## How to set up logical replication with Postgresql

<https://www.digitalocean.com/community/tutorials/how-to-set-up-logical-replication-with-postgresql-10-on-ubuntu-18-04>

The process of keeping database copies in sync is called replication

PostgreSQL supports for logical replication, in addition to physical replication. In a **logical replication** scheme, high-level write operations are streamed from a master database server into one or more replica database servers. In a **physical replication** scheme, binary write operations are instead streamed from master to replica, producing a byte-for-byte exact copy of the original content.

In this tutorial, you will configure logical replication with PostgreSQL 10 on two Ubuntu 18.04 servers, with one server acting as the master and the other as the replica.

## **Prerequisites**

To follow this tutorial, you will need:

* Two Ubuntu 18.04 servers, which we’ll name **db-master** and **db-replica**, each set up with a regular user account and sudo privileges. To set these up, follow [this initial server setup tutorial](https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-16-04).
* [Private networking enabled](https://www.digitalocean.com/docs/networking/private-networking/quickstart/) on your servers. Private networking allows for communication between your servers without the security risks associated with exposing databases to the public internet.
* PostgreSQL 10 installed on both servers, following Step 1 of [How To Install and Use PostgreSQL on Ubuntu 18.04](https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-ubuntu-18-04).

## **Step 1 — Configuring PostgreSQL for Logical Replication**

There are several configuration settings you will need to modify to enable logical replication between your servers. First, you’ll configure Postgres to listen on the private network interface instead of the public one, as exposing data over the public network is a security risk. Then you’ll configure the appropriate settings to allow replication to **db-replica**.

On **db-master**, open /etc/postgresql/12/main/postgresql.conf, the main server configuration file:

$ sudo vi /etc/postgresql/12/main/postgresql

Change:

...

#listen\_addresses = 'localhost' # what IP address(es) to listen on;

...

...

listen\_addresses = 'localhost, db\_master\_private\_ip\_address'

...

This makes **db-master** listen for incoming connections on the private network in addition to the loopback interface.

Next, find the following line:

/etc/postgresql/10/main/postgresql.conf

...

#wal\_level = replica # minimal, replica, or logical

...

Uncomment it, and change it to set the PostgreSQL [*Write Ahead Log*](https://www.postgresql.org/docs/current/static/wal-intro.html) (WAL) level to logical. This increases the volume of entries in the log, adding the necessary information for extracting discrepancies or changes to particular data sets:

/etc/postgresql/10/main/postgresql.conf

...

wal\_level = logical

...

The entries on this log will be consumed by the replica server, allowing for the replication of the high-level write operations from the master.

Save the file and close it.

Next, let’s set our firewall rules to allow traffic from **db-replica** to port 5432 on **db-master**:

* sudo ufw allow from db\_replica\_private\_ip\_address to any port 5432

Finally, restart the PostgreSQL server for the changes to take effect:

* sudo systemctl restart postgresql

With your configuration set to allow logical replication, you can now move on to creating a database, user role, and table.

## **Step 2 — Setting Up a Database, User Role, and Table**

To test the functionality of your replication settings, let’s create a database, table, and user role. You will create an example database with a sample table, which you can then use to test logical replication between your servers. You will also create a dedicated user and assign them privileges over both the database and the table.

First, open the [psql prompt](https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-ubuntu-18-04#using-postgresql-roles-and-databases) as the **postgres** user with the following command on both **db-master** and **db-replica**:

* sudo -u postgres psql

* sudo -u postgres psql

Create a new table called widgets with arbitrary fields on both hosts:

* CREATE TABLE widgets
* (
* id SERIAL,
* name TEXT,
* price DECIMAL,
* CONSTRAINT widgets\_pkey PRIMARY KEY (id)
* );

Copy

* CREATE TABLE widgets
* (
* id SERIAL,
* name TEXT,
* price DECIMAL,
* CONSTRAINT widgets\_pkey PRIMARY KEY (id)
* );