



# Installation Guide: Cardano node, DB-Sync, Postgres

## Main Idea

Install a node on your machine that synchronizes with the blockchain and receives all the information about the chain from one of the active nodes provided by IOHK. Then install DB-Sync, a tool included in Cardano's architecture that essentially creates a copy of the node. Then by creating a Postgres database, all the data from DB-Sync are passed as tables in a newly created database.

## Different Build Options

Cabal, Docker, Nix. This guide is for **Nix**.

Implemented on Ubuntu 20.04-LTS.

## Folder Structure

cardano-node & cardano-db-sync folders must be in the same level. Here they have been created in \$HOME.

## Prerequisites

### Install Nix package manager

```
curl -L https://nixos.org/nix/install > install-nix.sh
chmod +x install-nix.sh
./install-nix.sh
```

### Enable Flake

```
sudo mkdir -p /etc/nix
cat <<EOF | sudo tee /etc/nix/nix.conf # or /mnt/etc/nix/nix.conf
experimental-features = nix-command flakes
allow-import-from-derivation = true
EOF
```

### IOHK Binary (optional, enables faster installation):

```
sudo nano /etc/nix/nix.conf
# add the next two lines to the file
substituters = https://cache.nixos.org https://hydra.iohk.io
trusted-public-keys = iohk.cachix.org-1:DpRUyj7h7V830dp/i6Nti+NE02/nhblbov/8MW7Rqoo= hydra.iohk.io:f/Ea+s+dFdN+3Y/G+FDgSq+a5NEWhJGzdjvKNGv0

# sudo mkdir -p /etc/nix
# cat <<EOF | sudo tee /etc/nix/nix.conf # or /mnt/etc/nix/nix.conf
# substituters = https://cache.nixos.org https://hydra.iohk.io
```

```
# trusted-public-keys = iohk.cachix.org-1:DpRUyj7h7V838dp/i6Nti+NE02/nhblbov/8MW7Rqoo= hydra.iohk.io:f/Ea+s+dFdN+3Y/G+FDgSq+a5NEWhJGzdjvKNG
# EOF
```

## Node Installation

### Install Cardano node

```
git clone https://github.com/input-output-hk/cardano-node
cd cardano-node
git tag # find the latest stable version tag and use it to checkout
git checkout <latest-official-tag> -b tag-<latest-official-tag> # we use 1.35.0
nix-build -A scripts.mainnet.node -o mainnet-node-local
```

Or build and start (separate run command below) the node in one go with:

```
nix run github:input-output-hk/cardano-node#mainnet/node
```

### Create tmux windows for Node and DB-sync

```
tmux new -s cardano-node-run
tmux new -s cardano-db-sync-run
tmux a -t cardano-node-run # example: enter the node window
tmux a -t cardano-db-sync-run # example: enter the db-sync window
```

Use the windows to run the respective commands from the cardano-node and cardano-db-sync folders. The cardano node and the DB sync can this way be kept running in the background and be easily accessible.

### Start the node

NOTE: node syncing can take more than 30 hours.

Inside the tmux window and the cardano-node folder use the following command:

```
./mainnet-node-local/bin/cardano-node-mainnet
```

### Install Command Line Interface (CLI)

NOTE: CLI is a blockchain manager and not needed for the current installation, can be *skipped*.

```
cd cardano-node
nix build .#cardano-cli -o cardano-cli-build
```

Run CLI:

```
cd cardano-cli-build/bin
./cardano-cli <command you want> #./cardano-cli to see list of commands
```

## Postgres Installation

### Uninstall Postgres (for clean installation)

```
sudo apt-get --purge remove postgresql*
sudo rm -r /etc/postgresql/
sudo rm -r /etc/postgresql-common/
```

```
sudo rm -r /var/lib/postgresql/  
sudo userdel -r postgres  
sudo groupdel postgres
```

## Install Postgres

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -  
RELEASE=$(lsb_release -cs)  
echo "deb [arch=amd64] http://apt.postgresql.org/pub/repos/apt/ ${RELEASE}"-pgdg main | sudo tee /etc/apt/sources.list.d/pgdg.list  
sudo apt-get update  
sudo apt-get -y install postgresql-14 postgresql-server-dev-14 postgresql-contrib libghc-hdbc-postgresql-dev  
sudo systemctl restart postgresql  
sudo systemctl enable postgresql
```

## Connect to Postgres user

```
sudo -i -u postgres  
psql
```

If we get an error about the port 5432 being down, we do the following:

Go back to user\$ with ctrl-d.

