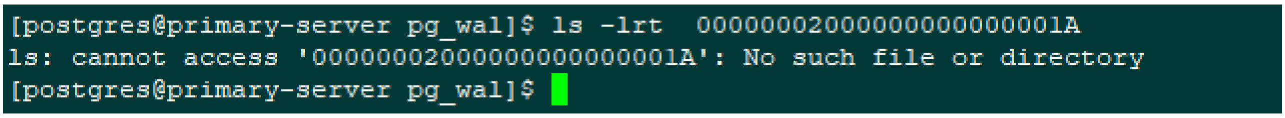
WAL file is missing or moved to Archive before applying to standby

Primary Log file:

ERROR: requested WAL segment 00000002000000000000001A has already been removed

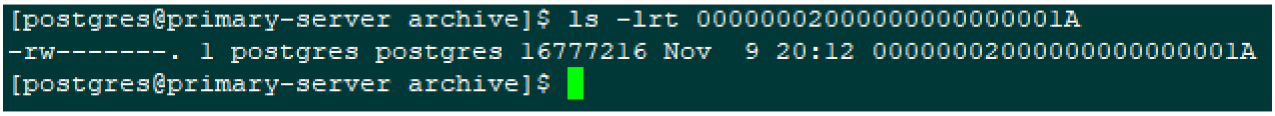
```
2024-11-09 21:58:35.170 PST [14032] ERROR:  requested WAL segment
00000002000000000000001A has already been removed

2024-11-09 21:58:35.170 PST [14032] STATEMENT:  START_REPLICATION SLOT
"new_slot_name" 0/1A000000 TIMELINE 2

2024-11-09 21:58:40.161 PST [14033] ERROR:  requested WAL segment
00000002000000000000001A has already been removed
```

file `00000002000000000000001A` in the `pg_wal` directory cannot be found, The file may have already been archived or deleted due to rotation.



```
[postgres@primary-server pg_wal]$ ls -lrt  00000002000000000000001A
ls: cannot access '00000002000000000000001A': No such file or directory
[postgres@primary-server pg_wal]$
```

WAL file `00000002000000000000001A` is present in the **archive directory** on the **primary server**. This is good news, as it means you have access to the missing WAL segment, which the **standby server** was trying to access.



```
[postgres@primary-server archive]$ ls -lrt 00000002000000000000001A
-rw-------. 1 postgres postgres 16777216 Nov  9 20:12 00000002000000000000001A
[postgres@primary-server archive]$
```

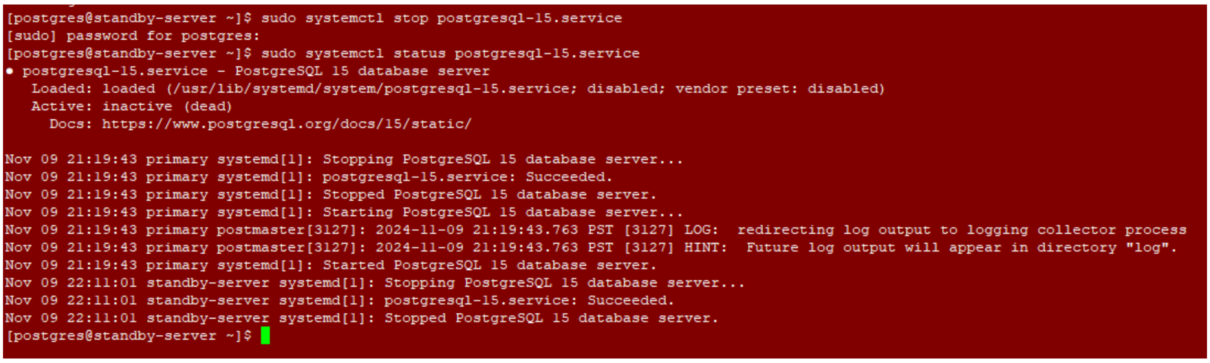
To resolve the issue, you need to restore this WAL file to the appropriate location on the **primary server** and ensure the **standby server** can use it.

