



# **Configuring EDB Postgres™ Advanced Server Streaming Replication**

**EDB Postgres™ Advanced Server 9.5**  
formerly Postgres Plus Advanced Server 9.5

**March 10, 2016**

**Configuring EDB Postgres Advanced Server Streaming Replication**  
by EnterpriseDB® Corporation  
**Copyright © 2016 EnterpriseDB Corporation. All rights reserved.**

EnterpriseDB Corporation, 34 Crosby Drive, Suite 100, Bedford, MA 01730, USA  
**T** +1 781 357 3390   **F** +1 978 589 5701   **E** [info@enterprisedb.com](mailto:info@enterprisedb.com) **www**.[enterprisedb.com](http://enterprisedb.com)

# Table of Contents

1	Introduction.....	4
1.1	Typographical Conventions Used in this Guide .....	5
2	Configuring Streaming Replication .....	6
2.1	Configuring the Master Node .....	7
2.2	Configuring the Standby Node .....	10
2.2.1	Confirming Replication from the Master to Standby .....	14
2.3	Manually Invoking Failover .....	15

# 1 Introduction

*Notice: The names for EDB's products have changed.*

*The product formerly referred to as Postgres Plus Advanced Server is now referred to as EDB Postgres Advanced Server (Advanced Server).*

*The product formerly referred to as Postgres Enterprise Manager (PEM) is now referred to as EDB Postgres Enterprise Manager (EDB Enterprise Manager).*

*Until a new version of this documentation is published, wherever you see an earlier version of a product name, you may substitute it with the current name. Name changes in software and software outputs will be phased in over time.*

EDB Postgres Advanced Server (Advanced Server) includes Streaming Replication functionality. You can use Streaming Replication to create a high-availability cluster that enables *hot standby*. Hot standby allows a standby node to respond to read-only queries while providing an environment that replicates the data of the master node. The server streams the change log (WAL records) to ensure that the data on the Standby node is updated as soon as possible, rather than waiting for a log file to fill before shipping an update to a standby node.

Streaming Replication does not provide a mechanism to monitor a database for a server failure or invoke failover. EDB Failover Manager provides the cluster health monitoring, node/database failure detection, and automatic failover mechanisms needed to create a high availability solution. For information about EnterpriseDB Failover Manager, please visit the EnterpriseDB website at:

<http://www.enterprisedb.com/products/edb-failover-manager>

In the following tutorial, we will walk you through configuring a simple Streaming Replication scenario with one Master node and one Standby node that enables hot standby. This tutorial assumes:

- You've installed Advanced Server on the Master node of a replication scenario.
- You've installed Advanced Server on the Standby node of a replication scenario.

For more information about installing Advanced Server, see the installation guide, available at:

<http://www.enterprisedb.com/products-services-training/products/documentation/enterpriseedition>

## 1.1 Typographical Conventions Used in this Guide

Certain typographical conventions are used in this manual to clarify the meaning and usage of various commands, statements, programs, examples, etc. This section provides a summary of these conventions.

In the following descriptions a *term* refers to any word or group of words that are language keywords, user-supplied values, literals, etc. A term's exact meaning depends upon the context in which it is used.

- *Italic font* introduces a new term, typically, in the sentence that defines it for the first time.
- Fixed-width (mono-spaced) font is used for terms that must be given literally such as SQL commands, specific table and column names used in the examples, programming language keywords, etc. For example, `SELECT * FROM emp;`
- *Italic fixed-width font* is used for terms for which the user must substitute values in actual usage. For example, `DELETE FROM table_name;`
- A vertical pipe | denotes a choice between the terms on either side of the pipe. A vertical pipe is used to separate two or more alternative terms within square brackets (optional choices) or braces (one mandatory choice).
- Square brackets [ ] denote that one or none of the enclosed terms may be substituted. For example, `[ a | b ]` means choose one of “a” or “b” or neither of the two.
- Braces {} denote that exactly one of the enclosed alternatives must be specified. For example, `{ a | b }` means exactly one of “a” or “b” must be specified.
- Ellipses ... denote that the preceding term may be repeated. For example, `[ a | b ] ...` means that you may have the sequence, “`b a a b a`”.

