# **PostgreSQL Installation, Security, Upgrade, High Availability, and Backup/Restore**

This README provides a comprehensive guide to installing PostgreSQL, securing your installation, upgrading PostgreSQL, setting up high availability, and configuring backup/restore procedures.

## **Table of Contents**

1. Installation
2. Security
3. Upgrade
4. High Availability
5. Backup and Restore

## **Installation**

Follow these steps to install PostgreSQL on an Ubuntu system.

### **1. Update the Package List**

bash

Copy

sudo apt update

### **2. Install PostgreSQL**

bash

Copy

sudo apt install postgresql postgresql-contrib

### **3. Verify the Installation**

bash

Copy

sudo systemctl status postgresql

### **4. Switch to the PostgreSQL User**

bash

Copy

sudo -i -u postgres

### **5. Access PostgreSQL**

bash

Copy

psql

## **Security**

Securing your PostgreSQL installation is essential for both performance and data protection.

### **1. Set a Strong Password for the postgres User**

bash

Copy

psql -c "ALTER USER postgres PASSWORD 'yourpassword';"

### **2. Configure pg\_hba.conf for Secure Authentication**

* Open pg\_hba.conf:

bash

Copy

sudo nano /etc/postgresql/12/main/pg\_hba.conf

* Use the following settings for secure local connections:

bash

Copy

local all postgres peer  
host all all 127.0.0.1/32 md5  
host all all ::1/128 md5

### **3. Use SSL for Remote Connections (Optional)**

Edit postgresql.conf to enable SSL:

bash

Copy

sudo nano /etc/postgresql/12/main/postgresql.conf

Add or uncomment the following:

bash

Copy

ssl = on  
ssl\_cert\_file = '/etc/ssl/certs/ssl-cert-snakeoil.pem'  
ssl\_key\_file = '/etc/ssl/private/ssl-cert-snakeoil.key'

## **Upgrade**

Upgrading PostgreSQL is crucial to keep the database secure and performant.

### **1. Upgrade PostgreSQL to a New Version**

* **Add the PostgreSQL repository** (if needed):

bash

Copy

sudo sh -c 'echo "deb <http://apt.postgresql.org/pub/repos/apt/> focal-pgdg main" > /etc/apt/sources.list.d/pgdg.list'  
wget --quiet -O - <https://www.postgresql.org/media/keys/ACCC4CF8.asc> | sudo apt-key add -  
sudo apt update

* **Install the new PostgreSQL version**:

bash

Copy

sudo apt install postgresql-13 postgresql-client-13

* **Upgrade the database** using pg\_upgrade (for major upgrades):

bash

Copy

sudo pg\_upgradecluster 12 main

* **Remove the old PostgreSQL version**:

bash

Copy

sudo apt-get remove postgresql-12

### **2. Verify the Upgrade**

bash

Copy

psql --version

## **High Availability**

High availability (HA) ensures that your PostgreSQL server remains operational even in the event of failure.

### **1. Configure Streaming Replication**

* **Set up Primary Server**:
  + Edit postgresql.conf:

bash

Copy

sudo nano /etc/postgresql/12/main/postgresql.conf

Add the following:

bash

Copy

wal\_level = replica  
max\_wal\_senders = 3  
hot\_standby = on

* + Edit pg\_hba.conf to allow replication:

bash

Copy

host replication all 192.168.x.x/32 md5

* **Set up Standby Server**:
  + Take a base backup from the primary:

bash

Copy

pg\_basebackup -h <primary\_ip> -D /var/lib/postgresql/12/main -U replication -P --wal-method=stream

* + Edit postgresql.conf on the standby:

bash

Copy

primary\_conninfo = 'host=<primary\_ip> port=5432 user=replication password=yourpassword'

* + Start the PostgreSQL server on the standby:

bash

Copy

sudo systemctl start postgresql

## **Backup and Restore**

Backing up and restoring PostgreSQL databases is critical for disaster recovery and system maintenance.

### **1. Backup the Database**

* **Full Database Backup**:

bash

Copy

pg\_dumpall > all\_databases\_backup.sql

* **Single Database Backup**:

bash

Copy

pg\_dump mydatabase > mydatabase\_backup.sql

* **Backup Using pg\_basebackup**:

bash

Copy

pg\_basebackup -D /path/to/backup -Ft -z -P

### **2. Restore the Database**

* **Restore from SQL Backup**:

bash

Copy

psql -f all\_databases\_backup.sql postgres

* **Restore from pg\_basebackup**:

Copy the backup files back to the data directory and start PostgreSQL.

bash

Copy

pg\_ctl -D /path/to/data\_directory start

### **3. Automate Backups**

You can automate backups using cron jobs. For example, to back up your databases every day at 2am, add the following to the crontab:

bash

Copy

crontab -e

bash

Copy

0 2 \* \* \* pg\_dumpall > /path/to/backup/all\_databases\_$(date +\%F).sql

## **Conclusion**

This guide provides essential steps for installing, securing, upgrading, and maintaining a PostgreSQL server. Regular backups, high availability setups, and security configurations are crucial to ensure your database remains robust, secure, and always available.