Steps to restore the WAL file and get replication working:

- **Stop the Standby Server (if it's currently running)**

First, stop the **standby server** if it's still running to avoid any conflicts or errors during the restoration process:

```
sudo systemctl stop postgre-15.service
```



```
[postgres@standby-server ~]$ sudo systemctl stop postgresql-15.service
[sudo] password for postgres:
[postgres@standby-server ~]$ sudo systemctl status postgresql-15.service
● postgresql-15.service - PostgreSQL 15 database server
   Loaded: loaded (/usr/lib/systemd/system/postgresql-15.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://www.postgresql.org/docs/15/static/

Nov 09 21:19:43 primary systemd[1]: Stopping PostgreSQL 15 database server...
Nov 09 21:19:43 primary systemd[1]: postgresql-15.service: Succeeded.
Nov 09 21:19:43 primary systemd[1]: Stopped PostgreSQL 15 database server.
Nov 09 21:19:43 primary systemd[1]: Starting PostgreSQL 15 database server...
Nov 09 21:19:43 primary postmaster[3127]: 2024-11-09 21:19:43.763 PST [3127] LOG:  redirecting log output to logging collector process
Nov 09 21:19:43 primary postmaster[3127]: 2024-11-09 21:19:43.763 PST [3127] HINT:  Future log output will appear in directory "log".
Nov 09 21:19:43 primary systemd[1]: Started PostgreSQL 15 database server.
Nov 09 22:11:01 standby-server systemd[1]: Stopping PostgreSQL 15 database server...
Nov 09 22:11:01 standby-server systemd[1]: postgresql-15.service: Succeeded.
Nov 09 22:11:01 standby-server systemd[1]: Stopped PostgreSQL 15 database server.
[postgres@standby-server ~]$
```

- **Copy the WAL file to the Primary Server's `pg_wal` Directory**

Now, copy the **WAL file** from the archive directory back into the **primary server's `pg_wal`** directory, where the active WAL files are stored.

```
 [postgres@primary-server archive]$ cp
/postgres/archive/00000002000000000000001A /postgres/pgdata/pg_wal/
[postgres@primary-server archive]$ ls -lrt
/postgres/pgdata/pg_wal/00000002000000000000001A
-rw-------. 1 postgres postgres 16777216 Nov  9 22:12
/postgres/standby/pg_wal/00000002000000000000001A
[postgres@primary-server archive]$
```

Make sure to replace `/path/to/archive/` and `/path/to/primary/data/pg_wal/` with the actual paths.

- **Ensure Correct File Permissions**

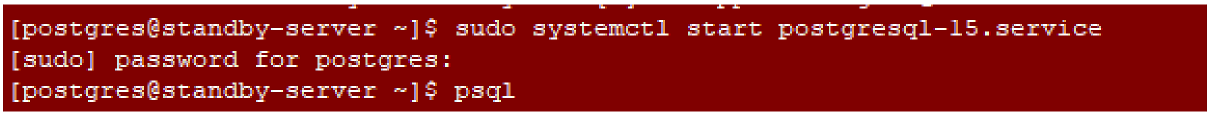
Ensure the **WAL file** you just copied has the correct permissions (it should be owned by the `postgres` user):

```
chown postgres:postgres /postgres/standby/pg_wal/00000002000000000000001A
chmod 600 /postgres/standby/pg_wal/00000002000000000000001A
```

- **Start the Standby Server**

Now, start the **standby server** and check if it successfully connects to the primary server and starts the replication process.

```
sudo systemctl start postgre-15.service
```

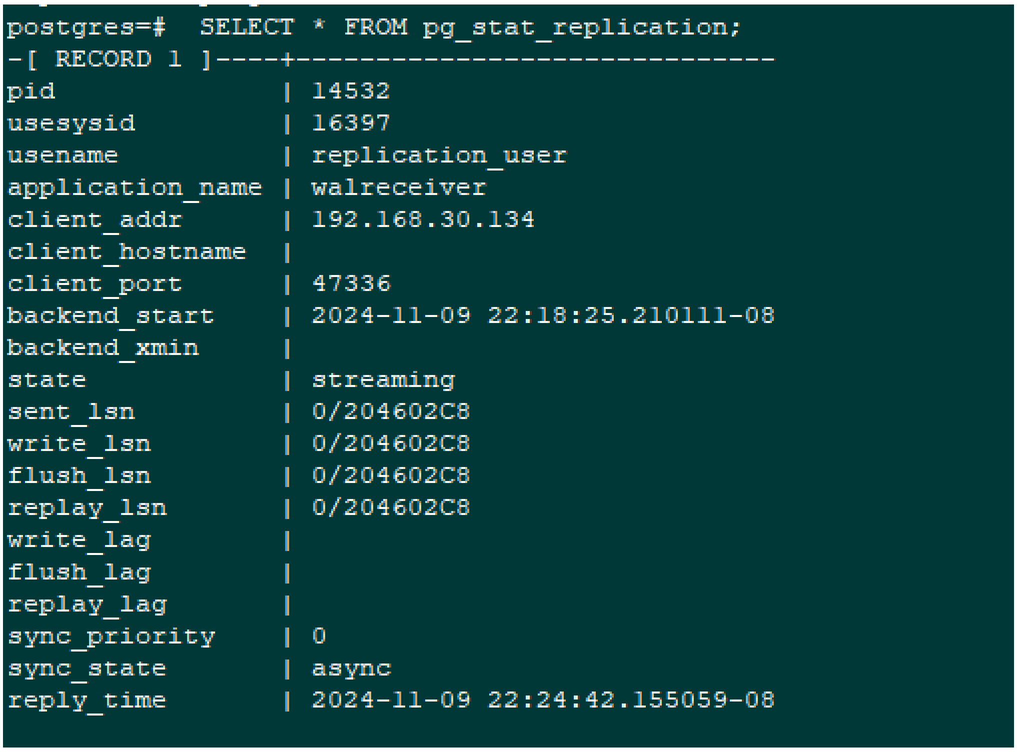


```
[postgres@standby-server ~]$ sudo systemctl start postgresql-15.service
[sudo] password for postgres:
[postgres@standby-server ~]$ psql
```

