Linux Server Configuration

REVIEW | CODE REVIEW 7 | HISTORY

README.md 7

```
1 # Linux Server Configuration
2
3 ### Project Description
4
```

AWESOME

Great job highlighting critical info on your project!

```
5 This is the third project in the Udacity Full-Stack Nanodegre
6
7 - IP Address: 54.93.190.254
```

AWESOME

Good job setting up your server to be accessible through an IP address but most o
setup is a must since asking users to write IP addresses all the time is usually cumb

Here is a resource containing some basic initial info about DNS

```
8
9 - Accessible SSH port: 2200
10
11 - SSH Connection: ssh -i ~/.ssh/[keyFile] grader@54.93.190.25
```

Rate this review

AWESOME

```
12
13 This summary will hopefully walk you through all the steps yo
14
15 # Walkthrough Steps
16
17 ### 1. The Baseline Installation of a Linux Distribution, Sec
18
19 ## Step 1: Get an Amazon LightSail Server
20
21 [Follow the link to start up your own LightSail Server](https
22
23 1. Sign up to the Amazon Service
24 2. Once logged in, under **Build a solution tab**, click **Bu
25 3. Create your instance:
26
27     a. Leave the Instance Location as the default (unless you
28
29     b. Select **Linux/Unix** under the **Pick your instance i
30
31     c. Click **OS Only** and select **Ubuntu 16.04 LTS**
32
33     d. Select the lowest tier of the payment plans. If you co
34
35     e. Give your instance a hostname. Can be anything you wan
36
37     f. Wait for it to start up. Once it says running, you're
38
39 ## Step 2: Configure LightSail Firewall
40
41 Before anything, make sure you configure your server firewall
42
43 1. Click on your instance name
44 2. Click on **Networking**
45 3. Under **Firewall**, click **Add Another**
46 4. Keep everything as default and simply enter the port numbe
47 5. Click Save
48
49 Now you are ready to start configuring your instance!
50
51 ## Step 3: Update, Secure!
52
53 1. In the same line as **Networking**, click on **Connect**
54 2. Click the shiny orange button **Connect using SSH**
55
56 You are now inside your instance using the LightSail terminal
57
58 3. Run the following commands:
59
60     ```
61     sudo apt-get update
62     sudo apt-get upgrade
63     sudo apt-get autoremove
64     ```
65
```

```
This will ensure your instance is updated to the latest versi

4. Run the following command:

    `sudo nano /etc/ssh/sshd_config`

Around line 5-6, change **PORT 22** to **PORT 2200**

Save the file: `CTRL + X Y Enter`

Now its time to configure the Uncomplicated Firewall (UFW)**

5. Back in the terminal, run the following commands:

    ```
    sudo ufw status                     # check the status of
    sudo ufw default deny incoming      # deny imcoming conne
    sudo ufw default allow outgoing     # allow outgoing conn
    sudo ufw allow 2200\tcp             # allow listening on
    sudo ufw allow www                  # allow HTTP connecti
    sudo ufw allow ntp                  # allow NTP connectio
    sudo ufw show added                 # double check if all
    sudo ufw enable                     # enable the firewall
    sudo ufw status                     # it should give an A
    ```

> *Warning:* When changing the SSH port, make sure that the f

Once done with those configurations, it is now time to create

Step 4: Create user Grader; Sudo! Sudo! Sudo!
1. Run the following command to create the user:

    `sudo adduser grader   #password: udacity`

2. To give the grader user permission to sudo, run the follow

    `sudo nano /etc/sudoers.d/grader`

3. Inside the now open file, add the following text:

      "grader ALL=(ALL:ALL) ALL"

4. Save and Exit the file: `CTRL + X Y Enter`

That's it! Now your grader user has sudo permissions! It is n

Step 5: Permission Granted!
1. On your local machine, inside a terminal (Recommended for

    `ssh-keygen`

2. Enter the path you want to save the key-pairs in:

    `/c/Users/yourDesktopName/.ssh/yourFileName`

3. Once the key is created, run the following command:
