Dec 31, 2021

JPC HA detailed install guide

Simple JChem PostgreSQL Cartridge highly available setup. It does not contain any specific configuration related to database size (e.g. number of structures), or performance tuning. Use it as a starting point for highly available setup followed by custom settings.

Table of contents:

Create highly available PostgreSQL using binary replication

Install postgres on both servers:

Install postgreSQL 12 database engine as root:

```
wget --quiet -O -
https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo
apt-key add -
```

```
echo "deb http://apt.postgresql.org/pub/repos/apt/
`lsb_release -cs`-pgdg main" |sudo tee
/etc/apt/sources.list.d/pgdg.list
```

```
apt update
```

```
apt -y install postgresql-12 postgresql-client-12
```

On a master machine:

As root.
Modify postgresql.conf to listen on all addresses:

```
listen_addresses = '*'
```

Create *rep* user in master db:

```
CREATE USER rep REPLICATION PASSWORD 'reppass';
```

Edit pg_hba file on master machine:

```
host replication rep 10.0.112.58/32 md5
```

Restart postgres on master machine:

```
systemctl restart postgresql.service
```

Optional: Make sure that slave can access postgresql open port or turn off the firewall (as root):

```
systemctl disable ufw.service
```

```
systemctl stop ufw.service
```

On a Slave machine:

Stop postgres as root:

```
systemctl stop postgresql.service
```

postgres userdelete old data files:

```
rm -rf /var/lib/postgresql/12/main/*
```

Structure of standby with replica splitter:

```
pg_basebackup -h 10.0.112.49 -D /var/lib/postgresql/12/main
-U rep -P -v -R -X stream -C -S pgstandby01
```

Start postgres:

```
systemctl start postgresql.service
```

Useful commands on replications status

Query statuses as postgres user.
Master:

```
psql -c "SELECT * FROM pg_replication_slots;"
```

```
psql -c "\x" -c "SELECT * FROM pg_stat_replication;"
```

Slave:

```
psql -c "\x" -c "SELECT * FROM pg_stat_wal_receiver;"
```

Install Cartridge

Execute the following steps on both cartridge instances

Install java

```
sudo apt install openjdk-11-jre-headless
```

Download jchem-psqlXXX .deb and install PostgreSQL Cartridge

```
sudo dpkg -i jchem-psql_21.19.0.r13679_amd64.deb
```

Install license file
Copy a valid ChemAxon license to /etc/chemaxon/license.cxl

Add a user that is going to have access to index data
Will use the already created (by jpc install script) jchem-psql user

Add a database that stores the JPC index data to the PostgreSQL database on master only

Since postgresql replication is in place it is needed only on the master

```
sudo su - postgres
```

```
createuser jchem-psql
```

```
createdb jpc -O jchem-psql
```

Create password

```
sudo -u postgres psql
```

```
\password jchem-psql
```

<enter password> in my case: jchem-jpc

Test connection:

```
psql -U jchem-psql -h localhost jpc
```

Configure cartridge on both master and slave

Set the following parameter values in the /etc/chemaxon/jchem-psql.conf

```
com.chemaxon.jchem.psql.scheme=crdb

com.chemaxon.jchem.psql.crdb.sqlBuilderProvider=POSTGRESQL

com.chemaxon.jchem.psql.crdb.jdbcUrl=jdbc:postgresql://localhost:5432/jpc

com.chemaxon.jchem.psql.crdb.user=jchem-psql

com.chemaxon.jchem.psql.crdb.password=jchem-jpc

com.chemaxon.jchem.psql.crdb.hazelcastConfigFile=/etc/chemaxon/hazelcast.xml
```

Change /etc/chemaxon/hazecast.xml file, modify the member list section to the ip4 address of your cartridge instances (in my case it is 10.0.112.49 and 10.0.112.58) .

```
<tcp-ip enabled="true">
<member-list>
<member>10.0.112.49</member>
<member>10.0.112.58</member>
</member-list>
</tcp-ip>
```

Initialize the service on master only

PostgreSQL service should be up and running before initialization of jchem-psql service.
Initialize jchem-psql service:

```
sudo service jchem-psql init
```

Start jchem-psql service on both master and slave:

```
sudo service jchem-psql start
```

Create a new user and a database for cartridge usage

On master
Create ubuntu user in the database as postrgres user

```
sudo su - postgres

createuser ubuntu

psql

\password ubuntu
```

<give the password>

```
\q
```

Create test database (still as postgres user)

```
createdb test -O ubuntu
```

Create extension

```
psql test
```

```
CREATE EXTENSION chemaxon_type;
CREATE EXTENSION hstore;
CREATE EXTENSION chemaxon_framework;
```

From now on we have a highly available JChem PostgreSQL Cartridge where the master node can be used for CRUD operations and the slave node only for read operations.

Security considerations

Make jchem-pqsl the owner of the config file

```
sudo chown jchem-psql /etc/chemaxon/jchem-psql.conf
```

Make it readable only to the owner

```
sudo chmod go-rw /etc/chemaxon/jchem-psql.conf
```

Do the same for hazelcast.xml

```
sudo chown jchem-psql /etc/chemaxon/hazelcast.xml
```

```
sudo chmod go-rw /etc/chemaxon/hazelcast.xml
```

Import 80 test molecules for testing purposes

Get test sdf with 80 molecules

```
wget https://shared.chemaxon.com/users/avolford/test80.sdf
```

Import it to test table:

Connect to test database

```
psql -U ubuntu -h localhost test
```

In test database import the sdf file

```
\set content `cat ~/test80.sdf`
```

Check fields in the sdf file

```
SELECT * FROM parse_sdf(:'content') limit 1;
```

Create ttest table with mol, id, pKa_ac1, pKa_bas1, logD_pH_7_4 columns

```
CREATE TABLE ttest AS
SELECT substring(molSrc,'.*M  END')::molecule('sample') AS mol,
props -> 'ID' AS id,
CAST(props -> 'pKa_ac1' AS FLOAT) AS pKa_ac1,
CAST(props -> 'pKa_bas1' AS FLOAT) AS pKa_bas1,
CAST(props -> 'logD_pH_7.4' AS FLOAT) AS logD_pH_7_4
FROM parse_sdf(:'content');
```

Create index

```
CREATE INDEX ttest_idx ON ttest USING sortedchemindex(mol);
```