Install and Configure the Database VM

Prerequisites

- Postgres 9.4
- PostGIS 2.1

Table of Contents

Configure PostgreSQL

Edit PosgreSQL Configuration for Application Server Use

1. Edit the ***postgresql.conf*** file to allow connections to the database:

```
# As root user or postgres user, open the postgresql.conf file with your favorite
editor
# use command sudo su postgres to change to postgres user
 > vi /etc/postgresql/9.4/main/postgresql.conf
```

2. Locate the section labeled "CONNECTIONS AND AUTHENTICATION," and edit the original to match the edited value below:

   Original:

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------
# - Connection Settings -
#listen_addresses = 'localhost'         # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost', '*' = all
                                        # (change requires restart)
#port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)


...

ssl = true                              # (change requires restart)
```

   Edited: 1) Set listen_addresses to '*', 2) make sure port is not commented out and, 3) comment out the ssl line. Refer to the following example:

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------
# - Connection Settings -
listen_addresses = '*'               # what IP address(es) to listen on;
                                     # comma-separated list of addresses;
                                     # defaults to 'localhost', '*' = all
                                     # (change requires restart)
port = 5432                          # (change requires restart)
max_connections = 100                # (change requires restart)
...
#ssl = true                          # (change requires restart)
```

3. After making configuration changes in this file, you'll need to restart the PostgreSQL server using the following command.

   ```
   > /etc/init.d/postgresql restart
   ```

   The following message will verify that the restart has successfully completed:

   ```
   * Restarting PostgreSQL 9.4 database server
   [ OK ]
   ```

4. Ensure PostgreSQL restarted successfully. Several parameter changes may fail due to the VM's kernel resources not being configured to support the PostgreSQL settings.

   ```
   $ service postgresql status
   9.4/main (port 5432): online
   ```

Configure Database Users

Configure the Postgres User

The postgres user should be set up as a nologin user.

1. Edit the ***pg_hba.conf*** (host-based authentication) file:

   ```
   # As the postgres user, open the pg_hba.conf file:
   # If you are not still the postgres user from step 1
   > sudo su postgres
   > vi /etc/postgresql/9.4/main/pg_hba.conf
   ```

2. Ensure the postgres user's authentication is set to 'peer'

```
# Database administrative login by Unix domain socket
local   all             postgres                                peer

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
 # IPv4 local connections:
host    all             all             127.0.0.1/32            md5
```

Configure the DCDS User

1. Creae the dcds user in the database. Be sure to replace YOURPASSWORD with an appropriate password of your choosing.

```
> sudo  su postgres
> psql -c "CREATE USER dcds with password 'YOURPASSWORD'"    #creates new user
dcds
```

2. Repeat step 2 in the previous section for adding the dcds user with the following line. Put it below the one for the postgres user, but before the "local all all peer" line

```
local   all             postgres                            peer
host    all             dcds        all                     md5
```

3. Restart the postgresql service to pick up the changes for the dcds user

```
# As postgres user:
> service postgresql restart
```

Verify the PostGIS Extension Installation

Test for the postgis extension, using the following commands:

```
> sudo  su postgres
> psql -c "SELECT name, default_version, installed_version FROM
pg_available_extensions WHERE name LIKE 'postgis%' or name LIKE 'address%';"
```

You should see a response to the following, most importantly, the 'postgis' entry:

```
name             | default_version | installed_version
------------------------------+-----------------+-------------------
 postgis_topology             | 2.2.2           |
 postgis                      | 2.2.2           | 2.2.2
 postgis_tiger_geocoder       | 2.2.2           |
 address_standardizer_data_us | 2.2.2           |
 address_standardizer         | 2.2.2           |
(5 rows)
```

At this point it is not clear whether the "address" entries are important. As long as the postgis entries are in the table, we know postgis is installed.

Configure the DCDS Database

For more information on the scripts below, see the **dcds-db/README.txt** file

1. Create the DCDS database:

    ```
    > sudo su postgres
    # Common database name is dcds, dbuser can also be dcds
     > ./create_db.sh <dbname> <dbuser>
    > ./create_data_dbs.sh <dbname>
    ```

2. Run all the sql scripts in the **dcds-db/changes** directory, if they exist

    ```
    > sudo  su - -s /bin/bash postgres
    > psql -f XXXX_Changes.sql <dbname>
      .....
    ```

3. From the directory **/dcds-db/scripts** :

```
> sudo  su postgres
# The 1 is the workspaceid parameter, and is assumed this is the first workspace
being created in this database
# Common workspace names include Production and Training
> ./create_workspace.sh <dbname> 1 <workspacename>

# The systemid is assumed to be the first system entry being created in this
database, so should be 1. The workspaceid should be the id of the workspace you
just created.
> ./create_system.sh <dbname> <hostname> <description> <systemid> <workspaceid>

# You can look up the orgtypeid choices in the orgtype table in baseline_data.sql
> ./create_org.sh <dbname> <orgName> <orgCounty> <orgState> <orgPrefix>
<orgTypeId>

# The last two parameters, orgid and workspaceid, assume you're using the first
org and workspace entries created in the above steps
> ./create_default_user.sh <dbname> <your email> 1 1

# Creates the Maps, Data, Weather, Tracking and Upload folders. workspaceid
should be the id you used for the create_workspace.sh script above
> ./create_default_folders.sh <dbname> <workspaceid>
```

NOTE: After creating a user using the "create_default_user" script, the user needs to be added to OpenAM and enabled.

4. Create the Weather Maps, from the directory **/dcds-db**:

```
> sudo  su - -s /bin/bash postgres
> cd datalayers/weather
> ./weather_layers.sh <dbname> <workspaceid>
```

5. Create the Base Maps, from the directory **/dcds-db**:

```
> sudo  su - -s /bin/bash postgres
> cd datalayers/maps
> ./maps_layers.sh <dbname>
```