```

```
    `cat ~/.ssh/yourFileName.pub`

4. Copy the whole line that is displayed.

5. Back on your LightSail terminal, run the following commanc

    ```
    sudo mkdir /home/grader/.ssh
    sudo touch /home/grader/.ssh/authorized_keys
    sudo nano /home/grader/.ssh/authorized_keys
    ```

    Within the opened file, paste the line you just copied ir

    Continue running commands to change file permissions for

    ```
    sudo chown grader:grader /home/grader/.ssh
    sudo chmod 700 /home/grader/.ssh
    sudo chmod 644 /home/grader/.ssh/authorized_keys
    ```

Alright! `grader` user is setup. Now, we just need one last t

Step 5: RootLogin: Permission Denied!
1. Run the following command:

    `sudo nano /etc/ssh/sshd_config`

2. Navigate the file until you find the following:

    PermitRootLogin - Change to **no**

    PasswordAuthentication - Change to **no**

Step 6: Restart the SSH Service
1. Run the following command:

    `sudo service ssh restart`

And, we are done with the first Part of the Configuration! We

> *Note:* By now, you might have lost access to your LightSai

2. Flask Application Deployment

Let's get into deploying our application for real now!

Step 1: Configure Timezone
1. Make sure the timezone is UTC by running the following cor

    `date`

2. Incase it is not UTC, run the following command:

    `sudo timedatectl set-timezone UTC`

Step 2: Apache time!
```

```
185 ## Step 2: Apache time!
186 We will now configure the Apache to serve Python mod_wsgi app
187
188 1. Run the following commands:
189
190     ```
191     sudo apt-get install apache2                    # install
192     sudo apt-get install libapache2-mod-wsgi-py3    # install
193     sudo a2enmod wsgi                               # enable
194     sudo service apache2 restart                    # restart
195     ```
196
197 > *Note:* If you have built your application with Python2, us
198
199
200 ## Step 3: PostgreSQL Time!
201 We will now configure the PostgreSQL package
202
203 1. Run the following commands:
204
205     ```
206     sudo apt-get install libpq-dev python-dev               #
207     sudo apt-get install postgresql postgresql-contrib      #
208     sudo nano /etc/postgresql/9.5/main/pg_hba.conf          #
209     ```
210
211     > *Note:* To ensure that remote connections to PostgreSQ
212
213 2. Inside the database, run the following:
214
215     ```
216     sudo -u postgres psql                                   #
217     CREATE USER catalog WITH PASSWORD 'catalog';            #
218     ALTER USER catalog CREATEDB;                            #
219     CREATE DATABASE catalog WITH OWNER catalog;             #
220     ```
221
222 ## Step 4: Git Time!
223 1. Install git (if not already installed). Run the following
224
225     `sudo apt-get install git`
226
227 ## Step 5: Deployment Time!
228 1. Clone the git repositroy for your Item Catalog Project & p
229
230     ```
231     cd /var/www
232     sudo mkdir catalog
233     sudo chown -R grader:grader catalog
234     cd catalog
235     sudo git clone https://github.com/a-g-hantash/item-catalo
236     ```
237
238 2. Change your files (database_setup.py, database_init.py, ap
239
240     a. Change all `create_engine` lines to:
241
242     `create_engine('postgresql://catalog:yourPassword@localho
243
244     b. Give the absolute path for the `client_secrets.json` f
```

```
    `/var/www/catalog/catalog/client_secrets.json`

    c. Import additional modules into your `app.py` file:


        import os
        import logging
        import psycopg2


    In the same file, in the main method, add:

        app.run()

    And remove:


         app.debug = True
         app.run(host='0.0.0.0', port=8080)


    d. Import additional modules into your `database_setup.py

        import os
        import sys
        import psycopg2


    e. Import additional modules into your `databse_init.py`

        import psycopg2

3. Create your _.wsgi_ file. Run the following commands:

