Install pgAdmin 4 on CentOS 7 & Fedora 33/32/31/30

By **Josphat Mutai** - November 11, 2020

pgAdmin is the leading Open Source feature-rich PostgreSQL administration and development platform that runs on Linux, Unix, Mac OS X, and Windows. pgAdmin can be used to manage PostgreSQL 9.2 and above. With the release of pgAdmin 4, there was migration from Bootstrap 3 to Bootstrap 4. In this article we will be installing pgAdmin 4 on CentOS 7 & Fedora 33/32/31/30.

For CentOS 8: How To Install pgAdmin 4 on CentOS 8 Linux

pgAdmin4 on CentOS 7 / Fedora 33/32/31/30 requirements

You need to have PostgreSQL installed on your system before you can install pgAdmin 4. Below are the guides to help you install PostgreSQL:

How to install PostgreSQL 12 on CentOS 7

Install PostgreSQL 13 on CentOS 7

How to install PostgreSQL on Fedora

Install pgAdmin 4 on CentOS 7

After installing PostgreSQL, you can begin the installation of pgAdmin 4 on CentOS 7. You need to add PostgreSQL RPM repository, which should have been done while installing PostgreSQL.

```
sudo yum -y install epel-release
sudo yum -y install
https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-
x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

Then install the pgAdmin package:

```
sudo yum -y update
sudo yum -y install pgadmin4
```

Install pgAdmin Fedora

Add PostgreSQL Yum Repository to your Fedora system by running the below command:

Fedora 33:

```
sudo dnf install
https://download.postgresql.org/pub/repos/yum/reporpms/F-33-
x86_64/pgdg-fedora-repo-latest.noarch.rpm
```

Fedora 32:

```
sudo dnf install
https://download.postgresql.org/pub/repos/yum/reporpms/F-32-
x86_64/pgdg-fedora-repo-latest.noarch.rpm
```

Fedora 31:

```
sudo dnf install
https://download.postgresql.org/pub/repos/yum/reporpms/F-31-
x86_64/pgdg-fedora-repo-latest.noarch.rpm
```

Fedora 30:

```
sudo dnf install
https://download.postgresql.org/pub/repos/yum/reporpms/F-30-
x86_64/pgdg-fedora-repo-latest.noarch.rpm
```

Fedora 29:

```
sudo dnf install
https://download.postgresql.org/pub/repos/yum/reporpms/F-29-
x86_64/pgdg-fedora-repo-latest.noarch.rpm
```

Once the repositories have been added, update system packages then install `pgadmin4`:

```
sudo dnf -y update && sudo dnf -y install pgadmin4
```

Among the dependencies installed is `pgadmin4-web` and `httpd` web server.

Configure pgAdmin 4 on CentOS 7 / Fedora 33/32/31/30

Now that we have pgAdmin 4 installed, let's configure it.

1. Start and enable httpd service to start on boot

```
sudo systemctl start httpd && sudo systemctl enable httpd
```

You can confirm service status by running:

```
sudo systemctl status httpd
```

2. Rename pgAdmin Apache configuration sample

```
sudo cp /etc/httpd/conf.d/pgadmin4.conf.sample
/etc/httpd/conf.d/pgadmin4.conf
```

3. Edit the file to add `VirtualHost` section, it should look like below:

For CentOS 7:

```
<VirtualHost *:80>
ServerName pgadmin.example.com
LoadModule wsgi_module modules/mod_wsgi.so
```

```
WSGIDaemonProcess pgadmin processes=1 threads=25
WSGIScriptAlias /pgadmin4 /usr/lib/python2.7/site-packages/pgadmin4-
web/pgAdmin4.wsgi

<Directory /usr/lib/python2.7/site-packages/pgadmin4-web/>
        WSGIProcessGroup pgadmin
        WSGIApplicationGroup %{GLOBAL}
        <IfModule mod_authz_core.c>
                # Apache 2.4
                Require all granted
        </IfModule>
        <IfModule !mod_authz_core.c>
                # Apache 2.2
                Order Deny,Allow
                Deny from All
                Allow from 127.0.0.1
                Allow from ::1
        </IfModule>
</Directory>
</VirtualHost>
```

For Fedora 31/30/29 – Nothing to change

```
# pgAdmin VirtualHost
LoadModule wsgi_module modules/mod_wsgi.so
WSGIDaemonProcess pgadmin processes=1 threads=25
WSGIScriptAlias /pgadmin4 /usr/lib/python3.7/site-packages/pgadmin4-
web/pgAdmin4.wsgi

<Directory /usr/lib/python3.7/site-packages/pgadmin4-web/>
        WSGIProcessGroup pgadmin
        WSGIApplicationGroup %{GLOBAL}
        <IfModule mod_authz_core.c>
```

```
                # Apache 2.4
                Require all granted
        </IfModule>
        <IfModule !mod_authz_core.c>
                # Apache 2.2
                Order Deny,Allow
                Deny from All
                Allow from 127.0.0.1
                Allow from ::1
        </IfModule>
</Directory>
```

Confirm configuration syntax to prevent any errors and restart httpd service.