## 2 Configuring Streaming Replication

The following section will walk you through the process of configuring a simple two-node replication scenario that uses streaming replication to replicate data from a Master node to a Standby node. The replication process for larger scenarios can be complex; for detailed information about configuration options, please see the PostgreSQL core documentation, available at:

<http://www.postgresql.org/docs/current/static/warm-standby.html#STREAMING-REPLICATION>

In the example that follows, we will use a .pgpass file to enable md5 authentication for the replication user. For more information about the supported authentication options, please see the PostgreSQL core documentation at:

<http://www.postgresql.org/docs/9.5/static/client-authentication.html>

The steps that follow configure a simple streaming replication scenario with one Master node and one Standby node, each running an installation of Advanced Server. In the example:

- The Master node resides on 192.168.2.206
- The Standby node resides on 192.168.2.217
- The replication user name is edbrepuser.

The pathnames and commands referenced in the examples are for Advanced Server hosts that reside on a CentOS 6.5 host – you may have to modify paths and commands for your configuration.

## 2.1 Configuring the Master Node

Connect to the master node of the replication scenario, and modify the `pg_hba.conf` file (located in the `data` directory under your Postgres installation), adding connection information for the replication user (in our example, `edbrepuser`):

```
host replication edbrepuser 192.168.2.217/32 md5
```

The connection information should specify the address of the Standby node of the replication scenario, and your preferred authentication method (see Figure 2.1).

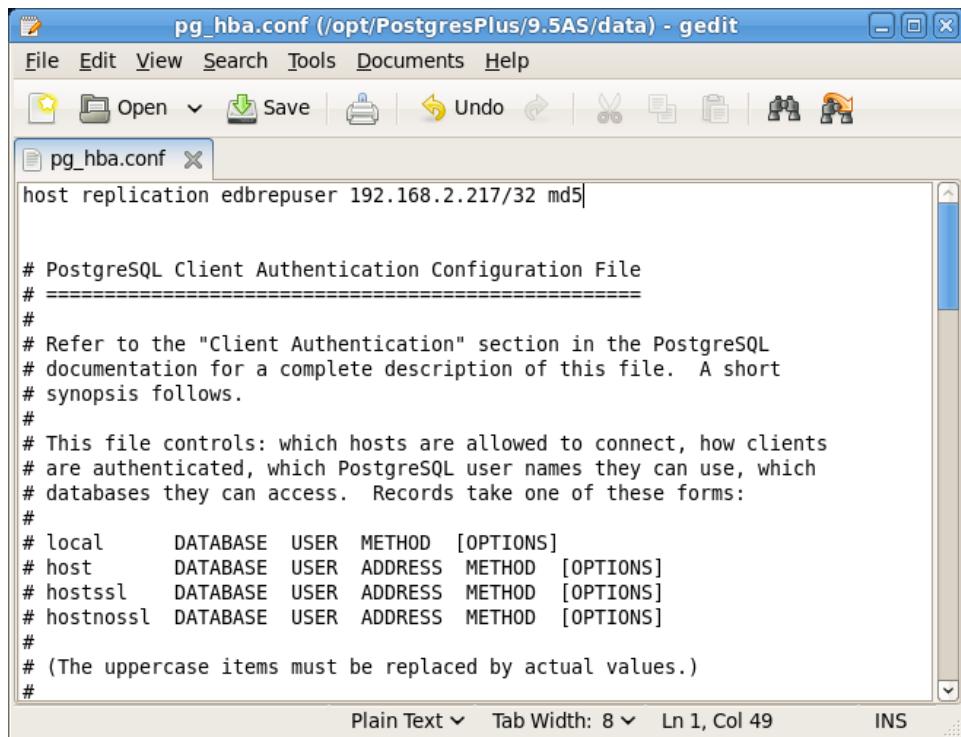


Figure 2.1 – Modifying the `pg_hba.conf` File

Then, modify the `postgresql.conf` file (located in the `data` directory, under your Advanced Server installation), adding the following replication parameter and values to the end of the file (see Figure 2.2):

```
wal_level = hot_standby
max_wal_senders = 8
wal_keep_segments = 128
archive_mode = on
archive_command = 'cp %p /tmp/%f'
```

