Listing all Resources in your Hierarchy

Resources in Google Cloud are organized into a hierarchy (https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy), with each node (Organizations, Folders, Projects, and so forth) having a reference to its parent. You can use that reference as a key filter term for scans to improve the consistency of resource searches.

You can grant users permissions using custom roles (https://cloud.google.com/iam/docs/creating-custom-roles). These roles operate on the principle of least privilege, and generally provide only the minimum necessary permissions required to do a particular task.

This scheme can be useful for isolating different user groups. For example:

- A large company with departments that shouldn't be able to inspect the resources of their peers.
- Contractors who are given permissions to a specific Project, but no other resources.

As a result of their restricted permissions, however, custom roles may cause many resources in your hierarchy to be omitted when executing a list operation. When performing searches as a user that has been granted a custom role, it can be difficult to tell why certain resources are not appearing.

To avoid this scenario, this page discusses the best practices for listing all of the resources managed by the Resource Manager API in your resource hierarchy. You can use this guidance to configure custom audit checks, or to create your own user experience on top of the Resource Manager API.

List all resources

When you scan your resource hierarchy to list every resource, you need strongly consistent results. If your scan misses resources or provides outdated results, it can be hard to tell that something has gone wrong. To make sure that you always get the most accurate and complete results, use a service account and perform a scan in the following way:

1. Grant a service account the `list` and `get` permissions for Organizations, Folders, and Projects on the Organization resource.
2. If you are listing Project and Folder resources, specify the parent resource in the filter string.
3. Run the `projects.list()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/list) method with this service account for each type of resource you want to find, and for any intermediate resources such as Folders.

List all resources example

The following example demonstrates how to list every resource node in your Organizations:

```
organizations = CloudResourceManager.Organizations.Search()
projects = emptyList()

parentsToList = queueOf(organizations)
while (parent = parentsToList.pop()) {
  // NOTE: Don't forget to iterate over paginated results.
  // TODO: handle PERMISSION_DENIED appropriately.
  projects.addAll(CloudResourceManager.Projects.List(
      "parent.type:" + parent.type + " parent.id:" + parent.id))
  parentsToList.addAll(CloudResourceManager.Folders.List(parent))
}
```

When building a custom user experience, you may also want to mix in search results and load the parent resources as needed (while also catching the `PERMISSION_DENIED` exception).

Search resources

If your scan is intended to search for a resource that was created some time ago, you can perform a faster scan that has eventual consistency rather than strong consistency. Note that this search method may omit some resources from the search result, particularly any resources that have been changed recently. To search for resources:

1. Use a service account that has the `get` permission for the resource you are searching for.

2. Run the `projects.list()` (https://cloud.google.com/resource-manager/reference/rest/v1/projects/list) method with this service account.

