

Configure and streaming replication EDB PostgreSQL 11 on Centos 7

Pre-Requisite:

1. Create Two Server. One is Master node and another for Worker node.
2. Stop & disable Firewall and Selinux
3. Yum update

Install EDB on PostgreSQL 11 for Master & Worker both server

Install the repository configuration

yum -y install <https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.rpm>

Replace 'USERNAME: PASSWORD' below with your username and password for the EDB repositories

Visit <https://www.enterprisedb.com/accounts/profile> to get your username and password

sed -i "s@<username>:<password>@Usernaem:Password@" /etc/yum.repos.d/edb.repo

sed -i "s@<username>:<password>@Bractech:caDnZ4qQvEGwvvBV@" /etc/yum.repos.d/edb.repo

Install EPEL repository

yum -y install <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

Install selected packages

yum -y install edb-as11-server

Initialize Database cluster

PGSETUP_INITDB_OPTIONS="-E UTF-8" /usr/edb/as11/bin/edb-as-11-setup initdb

Start Database cluster

systemctl start edb-as-11

systemctl enable edb-as-11

systemctl status edb-as-11

Connect to the database server

sudo su – enterprisedb

psql -p 5444 -d postgres

Change your user default password

```
ALTER USER enterprisedb WITH PASSWORD 'Set_passwd_here';
```

Exit

Change Config File

```
cd /var/lib/edb/as11/data/
```

```
vim pg_hba.conf
```

or

```
vi /var/lib/edb/as11/data/pg_hba.conf
```

Add the following line to the marked location:

"local" is for Unix domain socket connections only

```
local    all         all                                peer
```

```
host     all         all          0.0.0.0/0          trust
```

IPv4 local connections:

```
host     all         all          127.0.0.1/32        ident
```

```
host     all         all          [Host machine IP]   md5
```

Start/Stop/Reload edb-as-11 service

```
systemctl reload edb-as-11
```

```
systemctl restart edb-as-11
```

```
systemctl status edb-as-11
```

Same procedure configures on Worker Node server.

After that Configure PostgreSQL on Master Server Node -

Create password for replication user

```
su - enterprisedb
```

```
[root@edb-master ~]# psql -p 5444 -d postgres
```

```
postgres=# CREATE USER replication REPLICATION LOGIN CONNECTION LIMIT 1 ENCRYPTED PASSWORD '1234';
```

Exit -> exit

Then go to pg_hba.conf file set the worker node ip as a replication Ip.

```
#cd /var/lib/edb/as11/data/
```

```
# vim pg_hba.conf
```

```
# TYPE DATABASE      USER      ADDRESS      METHOD
```

```
# "local" is for Unix domain socket connections only
```

```
local    all                all                                peer
```

```
host     all                all                0.0.0.0/0          trust
```

```
# IPv4 local connections:
```

```
host     all                all                127.0.0.1/32       ident
```

```
host     all                all                machine_host_ip     md5
```

```
# IPv6 local connections:
```

```
host     all                all                ::1/128             ident
```

```
# Allow replication connections from localhost, by a user with the
```

```
# Replication privilege.
```

```
local     replication        all                                peer
```

```
host     replication        all                127.0.0.1/32       ident
```

```
host     replication        all                ::1/128             ident
```

```
host     replication        all                slave_host_ip       md5
```

Save & exit

Edit postgresql.conf file and Uncomment

```
# vim postgresql.conf
```

```
listen_addresses = 'localhost, host_IP'
```

```
port = 5444
```

```
wal_level = replica
```

```
max_wal_senders = 10
```

```
wal_keep_segments = 0
```

```
hot_standby = on
```

Save and exit

```
systemctl restart edb-as-11
```

Go to Worker node -

```
cd /var/lib/edb/as11/data/
```

Remove everything from that directory: If you want before deleting the data directory you can copy this directory using another directory name.

```
cp -R /var/lib/edb/as11/data/ /var/lib/edb/as11/data-orig
```

```
rm -rfv *
```

***Now copy the data from the pg-master server to the pg-slave server's data_directory:

```
/usr/edb/as11/bin/pg_basebackup -h master_ip -D /var/lib/edb/as11/data/ -P -U replication --wal-method=fetch
```

```
chown enterprisedb. * -R
```

*** If the `pg_basebackup` command is not found, Please go to the `pg_basebackup` file location.

My PC `pg_basebackup` file directory location: Other way set environment path variable

```
cd /usr/edb/as11/bin/ (2nd Option If 1st command not working)
```

(Enter user password)

For replication change the Slave Server conf file, You can login as an enterprisedb user:

```
[root@edb-slave ~]# su - enterprisedb
```

```
[root@edb-slave ~]# systemctl stop edb-as-11
```

Now edit vim postgresql.conf

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

```
listen_addresses = '*'
```

```
wal_level = replica
```

```
max_wal_senders = 10
```

```
wal_keep_segments = 64
```

```
hot_standby = on
```

Save & exit

***Now create a recovery.conf file in the data_directory -**

```
cd /var/lib/edb/as11/data/
```

```
vim recovery.conf
```

```
standby_mode = on
```

```
primary_conninfo = 'host=172.16.163.130 (masterip)port=5444 user=replication password=1234  
application_name=slave1'
```

```
recovery_target_timeline = 'latest'
```

```
restore_command = 'scp enterprisedb@172.16.163.130:/var/lib/edb/edb-logs/archive/ %f %p'
```

Save & Exit

Now change Master Node postgresql.conf file

```
# cd /var/lib/edb/as11/data
```

```
# vim postgresql.conf
```

```
archive_mode = on          # enables archiving; off, on, or always
```

```
# (change requires restart)
```

```
archive_command = 'cp %p /var/lib/edb/edb-logs/archive/%f'
```

Restart both master and worker node server

```
systemctl restart edb-as-11
```

Generating SSH Keys:

#For Master

```
su -u enterprisedb
```

```
$ ssh-keygen
```

Keep pressing <Enter>. SSH key should be generated.

From the pg-master server, copy your SSH key to the pg-slave server:

```
$ ssh-copy-id enterprisedb@192.168.25.132
```

From the pg-slave server, copy your SSH key to the pg-master server:

For Check Slave server ssh-key

```
ssh enterprisedb@192.168.25.132
```

Doing same on Worker node Server.

Testing Replication:

Create Database

```
postgres# create database devtest;
```

Connect devtest database

```
postgres# \c devtest;
```

```
CREATE TABLE users (
```

```
name VARCHAR(30),
```

```
country VARCHAR(2)
```

```
);
```

SQL commands to insert dummy data into the user's 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('Keshu', 'CA');
```

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

```
postgres#\x off
```

```
select * from users;
```

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

```
su - enterprisedb
```

```
psql -p 5444 -d postgres
```

Now try to select the data we just added:

```
select * from users;
```

Then you check from your worker node that all the created database also seen from this node. Then configuration is successfully completed.