## Configuring EDB Postgres Advanced Server Streaming Replication

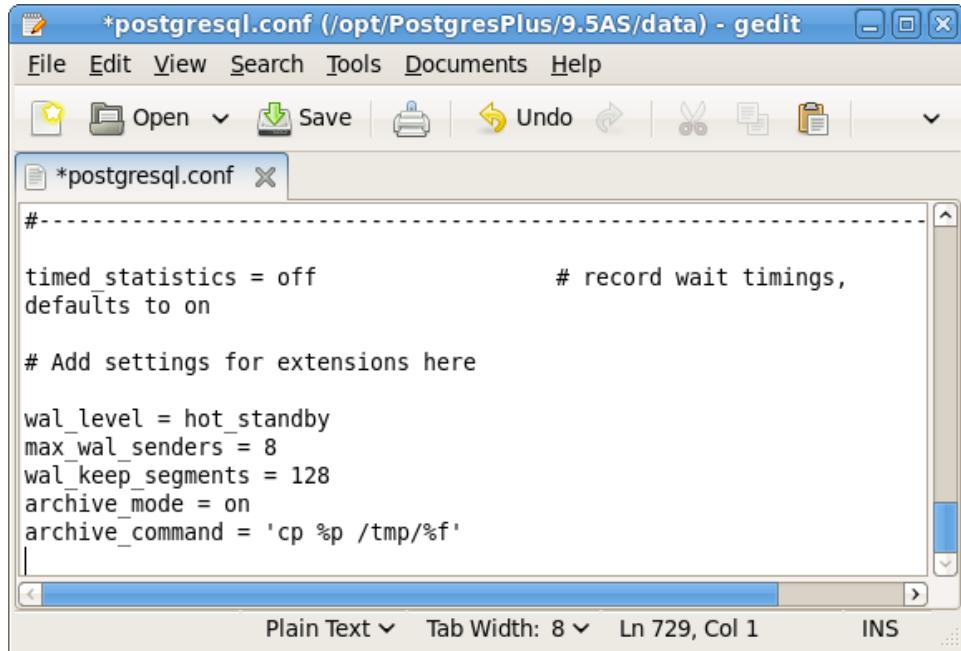


Figure 2.2 – Modifying the postgresql.conf File

Save and close the configuration file, and use your platform specific command to restart the server (see Figure 2.3). For example:

```
/etc/init.d/ppas-9.5 restart
```

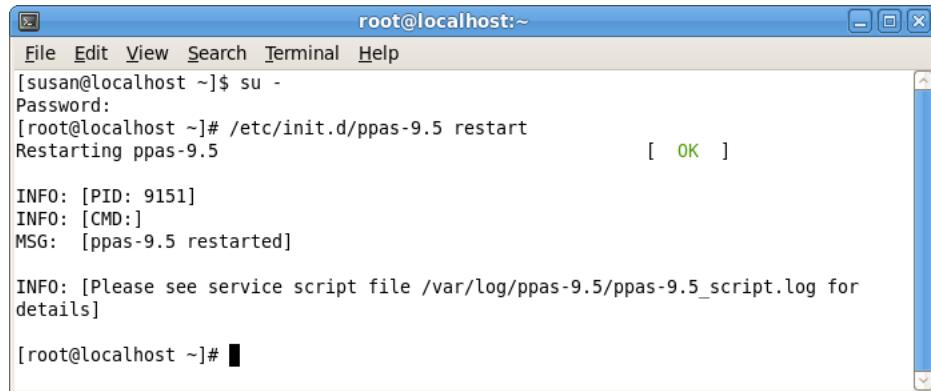


Figure 2.3 – Restarting the ppas-9.5 Service

Assume the identity of the `enterprisedb` database superuser:

```
su - enterprisedb
```

