

All things PostgreSQL from MinervaDB

 Follow



Photo by Deon De Villiers on Unsplash

Easy Steps to Configure Hot Standby with Logical Replication in PostgreSQL 16



Shiv Iyer

Jul 7, 2024 •  3 min read



1



Setting up a hot standby in PostgreSQL 16 using logical replication involves several steps, including configuring the primary and standby servers, setting up the replication slots, and ensuring the replication is working correctly. Here is a step-by-step guide to configuring a hot standby using logical replication in PostgreSQL 16.

Step 1: Configure the Primary Server

1. **Install PostgreSQL 16:** Ensure PostgreSQL 16 is installed on your primary server. You can download and install it from the official PostgreSQL website or your system's package manager.
2. **Modify the Configuration Files:** Edit the `postgresql.conf` file to enable logical replication and set the necessary parameters.

COPY

```
wal_level = logical
max_replication_slots = 10
max_wal_senders = 10
```

Edit the `pg_hba.conf` file to allow replication connections from the standby server.

COPY

```
host      replication      replicator      <standby_ip_address>/32      md5
```

3. **Create a Replication Role:** Create a role for replication on the primary server.

COPY

```
CREATE ROLE replicator WITH REPLICATION PASSWORD 'yourpassword' LOGIN;
```

Step 2: Set Up the Standby Server

1. **Install PostgreSQL 16:** Ensure PostgreSQL 16 is installed on your standby server.
2. **Initial Data Synchronization:** Perform an initial data synchronization from the primary server to the standby server. This can be done using `pg_basebackup`.

COPY

```
pg_basebackup -h <primary_ip_address> -D /var/lib/postgresql/16/main -U
```



3. **Configure the Standby Server:** Edit the `postgresql.conf` file on the standby server.

COPY

```
hot_standby = on
```

Step 3: Set Up Logical Replication

1. **Create a Publication on the Primary Server:** Create a publication for the tables you want to replicate.

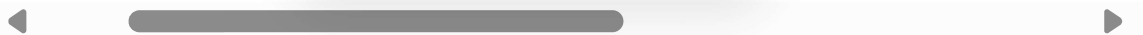
COPY

```
CREATE PUBLICATION my_publication FOR ALL TABLES;
```

2. **Create a Subscription on the Standby Server:** Create a subscription to the publication from the primary server.

COPY

```
3SCRIPTION my_subscr<input>_ip_address> dbname=mydb
```



Step 4: Verify the Configuration

1. **Check the Replication Status:** On the standby server, you can check the replication status using:

```
SELECT * FROM pg_stat_subscription;
```

COPY

2. **Verify Data Replication:** Insert some data into the primary server's database and verify that it appears on the standby server.

```
-- On Primary Server
INSERT INTO test_table (column1, column2) VALUES ('value1', 'value2');

-- On Standby Server
SELECT * FROM test_table;
```

COPY

Step 5: Monitoring and Maintenance

1. **Monitor Replication Lag:** Regularly monitor the replication lag to ensure the standby server is in sync with the primary server.

```
SELECT * FROM pg_stat_replication;
```

COPY

2. **Handle Failover:** In case of primary server failure, you can promote the standby server to be the new primary.

```
pg_ctl promote -f
```

COPY

References

- [PostgreSQL Documentation on Logical Replication](#)
- [Setting Up Logical Replication](#)
- [PostgreSQL pg_hba.conf File Documentation](#)
- [PostgreSQL CREATE SUBSCRIPTION Documentation](#)

By following these steps, you should be able to set up a hot standby using logical replication in PostgreSQL 16, ensuring high availability and disaster recovery capabilities for your PostgreSQL databases.

Unnest Function in PostgreSQL: Transform Arrays into Rows

Learn how PostgreSQL's UNNEST function transforms arrays into rows for...

[minervadb.xyz](#)

<https://minervadb.xyz/optimizing-query-performance-tips-for-troubleshooting-postgresql-statistics-and-cost-estimation/>

<https://minervadb.xyz/optimizing-postgresql-performance-navigating-the-use-of-bind-variables-in-queries/>



Subscribe to our newsletter

Read articles from **All things PostgreSQL from MinervaDB** directly inside your inbox. Subscribe to the newsletter, and don't miss out.

SUBSCRIBE

PostgreSQL

MySQL

SQL

Data Science

Databases

software development

Software Engineering

data structures

analytics

big data

Artificial Intelligence

Machine Learning

Cloud

Security

MORE ARTICLES

Shiv Iyer

Shiv Iyer

How to Use Cumulative Aggregation in PostgreSQL: A Step-by-Step Retail Use Case

Cumulative Aggregation in PostgreSQL Cumulative aggregation, also referred to as running totals or c...

Learn PostgreSQL Embedded JOINS: Examples and Techniques

In PostgreSQL, embedding JOINS within JOINS refers to the practice of chaining multiple JOIN operati...

Shiv Iyer



Easy Steps to Transfer Data from Amazon EMR to Redshift

To load data from Amazon EMR (Elastic MapReduce) to Amazon Redshift, you need to follow a series of ...

©2025 All things PostgreSQL from MinervaDB

[Archive](#) • [Privacy policy](#) • [Terms](#)

