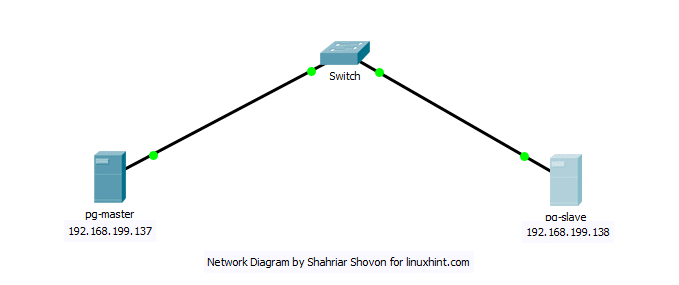
**PostgreSQL Replication**

### Install the postgreSQL on the both servers make sure both server have same verison of the postgresql

**Setting Up the Master PostgreSQL Server:**

## **Network Diagram:**

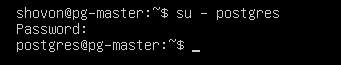
This is the network diagram for the PostgreSQL ****Master****/****Slave**** replication setup. Here I have two servers, ****pg-master**** is the ****Master**** PostgreSQL server and ****pg-slave**** is the ****Slave**** PostgreSQL server. Of course, you can have more ****Slave****server, but for the sake of simplicity I am going to have one ****Slave**** server.



My ****pg-master**** PostgreSQL ****Master**** server has the IP address ****192.168.199.137**** and the ****pg-slave****PostgreSQL ****Slave**** server has the IP address ****192.168.199.138****. Remember these and make changes where necessary for your setup.

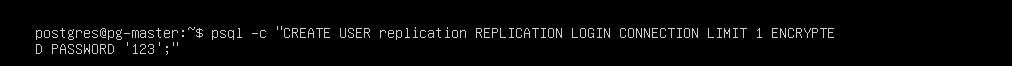
Now login as the ****postgres**** user:

*$*su - postgres



Now create a new user ****replication****:

$ psql -c "CREATE USER replication REPLICATION LOGIN CONNECTION LIMIT 1 ENCRYPTED  
PASSWORD 'YOUR\_PASSWORD';"



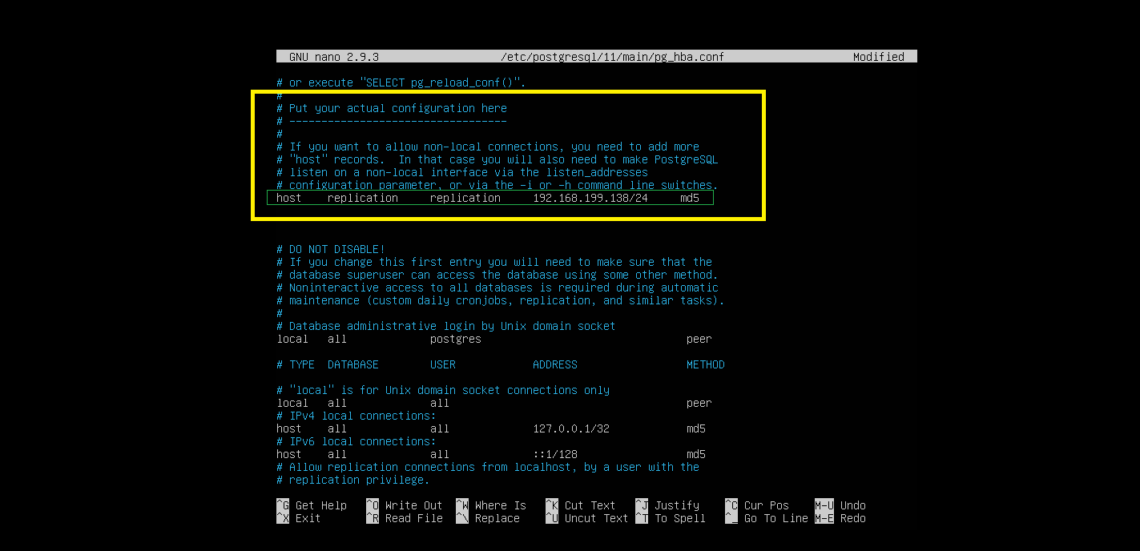
Now open ****/etc/postgresql/11/main/pg\_hba.conf**** with ****nano****:

$ nano /etc/postgresql/11/main/pg\_hba.conf

IMG_258

Add the following line to the marked location:

host    replication     replication   192.168.199.138/24   md5



Now open the main PostgreSQL configuration file with ****nano****:

$ nano /etc/postgresql/11/main/postgresql.conf

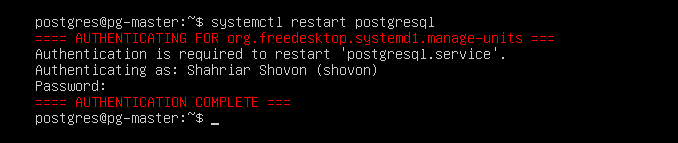
IMG_260

Now find and change the following settings. If any line is commented out, uncomment it (removing #) as necessary.

listen\_addresses = 'localhost,192.168.199.137'  
wal\_level = replica  
max\_wal\_senders = 10  
wal\_keep\_segments = 64

Now restart PostgreSQL server on your ****pg-master****server:

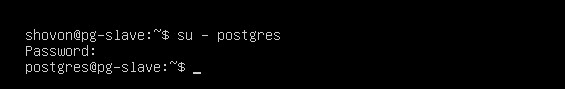
$ systemctl restart postgresql



### **Configuring the Slave Server:**

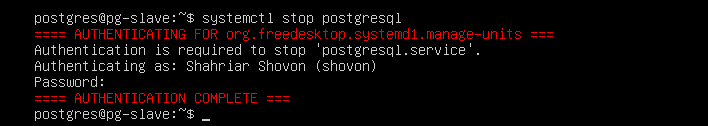
On the ****pg-slave**** server login as ****postgres**** user:

*$*su - postgres



Stop the PostgreSQL service on the ****pg-slave**** server:

$ systemctl stop postgresql



Now open ****/etc/postgresql/11/main/pg\_hba.conf**** with ****nano****:

$ nano /etc/postgresql/11/main/pg\_hba.conf

IMG_258

Add the following line as you did on the ****pg-master**** server:

host    replication     replication     192.168.199.137/24   md5



Now open the main PostgreSQL configuration file with ****nano****:

$ nano /etc/postgresql/11/main/postgresql.conf

IMG_260

Now find and change the following settings. If any line is commented out, uncomment it (removing #) as necessary.

listen\_addresses = 'localhost,192.168.199.138'  
wal\_level = replica  
max\_wal\_senders = 10  
wal\_keep\_segments = 64  
hot\_standby = on

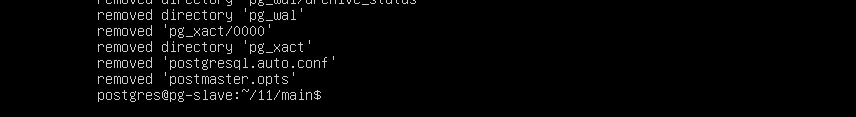
Now go to your ****data\_directory****:

$ cd /var/lib/postgresql/11/main



Remove everything from that directory:

$ rm -rfv \*

IMG_262 

Now copy the data from the ****pg-master**** server to the ****pg-slave**** server’s ****data\_directory****:

$ pg\_basebackup -h 192.168.199.137 -D /var/lib/postgresql/11/main/ -P -U  
replication *--wal-method=fetch*

IMG_264

Type in the password for the ****postgres**** user of the ****pg-master**** server and press ****<Enter>****.



Now create a ****recovery.conf**** file in the ****data\_directory**** with nano:

*$*nano recovery.conf

IMG_266

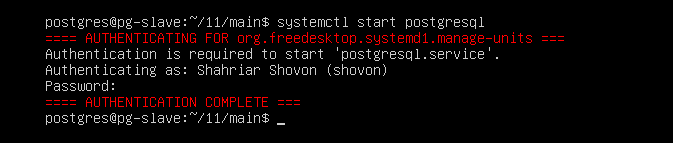
Now add the following line to it:

standby\_mode          = 'on'  
primary\_conninfo      = 'host=192.168.199.137 port=5432 user=replication password=123'  
trigger\_file = '/tmp/MasterNow'



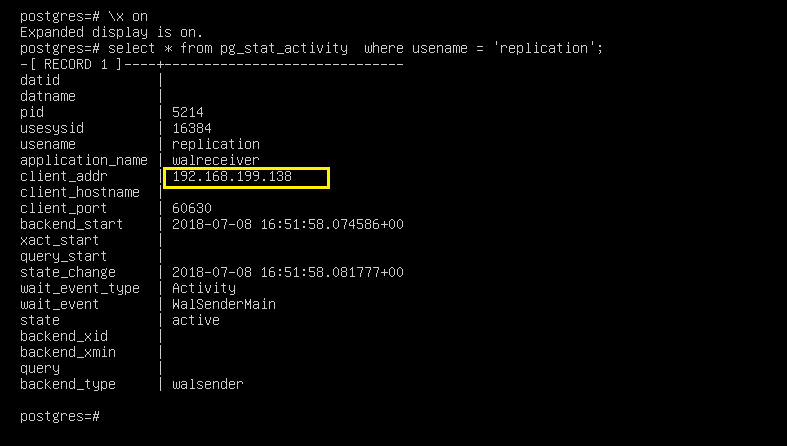
Start the PostgreSQL ****Slave**** server:

$ systemctl start postgresql



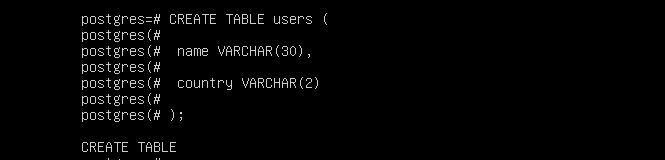
### **Testing Replication:**

Now on the ****pg-master**** server, you can see that the ****Slave**** server is detected.



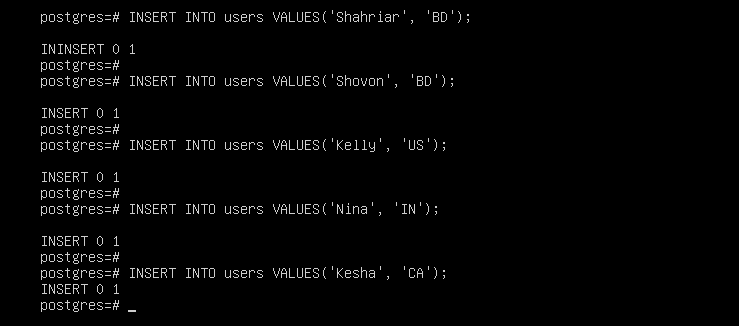
SQL command for creating ****users**** table:

CREATE TABLE users (  
name VARCHAR(30),  
country VARCHAR(2)  
);



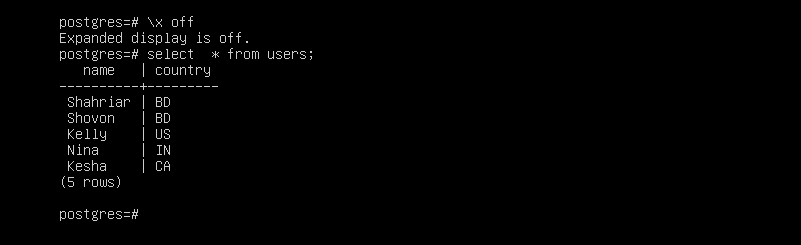
SQL commands to insert dummy data into the ****users**** table:

INSERT INTO users VALUES('Shahriar', 'BD');  
INSERT INTO users VALUES('Shovon', 'BD');  
INSERT INTO users VALUES('Kelly', 'US');  
INSERT INTO users VALUES('Nina', 'IN');  
INSERT INTO users VALUES('Kesha', 'CA');



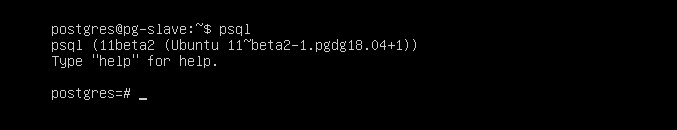
As you can see, the data is correctly added to the ****Master**** server ****pg-master****:

# \x off  
# select \* from users;



Now from the ****Slave**** server ****pg-slave****, login to the PostgreSQL console:

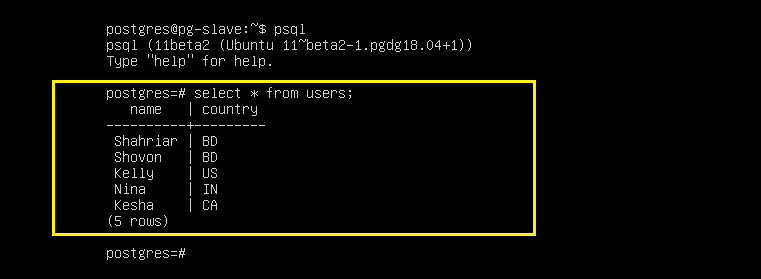
$ psql



Now try to select the data we just added:

$ select \* from users;

As you can see the data is displayed in the ****Slave**** server. It means replication is working perfectly.



Refrence link : <https://linuxhint.com/setup_postgresql_replication/>

<https://narasimmantech.com/setting-up-postgresql-streaming-replication-for-high-availability-and-backup/>

<https://www.cherryservers.com/blog/how-to-set-up-postgresql-database-replication>