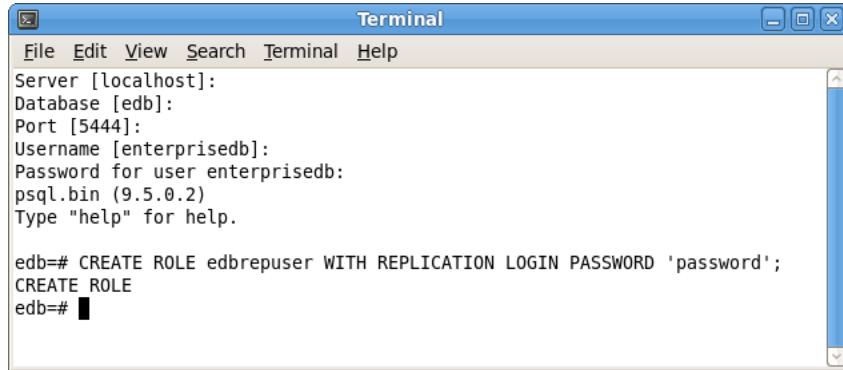
Then, start a psql session, connecting to the `edb` database:

```
psql -d edb
```

## Configuring EDB Postgres Advanced Server Streaming Replication

At the `psql` command line, create a user with the `replication` attribute (see Figure 2.4):

```
CREATE ROLE edbrepuser WITH REPLICATION LOGIN PASSWORD  
'password';
```



The screenshot shows a terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu is a configuration section with the following settings:

- Server [localhost]:
- Database [edb]:
- Port [5444]:
- Username [enterprisedb]:
- Password for user enterprisedb:  
psql.bin (9.5.0.2)
- Type "help" for help.

At the bottom of the window, the PostgreSQL prompt `edb=#` is followed by the command:

```
edb=# CREATE ROLE edbrepuser WITH REPLICATION LOGIN PASSWORD 'password';  
CREATE ROLE  
edb=#[REDACTED]
```

Figure 2.4 – Creating the `edbrepuser` role.

For more information about the options available with the `CREATE ROLE` command, see the PostgreSQL core documentation at:

<http://www.postgresql.org/docs/9.5/static/sql-createrole.html>

## 2.2 Configuring the Standby Node

Connect to the Standby server, and assume the identity of the database superuser `enterprisedb`:

```
su - enterprisedb
```

With your choice of editor, create a `.pgpass` file in the home directory of the `enterprisedb` user. The `.pgpass` file holds the password of the replication user in plain-text form; if you are using a `.pgpass` file, you should ensure that only trusted users have access to the `.pgpass` file:

Add an entry that specifies connection information for the replication user (see Figure 2.5):

```
*:5444:*:edbrepuser:password
```

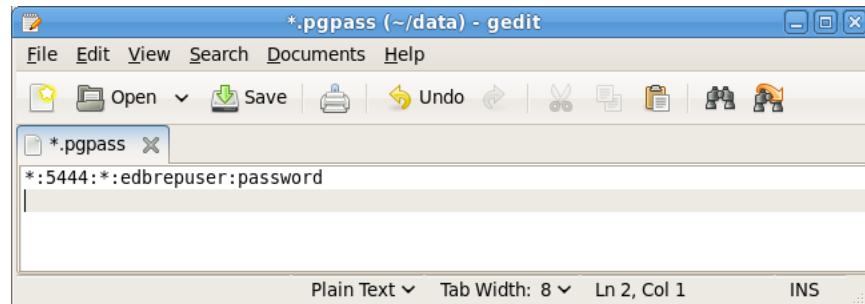


Figure 2.5 – Modifying the `.pgpass` File

Save and close the `.pgpass` file. The server will enforce restrictive permissions on the `.pgpass` file; use the following command to change the file permissions:

```
chmod 600 .pgpass
```

