How to Install Postgres for Ubuntu Linux

launchschool.com/blog/how-to-install-postgres-for-linux

By Albert Agram

Note: For Postgres installation on Mac OS X, see How to Install Postgres for Mac OS X.

Introduction

For this post, we will go through the steps and options available to install Postgres on a Linux based system, specifically on Ubuntu 14.04 LTS. We will cover installation options using `apt-get` and a `Grapgical Installer` . Let’s dive in!

Install Postgres Database

All Linux distributions come bundled with an Advanced Packaging Tool or APT, to handle installation and removal of software. This acts as a kind of user interface and simplifies the process of installing software on Debian/Linux distributions. `apt-get` is one of the common tools for dealing with packages on Linux systems. You can type `apt-get --help` to view usage and available options. `APT` relies on repositories that contain free software available for installation. PostgreSQL is available on Ubuntu’s `apt` repository by default, thus we can install it easily.

Let’s start by updating our `apt-get`’s list of packages:

```
sudo apt-get update
```

Enter your admin password when prompted and wait for the process to complete.

Next, we’ll install Postgres by entering the following command in our terminal:

```
sudo apt-get install postgresql postgresql-contrib libpq-dev
```

What this does is to install additional modules and useful tools found in the contrib directory of the PostgreSQL distribution.

Enter `y` when prompted “Do you want to continue? [Y/n]” and wait for the installation to complete.

By default Postgres creates a `postgres` user and is the only user who can connect to the server. We’ll want to create ourselves on the server with superuser capabilities with the same name as our login name:

```
sudo -u postgres createuser --superuser $USER
```

Enter your desired password when prompted.

Next, we'll have to create a database with the same name as our login name since this is what Postgres expects by default when connecting to the server with your login name:

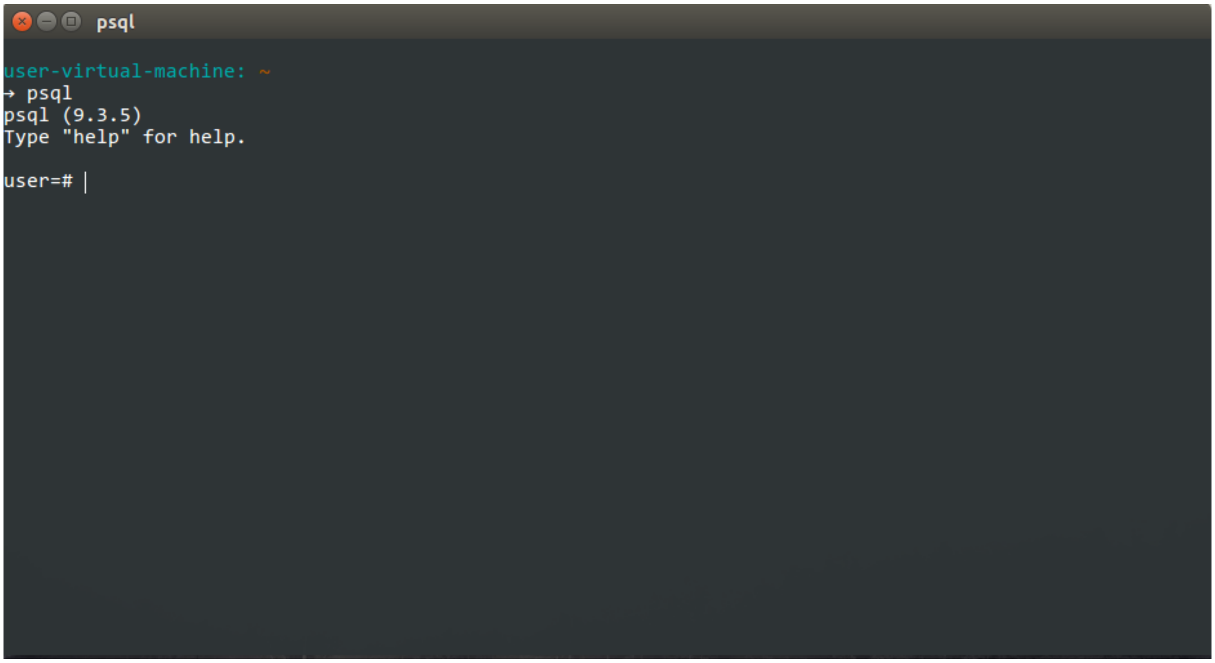
```
sudo -u postgres createdb $USER
```

Navigate to your home directory and enter the following command to create the `.psql_history` in order to save your history:

```
touch .psql_history
```

Type `psql` on your terminal to connect to the server:

```
psql
```



```
user-virtual-machine: ~
→ psql
psql (9.3.5)
Type "help" for help.

user=# |
```

Enter `\q` to quit and return to your terminal.

Set up Postgres to work with a Rails app

First, install the `pg` gem:

```
gem install pg
```

Make sure you include the `pg` gem in your Gemfile, and run

```
bundle install
```

Now create a Postgres user we’ll use in the Rails app. Use the same username that you use for your development work.

```
sudo -i -u postgres psql
```

Now you are in Postgres command line, run:

```
postgres=# CREATE ROLE <your_username> WITH CREATEDB LOGIN PASSWORD <your_password>;
```

Now, set up your `config/database.yml` file to point to your Posgres database.

```
development:
  adapter: postgresql
  encoding: unicode
  database: myapp_dev
  host: localhost
  pool: 5
  username: <your_username>
  password: <your_password>

test:
  adapter: postgresql
  encoding: unicode
  database: myapp_test
  host: localhost
  pool: 5
  username: <your_username>
  password: <your_password>
```

Let’s create the development and test databases:

```
rake db:create:all
```

Now you can run pending migrations, if there are any.

```
rake db:migrate
```