

**POSTGRESQL GCS SETUP GUIDE**

**The Below requirements are needed to establish a connection postgresql gcs .**

**Prerequisites**

Prerequisiteslink  
To connect your Google Cloud SQL for PostgreSQL database to Fivetran, you need:  
  
PostgreSQL version 9.6 - 13  
Your database's port (usually 5432)  
TLS enabled on your database. Follow Google Cloud's TLS setup instructions to enable TLS on your database.

**Setup Guide**

1. Choose connection method (TLS required)

* IMPORTANT: You must have TLS enabled on your database to connect to Fivetran.
* Google Cloud SQL products require client certificates to connect to databases that are configured to require SSL. However, Fivetran only supports server certificate authentication.
* To connect Fivetran to your Google Cloud SQL for PostgreSQL database, do one of the following:

1. Allow unsecured connections

* Allow unsecured connections to your database.
* Go to your Google Cloud Platform SQL dashboard.
* Click on the name of the database that you want to connect to Fivetran.
* NOTE: If you plan to create a read replica to connect to Fivetran, then edit the master database you plan to replicate. The master database's settings will be propagated to the replica.
* In the left menu, go to the Connections tab.
* Scroll down to the SSL section.
* If you see Only secured connections are allowed to connect to this instance, click Allow unsecured connections.
* If you see Unsecured connections are allowed to connect to this instance, you do not need to change anything.

1. Create Google Cloud Virtual Machine

* Create a Google Cloud Virtual Machine to act as a proxy to connect. Use one of the following options:
* Google Cloud SQL Proxy
* Certificate forwarding using stunnel
* SSH tunneling with client certificates configured

1. Create read replica (optional)

* If you'd like, create a read replica for Fivetran's exclusive use. Using a read replica allows Fivetran to integrate your data without putting unnecessary load on or interrupting the queries running on your master server. We recommend that you connect a read replica to Fivetran, but it's not required.
* If you already have a read replica or want to connect Fivetran to your master database, skip ahead to Step 3.
* Go to your Google Cloud Platform SQL dashboard.
* Click on the name of your master database.
* In the left menu, go to the Replicas tab.
* Click Create Read Replica.
* On the Create read replica page, enter an Instance ID for the replica.
* Click Done.
* Click Create. It will take a few minutes for the read replica to be created.

1. Enable database access

* Grant Fivetran's data processing servers access to your database.
* In your Google Cloud Platform SQL dashboard, click on your master database or read replica.
* In the left menu, go to the Connections tab.
* In the Connectivity section, click Add network.
* In the New network window, create a network for Fivetran. What you enter in the Network field depends on whether you're connecting directly or using an SSH tunnel.
* If you're connecting directly, enter Fivetran's IPs for your database's region.
* If you're connecting using an SSH tunnel, enter {your-ssh-tunnel-server-ip-address}/32.
* Click Save. It will take a few minutes for the database to be updated with the new settings.
* In the left menu, go to the Overview tab.
* In the Connect to this instance section, find your database's public IP address and make a note of it. You will need it to configure Fivetran.

1. Create user

* Create a database user for Fivetran's exclusive use.
* Open a connection to your PostgreSQL master database in a PostgreSQL console (such as a SQL workbench or psql).
* Create a user for Fivetran by executing the following SQL command. Replace <username> and some-password with a username and password of your choice.
* CREATE USER <username> PASSWORD 'some-password';
* content\_copy

1. Grant read-only access

* Grant the Fivetran user read-only access to all tables by running the following commands. To grant access to a schema other than PostgreSQL's default public schema, replace public with the schema name.
* GRANT USAGE ON SCHEMA "public" TO <username>;
* GRANT SELECT ON ALL TABLES IN SCHEMA "public" TO <username>;
* ALTER DEFAULT PRIVILEGES IN SCHEMA "public" GRANT SELECT ON TABLES TO <username>;
* content\_copy
* NOTE: The last command makes sure that any future tables will be accessible to Fivetran.
* If you want to grant access to multiple schemas, you must run these three commands for each schema.