Relinquish the identity of the database superuser:

```
exit
```

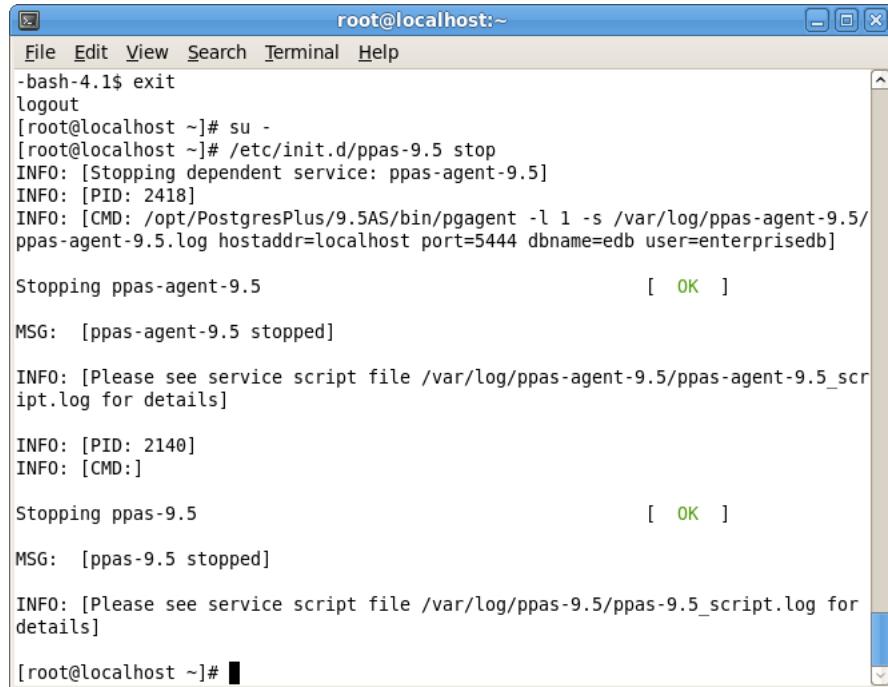
Then, assume superuser privileges:

```
su -
```

You must stop the database server before replacing the data directory on the Standby node with the data directory of the Master node (see Figure 2.6). Use the command:

```
/etc/init.d/ppas-9.5 stop
```

## Configuring EDB Postgres Advanced Server Streaming Replication



The screenshot shows a terminal window titled "root@localhost:~". The session starts with the command "exit", followed by "logout". Then, the user runs "/etc/init.d/ppas-9.5 stop". The output shows the service stopping, with log messages indicating it's stopping dependent services and the ppas-agent process. It also mentions a PID of 2418 and a CMD of "/opt/PostgresPlus/9.5AS/bin/pgagent -l 1 -s /var/log/ppas-agent-9.5/ppas-agent-9.5.log hostaddr=localhost port=5444 dbname=edb user=enterprisedb". The process is marked as "OK". Next, the user stops the ppas-agent-9.5 service, which is also marked as "OK". Finally, the user stops the ppas-9.5 service, which is also marked as "OK". The session ends with "[root@localhost ~]#".

Figure 2.6 – Stopping the Database Server

The following command deletes the existing `data` directory, and any information currently stored in the `data` cluster on the Standby node:

```
rm -rf /opt/PostgresPlus/9.5AS/data
```

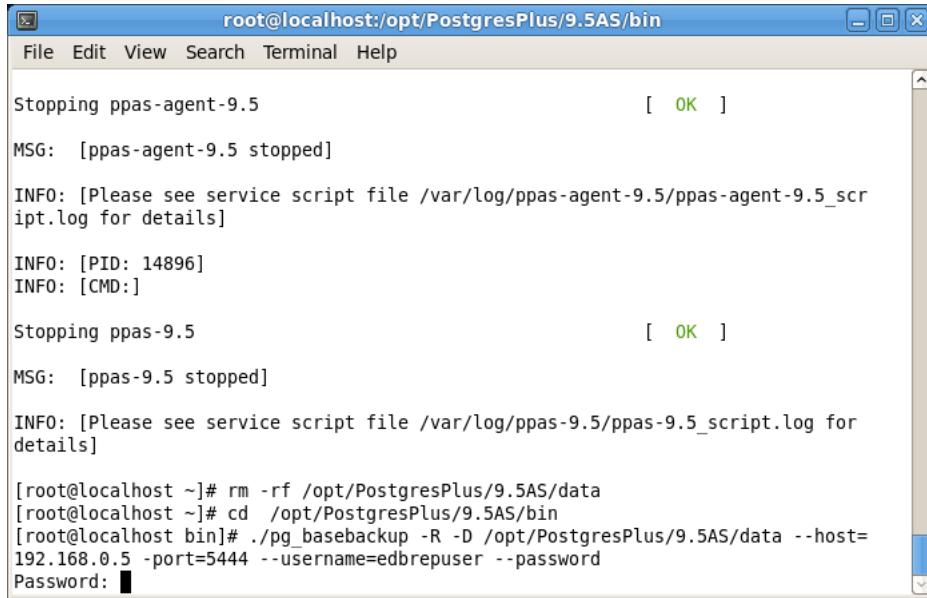
After deleting the `data` directory on the Standby node, navigate into the `bin` directory under your Advanced Server installation:

```
cd /opt/PostgresPlus/9.5AS/bin
```

