Connect to the PostgreSQL database and create a new user:

Source:<https://connect.boundlessgeo.com/docs/suite/4.8/dataadmin/pgGettingStarted/firstconnect.html>

Connecting as the Postgres default user by using either two following commands:

psql -U postgres -W

OR

sudo -u postgres psql postgres

Then create a new user using the following command:

CREATE ROLE bsamani WITH LOGIN PASSWORD 'Pass!@#$1234';

Next Create a database:

CREATE DATABASE mmgisdb;

Then grant all the privileges on the database to the user:

GRANT ALL PRIVILEGES ON DATABASE mmgisdb TO bsamani;

Or using this:

ALTER ROLE bsamani CREATEDB;

ALTER ROLE bsamani REPLICATION;

ALTER ROLE bsamani SUPERUSER;

ALTER ROLE bsamani CREATEROLE;

Then verify all these changes by checking the user table or other tables:

SELECT \* FROM pg\_user;

usename | usesysid | usecreatedb | usesuper | usecatupd | userepl | passwd | valuntil | useconfig

----------+----------+-------------+----------+-----------+---------+----------+----------+-----------

postgres | 10 | t | t | t | t | \*\*\*\*\*\*\*\* | |

bsamani | 16384 | t | t | t | t | \*\*\*\*\*\*\*\* | |

(2 rows)

To verify that a user has a valid password:

SELECT \* FROM pg\_shadow;

usename | usesysid | usecreatedb | usesuper | usecatupd | userepl | passwd | valuntil | useconfig

----------+----------+-------------+----------+-----------+---------+-------------------------------------+----------+-----------

postgres | 10 | t | t | t | t | | |

bsamani | 16384 | t | t | t | t | md56789114ba5a60c90b40e2f058142b731 | |

(2 rows)

Most commonly error:

<!DOCTYPE html><html><head><title></title><link rel="stylesheet" href="/stylesheets/style.css"></head><body><h1>database &quot;bsamani&quot; does not exist</h1><h2></h2><pre>error: database &quot;bsamani&quot; does not exist

at Connection.parseE (/var/www/html/mmgis/API/node\_modules/pg/lib/connection.js:553:11)

at Connection.parseMessage (/var/www/html/mmgis/API/node\_modules/pg/lib/connection.js:378:19)

at Socket.&lt;anonymous&gt; (/var/www/html/mmgis/API/node\_modules/pg/lib/connection.js:119:22)

at emitOne (events.js:116:13)

at Socket.emit (events.js:211:7)

at addChunk (\_stream\_readable.js:263:12)

at readableAddChunk (\_stream\_readable.js:250:11)

at Socket.Readable.push (\_stream\_readable.js:208:10)

at TCP.onread (net.js:597:20)</pre></body></html>

Solution:

This issue was coming from the setup for the used database username and password. For security reasons, I was setting up a .env file to store all database credentials and from there read the database username and password for connecting to the database, but for some reasons the pg-promise node package wasn’t able to read that file and was returning to me the above error message on the browser. I’ll search more to see how I can fix this problem.

**PostgreSQL server configuration**

Configuration of postgres config. file which is pg\_hba.conf that is located in

/var/lib/pgsql/data directory on Linux RedHat E7.5.

More about this file in: <https://www.postgresql.org/docs/9.1/static/auth-pg-hba-conf.html>

Peer

* Obtain the client's operating system user name from the operating system and check if it matches the requested database user name. This is only available for local connections.

Ident

* Obtain the operating system user name of the client by contacting the ident server on the client and check if it matches the requested database user name. Ident authentication can only be used on TCP/IP connections. When specified for local connections, peer authentication will be used instead.

md5

* Require the client to supply an MD5-encrypted password for authentication.

Trust

* Allow the connection unconditionally. This method allows anyone that can connect to the PostgreSQL database server to login as any PostgreSQL user they wish, without the need for a password or any other authentication.

# =====================================================================

# 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 ident

# IPv6 local connections:

host all all ::1/128 ident

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

# replication privilege.

#local replication postgres peer

#host replication postgres 127.0.0.1/32 ident

#host replication postgres ::1/128 ident

# ======================================================================

# ====> psql: FATAL: Peer authentication faild for user "postgres"

# ======================================================================

# Change it to ===========================================================

# TYPE DATABASE USER ADDRESS METHOD

# "local" is for Unix domain socket connections only

local all all md5

# IPv4 local connections:

host all all 127.0.0.1/32 ident

# IPv6 local connections:

host all all ::1/128 ident

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

# replication privilege.

#local replication postgres peer

#host replication postgres 127.0.0.1/32 ident

#host replication postgres ::1/128 ident

# ======================================================================

# ====> psql: fe\_sendauth: no password supplied

# ======================================================================

# Change it to ===========================================================

# TYPE DATABASE USER ADDRESS METHOD

# "local" is for Unix domain socket connections only

local all all trust

# IPv4 local connections:

host all all 127.0.0.1/32 ident

# IPv6 local connections:

host all all ::1/128 ident

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

# replication privilege.

#local replication postgres peer

#host replication postgres 127.0.0.1/32 ident

#host replication postgres ::1/128 ident

# ======================================================================

# ====> After running 'psql -U postgres -W' it connects to the database

# ======================================================================

**Install load balancer “PM2” for Node applications**

1. Install PM2
2. Reboot the OS
3. Go to app directory
4. Run pm2 start app.js
5. Run pm2 save
6. Run pm2 startup system
7. Check pm2 status by running systemctl status pm2-<username>