```
$ sudo  httpd -t
Syntax OK
$ sudo systemctl restart httpd
```

4. Create pgAdmin data directories:

```
sudo mkdir -p /var/lib/pgadmin4/ /var/log/pgadmin4/
```

5. Edit `config_local.py` and add the following settings. In most cases, the default file locations should be appropriate:

For CentOS 7:

```
sudo vi /usr/lib/python2.7/site-packages/pgadmin4-web/config_distro.py
```

For Fedora 30/29:

```
sudo vi /usr/lib/python3.7/site-packages/pgadmin4-web/config_distro.py
```

Add:

```
LOG_FILE = '/var/log/pgadmin4/pgadmin4.log'
SQLITE_PATH = '/var/lib/pgadmin4/pgadmin4.db'
SESSION_DB_PATH = '/var/lib/pgadmin4/sessions'
STORAGE_DIR = '/var/lib/pgadmin4/storage'
```

6. Run the following command to create the configuration database:

For CentOS 7:

```
sudo python /usr/lib/python2.7/site-packages/pgadmin4-web/setup.py
```

For Fedora 30/29:

```
sudo python3 /usr/lib/python3.7/site-packages/pgadmin4-web/setup.py
```

This will ask you to Enter the `email address` and the `password` to use for the initial pgAdmin user account.

```
Email address: admin@example.com
Password: <INPUT PASSWORD>
Retype password:<Confirm PASSWORD>
pgAdmin 4 - Application Initialisation
==========
```

Set permissions for pgAdmin directories to `apache` user

```
sudo chown -R apache:apache /var/lib/pgadmin4 /var/log/pgadmin4
```

Configure SELinux

If you have SELinux running in enforcing mode, create and apply a policy to allow Apache user access pgAdmin directories.

```
sudo semanage fcontext -a -t httpd_sys_rw_content_t
"/var/lib/pgadmin4(/.*)?"
```

