Lab 14.1: Setting up Postgresql
IN705 Databases Three

Introduction

We did our initial application development using SQLite. This is a common practice and lets us defer setting up a database server until it is needed. A small, lightly used application could use SQLite as a prodeuction database, but most applications need the performance and scalability supplied by a proper DBMS.

In this lab we will install PostgreSQL, one of the most widely used DBMSs. We will configure our Rails application to use PostgreSQL and create our database for it.

1 Install and configure PostgreSQL

Installing PostgreSQL on our EC2 servers is easy. Just enter the command

```
sudo apt-get install postgresql
```

We will also need some client libraries for our Rails application. Install these as well

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

Now we will create a user for our application's database. PostgreSQL users *peer based authentication* for typical admin operations. If you are logged into the server under a particular account, and if that account has the correct database privileges, then you are allowed to exectute the operations. The default database admin user is called `postgres` so we will switch to that system user and then create our new database account for the appliction.

```
sudo su - postgres
```

Now we are able to use the command `createuser`.

```
createuser -P -d sqrawler
```

This creates a database user named `sqrawler`. The `-d` option means that this user can create databases, and the `-P` option tells `createuser` to prompt us for a password for the new user.

Since Postgres uses peer based authentication by default, we need to modify the server configuration before our news user account will work with our Rails application. We have to add a line to the PostgreSQL server's configuration file.

File: /etc/postgresql/9.3/main/pg_hba.conf

```
local   all             sqrawler                                md5
local   all             all                                     peer # <-this line aleady exists
```

Restart the database server with the command `sudo service postgresql restart`.

2 Configuring Rails to use PostgreSQL

You need to do this work on the master branch of your git repository. You could either clone a fresh copy or work in one of your current working directories.

First, you need to edit your `Gemfile` to include the needed Postgres library. Add the line

```
gem 'pg'
```

and then run

```
bundle install
```

to load the needed gem.

Now edit the `config/database.yml` file. Modify the production settng to look like this

File: config/database.yml

```
production:
  adapter: postgresql
  encoding: unicode
  database: sqrawler
  pool: 5
  username: sqrawler
  password: sqrawl2014
```

Once this is done you will be able to create the database and its tables with the commands

```
rake db:create RAILS_ENV=production
```

```
rake db:migrate RAILS_ENV=production
```

Commit your changes and push them to Github. Then, pull the new code into your live working directory, use `git pull` to retrieve your new code, and restart the web server. You will now be serving your application using your new (but empty) PostgreSQL database.