# PostgreSQL Streaming Replication

## Introduction

Streaming replication in PostgreSQL is a robust and efficient method for replicating data from a primary database to one or more standby databases. It ensures high availability, disaster recovery, and scalability by distributing read workloads to replicas. This document provides a comprehensive guide for setting up streaming replication in PostgreSQL 17.

## Prerequisites

1. Two or more servers: One will act as the primary server and the others as standby servers.  
2. PostgreSQL 17 installed on all servers.  
3. Proper network connectivity between the servers.  
4. Basic understanding of PostgreSQL configuration and administration.  
5. Sufficient disk space for Write-Ahead Logging (WAL) files.

## Configuration Steps

### 1. Configure the Primary Server

a. Edit the `postgresql.conf` file:  
 - Enable WAL archiving and set WAL level to 'replica':  
 wal\_level = replica  
 max\_wal\_senders = 10  
 wal\_keep\_size = 128MB  
 - Adjust `listen\_addresses` to allow connections from standby servers:  
 listen\_addresses = '\*'

b. Edit the `pg\_hba.conf` file to allow replication connections:  
 host replication all <standby\_server\_ip>/32 md5

c. Restart the PostgreSQL service to apply changes:  
 sudo systemctl restart postgresql

### 2. Configure the Standby Server

a. Stop the PostgreSQL service:  
 sudo systemctl stop postgresql

b. Use `pg\_basebackup` to copy data from the primary server:  
 pg\_basebackup -h <primary\_server\_ip> -D /var/lib/postgresql/17/main -U replication -P --wal-method=stream

c. Create a `standby.signal` file in the data directory to enable standby mode:  
 touch /var/lib/postgresql/17/main/standby.signal

d. Configure `primary\_conninfo` in `postgresql.conf`:  
 primary\_conninfo = 'host=<primary\_server\_ip> port=5432 user=replication password=<replication\_password>'

e. Restart the PostgreSQL service:  
 sudo systemctl start postgresql

### 3. Verifying Replication

a. Check replication status on the primary server:  
 SELECT \* FROM pg\_stat\_replication;

b. Check the standby server logs for successful connection messages.

## Monitoring Replication

Use the following tools and queries to monitor replication:  
- Query `pg\_stat\_replication` on the primary server to view active connections.  
- Monitor WAL files and lag using tools like `pg\_stat\_wal\_receiver` on standby servers.  
- Set up automated alerts using monitoring tools like Prometheus and Grafana.

## Best Practices

1. Regularly monitor replication lag to ensure data consistency.  
2. Use a dedicated replication user with minimal privileges.  
3. Test failover procedures periodically in a staging environment.  
4. Keep backups even with replication to avoid accidental data loss.  
5. Use SSL for secure connections between primary and standby servers.

## Conclusion

Streaming replication in PostgreSQL is a powerful tool for achieving high availability and scalability. By following the steps outlined in this document, you can set up and manage a robust replication environment. Regular monitoring and adherence to best practices will ensure the reliability of your PostgreSQL database.