```
sudo semanage fcontext -a -t httpd_sys_rw_content_t
"/var/log/pgadmin4(/.*)?"
sudo restorecon -R /var/lib/pgadmin4/
sudo restorecon -R /var/log/pgadmin4/
```

Restart httpd service.

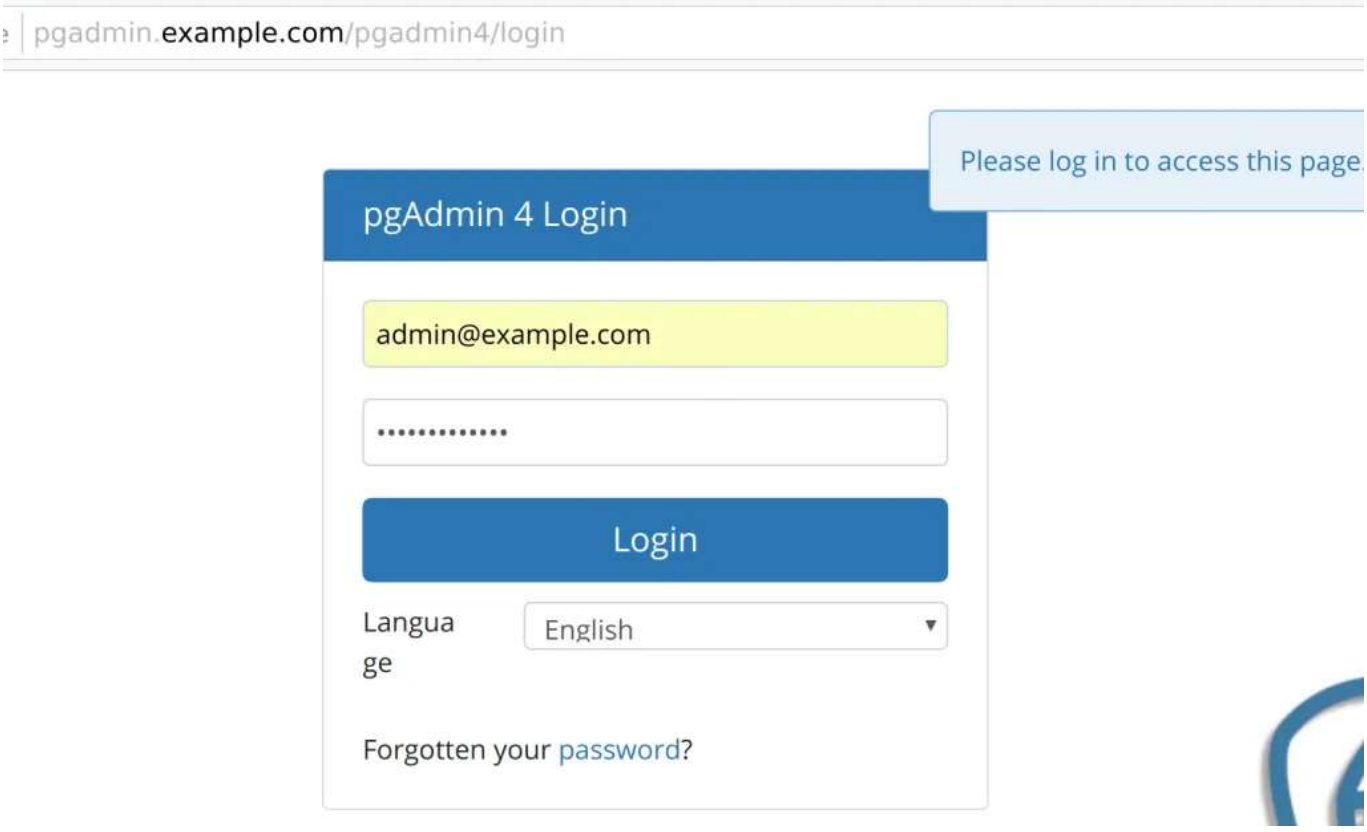
```
sudo systemctl restart httpd
```

Access pgAdmin 4 Web Interface

if you have an active firewall service, allow http port