Use the `pg_basebackup` utility to copy the `data` directory of the Master node to the Standby node (see Figure 2.7):

```
./pg_basebackup -R -D /opt/PostgresPlus/9.5AS/data  
--host=192.168.2.206 --port=5444  
--username=edbrepuser --password
```

## Configuring EDB Postgres Advanced Server Streaming Replication



```
root@localhost:/opt/PostgresPlus/9.5AS/bin
File Edit View Search Terminal Help
Stopping ppas-agent-9.5 [ OK ]
MSG: [ppas-agent-9.5 stopped]
INFO: [Please see service script file /var/log/ppas-agent-9.5/ppas-agent-9.5_script.log for details]
INFO: [PID: 14896]
INFO: [CMD:]

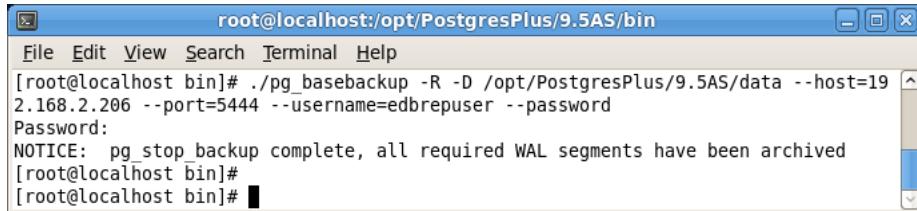
Stopping ppas-9.5 [ OK ]
MSG: [ppas-9.5 stopped]
INFO: [Please see service script file /var/log/ppas-9.5/ppas-9.5_script.log for details]
[root@localhost ~]# rm -rf /opt/PostgresPlus/9.5AS/data
[root@localhost ~]# cd /opt/PostgresPlus/9.5AS/bin
[root@localhost bin]# ./pg_basebackup -R -D /opt/PostgresPlus/9.5AS/data --host=192.168.0.5 -port=5444 --username=edbrepuser --password
Password: [REDACTED]
```

Figure 2.7 – Replacing the data Directory

The call to `pg_basebackup` specifies the IP address of the Master node and the name of the replication user created on the Master node. For more information about the options available with the `pg_basebackup` utility, see the PostgreSQL core documentation at:

<http://www.postgresql.org/docs/9.5/static/app-pgbasebackup.html>

When prompted by `pg_basebackup`, provide the password associated with the replication user (see Figure 2.8).



```
root@localhost:/opt/PostgresPlus/9.5AS/bin
File Edit View Search Terminal Help
[root@localhost bin]# ./pg_basebackup -R -D /opt/PostgresPlus/9.5AS/data --host=192.168.2.206 -port=5444 --username=edbrepuser --password
Password:
NOTICE: pg_stop_backup complete, all required WAL segments have been archived
[root@localhost bin]#
[root@localhost bin]#
```

Figure 2.8 – Invoking `pg_basebackup`

After copying the data directory, change ownership of the directory to the database superuser (`enterprisedb`):

```
chown -R enterprisedb /opt/PostgresPlus/9.5AS/data
```

## Configuring EDB Postgres Advanced Server Streaming Replication

Navigate into the data directory:

```
cd /opt/PostgresPlus/9.5AS /data
```

With your choice of editor, create a file named `recovery.conf` (in the `/opt/PostgresPlus/9.xAS/data` directory) that includes:

```
standby_mode = on
primary_conninfo = 'host=192.168.0.5 port=5444
user=edbrepuser password=password'
```

Please note: the `primary_conninfo` parameter specifies connection information for the replication user on the master node of the replication scenario. Save and close the file.