- **Check Replication Status**

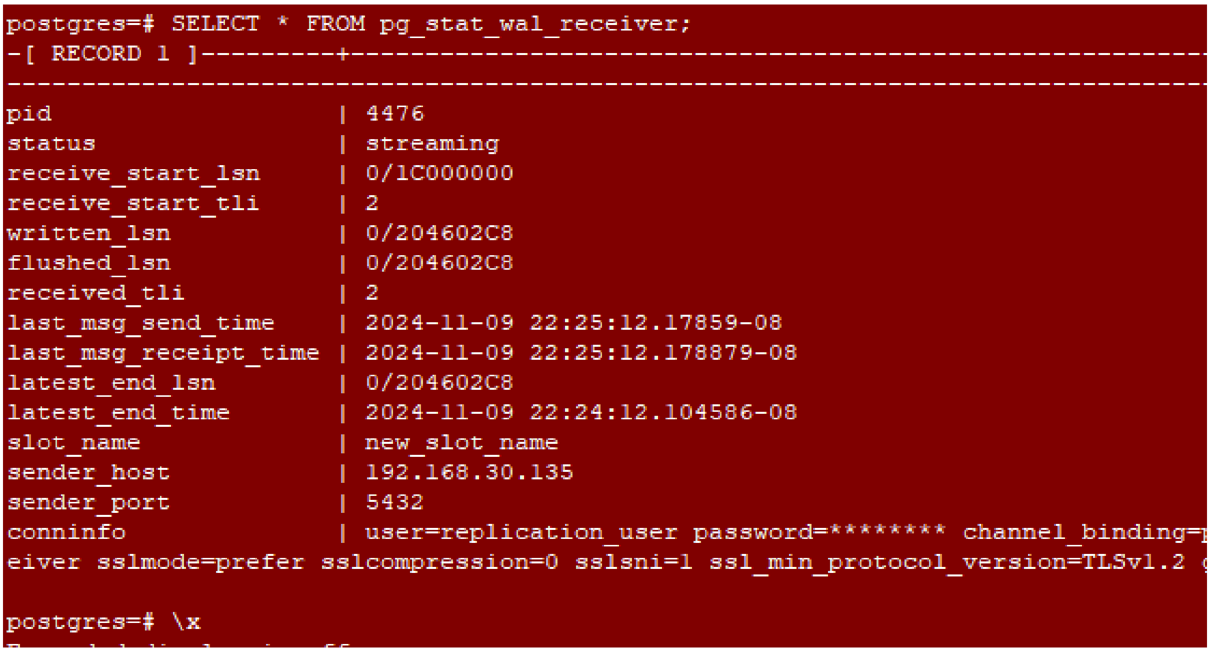
Once the **standby server** is up, check the replication status again on the **primary server**:

```
SELECT * FROM pg_stat_replication;
```



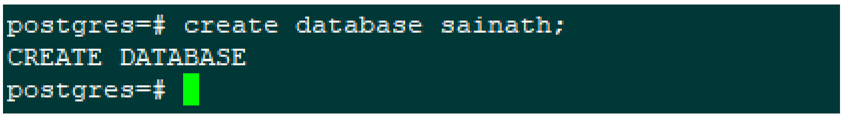
```
postgres=#  SELECT * FROM pg_stat_replication;
-[ RECORD 1 ]----+-----------------------------
pid              | 14532
usesysid         | 16397
usename          | replication_user
application_name | walreceiver
client_addr      | 192.168.30.134
client_hostname  |
client_port      | 47336
backend_start    | 2024-11-09 22:18:25.210111-08
backend_xmin     |
state            | streaming
sent_lsn         | 0/204602C8
write_lsn        | 0/204602C8
flush_lsn        | 0/204602C8
replay_lsn       | 0/204602C8
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
reply_time       | 2024-11-09 22:24:42.155059-08
```

If everything is set up correctly, the **standby server** should now appear in the replication status and be connected.



