Table of Contents

ISSUE-TRACKER INSTALLATIONS AND CONFIGURATION GUIDE

1. POSTRES RELATED INSTALLATIONS AND CONFIGURATIONS

1.1. Configure the Ubuntu repositories

Configure the Ubuntu repositories

```
sudo add-apt-repository "deb http://apt.postgresql.org/pub/repos/apt/ xenial-pgdg main"

sudo apt-get update
sudo apt-get install postgresql-9.6
```

1.2. Add the media keys

Add the media keys as follows:

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

1.3. Install the postgre package with apt

Install the postgre package with apt

```
# update your repos
sudo apt-get update

# install the postgresql binary
sudo apt-get install postgresql postgresql-contrib

# enable postgre
sudo update-rc.d postgresql enable
```

1.4. Change the postgre user password

Configure the Ubuntu repositories

```
sudo passwd postgres
# Type a pw - add to your password manager !!!

# and verify
su - postgres
```

1.4.1. start the postgreSQL

Start the postgreSQL by issuing the following command

```
sudo /etc/init.d/postgresql start
```

1.4.2. Start the psql client as the postgres shell user

Start the psql client as the postgres shell user
source:
http://dba.stackexchange.com/a/54253/1245

```
sudo su - postgres
# start the psql client
psql

# the psql prompt should appear as
# postgres=#

# list the databases
\l
```

```
#and quit
\q
```

1.4.3. Create the pgsql user

Create the pgsql user and grant him the privileges to create dbs and to connect to the postgres db.
You could alternatively configure different way of authenticatio according to the options provided in this stackoverflow answer:
http://stackoverflow.com/a/9736231/65706

```
# create the pgsql user to be the same as the shell
# user you are going to execute the scripts with
sudo su - postgres  -c "psql -c 'CREATE USER '$USER' ;'"

# grant him the priviledges
sudo su - postgres  -c "psql -c 'grant all privileges on database postgres to '$USER' ;'"

# grant him the privilege to create db's
sudo su - postgres  -c "psql -c 'ALTER USER '$USER' CREATEDB;'"

sudo su - postgres  -c 'psql -c "select * from information_schema.role_table_grants
 where grantee='"'"$USER"'"';"'
```

1.4.4. add the uuid generation capability enabling extension

add the uuid generation capability enabling extension

```
sudo su - postgres  -c "psql template1 -c 'CREATE EXTENSION IF NOT EXISTS \"uuid-ossp\";'"

sudo su - postgres  -c "psql template1 -c 'CREATE EXTENSION IF NOT EXISTS \"pgcrypto\";'"
```

1.4.5. Install the dblink extension as follows

Install the dblink extension as follows

```
sudo su - postgres  -c "psql template1 -c 'CREATE EXTENSION IF NOT EXISTS \"dblink\";' "
```

1.5. Install the perl modules

Install the perl module by first installing the server development package

```
# check which server development packages are available
sudo apt-cache search postgres | grep -i server-dev | sort

# install it
sudo apt-get install -y postgresql-server-dev-9.6

# install the DBD::Pg module
sudo perl -MCPAN -e 'install DBD::Pg'

sudo perl -MCPAN -e 'Tie::Hash::DBD'
```

2. FRONT END-INSTALLATIONS

2.1. Install NodeJS on Ubuntu

From the following page:
https://nodejs.org/en/download/package-manager/#debian-and-ubuntu-based-linux-distributions

```
sudo apt-get install -y build-essential
sudo apt-get install -y nodejs
```

2.2. Install npm

Install npm by issuing the following command:

```
sudo apt-get install npm
```

2.3. Install web-pack

Install webpack globally by issueing the following command:

```
sudo npm install -g webpack
```

2.4. Install bower

Install the bower package globally as follows:

```
npm config set prefix /usr/local
npm install -g bower
sudo npm install -g bower
which bower
```

2.5. Install the jcoud

Install the jcloud as follows

3. PHPPGADMIN INSTALLATION AND CONFIGURATION

3.1. Install Apache

Install the apache binaries as follows:

```
# update first your package manager
sudo apt update
# install apache
sudo apt install -y apache2 apache2-utils
systemctl status apache2
sudo systemctl enable apache2
apache2 -v
sudo chown www-data:www-data /var/www/html/ -R
```

3.2. Configure Apache

Configure apache as follows:

```
systemctl status apache2
sudo systemctl enable apache2
apache2 -v
sudo chown ysg:ysg /var/www/html/ -R
```

3.3. Install and configure php

Install and configure php as follows:

```
sudo apt install php7.1 libapache2-mod-php7.1 php7.1-mysql php-common php7.1-cli php7.1-common php7.1-json php7.1-opcache php7.1-
readline
sudo a2enmod php7.1
sudo apt-get install -y php-pgsql
sudo systemctl restart apache2
php --version
sudo vim /var/www/html/info.php
```

3.4. Clone the phppgadmin project

Clone the phppgadmin project with git as follows:

```
git clone git://github.com/phppgadmin/phppgadmin.git
sudo chown ysg:ysg /var/www/html/ -R
```

3.5. Enable pg_hba file

Edit the pg_hba.conf file to allow TCP/IP connections from the phppg application layer.

```
# check the following example:
dat/etc/postgresql/9.6/main/pg_hba.conf
```

3.6. Copy the phppgadmin dist conf file

As follows:

```
cd /var/www/html/phppgadmin/
sudo cp -v conf/config.inc.php-dist conf/config.inc.php


sudo systemctl restart apache2
```

3.7. Test the connection

Navigate to the following url:
http://192.168.56.120/phppgadmin/
Use the credentials of the ysg (your postgres user)