# Understanding Multiple `pg\_basebackup` Processes in PostgreSQL

## 1. Why Are Two `pg\_basebackup` Processes Created?

When you run `pg\_basebackup`, PostgreSQL creates two processes. This is expected behavior and occurs due to PostgreSQL's process-based architecture.

### Process Overview

When executing the command:  
`/usr/lib/pgsql-15.3/bin/pg\_basebackup --checkpoint=fast -D /data/postgresql-15/data -h 172.25.4.7 -p 5432 -Xs -R -P`, PostgreSQL spawns two processes:

1. \*\*Parent Process\*\*  
 - Manages the overall backup.  
 - Requests data from the primary server.  
 - Spawns a child process for efficient data handling.

2. \*\*Child Process\*\*  
 - Receives and writes data to the replica’s disk.  
 - Handles actual data transfer from the primary.  
 - Manages WAL streaming if the `-Xs` flag is enabled.

## 2. How to Identify the Two Processes?

Run the following command:  
  
```sh  
ps -ef | grep pg\_basebackup  
```

Example output:  
```  
postgres 1446964 1 0 Mar24 ? 00:01:19 /usr/lib/pgsql-15.3/bin/pg\_basebackup ...  
postgres 1449739 1446964 0 Mar24 ? 00:02:11 /usr/lib/pgsql-15.3/bin/pg\_basebackup ...  
```

## 3. Why Does `pg\_basebackup` Use Two Processes?

PostgreSQL’s architecture is \*\*process-based\*\*, meaning each task runs as a separate process.  
  
- \*\*Parent Process\*\*: Establishes the connection and controls the backup execution.  
- \*\*Child Process\*\*: Handles data transfer and WAL streaming.

## 4. How to Confirm Both Processes Are Working Correctly?

Check data transfer progress with:  
  
```sh  
ls -lh /data/postgresql-15/data  
```

Monitor PostgreSQL logs:  
  
```sh  
tail -f /var/lib/pgsql/15/data/log/postgresql.log  
```

Check streaming WAL status on the primary server:  
  
```sql  
SELECT \* FROM pg\_stat\_replication;  
```

## 5. What Happens If One Process Fails?

- \*\*If the child process fails\*\*, the parent will report an error and stop.  
- \*\*If the parent process fails\*\*, the entire backup process terminates.

## 6. Should You Be Concerned?

✅ \*\*No, this is normal behavior.\*\* PostgreSQL always creates multiple processes for efficiency.

## ✅ Conclusion

| \*\*Process\*\* | \*\*Function\*\* |  
|------------|-------------|  
| \*\*Parent Process\*\* | Manages `pg\_basebackup` execution |  
| \*\*Child Process\*\* | Handles data transfer & WAL streaming |

=============Why Does pg\_basebackup Use Two Processes?=========

This happens because of PostgreSQL’s **process-based architecture**. Each connection and task in PostgreSQL is handled by separate processes.

**3.1 Role of the Parent Process**

1. Handles overall backup control.
2. Establishes a connection with the primary PostgreSQL server at 172.25.4.7:5432.
3. Initiates the data transfer process by requesting data from the primary.
4. Spawns a child process to handle the data reception.

**3.2 Role of the Child Process**

* Takes care of receiving and writing the data to /data/postgresql-15/data.
* If -Xs (WAL streaming) is enabled, it also sets up a separate WAL streaming connection.
* Ensures that backup progress and performance are not blocked by the parent process.