1. Restrict access to tables (optional)

* If you want to limit Fivetran's access to your tables, grant the Fivetran user access to only the tables that you would like to sync. You must individually grant access to each table that you want to sync. You cannot grant access to all tables and then revoke access for a subset of tables.
* Ensure that the Fivetran user has access to the schema that contains your table(s).
* GRANT USAGE ON SCHEMA "some\_schema" TO <username>;
* content\_copy
* Revoke any previously granted permission to all tables in that schema.
* ALTER DEFAULT PRIVILEGES IN SCHEMA "some\_schema" REVOKE SELECT ON TABLES FROM <username>;
* REVOKE SELECT ON ALL TABLES IN SCHEMA "some\_schema" FROM <username>;
* content\_copy
* Repeat the following command for each table that you want Fivetran to sync.
* GRANT SELECT ON "some\_schema"."some\_table" TO <username>;
* content\_copy
* By default, any tables that you create in the future will not be accessible to the Fivetran user. To grant access to new tables, run the following command.
* ALTER DEFAULT PRIVILEGES IN SCHEMA "some\_schema" GRANT SELECT ON TABLES TO <username>;
* content\_copy

1. Restrict access to columns (optional)

* If you want to limit Fivetran's access to the columns in your tables, grant the Fivetran user access to only certain columns. You must individually grant access for each column that you want to sync.
* Revoke any previously granted permission to read all columns in that table.
* REVOKE SELECT ON "some\_schema"."some\_table" FROM <username>;
* content\_copy
* Grant permission to the specific columns you want to sync (for example, some\_column and other\_column).
* IMPORTANT: You must grant us access to the hidden system column xmin. We use the xmin column for incremental updates.
* GRANT SELECT (xmin, "some\_column", "other\_column") ON "some\_schema"."some\_table" TO <username>;
* content\_copy
* Once you restrict access to columns within a table, the Fivetran user will not have access to any new columns added to that table in the future. To grant access to new columns, you must rerun the command above.

1. Choose incremental sync mechanism

* To keep your data up to date after the initial sync, we use one of the following incremental sync methods:
* logical replication with the test\_decoding plugin Beta
* logical replication with the pgoutput plugin Beta
* XMIN
* Each of these methods keeps a record of recent data changes, which allows Fivetran to update only the data that has changed since our last sync. We support the XMIN method on Google Cloud PostgreSQL versions 9.6 or later and logical replication with the test\_decoding or pgoutput plugin for versions 12 or later.
* TIP: We recommend using logical replication with the test\_decoding plugin as your incremental update mechanism. Learn more in our Updating data documentation.
* Choose logical replication with the test\_decoding plugin, logical replication with the pgoutput plugin, or XMIN as your incremental update mechanism.
* To enable logical replication with the test\_decoding plugin, proceed to the test\_decoding section.
* To enable logical replication with the pgoutput plugin, proceed to the pgoutput section.
* To enable XMIN, proceed to the XMIN section.

