



## pg\_basebackup

Many people around the globe have used PostgreSQL streaming replication successfully and enjoy its simplicity. `pg_basebackup` is the most commonly used tool to create an initial copy of the database instance around.

Typically a base backup is created as follows:

```
pg_basebackup -h master.example.com -D /slave
```

For a very basic backup this can be enough.

However, there are two issues with this command, which are often forgotten or simply not known by users.

### **Waiting for a checkpoint**

The first important thing here is that `pg_basebackup` has to wait for a checkpoint on the master. In case of a very large database, people might not notice this delay. However, in case of a small database, it is pointless to wait for a couple of minutes.

To speed up the process, the following command can help:

```
pg_basebackup -h master.example.com -D /slave --checkpoint=fast
```

### **Consistent backups**

However, there is more. The backup we have just created cannot be used without some transaction log stored in some archive.

To solve the problem, `pg_basebackup` allows us to use “`--xlog-method=stream`”. It opens a second stream, which fetches the transaction log created while the base backup is running.

This is how it works:

```
pg_basebackup -h master.example.com -D /slave --checkpoint=fast --xlog-method=stream
```

In this case the backup can be used without a `recovery.conf` file. You can start the database directly and you will have a ready to use database instance.

By the way: This method is perfect for quickly cloning test machines for your developers.

In addition to that `--xlog-method=stream` is a wonderful additional safety net because the backup is actually usable.