to set the policy for.

4. Click **Google Cloud Platform - Define Resource Restriction**.
5. Click **Edit**.
6. Under **Applies to**, select **Customize**.
7. Under **Policy values**, select **Custom**.
8. In the **Policy value** box, enter the `in` prefix and a value group (#value_groups) location string, then press **Enter**. For example, `in:us-locations` or `in:us-west1-locations`. You can enter multiple location strings.
    a. You can also enter specific zone, region, or multi-region locations as location strings. For a list of available locations, see the Resource Locations Supported Services (https://cloud.google.com/resource-manager/docs/organization-policy/defining-locations-supported-services) page.
9. Click **Save**. A notification will appear to confirm the policy update.

To learn about using constraints in organization policies, see Using Constraints (https://cloud.google.com/resource-manager/docs/organization-policy/using-constraints#list-constraint).

Using inheritance in organization policy

You can refine your organization policy to inherit the organization policy from the resource's parent nodes. Inheritance (https://cloud.google.com/resource-manager/docs/organization-policy/understanding-hierarchy) gives you granular control over the organization policies used throughout your resource hierarchy.

To enable inheritance on a resource node, set `inheritFromParent = true` in the organization policy `.yaml` file. For example:

```
etag: BwVtXec438Y=
constraint: constraints/gcp.resourceLocations
listPolicy:
  deniedValues:
    "us-west1"
  inheritFromParent: true
```

Example error message

Services that support the resource location constraint are prevented from creating new resources in locations that would violate the constraint. If a service attempts to create a resource in a location that violates the constraint, the attempt will fail and an error message will be generated.

This error message will have this format: `LOCATION_IN_REQUEST violates constraint constraints/gcp.resourceLocations on the resource RESOURCE_TESTED.`

In the following example, a Compute Engine resource fails to create a new instance (https://cloud.google.com/compute/docs/instances) due to policy enforcement:

```
Location ZONE:us-east1-b violates constraint constraints/gcp.resourceLocations
on the resource
projects/policy-violation-test/zones/us-east1-b/instances/instance-3.
```

Stackdriver and Cloud Audit Logs log entry:

```
{
 insertId: "5u759gdngec"
 logName: "projects/policy-violation-test/logs/cloudaudit.googleapis.com%2Factivity"
 protoPayload: {
  @type: "type.googleapis.com/google.cloud.audit.AuditLog"
  authenticationInfo: {…}
  authorizationInfo: [6]
  methodName: "beta.compute.instances.insert"
  request: {…}
  requestMetadata: {…}
  resourceLocation: {…}
  resourceName: "projects/policy-violation-test/zones/us-east1-b/instances/instance-
  response: {
   @type: "type.googleapis.com/error"
   error: {
    code: 412
    errors: [
     0: {
      domain: "global"
      location: "If-Match"
      locationType: "header"
      message: "Location ZONE:us-east1-b violates constraint constraints/gcp.resourc
      reason: "conditionNotMet"
     }
    ]
    message: "Location ZONE:us-east1-b violates constraint constraints/gcp.resourceL
```

```
   }
  }
  serviceName: "compute.googleapis.com"
  status: {
   code: 3
   message: "INVALID_ARGUMENT"
  }
 }
 receiveTimestamp: "2019-06-14T03:04:23.660988360Z"
 resource: {
  labels: {…}
  type: "gce_instance"
 }
 severity: "ERROR"
 timestamp: "2019-06-14T03:04:22.783Z"
}
```

Value groups

Value groups are collections of groups and locations that are curated by Google to provide a simple way to define your resource locations. Value groups include many related locations and are expanded over time by Google without needing to change your organization policy to accommodate the new locations.

To use value groups in your organization policy, prefix your entries with the string `in:`. For more information on using value prefixes, see Using Constraints (https://cloud.google.com/resource-manager/docs/organization-policy/using-constraints#hierarchy_subtree). Group names are not validated on the call to set the organization policy. If the group name provided does not exist, no new values will be added to the effective organization policy, unless that group name is created by Google later.

The following table contains the current list of available groups:

Group	Details	Direct members

Group	Details	Direct members
Asia	All locations within Asia: `in:asia-locations`	*Groups:* • `asia-east1-locations` • `asia-east2-locations` • `asia-northeast1-locations` • `asia-northeast2-locations` • `asia-south1-locations` • `asia-southeast1-locations` *Values:* • `asia`
Taiwan	All locations within Taiwan: `in:asia-east1-locations`	*Values:* • `asia-east1` • `asia-east1-a` • `asia-east1-b` • `asia-east1-c`
Hong Kong	All locations within Hong Kong: `in:asia-east2-locations`	*Values:* • `asia-east2` • `asia-east2-a` • `asia-east2-b` • `asia-east2-c`
Tokyo	All locations within Tokyo: `in:asia-northeast1-locations`	*Values:* • `asia-northeast1` • `asia-northeast1-a` • `asia-northeast1-b` • `asia-northeast1-c`

