

# PostgreSQL

## Installation

Exemple sous debian :

```
apt install postgresql
```

## Configuration

Dans `/etc/postgresql/<XX>/main/pg_hba.conf` ajout de :

```
host    all all    127.0.0.1/32    password
```

Redémarrage de postgresql

```
systemctl restart postgresql.service
```

## Création de la base

```
su postgres
psql
create database esupsignature;
create USER esupsignature with password 'esup';
grant ALL ON DATABASE esupsignature to esupsignature;
# Chgt du owner requis à partir de postgresql 15 -> permission création des tables
ALTER DATABASE esupsignature OWNER TO esupsignature;
```

## Ajustements

L'application a été développée avec la fonction de lecture/écriture des blobs dans une transaction par streaming de PostgreSQL. CF <https://www.postgresql.org/docs/11/static/lo.html>

Pour une bonne gestion des blob de cette application, il faut ajouter dans PostgreSQL un trigger sur la base de données sur la table `big_file`. La fonction `lo_manage` est nécessaire ici.

Sous debian :

```
apt-get install postgresql-contrib
```

Puis la création de l'extension `lo` se fait via un super-user:

```
su postgres
psql
\c esupsignature
CREATE EXTENSION lo;
```

Pour vérifier que l'extension est bien installée (toujours dans `psql`)

```
esupsignature=# \dx
```

Retourne :

#### Liste des extensions installées

Nom	Version	Schéma	Description
lo	1.1	public	Large Object maintenance
plpgsql	1.0	pg_catalog	PL/pgSQL procedural language

(2 lignes)



Après cette étape, il faut passer directement à la phase [Configuration](#). La suite se passera après le premier démarrage de l'application

Enfin, **seulement après avoir démarré une première fois esup-signature**, ajout du trigger à l'aide du script suivant :

```
CREATE TRIGGER t_big_file BEFORE UPDATE OR DELETE ON big_file FOR EACH ROW EXECUTE PROCEDURE lo_manage  
(binary_file);
```