1. Logical replication with the test\_decoding plugin Beta

* IMPORTANT: You can only enable logical replication with the test\_decoding plugin if your Google Cloud PostgreSQL version is 9.6 or later.
* To enable logical replication with the test\_decoding plugin, follow these steps:
* Connect to your master database. You cannot enable logical replication on a read replica.
* Ensure that your server has ample free space for the logs. Logs that Fivetran has already processed are released. However, logs are not released if replication stops (for example, if we lose access). In this case, logs may accumulate on your server and consume additional storage. The amount of additional disk space consumed by these logs is proportional to the amount of changes committed on the server. If a lost connection can't be resumed quickly enough, you can drop the replication slot, which releases the storage of unconsumed logs. You would then need to do a full re-sync of your connector to reset the cursor in the replication slot.
* In your Google Cloud platform SQL dashboard, set the cloudsql.logical\_decoding flag to on.
* Click Done to save the change.
* Go into your Cloud PostgreSQL database as a cloudsqlsuperuser (postgres by default in Google Cloud SQL).
* IMPORTANT: If your superuser doesn't have permission to create a replication slot, run the alter user postgres with replication; command before you do the following steps.
* Set the statement\_timeout parameter in your database to either 0 to disable the timeout or greater than 5 minute.
* Set the wal\_sender\_timeout parameter in your database to 0 to disable the timeout mechanism.
* NOTE: Disabling the timeout mechanism ensures that the database won't end the wal\_sender process before Fivetran is able to establish a connection. This helps us stay up to date with the latest changes in the database’s WAL slot.
* Create a logical replication slot for the database you want to sync by running the following command. You must use the output plugin test\_decoding supplied in the postgresql-contrib subpackage.
* IMPORTANT: The replication slot name fivetran\_replication\_slot quoted throughout this guide is used purely as an example. The actual replication slot name should be unique for every connector using the same PostgreSQL cluster. Replication slot names cannot start with a number.
* SELECT pg\_create\_logical\_replication\_slot('fivetran\_replication\_slot', 'test\_decoding');
* content\_copy
* Grant permission to the Fivetran user for reading the replication slot.
* ALTER ROLE fivetran WITH REPLICATION;
* content\_copy
* Log in as the Fivetran user.
* Verify that the Fivtran user can read the replication slot by running the following command.
* SELECT count(\*) FROM pg\_logical\_slot\_peek\_changes('fivetran\_replication\_slot', null, null);
* content\_copy
* If the query succeeds, then permissions are sufficient.

1. Logical replication with the pgoutput plugin Beta

* IMPORTANT: You can only enable logical replication with the pgoutput plugin if your Google Cloud PostgreSQL version is 10 or later.
* To enable logical replication with the pgoutput plugin, follow these steps:
* Connect to your master database. You cannot enable logical replication on a read replica.
* Ensure that your server has ample free space for the logs. Logs that Fivetran has already processed are released. However, logs are not released if replication stops (for example, if we lose access). In this case, logs may accumulate on your server and consume additional storage. The amount of additional disk space consumed by these logs is proportional to the amount of changes committed on the server. If a lost connection can't be resumed quickly enough, you can drop the replication slot, which releases the storage of unconsumed logs. You would then need to do a full re-sync of your connector to reset the cursor in the replication slot.
* In your Google Cloud platform SQL dashboard, set the cloudsql.logical\_decoding flag to on.
* Click Done to save the change.
* Go into your Cloud PostgreSQL database as a cloudsqlsuperuser (postgres by default in Google Cloud SQL).
* IMPORTANT: If your superuser doesn't have permission to create a replication slot, run the alter user postgres with replication; command before you do the following steps.
* Ensure that the statement\_timeout setting on your server is either 0 (the default value to disable the timeout) or greater than 5 minute.
* Set the wal\_sender\_timeout parameter in your database to 0 to disable the timeout mechanism.
* NOTE: Disabling the timeout mechanism ensures that the database won't end the wal\_sender process before Fivetran is able to establish a connection. This helps us stay up to date with the latest changes in the database’s WAL slot.
* Create a logical replication slot for the database you want to sync by running the following command. You must use the standard output plugin pgoutput.
* IMPORTANT: The replication slot name fivetran\_pgoutput\_slot quoted throughout this guide is used purely as an example. The actual replication slot name should be unique for every connector using the same PostgreSQL cluster. Replication slot names cannot start with a number.
* SELECT pg\_create\_logical\_replication\_slot('fivetran\_pgoutput\_slot', 'pgoutput');
* content\_copy
* Create a publication for your tables. If you want, you can create a publication for only certain tables so that you add or remove tables from the publication later on. Only changes from tables in the publication are replicated to Fivetran. Each database can have multiple distinct publications. You must have CREATE privileges or above to run this command.
* IMPORTANT: The publication name fivetran\_pub quoted throughout this guide is used purely as an example. The actual publication name should be unique for every database and cannot start with a number.
* CREATE PUBLICATION fivetran\_pub FOR TABLE table2, table4, table8;
* content\_copy
* To add or remove a table from a publication, run the following command. You must have the ownership rights over the table(s).
* ALTER PUBLICATION fivetran\_pub ADD/DROP TABLE table\_name;
* content\_copy
* Alternatively, you can create a publication for all of your tables. However, you cannot remove any table from this publication later on. You must have cloudsqlsuperuser privileges to run this command.
* CREATE PUBLICATION fivetran\_pub FOR ALL TABLES;
* content\_copy
* (Optional) You can choose which operations to include in the publication. For example, the following publication includes only INSERT and UPDATE operations.
* CREATE PUBLICATION insert\_only\_pub FOR TABLE table1 WITH (publish = 'INSERT, UPDATE');
* content\_copy
* Verify that your chosen tables are in the publication.
* SELECT \* FROM pg\_publication\_tables.
* content\_copy
* Grant the Fivetran user permission to read the replication slot.
* ALTER ROLE fivetran WITH REPLICATION;
* content\_copy
* Log in as the Fivetran user.
* Verify that the Fivetran user can read the replication slot by running the following command. Replace fivetran\_pgoutput\_slot with your replication slot name and fivetran\_pub with the publication name.
* SELECT count(\*) FROM pg\_logical\_slot\_peek\_binary\_changes('fivetran\_pgoutput\_slot', null, null, 'proto\_version', '1', 'publication\_names', 'fivetran\_pub');
* content\_copy
* If the query succeeds, then permissions are sufficient.