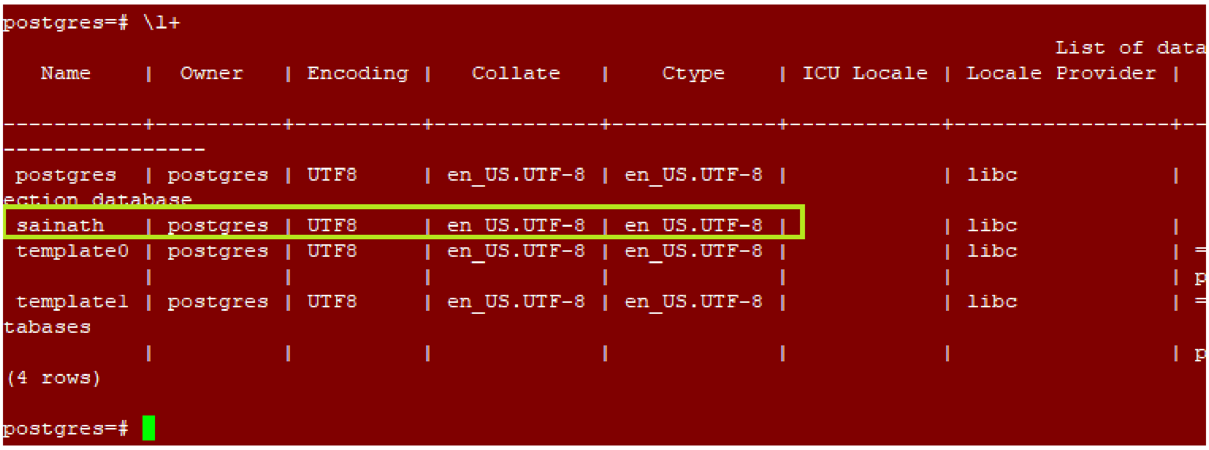
```
postgres=# SELECT * FROM pg_stat_wal_receiver;
-[ RECORD 1 ]---------+-----------------------------------------------------
-------------------------------------------------------------------------------
pid                   | 4476
status                | streaming
receive_start_lsn     | 0/1C000000
receive_start_tli     | 2
written_lsn           | 0/204602C8
flushed_lsn           | 0/204602C8
received_tli          | 2
last_msg_send_time    | 2024-11-09 22:25:12.17859-08
last_msg_receipt_time | 2024-11-09 22:25:12.178879-08
latest_end_lsn        | 0/204602C8
latest_end_time       | 2024-11-09 22:24:12.104586-08
slot_name             | new_slot_name
sender_host           | 192.168.30.135
sender_port           | 5432
conninfo              | user=replication_user password=******** channel_binding=
eiver sslmode=prefer sslcompression=0 sslsni=1 ssl_min_protocol_version=TLSv1.2

postgres=# \x
```

Primary



```
postgres=# create database sainath;
CREATE DATABASE
postgres=#
```

Standby in Sync



```
postgres=# \l+
                                                                                     List of data
   Name     |  Owner    | Encoding |   Collate     |    Ctype      | ICU Locale | Locale Provider |

-----------+-----------+----------+-------------+-------------+------------+-----------------+--
----------------
 postgres   | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
ection database
 sainath    | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            | =
           |          |          |             |             |            |                 | p
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            | =
tabases
           |          |          |             |             |            |                 | p
(4 rows)

postgres=#
```

Alternative Option (Reinitialize Replication)

If restoring the WAL file doesn't work or if you prefer starting fresh, you could also reinitialize the **standby server** by taking a new **base backup**:

1. **Stop the Standby Server**.
2. Use `pg_basebackup` from the **primary server** to take a new base backup:

   On the **standby server**:

   ```
   pg_basebackup -h 192.168.30.135 -U replication_user -D
   /postgres/standby/data --write-recovery-conf
   ```

3. Once the base backup is complete, start the **standby server** again.