How to Install and Use a PostgreSQL Database on Linux

Introduction

PostgreSQL is a powerful, open source object-relational database. In this document, we will demonstrate how to install and use PostgreSQL at a high-level. Basic knowledge of Linux and relational databases is assumed.

Determine the Linux Distribution and Version

Login to your target Linux server as root. In our example, the hostname is "hp-mini".

```
[rana@alfred ~]$ ssh root@hp-mini
root@hp-mini's password:
Last login: Fri Feb 18 09:03:41 2022
[root@hp-mini ~]#
```

Type out the /etc/os-release file to determine the Linux distribution and version.

```
[root@hp-mini ~]# cat /etc/os-release
NAME="CentOS Stream"
VERSION="8"

    ...

ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="8"
PLATFORM_ID="platform:el8"
PRETTY_NAME="CentOS Stream 8"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:8"
HOME_URL="https://centos.org/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"
REDHAT_SUPPORT_PRODUCT="Red Hat Enterprise Linux 8"
REDHAT_SUPPORT_PRODUCT_VERSION="CentOS Stream"
[root@hp-mini ~]#
```

We have CentOS Version 8.

Create a New Linux User

As root, create a new user “rana” and set their password.

```
[root@hp-mini ~]# useradd rana
[root@hp-mini ~]# passwd rana
Changing password for user rana.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@hp-mini ~]#
```

Grant sudo administrative privileges to the new user by adding them to the “wheel” group. By default on CentOS 8, users who belong to the wheel group are allowed to use the sudo command.

```
[root@hp-mini ~]# usermod -aG wheel rana
[root@hp-mini ~]# id rana
uid=1064(rana) gid=1064(rana) groups=1064(rana),10(wheel)
[root@hp-mini ~]#
```

Login as the new user.

```
[rana@alfred ~]$ ssh rana@hp-mini
rana@hp-mini's password:
Last login: Fri Feb 18 09:51:53 2022 from 198.179.132.196
[rana@hp-mini ~]$
```

Test running a privileged command with sudo. You will be prompted for your password the first time each session and periodically after that.

```
[rana@hp-mini ~]$ sudo cat /etc/passwd
[sudo] password for rana:

     <contents of file>

[rana@hp-mini ~]$
```

Install PostgreSQL

PostgreSQL is available from CentOS 8's software repository, with several versions to choose from. The appropriate packages and dependencies for each version are bundled together into a "module stream".

For the installation we will use DNF, the default package manager for CentOS 8.

First, list the available postgresql module streams.

```
[rana@hp-mini ~]$ dnf module list postgresql

   •••

Name                      Stream                     Profiles
postgresql                9.6                        client, server
postgresql                10 [d]                     client, server
postgresql                12                         client, server
postgresql                13                         client, server

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
[rana@hp-mini ~]$
```

Enable the module stream for version 13, the most recent version. This will make the software available to us.

```
[rana@hp-mini ~]$ sudo dnf module enable postgresql:13

Last metadata expiration check: 1:52:15 ago on Fri 18 Feb 2022
Dependencies resolved.
================================================================
 Package                                   Version
================================================================
Enabling module streams:
 postgresql                                   13

Transaction Summary
================================================================

Is this ok [y/N]: y
Complete!
[rana@hp-mini ~]$
```

Install the “postgresql-server” package, which will also install its dependency package “postgresql”.

```
[rana@hp-mini ~]$ sudo dnf install postgresql-server

Last metadata expiration check: 3:09:39 ago on Fri 18 Feb 2022
Dependencies resolved.
================================================================
 Package        Architecture        Version
Installing:
 postgresql-server   x86_64   13.52.module_el8.6.0+1044+ed943ce5
Installing dependencies:
 postgresql          x86_64   13.52.module_el8.6.0+1044+ed943ce5
Transaction Summary
================================================================
Install  2 Packages
Total download size: 7.2 M
Installed size: 28 M
Is this ok [y/N]: y
Downloading Packages:
(1/2):
postgresql-13.5-2.module_el8.6.0+1044+ed943ce5.x86_64.rpm
(2/2):
postgresql-server-13.5-2.module_el8.6.0+1044+ed943ce5.x86_64.rpm
----------------------------------------------------------------
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction

   ...

Installed:
  postgresql-13.5-2.module_el8.6.0+1044+ed943ce5.x86_64
  postgresql-server-13.5-2.module_el8.6.0+1044+ed943ce5.x86_64
Complete!
[rana@hp-mini ~]$
```

Post-Installation Configuration

Initialize the database using the “postgresql-setup” script. This script is included with the distribution.

```
[rana@hp-mini ~]$ sudo postgresql-setup --initdb
 * Initializing database in '/var/lib/pgsql/data'
 * Initialized, logs are in /var/lib/pgsql/initdb_postgresql.log
[rana@hp-mini ~]$
```

Use systemctl to manually start the postgresql service. Then, enable the service so it will start automatically at boot.

```
[rana@hp-mini ~]$ sudo systemctl start postgresql

[rana@hp-mini ~]$ sudo systemctl enable postgresql
Created symlink /etc/systemd/system/multi-user.target.wants/
postgresql.service ->/usr/lib/systemd/system/postgresql.service.
[rana@hp-mini ~]$
```

Login to PostgreSQL

The installation procedure will have automatically created a Linux user account called “postgres”.

