Google Cloud

Configuring Serverless VPC Access for Cloud Run Functions

Course reading

This is a brief document that describes how you can configure Serverless VPC Access for your Cloud Run functions. By configuring Serverless VPC Access, you enable Cloud Run functions to connect to your internal resources in your VPC network.

Confidential + proprietary

What is Serverless VPC Access

Serverless VPC Access is a configuration that you create in your Google Cloud project to connect Cloud Run functions directly to your VPC network. This configuration enables access to Compute Engine VM instances, Memorystore, and other resources with an internal IP address that are in your VPC network.

With Serverless VPC Access, you can send requests and receive responses to and from your VPC network using internal DNS and internal IP addresses, so that traffic is not exposed to the internet.

A Virtual Private Cloud (VPC) network is a virtual version of a physical network, implemented inside Google's production network. It's a global resource that consists of a list of regional virtual subnetworks (subnets) in data centers, all connected by a global wide area network.

Configuring Serverless VPC Access

To configure Serverless VPC Access:

1. Enable the Serverless VPC Access API.
2. Create a Serverless VPC Access connector in your Google Cloud Project.
3. Attach the connector to a VPC network and region. The region that is configured for the connector must match the region where your Cloud Run functions are deployed.

A Serverless VPC Access connector is a resource that handles traffic between your serverless Cloud Run functions environment and your VPC network.

Configure the connector with an unused /28 subnet or non-overlapping /28 CIDR range. The subnet or CIDR range must be used exclusively by the connector and no other resources.

After the connector is created, you use the connector by deploying each function specifying the connector name and optional flags.

```
$ gcloud functions deploy FUNCTION_NAME --vpc-connector
CONNECTOR_NAME FLAGS...
```

Restricting connector access

You can restrict your connector's access to resources in your VPC network by using firewall rules.

Using ingress rules

When connecting to a standalone VPC network or a Shared VPC network that has the connector in the host project, an implicit firewall rule with priority 1000 is automatically created on your VPC network to allow ingress from the connector's subnet or CIDR range to all resources in the VPC network.

Connector subnet/CIDR range

To override the implicit firewall rule that Serverless VPC Access creates on your VPC network by default, create an ingress firewall rule with priority lower than 1000 to deny ingress from the connector network tag or CIDR range.

Source connector tag/CIDR range

You can then restrict connector access by creating ingress rules on the destination resource.

To allow connector traffic to the resource that should receive connector traffic, create another ingress firewall rule with a lower priority than that of the previous rule, targeting the resource in your VPC network that you want the VPC connector to access.

Source connector tag/CIDR range *Target resource tag*

Using egress rules

You can also restrict connector access by creating egress rules on the connector in the VPC.

Create an egress firewall rule on your connector to prevent it from sending outgoing traffic.

Target connector tag

Then, to allow egress traffic from your connector to a specific destination CIDR range, create another firewall rule with a lower priority than that of the previous rule.

Set the destination range to the CIDR range of the resource in your VPC network that you want your connector to be able to access.

Target connector tag *Destination CIDR range*

Connecting to a Shared VPC network

You can connect Cloud Run functions directly to a Shared VPC network by using Serverless VPC Access. This allows Cloud Run functions to access resources in the Shared VPC network, such as Compute Engine VM instances, and any other resources with an internal IP address.

For a Shared VPC, Serverless VPC Access connectors can be configured in two different ways:

- Create a connector in each service project with Cloud Run functions that need access to the resources in your network.
- Create a shared connector in the host project.

To configure Serverless VPC Access for a Shared VPC:

1. Enable the Serverless VPC Access API.
2. Create a Serverless VPC Access connector in each service project or in the host project.
3. Configure the connector and subnet.
4. Perform additional configuration depending on the connector being in the host or service projects.
5. Configure your Cloud Run functions to use the connector to send traffic to the VPC network.

Configuring a connector in a host project

To configure a connector in a host project:

1. Enable the Cloud Functions API for the service project. This is required to add IAM roles and for the service project to use Cloud Run functions.

2. To provide access to the connector, grant the service project's Cloud Functions Service Agent the *Serverless VPC Access User* IAM role on the host project.

3. To make the connector discoverable, grant the *Serverless VPC Access Viewer* IAM role on the host project, and the *Compute Network Viewer* role on the service project.

Configuring a connector in a service project

To configure a connector in a service project:

1. Create firewall rules to allow requests from NAT and health check IP ranges to reach the connector and to be reached by the connector. These ranges are used by the Google infrastructure underlying Cloud Run functions.

2. Optionally limit the scope of the firewall rules to target specific resources in the network.

3. Grant the *Compute Network User* role in the host project to the service project's *cloudservices* and *vpcaccess* service accounts. This is required for each service project that uses the VPC connector.

Connector in service project versus host project

There are advantages in setting up connectors in each service project with Cloud Run functions that need access to resources in the Shared VPC network. There are also other advantages in

setting up connectors in the host project.

Connector in service project	Connector in host project
Isolation Each connector has dedicated bandwidth and is unaffected by bandwidth use of connectors in other service projects.	Centralized network management Allows centralized management of network resources in the host project.
Chargebacks Charges incurred by connectors are associated with the service project containing the connector.	IP address space Preserves IP address space as each connector instance requires an IP address.
Security Limit the access scope of the services in the project with firewall rules. (principle of least privilege)	Maintenance Reduces maintenance because each connector may be used by multiple services across service projects.
Team independence Separate teams can create and manage the connectors in each service project.	Idle time and cost Fewer connectors may reduce connector idle time and the cost when not serving traffic.

Related documentation

For more information, view the documentation at the links provided:

Configuring networking for Cloud Run functions

Serverless VPC Access