1. XMIN

* You do not need to do any additional configuration for the XMIN method.

1. Finish Fivetran configuration

* In your connector setup form, enter a destination schema prefix. This prefix applies to each replicated schema and cannot be changed once your connector is created.
* In the Host field, enter your database's IP that you found in Step 3 (for example, 1.2.3.4). Alternately, you can also enter your database host's domain (for example, your.server.com).
* Enter your database instance's port number. The port will be 5432, unless you changed the default.
* Enter the Fivetran-specific user that you created in Step 4.
* Enter the password for the Fivetran-specific user that you created in Step 4.
* Enter the name of your database (for example, your\_database).
* Choose your connection method. If you selected Connect via an SSH tunnel, provide the following information:
* SSH hostname (do not use a load balancer's IP address/hostname)
* SSH port
* SSH user
* Click Save & Test. Fivetran tests and validates our connection to your Google Cloud SQL for PostgreSQL database. Upon successful completion of the setup tests, you can sync your data using Fivetran.

1. Setup tests

* Fivetran performs the following tests to ensure that we can connect to your Google Cloud PostgreSQL database and that it is properly configured:
* The Connecting to SSH Tunnel Test validates the SSH tunnel details you provided in the setup form. It then checks that we can connect to your database using the SSH Tunnel. (We skip this test if you aren't connecting using SSH.)
* The Connecting to Host Test validates the database credentials you provided in the setup form. The test verifies that the host is not private and then checks the connectivity to the host.
* The Validating Certificate Test generates a pop-up window where you must choose which certificate you want Fivetran to use. It then validates that certificate and checks that we can connect to your database using TLS. (We skip this test if you aren't connecting directly.)
* The Connecting to Database Test checks that we can access your database.
* The Connecting to WAL Replication Slot Test confirms that the database associated with the replication slot matches the name you supplied in the setup form. It then verifies that the replication slot uses the pgoutput if you selected WAL with pgoutput update method, or the test\_decoding plugin if you selected WAL with test\_decoding update method. Lastly, it makes sure that the Fivetran user has replication privileges. (We skip this test if you selected XMIN as your incremental update mechanism.)
* The Publication Test verifies that the supplied publication name exists in your database. (We skip this test if you selected XMIN or WAL with test\_decoding plugin as your incremental update mechanism.)
* The Checking wal\_sender\_timeout Value Test checks that your database's wal\_sender\_timeout value is 0. (We skip this test if you selected XMIN as your incremental update mechanism.)
* The Checking statement\_timeout Value Test checks that we can access your database's pg\_settings table. It then verifies that the statement\_timeout value is either 0 or greater than 5 minutes.
* NOTE: The tests may take a few minutes to finish running.