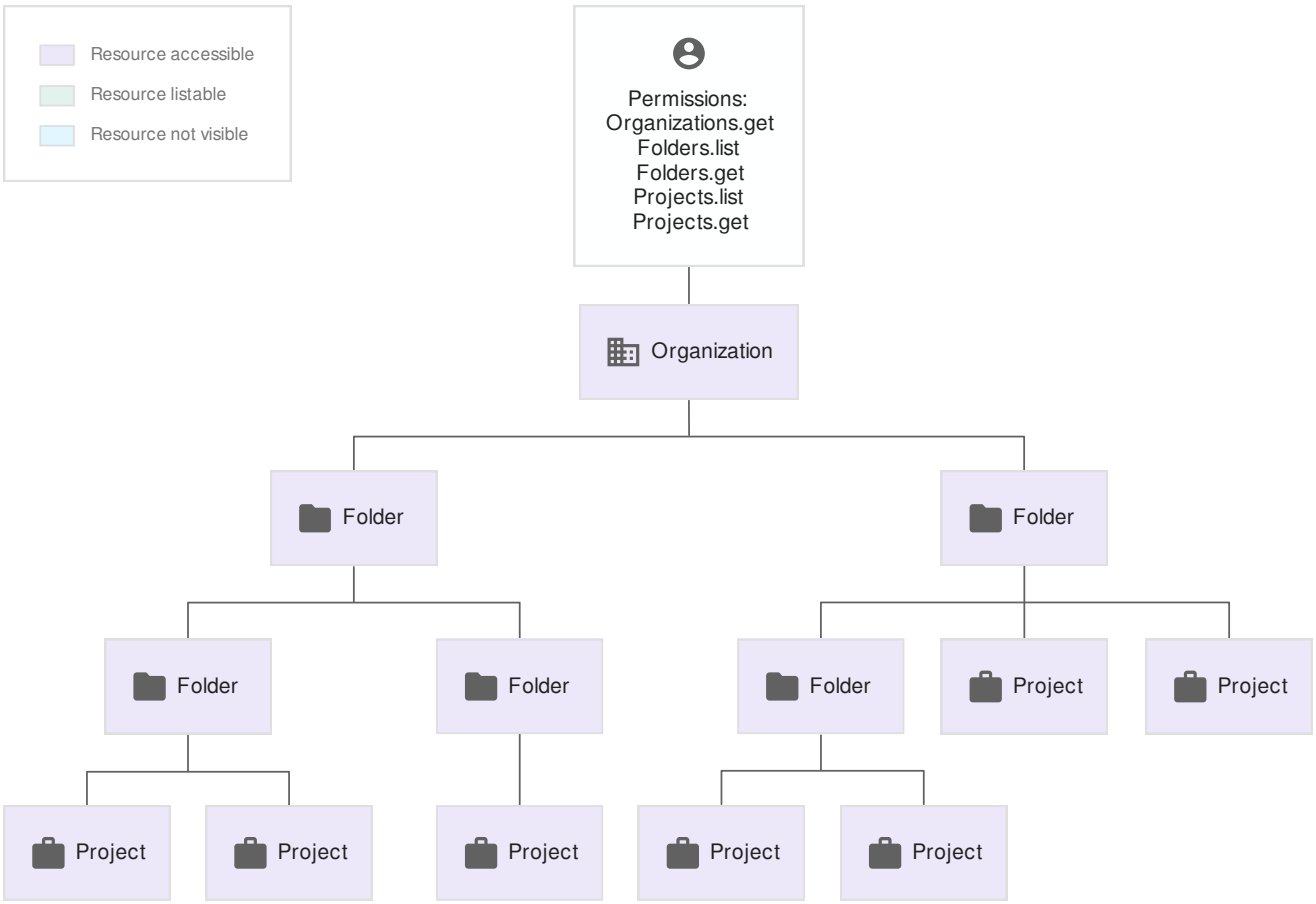
Troubleshooting omitted resources

If you are developing a scanning tool, we recommend that you use `list` and `get` permissions granted at the Organization level. This avoids issues caused by the user having partial permissions, which results in some resources being omitted from the list.

