Postgres Exercise

Install and setup PostgreSQL. Everything is explained in the digital ocean tutorial linked below. Just work through as much of the tutorial as you can in the time available.
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-ubuntu-16-04

NGINX Exercise

Basic Objectives

1. Installing the standard version of NGINX is easy:
   ```
   sudo apt install nginx
   ```

2. Start NGINX
   ```
   sudo nginx
   ```

   NOTE: If you get the error:
   ```
   bind() to 0.0.0.0:80 failed (98:Address already in use)
   ```
   type: `sudo fuser -k 80/tcp` (kills any other processes bound to port 80)

3. Make a request to the server in your browser to check the server is installed and started properly. Do this by typing your server's IP address in to the browser's address bar.

4. Find the main nginx configuration file and have a glance over it. (Hint: Google!)
   You should see all the main contexts we discussed, but no server config contexts.

5. Use the main config file to find and open the default "server" (otherwise referred to as a "virtual host") config file for editing, You will need to open the file with sudo. (Hint: Read the comments.)

6. Make the webserver return a message. There are two lines of configs you will need to add the following lines inside the server context (at the end of the context):

   ```
   location /hello {
        default_type text/html;
        return 200 "Hello World!";
   }
   ```

   This tells the webserver that for requests directed to /hello we want to return a HTTP 200 OK response with a short message that will be of the type "text/html".

7. Reload NGINX with the new configuration files:
   ```
   sudo nginx -s reload
   ```

8. Type your server's IP into your web browsers public address bar. Do you see your message? Ask for help if you don't, we'll help you debug.

9. Can you find the logs? Open up the access log, can you see the request you just made to the server? It should tell you your public IP address.
   (Hint: Look back at the main config file to find the logs)

NGINX Exercise

Advanced Objectives

SSL (Secure Sockets Layer) provides a means of securely communicating over the web I.e. way of assuring that you the consumer are indeed using the website you intend to use, and not a hacker's imitation of it. We use SSL certificates in the MOH server to securely communicate with the tablet devices. Read more about it here. Traditionally you would pay your web server host, or some similar company, money to validate your SSL certificates. Lets Encrypt now offers free automated SSL certificates. This is what we use to secure the Meerkat software system.

We will install a script called **`certbot t`**hat generates these Lets Encrypt certificates. We will then secure our new NGINX server and check that we have a secure connection in our web browser.

1. You will need a special domain name, ask Jonathan/Gunnar to create it for you.

2. Install certbot. These instructions are taken from here. Because the script isn't available in the standard Ubuntu app repositories, we will first have to add the correct application repository.

   ```
   sudo apt update
   sudo apt install software-properties-common
   sudo add-apt-repository ppa:certbot/certbot
   ```

   Now that the repository is added, we can install certbot as usual:

   ```
   sudo apt update
   sudo apt install certbot
   ```

3. Generate the certificate. Certbot will create a file accessible from the outside world through the NGINX server. Let's Encrypt will then try to use the file to assert that you have ownership over the server/ip address.

   ```
   sudo certbot certonly -a webroot -d <your domain>
                         --webroot-path=/var/www/html
   ```

4. Install the certificate with NGINX. We need to update the NGINX configs to accept https connections (which happen on port 443) and check the ssl certificates. Just add these lines to the top of the server context you edited earlier:

   ```
   listen 443 ssl default_server;
   ssl_certificate /etc/letsencrypt/live/<your domain>/fullchain.pem;
   ssl_certificate_key /etc/letsencrypt/live/<your domain>/privkey.pem;
   ```

5. Reload nginx.

6. Check the certificate. Go to https://<your domain> in the browser again, this time when you connect you should see a little green lock symbol appear the the address bar. Click it to get details of your secure connection.

7. Can you force all HTTP requests to be HTTPS requests? (Hint: Google!)