Change ownership of the `recovery.conf` file to `enterprisedb`:

```
chown enterprisedb:enterprisedb recovery.conf
```

The `data` file has been copied from the Master node, and will contain the replication parameters specified on the Master. You must modify the `postgresql.conf` file on the Standby node, specifying new values at the end of the file (see Figure 2.9):

```
wal_level = hot_standby
max_wal_senders = 8
wal_keep_segments = 128
hot_standby = on
```

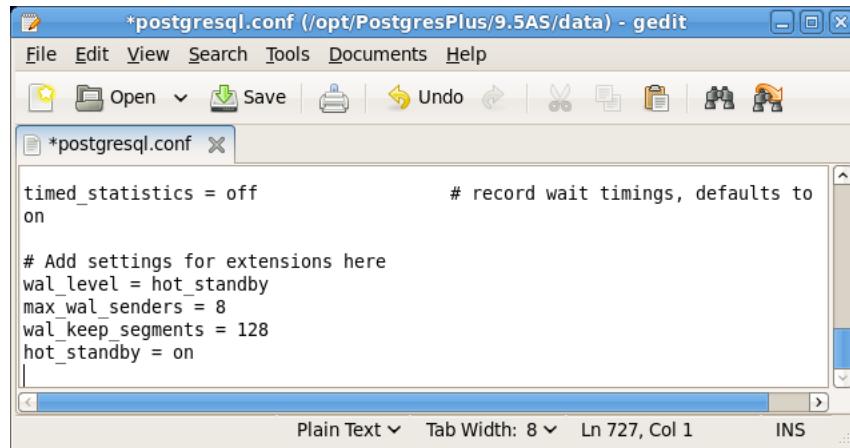


Figure 2.9 – Modifying the `postgresql.conf` File

Save the file and restart the server:

```
/etc/init.d/ppas-9.5 start
```

At this point, the Master node will be replicating data to the Standby node.

For more information about the options available with the `pg_basebackup` utility, see the PostgreSQL core documentation at:

<http://www.postgresql.org/docs/9.5/static/app-pgbasebackup.html>

### 2.2.1 Confirming Replication from the Master to Standby

You can confirm that the server is running and replicating by entering the command:

```
ps -ef | grep postgres
```

If replication is running, the Standby server will echo:

```
501 42054 1 0 07:57 pts/1 00:00:00
/opt/PostgresPlus/9.2AS/bin/edb-postgres -D
/opt/PostgresPlus/9.2AS/data
501 42055 42054 0 07:57 ? 00:00:00 postgres: logger process
501 42056 42054 0 07:57 ? 00:00:00 postgres: startup
process recovering 00000001000000000000000000000004
501 42057 42054 0 07:57 ? 00:00:00 postgres: checkpointer
process
501 42058 42054 0 07:57 ? 00:00:00 postgres: writer process
501 42059 42054 0 07:57 ? 00:00:00 postgres: stats
collector process
501 42060 42054 0 07:57 ? 00:00:00 postgres: wal receiver
process streaming 0/4000150
501 42068 42025 0 07:58 pts/1 00:00:00 grep postgres
```

If you connect to the Standby with the `psql` client and query the `pg_is_in_recovery()` function, the server will reply:

```
edb=# select pg_is_in_recovery();
pg_is_in_recovery
-----
t
(1 row)
```

Any entries made to the Master node will be replicated to the Standby node. The Standby node will operate in read-only mode; while you can query the Standby server, you will not be able to add entries directly to the database that resides on the Standby node.

## 2.3 Manually Invoking Failover

To promote the Standby to become the Master node, assume the identity of the cluster owner (enterprisedb):

```
su - enterprisedb
```

Then, invoke pg\_ctl:

```
/opt/PostgresPlus/9.5AS/bin/pg_ctl promote -D /  
opt/PostgresPlus/9.5AS /data/
```

Then, if you connect to the Standby node with psql, the server will confirm that it is no longer a standby node:

```
edb=# select pg_is_in_recovery();  
pg_is_in_recovery  
-----  
f  
(1 row)
```

For more information about configuring and using streaming replication, please refer to Chapter 25 of the PostgreSQL core documentation, available at:

<http://www.postgresql.org/docs/current/static/warm-standby.html>