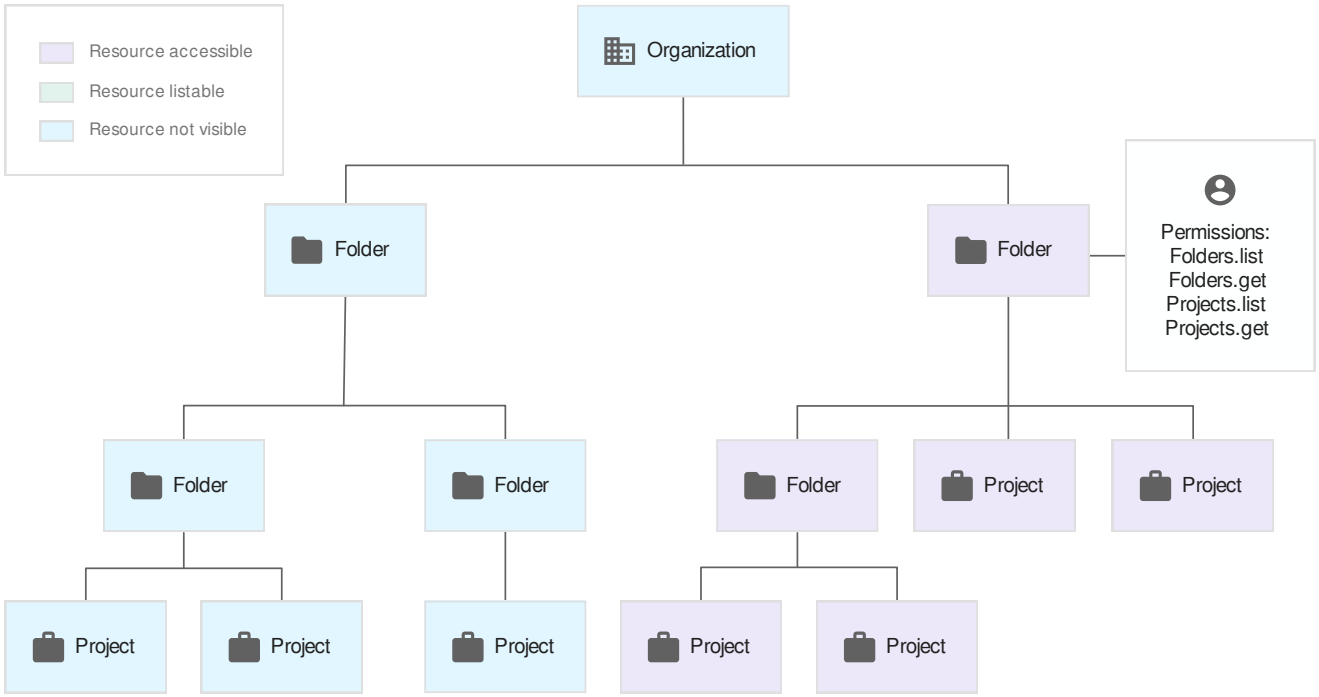
If you are designing a custom user experience that checks user permissions, there is no easy solution. If a user does not have Organization-level permissions, they will need certain permissions on every resource for it to appear. If a user is missing permissions on a resource somewhere in the hierarchy, some resources may not appear.

If a user has the `list` permission but not the `get` permission for a particular resource, that resource won't be visible at all in the Google Cloud Console. However, the resource will be returned in a search using the API or `gcloud` command-line tool that specifies the resource's parent. This disparity between the Google Cloud Console and other methods is a common source of confusion when trying to scan the resource hierarchy.

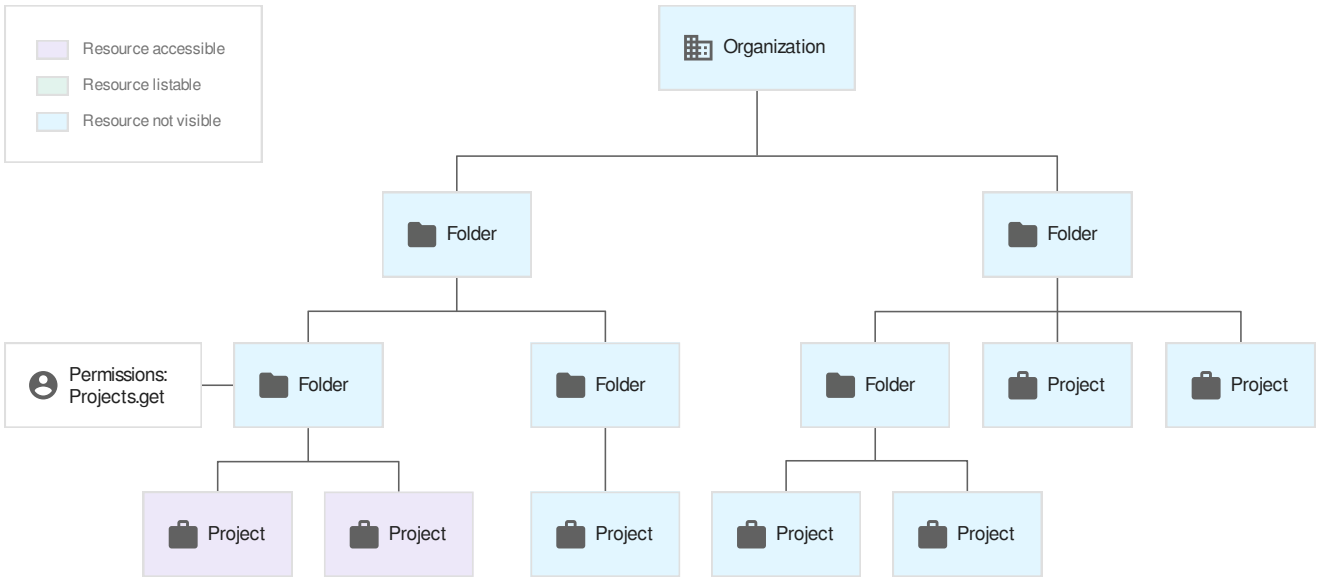
The following diagrams demonstrate some common configurations of permissions, and how they change what resources are visible to a user running a search.