Switch to the “postgres” user account.

```
[rana@hp-mini ~]$ sudo -i -u postgres
[postgres@hp-mini ~]$
```

Type “psql” to access a postgres prompt.

```
[postgres@hp-mini ~]$ psql
psql (13.5)
Type "help" for help.
postgres=#
```

Exit by typing “\q”.

```
postgres=# \q
[postgres@hp-mini ~]$
```

Return to your own account.

```
[postgres@hp-mini ~]$ exit
logout
[rana@hp-mini ~]$
```

Create a New User, Role, and Database

PostgreSQL uses roles for user authentication and authorization. PostgreSQL roles typically have a matching Linux user account. Thus the terms “role” and “user” are used somewhat interchangeably.

By default, PostgreSQL also expects a user to be able to access a database with the same name as that user.

Taken altogether, this means we will want to create a Linux user, a PostgreSQL role, and a database all with the same name (“demo” in this example).

Create a new Linux “demo” user account.

```
[rana@hp-mini ~]$ sudo adduser demo
[rana@hp-mini ~]$
```

Switch to the “postgres” account and create the matching user/role.

```
[rana@hp-mini ~]$ sudo -i -u postgres
[postgres@hp-mini ~]$ createuser --interactive
Enter name of role to add: demo
Shall the new role be a superuser? (y/n) y
[postgres@hp-mini ~]$
```

Still from the “postgres” account, create a new database with the same name.

```
[postgres@hp-mini ~]$ createdb demo
[postgres@hp-mini ~]$
```

Connect to the Database

Exit from the “postgres” account. Then, switch to the “demo” account and connect to the database.

```
[postgres@hp-mini ~]$ exit
logout
[rana@hp-mini ~]$ sudo -i -u demo psql
psql (13.5)
Type "help" for help.
demo=#
```

Display database connectivity information.

```
demo=# \conninfo
You are connected to database "demo" as user "demo" via socket
in "/var/run/postgresql" at port "5432".
demo=#
```

Create a New Table

Our table example will be extremely simple: it will list just various types of fruit and an associated quantity. For example: “Bananas”, “30”.

Create a new table “fruit” with columns for the “type” of fruit and “quantity”.

```
demo=# create table fruit
demo=#(type varchar (50) PRIMARY KEY,
demo=# quantity integer NOT NULL);
CREATE TABLE
demo=#
```

Display the table information.

```
demo=# \d
       List of relations
 Schema | Name  | Type  | Owner
--------+-------+-------+-------
 public | fruit | table | demo
(1 row)
```

Import Data

To populate our table, we will import from a small pre-existing CSV file.

Display the file. Note that the data is comma-separated and we have a header.

```
[rana@hp-mini ~]$ cat fruit.csv
Type, Quantity
Bananas,30
Bartlett Pears,30
Canteloupes,10
Fuji Apples,50
Honeycrisp Apples,62
Kiwis,14
Lemons,12
Nectarines,20
Pink Lady Apples,75
Valencia Oranges,33
[rana@hp-mini ~]$
```

Import the data in the “fruit” table using the SQL copy command. Specify the file name, the comma delimiter, the file format of csv, and the fact that we have a header.

```
demo=# copy fruit
demo=# from '/home/rana/fruit.csv'
demo=# delimiter ','
demo=# csv
demo=# header;
COPY 10
demo=#
```

Verify the data by selecting everything from the table.

```
demo=# select * from fruit;
       type        | quantity
-------------------+----------
 Bananas           |       30
 Bartlett Pears    |       30
 Canteloupes       |       10
 Fuji Apples       |       50
 Honeycrisp Apples |       62
 Kiwis             |       14
 Lemons            |       12
 Nectarines        |       20
 Pink Lady Apples  |       75
 Valencia Oranges  |       33
(10 rows)
```

Query, Add, Update, and Delete Data

Query for just the “Apples” data.

```
demo=# select * from fruit where type like '%Apples';
       type        | quantity
-------------------+----------
 Fuji Apples       |       50
 Honeycrisp Apples |       62
 Pink Lady Apples  |       75
(3 rows)
```

Insert a new row and query to see the result.

```
demo=# insert into fruit values ('Gala Apples', '12');
INSERT 0 1
demo=#

demo=# select * from fruit where type like '%Apples';
       type        | quantity
-------------------+----------
 Fuji Apples       |       50
 Honeycrisp Apples |       62
 Pink Lady Apples  |       75
 Gala Apples       |       12
(4 rows)
```

Update the “Gala Apples” quantity and query to see the result.

```
demo=# update fruit set quantity='24' where type='Gala Apples';
UPDATE 1
demo=#

demo=# select * from fruit where type like '%Apples';
       type        | quantity
-------------------+----------
 Fuji Apples       |       50
 Honeycrisp Apples |       62
 Pink Lady Apples  |       75
 Gala Apples       |       24
(4 rows)
```

Delete the “Gala Apples” row and query to see the result.

```
demo=# delete from fruit where type='Gala Apples';
DELETE 1
demo=#
demo=# select * from fruit where type like '%Apples';
       type        | quantity
-------------------+----------
 Fuji Apples       |       50
 Honeycrisp Apples |       62
 Pink Lady Apples  |       75
(3 rows)
```

Conclusion

For more information, refer to the official PostgreSQL documentation.

Author: Rana Jones
Date: 2/25/2022