```

AWESOME

Excellent Markdown usage to make your README more readable! It sure makes pr
stand out.

```
    ```
    cd /var/www/catalog
    sudo touch /var/www/catalog/catalog.wsgi
    sudo nano /var/www/catalog/catalog.wsgi
    ```

   a. Inside the file, write the following:

    ```
    import sys
    import logging

    logging.basicConfig(stream=sys.stderr, level=logging.DEBU
    sys.path.insert(0, "/var/www/catalog/")
    sys.path.append('/var/www/catalog/catalog/')
    sys.path.append('/usr/local/bin/python3.5/site-packages')
```
```

```
296
297
298     from catalog import app as application
299     application.secret_key = 'super_secret_key
300     ```
301
302     b. Save the file: `CTRL + X, Y, Enter`
303
304 4. Configure a New VirtualHost. Run the following commands:
305
306     `sudo nano /etc/apache2/sites-available/catalog.conf`
307
308     a. Inside the file, write the following:
309
310     ```
311     WSGIApplicationGroup %{GLOBAL}
312     WSGIRestrictEmbedded On
313
314     <VirtualHost *:80>
315         ServerName 54.93.190.254
```

AWESOME

```
316         ServerAlias ec2-54-93-190-254.eu-central-1.compute.ar
317         ServerAdmin grader@54.93.190.254
318         WSGIDaemonProcess catalog
319         WSGIProcessGroup catalog
320         WSGIScriptAlias / /var/www/catalog/catalog.wsgi
321
322         <Directory /var/www/catalog/catalog/>
323             Order allow,deny
324             Allow from all
325         </Directory>
326
327         Alias /static /var/www/catalog/catalog/static
328
329         <Directory /var/www/catalog/catalog/static/>
330             Order allow,deny
331             Allow from all
332         </Directory>
333
334         ErrorLog ${APACHE_LOG_DIR}/error.log
335         LogLevel warn
336         CustomLog ${APACHE_LOG_DIR}/access.log combined
337     </VirtualHost>
338     ```
339
340      b. Save the file: `CTRL + X, Y, Enter`
341
342 5. Install all the dependencies
343
```

AWESOME

Good job adding a high-level system dependency list

```
344     ```
345     sudo apt-get install python3-flask
346     sudo apt-get install python3-sqlalchemy
347     sudo apt-get install python3-psycopg2
348     sudo apt-get install python3-oauth2client
349     sudo apt-get install python3-httplib2
350     ```
351
352 6. Rename app.py. Run the following command:
353
354     `mv app.py __init__.py`
355
356 7. Enable the catalog virtual host and disable the default si
357
358     ```
359     sudo a2ensite catalog.conf
360     sudo service apache2 restart
361     sudo a2dissite 000-default.conf
362     sudo service apache2 restart
363     ```
364 8. Enable the virtual site. Run the following command:
365
366     ```
367     sudo a2ensite catalog
368     sudo service apache2 restart
369     ```
370
371
372 **And all done! If you visit your application URL or your IP
373
374
375 ## References
376
377 **Udacity Forums**
378
```

AWESOME

Great job documenting some references. It is essential to make your documentatio

```
379 **GitHub**
380 **Special Thanks to *
381 https://github.com/a-g-hantash/linux-server-configuration
382 https://github.com/iliketomatoes/linux_server_configuration
383 https://github.com/rrjoson/udacity-linux-server-configuratio
384 https://github.com/twhetzel/ud299-nd-linux-server-configurat
385 https://github.com/stueken/FSND-P5_Linux-Server-Configuratio
386 https://github.com/kongling893/Linux-Server-Configuration-UD
387 https://github.com/louiscollinsjr/udacity-linux-server-confi
388
389
390 * for a very helpful README**
391
```

RETURN TO PATH