In this example, all required permissions are granted in the Organization resource. Therefore, the entire hierarchy is visible when performing a list or search.

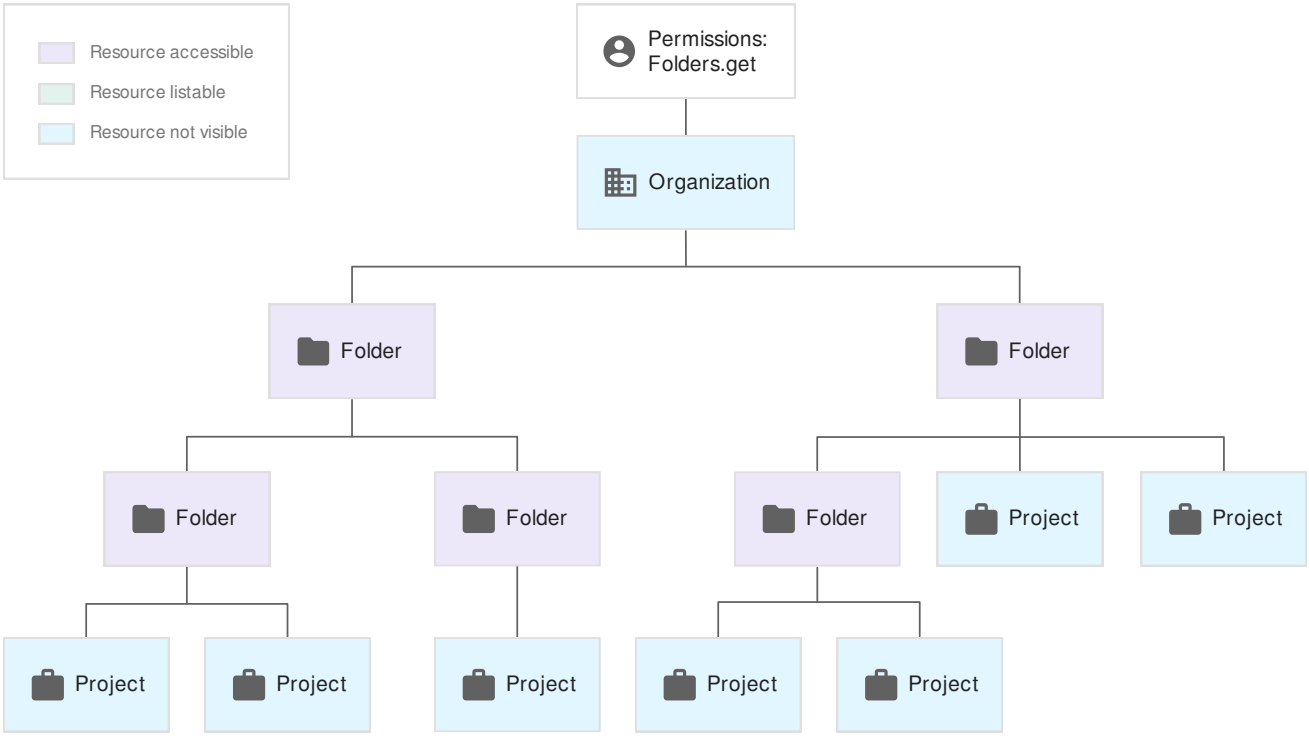


The user in this example has all required permissions except for `resourcemanager.organizations.get`, but they are granted those permissions at the Folder level. This permissions gap gives them full visibility on list or search of that part of the hierarchy, but not the other half.

