postgres_fdw

Last updated by | Lisa Liu | Nov 6, 2020 at 10:34 AM PST

Postgres_fdw

NOTE

1. FDW currently is not supported across regions. Meaning you can connect to a target server which is in the same region and served from the same control ring.
2. When we setup the FDW for a target server in same region, we have to whitelist IP of a TR of a source server. Note that every restart\failover might not change the IP as the server would potentially move to other nodes but remain in the same TR. But there are high changes that server could move to a different TR during update SLO operations. If you don't want to do this its recommended to use allow all azure services.
3. There is a way you can assign a static private Ip to a postgres server using private links feature but currently connecting to target server with a private link from a source azure postgres server using FDW is currently not supported

dblink and postgres_fdw allow you to connect from one PostgreSQL server to another, or to another database in the same server. The receiving server needs to allow connections from the sending server through its firewall and all the servers must be on the same region. When using these extensions to connect between Azure Database for PostgreSQL servers, this can be done by setting "Allow access to Azure services" to ON. This is also needed if you want to use the extensions to loop back to the same server. The "Allow access to Azure services" setting can be found in the Azure portal page for the Postgres server, under Connection Security. Turning "Allow access to Azure services" ON puts all Azure IPs on the allow list.

Currently, outbound connections from Azure Database for PostgreSQL are not supported, except for connections to other Azure Database for PostgreSQL servers.

Source: https://docs.microsoft.com/en-us/azure/postgresql/concepts-extensions

FDW on same region:

TARGET: that has data.(POSTGRESSERVER3)

SOURCE: that will pull the data.(POSTGRESSERVER2)

RUN the following commands on SOURCE database server.

CREATE postgres_fdw extension on both source and target using below command.

```
create extension postgres_fdw;
```

CREATE server that connects to the TARGET:

```
create server fdwwestusservers foreign data wrapper postgres_fdw options(host 'postgresserver3.database.windows.net', port '5432', dbname 'postgres');
```

USERMAPPING:

create user mapping for current_user server fdwwestusservers options (user'rajaniserver2@postgresserver2', password 'iwillnottell');-done

CREATE SCHEMA: create schema testuser;-done

IMPORTDATA:

> import foreign schema public from server fdwwestusservers into testuser;

Query the data: select * from testuser.testtable;--done

<IMG width=1108 height=647

How good have you found this content?