# **Setting up WINTER-WINN database replication:**

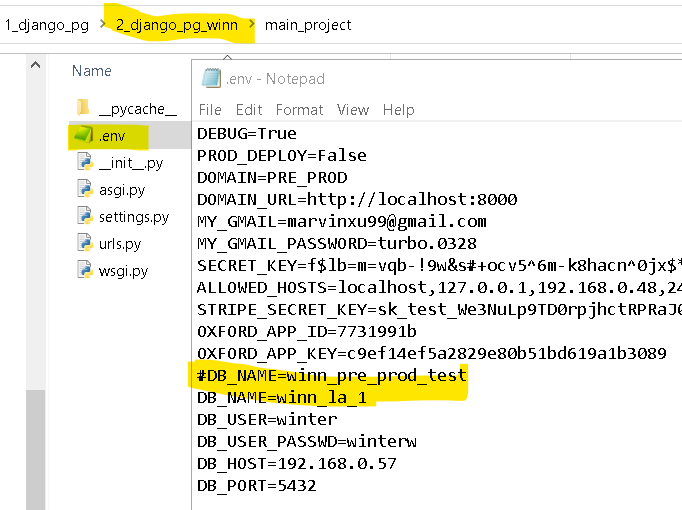
**DB\_MASTER\_IP**: 192.680.48 (Ubuntu 20.4/pg13,

**DB\_REPLICA\_IP**: 192.680.57 (Raspberry Pi4, Bullseye/pg13),

## **1. Ensure the database on the replica machine is the same as the master machine. If not, use the following method to copy from master to replica.**

Method #1

Temporary change the DB settings pointing to the REPLICA machine. Then migrate



Method #2 – dump file copying

### **The most simple case is dumping and restoring on the same server:**

$ Sudo su - postgres

$ pg\_dump -h **localhost** -Fc test > /home/postgres/dump.sql

$ pg\_restore -h **localhost** test < /home/postgres/dump.sql

### **Or with a plain text dump:**

$ pg\_dump -h **localhost** -f /home/postgres/dump.sql test

$ psql -h **localhost** -f /home/postgres/dump.sql test

1. Dump database winn\_la\_1 on DB\_MASTER\_IP to file winn\_la\_1.sql

$pg\_dump -U postgres -h localhost -f winn\_la\_1.sql winn\_la\_1

1. **Copy the file winn\_la\_1.sql to DB\_REPLICA\_IP machine**
2. **On the DB\_REPLICA\_IP machine**

**$sudo -u postgres psql**

**#CREATE DATABASE winn\_la\_1;**

**# \q**$psql -h localhost -f winn\_la\_1.sql winn\_la\_1

$ *# dump a remote database to your local machine*

$ pg\_dump -h **remotedb.mydomain.com** -f /home/postgres/dump.sql test

$ *# dump a local database and write to a remote machine*

$ pg\_dump -h **remotedb.mydomain.com** test | ssh postgres@**remotedb.mydomain.com** 'cat > dump.sql'

$ *# dump a remote database and write to the same remote machine*

$ pg\_dump -h **remotedb.mydomain.com** test | ssh postgres@**remotedb.mydomain.com** 'cat > dump.sql'

$ *# or a different remote machine*

$ pg\_dump -h **remotedb1.mydomain.com** test | ssh postgres@**remotedb2.mydomain.com** 'cat > dump.sql'

## **Setting up publication on the Master machine**

**WAL\_LEVEL** need to be set to **logical** before you can create publication.

**Set wal-level = logical in /etc/postgresql/13/main/postgresql.conf**

select \* from pg\_settings where name ='wal\_level'

$sudo -u postgres psql

# \c winn\_la\_1

# CREATE PUBLICATION winn\_publication\_all FOR ALL TABLES;

1. **Creating a subscription on the Replica machine**

$ sudo -u postgres psql

# \c winn\_la\_1

# CREATE SUBSCRIPTION pi\_subscription CONNECTION 'host=192.168.0.48 port=5432 password=my\_password user=sammy dbname=winn\_la\_1' PUBLICATION winn\_publication\_all;

1. On the replica machine (raspberry Pi), check if it is working by using:

$sudo tail -f /var/log/postgresql/postgresql-13-main.log

To check if the **replica** has connected to the **primary** and the **primary** is streaming, connect to the **primary** database cluster by running:

$ sudo -u postgres psql

# SELECT client\_addr, state FROM pg\_stat\_replication;

Ensure winter has the role of REPLCATION:

**ALTER USER <user> REPLICATION;**

**SELECT \* FROM pg\_roles;**

Use pgadmin4 to configure

DROP a subscription:

alter subscription <sub1> disable;

alter subscription <sub1> set (slot\_name = none);

drop subscription <sub1>;

postgres> SHOW wal\_level