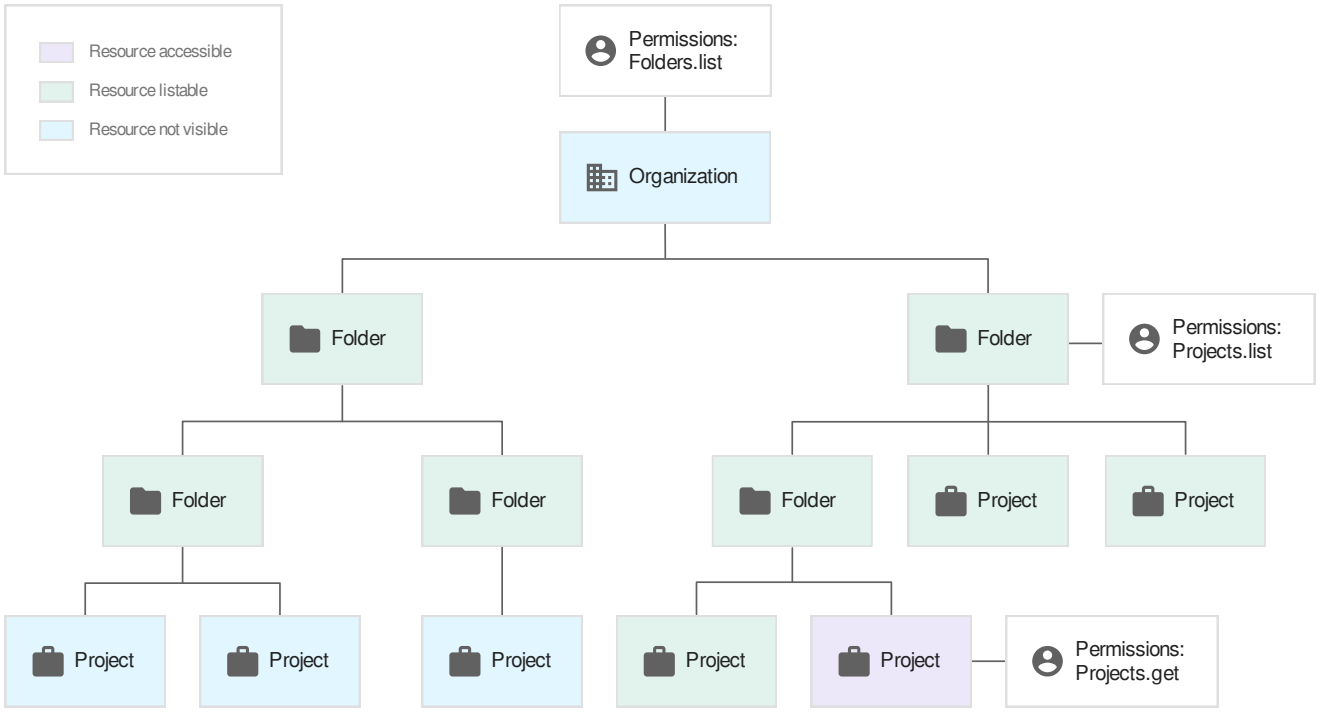


This example shows the experience of a user with only the `resourcemanager.projects.get` permission granted at the Folder resource level. They are able to see the Projects underneath

that Folder in the hierarchy, but only by searching. Using the list functionality will not return any results.

