Database VM

Configure and Install NICS Database:

Prerequisites

- Postgres 9.4
- Postgis 2.1

Configure PostgreSQL

Edit PostgreSQL configuration for Application Server Use

a. Edit the postgresql.conf file to allow external 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
```

b. 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: Set listen_addresses to '*', make sure port is not commented out and comment out the ssl line ,

```
#--------------------------------------------------------------------------
----
# 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)
```

c. After most configuration changes in this file, you'll need to restart the PostgreSQL server.

```
> /etc/init.d/postgresql restart
 * Restarting PostgreSQL 9.4 database server
[ OK ]
```

d. 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 Postgres Users

Configure the Postgres User

The postgres user should have been set up as a nologin user.

a. 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
```

b. 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 NICS User

a. Create nics user in the database. Be sure to replace YOURPASSWORD with a proper password of your choosing.

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

b. Repeat step 2 in the previous section for adding the nics 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             nics        all                     md5
```

c. Restart the postgresql service to pick up the changes for the nics user

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

Ensure you have successfully installed the postgis extension

1. Test for the postgis extension:

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

a. You should see a response similar 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 NICS Database

a. For more information on the scripts below, see the nics-db/README.txt file
b. Create the NICS database : /nics-db

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

c. ./nics-db/scripts : **NOTE:** After creating a user using the "create_default_user" script, the user needs to be added to OpenAM and enabled.

```
> 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> <workspacename> 1

# The 1 is the systemid, and is assumed to be the first system entry being
created in this database
> ./create_system.sh <dbname> <hostname> <description> 1

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

d. Run all the sql scripts in the nics-db/changes directory, if they exist

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

e. Create the Weather Maps : /nics-db

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

f. Create the Base Maps : /nics-db

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