Sharing across perimeters with bridges

This page describes how perimeter bridges can be used to allow projects and services in different service perimeters to communicate.

Before you begin

- Read Overview of VPC Service Controls (https://cloud.google.com/vpc-service-controls/docs/overview)
- Read Service Perimeter Configuration (https://cloud.google.com/vpc-service-controls/docs/service-perimeters#stages)

Service perimeter bridges

While a project can be assigned to only one service perimeter, you may want your project to be able to communicate with projects in another perimeter. You can enable communication to services and share data across service perimeters by creating a **perimeter bridge** (https://cloud.google.com/vpc-service-controls/docs/create-perimeter-bridges).

A perimeter bridge allows projects in different security perimeters to communicate. Perimeter bridges are bidirectional, allowing projects from each service perimeter equal access within the scope of the bridge. However, the access levels and service restrictions of the project are controlled solely by the service perimeter that the project belongs to. A project can have multiple bridges connecting it to other projects.

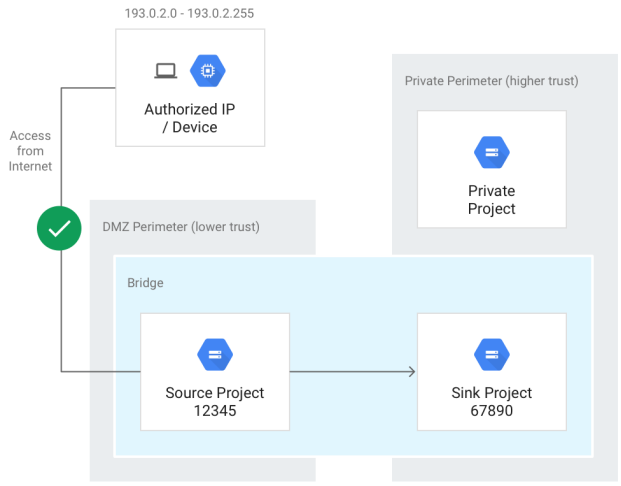
A project from one service perimeter cannot indirectly gain access to projects in other perimeters. For example, assume we have three projects: A, B, and C. Each project belongs to a different service perimeter. A and B share a perimeter bridge. B and C also share a bridge. While data can move between A and B, as well as between B and C, nothing can pass between A and C because the two projects are not directly connected by a perimeter bridge.

A project must belong to a service perimeter before it can be connected to another project using a perimeter bridge.

Perimeter bridges cannot include projects from different organizations. The projects connected by a perimeter bridge must belong to service perimeters that are in the same organization.

Example of perimeter bridges

For a broader example of how perimeter bridges work, consider the following setup:



The goal is to allow copies between the Cloud Storage buckets in the DMZ Perimeter and only the buckets in the Sink Project but not allow any VMs in the DMZ Perimeter access to data in Storage buckets in the Private Project.

Using the following command, a perimeter bridge (**Bridge**) is created, specifying that project A and project B are to be connected by the perimeter bridge.

Note: In the example command and the previous diagram, projects A and B are represented by their project numbers, 12345 and 67890, as the project numbers are required for the `resources` option.

```
gcloud access-context-manager perimeters create Bridge \
  --title="Perimeter Bridge" --perimeter-type=bridge \
  --resources=projects/12345,projects/67890
```

The perimeter bridge boundary is not directional. This means copies from DMZ Perimeter to Private Perimeter and from Private Perimeter to DMZ Perimeter are both allowed. To provide some directional control, it's best to combine perimeters with Cloud IAM permissions on the service account or identity that is executing the copy operation.

Last updated March 29, 2019.