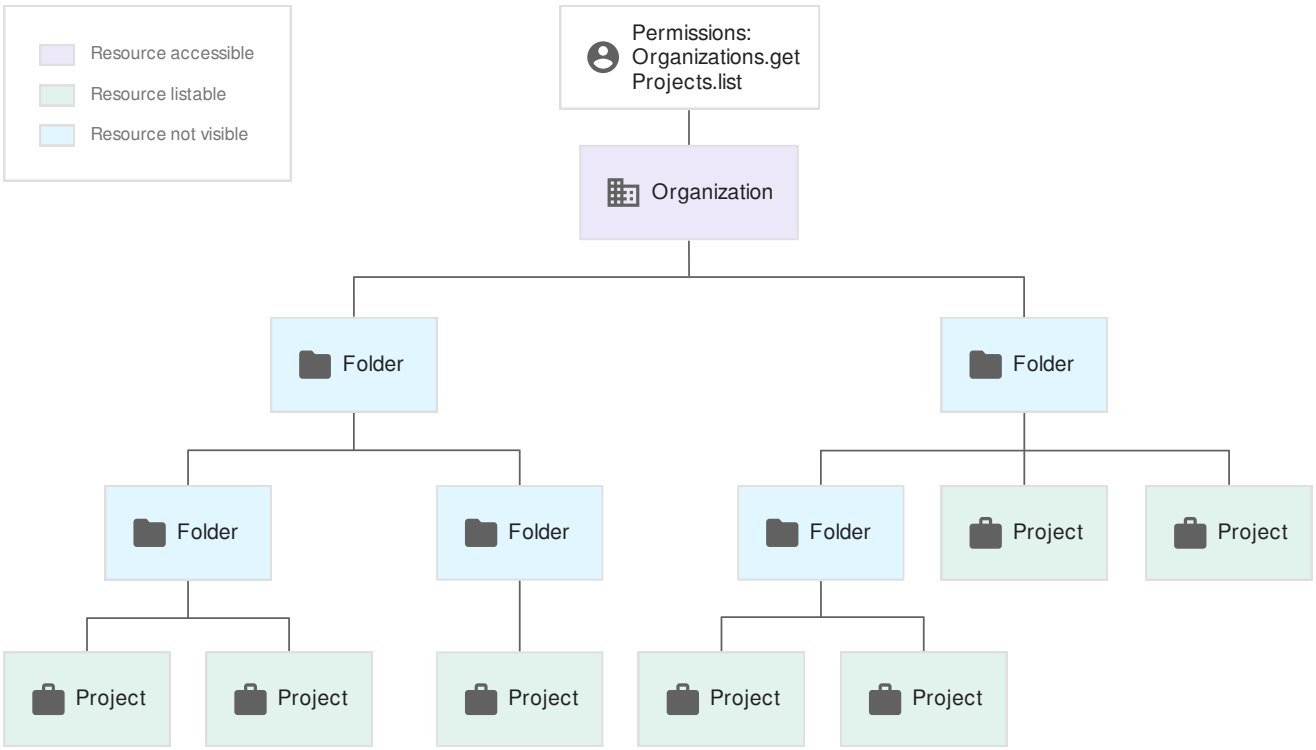


This example shows the same issue as above, where the granted permissions only allow a user to find their Folder resources by searching. Using the list functionality will not return any results.



The user in this example has a mix of permissions throughout their Organization. They can list folders from the Organization level, which allows them to find them with searches that specify the parent resource throughout the hierarchy. They can list Project resources for one Folder, but not the other, and they have `resourcemanager.projects.get` permission on one Project at the bottom of the hierarchy.

The result is that they aren't able to return the Projects on the left side of this resource hierarchy. They can list the Projects on the right side only by using a search that specifies the parent resource, and only one Project is visible when viewed in the Cloud Console.



In this example, the user can get the Organization resource and list Project resources by specifying the parent throughout the hierarchy. However, they do not have permission to list or search any of the intermediate Folders. Their Projects are searchable if the user happens to know the ID of its parent Folder. The Folders are not visible at all to this user, and so they will not be able to discover the ID if they don't already have it. The only resource that will appear in the Cloud Console is the Organization.

When designing your custom user experience, it's important to be aware of situations similar to the above. You can use a combination of listing and searching to render the resource hierarchy. You should also consider how to communicate to users that they are missing permissions that would allow them to see the whole resource hierarchy.

Last updated December 4, 2019.