```
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --reload
```

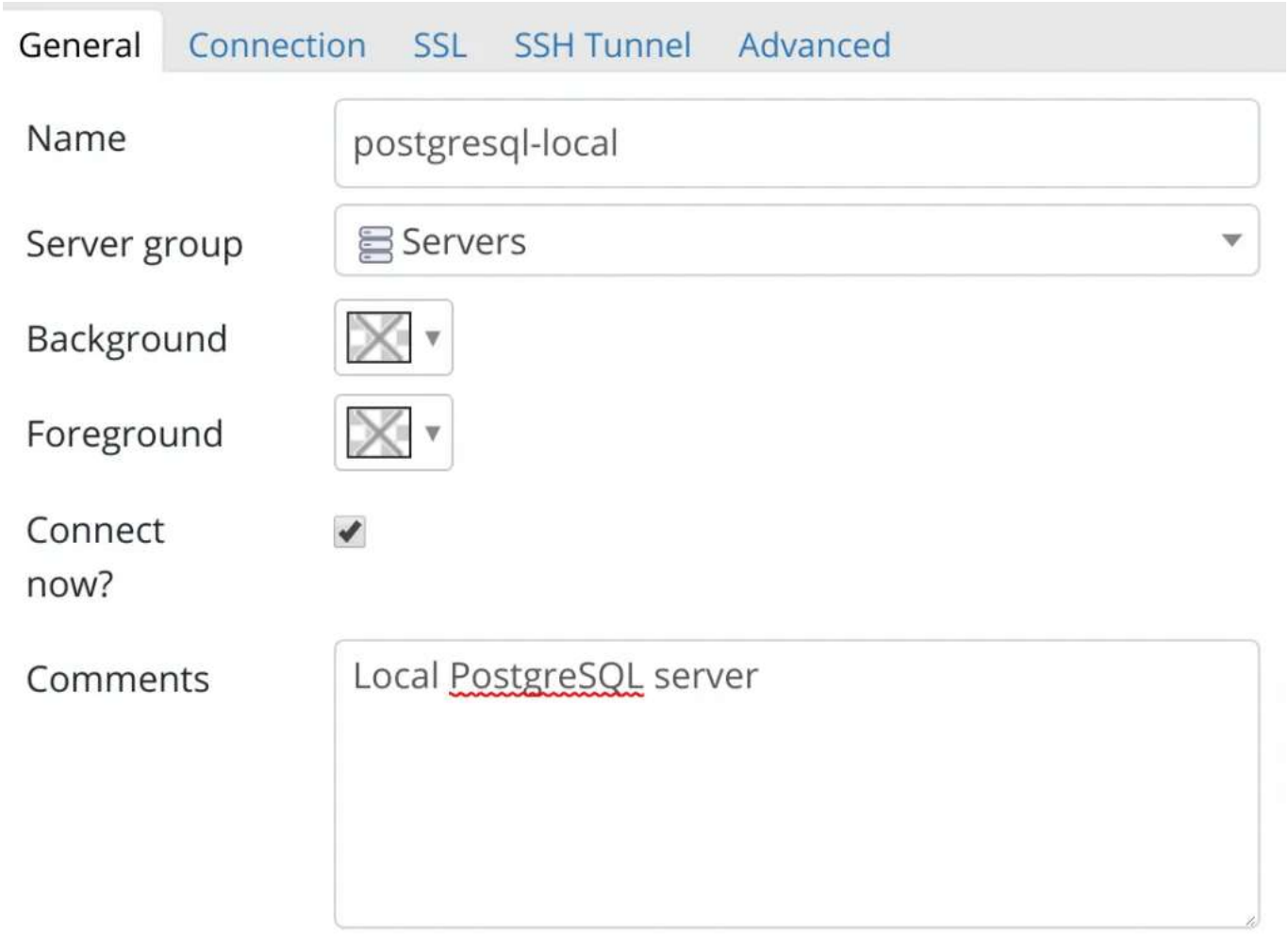
Open `http://pgadmin.example.com/pgadmin4` to log in to the pgAdmin with the credentials created step `6` above.



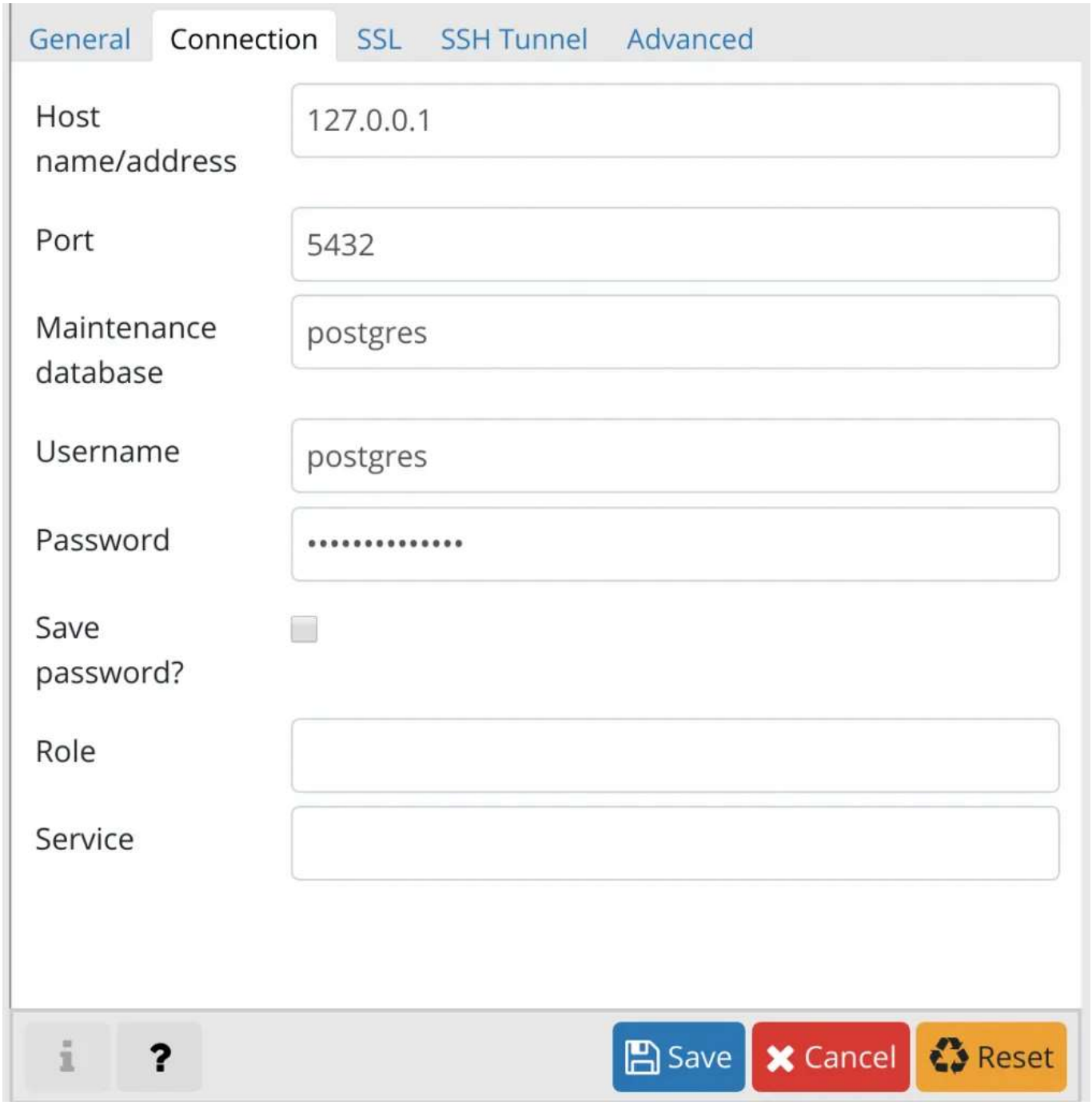
On the first page of pgAdmin, add a PostgreSQL server to administer with pgAdmin by clicking on **"Add New Server"**. This can be local or a remote

PostgreSQL server.

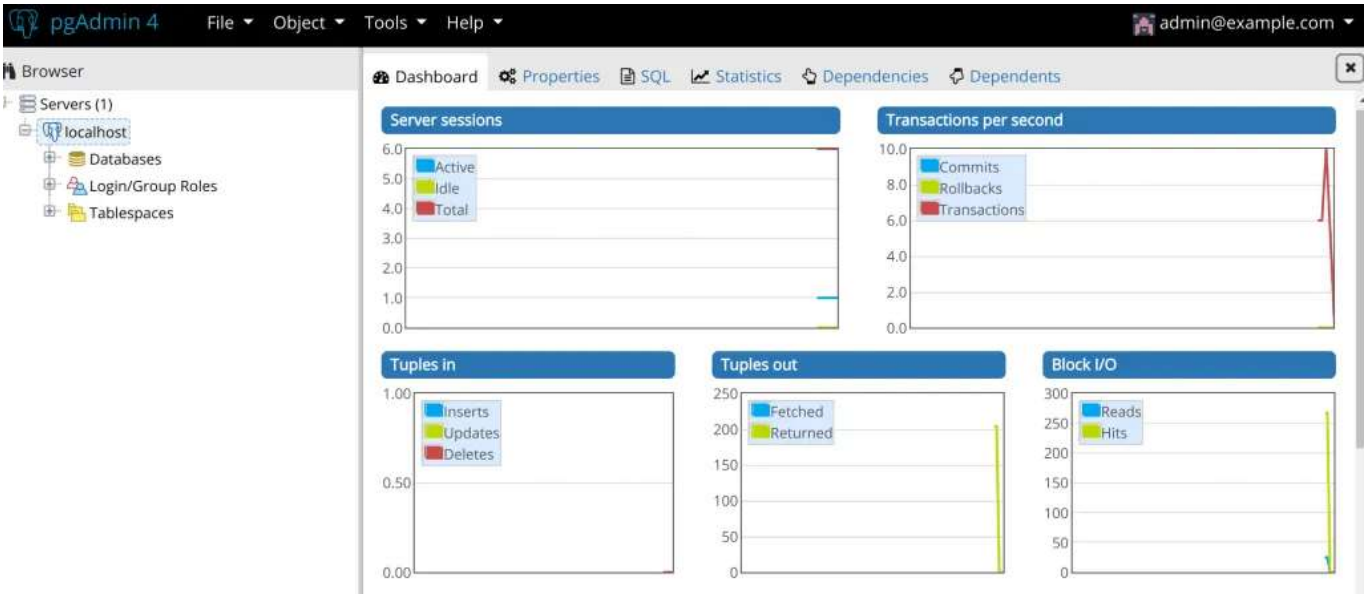
Under the **"General"** section, give the server a name & description.



Under **"Connection"** tab, provide access details – DB host, DB user and Password.



When done, Click **Save** button to save the configurations. If you were successful adding the server, the name will appear in the left sidebar. Select the server to see database summary information and make changes.



I hope our article was helpful in installing pgAdmin 4 on CentOS 7 and Fedora server/Desktop.

Other guides on installing pgAdmin 4:

How to Install pgAdmin4 on FreeBSD 12

Install pgAdmin4 on Ubuntu

Install pgAdmin4 on Debian