```
pg_lsclusters # get the Ver no. for next command  
sudo pg_ctlcluster <<Ver no.(1st column)> main start
```

Get back to postgres and make the user\$ as SUPERUSER:

```
sudo -i -u postgres  
psql  
CREATE ROLE <user> SUPERUSER LOGIN; # as <user> use your user name  
ALTER USER <user> PASSWORD 'YourpasswdButKeeptheQuotes';  
ctrl-d  
su-<user> # we change back to <user>, to whom we've given superuser privileges  
psql
```

The expected result is an error message about database <user> not existing. That is because psql needs an argument of the database name and default is username, e.g "psql postgres" will work because there is a (default) postgres database. We can safely ignore it for now. If this error is accompanied by the port being down error, while the port is not down, we can again ignore it.

## DB-Sync

**NOTE:** Make sure that cardano-node & cardano-db-sync folders are created on the same level. Use "tree -L 1" to check.

## Install DB-Sync

```
git clone https://github.com/input-output-hk/cardano-db-sync  
cd cardano-db-sync  
git tag  
git checkout <latest-official-tag> -b tag-<latest-official-tag> # we use 13.0.0  
nix-build -A cardano-db-sync -o db-sync-node # installation takes time  
nix-build -A cardano-db-sync-extended -o db-sync-node # to add the extended version  
ls db-sync-node/bin # confirm extended version
```

Next steps should be performed **after** the node is **fully** synced:

```
PGPASSFILE=config/pgpass-mainnet scripts/postgresql-setup.sh --createdb # result "All good!"
```

## Restore database from Snapshot

**NOTE:** If you do not want to restore from snapshot but sync the full database, skip this section.

**NOTE:** Restoring from snapshot drops the existing database!

**Suggestion:** Run this section in the tmux window for cardano-db-sync.

Download the snapshot from [here](#) (.tgz) in the cardano-db-sync folder. Do not unzip.

Create empty folders ledger-state/mainnet in cardano-db-sync folder.

```
wget https://update-cardano-mainnet.iohk.io/cardano-db-sync/13/db-sync-snapshot-schema-13-block-7519843-x86_64.tgz
mkdir ledger-state/mainnet
```

Now restore from snapshot. This will tar the zipped folder in /tmp. The snapshot used has 81 GB size.

```
PGPASSFILE=config/pgpass-mainnet scripts/postgresql-setup.sh --restore-snapshot \
db-sync-snapshot-schema-13-block-7519843-x86_64.tgz ledger-state/mainnet
# warning: online instructions are not 100% accurate.
```

After restoring, continue with syncing with the next section, as normal.

## Run DB-Sync

**NOTE:** DB-Sync can take more than 24 hours to fully sync (w/o snapshot). The following “Postgres Run” step can be executed from the beginning.

Inside the tmux window and the cardano-db-sync folder use the following command:

```
# IMPORTANT NOTE: do not use extended if restoring from snapshot
PGPASSFILE=config/pgpass-mainnet db-sync-node/bin/cardano-db-sync-extended \
--config config/mainnet-config.yaml \
--socket-path ../cardano-node/state-node-mainnet/node.socket \
--state-dir ledger-state/mainnet \
--schema-dir schema/
```

## Run Postgres

Confirm that the *cexplorer* database is created and receives data from DB-Sync.

As the user do:

```
psql cexplorer
\dt # see the tables
```

Example query:

```
select * from epoch limit 2;
```

Go back to \$USER:

```
ctrl+d
```