Group	Details	Direct members
Osaka	All locations within Osaka: `in:asia-northeast2-locations`	*Values:* • `asia-northeast2` • `asia-northeast2-a` • `asia-northeast2-b` • `asia-northeast2-c`
Mumbai	All locations within Mumbai: `in:asia-south1-locations`	*Values:* • `asia-south1` • `asia-south1-a` • `asia-south1-b` • `asia-south1-c`
Singapore	All locations within Singapore: `in:asia-southeast1-locations`	*Values:* • `asia-southeast1` • `asia-southeast1-a` • `asia-southeast1-b` • `asia-southeast1-c`
Australia	All locations within Australia: `in:australia-locations`	*Groups:* • `australia-southeast1-locations`
Sydney	All locations within Sydney: `in:australia-southeast1-locations`	*Values:* • `australia-southeast1` • `australia-southeast1-a` • `australia-southeast1-b` • `australia-southeast1-c`

Group	Details	Direct members
Europe	All locations within Europe: `in:europe-locations`	*Groups:* • `europe-north1-locations` • `europe-west1-locations` • `europe-west2-locations` • `europe-west3-locations` • `europe-west4-locations` • `europe-west6-locations` *Values:* • `EU` • `eu` • `europe-west`
Finland	All locations within Finland: `in:europe-north1-locations`	*Values:* • `europe-north1` • `europe-north1-a` • `europe-north1-b` • `europe-north1-c`
Belgium	All locations within Belgium: `in:europe-west1-locations`	*Values:* • `europe-west1` • `europe-west1-b` • `europe-west1-c` • `europe-west1-d`
London	All locations within London: `in:europe-west2-locations`	*Values:* • `europe-west2` • `europe-west2-a` • `europe-west2-b` • `europe-west2-c`

Group	Details	Direct members
Frankfurt	All locations within Frankfurt: `in:europe-west3-locations`	*Values:* • `europe-west3` • `europe-west3-a` • `europe-west3-b` • `europe-west3-c`
Netherlands	All locations within Netherlands: `in:europe-west4-locations`	*Values:* • `europe-west4` • `europe-west4-a` • `europe-west4-b` • `europe-west4-c`
Zurich	All locations within Zurich: `in:europe-west6-locations`	*Values:* • `europe-west6` • `europe-west6-a` • `europe-west6-b` • `europe-west6-c`
North America	All locations within North America: `in:northamerica-locations`	*Groups:* • `northamerica-northeast1-locations` • `us-locations` *Values:* • `nam3`
Montréal	All locations within Montréal: `in:northamerica-northeast1-locations`	*Values:* • `northamerica-northeast1` • `northamerica-northeast1-a` • `northamerica-northeast1-b` • `northamerica-northeast1-c`

Group	Details	Direct members
United States	All locations within the United States: `in:us-locations`	*Groups:* • `us-central1-locations` • `us-central2-locations` • `us-east1-locations` • `us-east4-locations` • `us-west1-locations` • `us-west2-locations` *Values:* • `US` • `nam3` • `us` • `us-central`
Iowa	All locations within Iowa: `in:us-central1-locations`	*Values:* • `us-central1` • `us-central1-a` • `us-central1-b` • `us-central1-c` • `us-central1-f`
Oklahoma	All locations within Oklahoma: `in:us-central2-locations`	*Values:* • `us-central2` • `us-central2-a` • `us-central2-b` • `us-central2-c` • `us-central2-d`

Group	Details	Direct members
South Carolina	All zones within South Carolina: `in:us-east1-locations`	*Values:* • `us-east1` • `us-east1-a` • `us-east1-b` • `us-east1-c` • `us-east1-d`
Northern Virginia	All locations within Northern Virginia: `in:us-east4-locations`	*Values:* • `us-east4` • `us-east4-a` • `us-east4-b` • `us-east4-c`
Oregon	All locations within Oregon: `in:us-west1-locations`	*Values:* • `us-west1` • `us-west1-a` • `us-west1-b` • `us-west1-c`
Los Angeles	All locations within Los Angeles: `in:us-west2-locations`	*Values:* • `us-west2` • `us-west2-a` • `us-west2-b` • `us-west2-c`
South America	All locations within South America: `in:southamerica-locations`	*Groups:* • `southamerica-east1-locations`
São Paulo	All locations within São Paulo: `in:southamerica-east1-locations`	*Values:* • `southamerica-east1` • `southamerica-east1-a` • `southamerica-east1-b` • `southamerica-east1-c`

Authentication

Organization Policy Service uses OAuth 2.0 (http://tools.ietf.org/html/rfc6749) for API authentication and authorization. To get an OAuth 2.0 bearer token:

1. Go to the OAuth 2.0 Playground page (https://developers.google.com/oauthplayground/).
2. In the **Step 1** list of scopes, select the **Cloud Resource Manager API v2** > **https://www.googleapis.com/auth/cloud-platform**, and then click **Authorize APIs**.
3. On the **Sign in with Google** page that appears, select your account and sign in.
4. To provide access to **Google Oauth 2.0 Playground**, click **Allow** on the prompt that appears.
5. In **Step 2**, click **Exchange authorization code for tokens**.
6. At the bottom of the **Request / Response** pane on the right, your access token string is displayed:

```
{
  "access_token": "ACCESS_TOKEN",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

   Where ***ACCESS_TOKEN*** is the OAuth 2.0 bearer token string that you can use for API authorization.

Last updated November 19, 2019.