Capping CS/IT/IS Project
CMPT 475-200

Documentation for Production Instance of Ubuntu 16.04 Running Apache2, PHP, and PostgreSQL on Google Server Platform

ABS Life Coach
Abroad Squad + Chris

Table of Contents

About This Document 2

Getting Started 2

Services currently running 2

Tasks In Progress 2

Installations and Related Service Info 2
- Apache 2
  - Installation 2
  - Apache .htaccess setup 3
- PHP 4
  - PHP Installation 4
  - (Link) for installation/setup. Skip to step 3 for PHP setup 4
  - Mcrypt Installation 4
- PostgreSQL 4
  - Installation 4
  - Database Backup 5

Security 5
- PHP Mcrypt 5
- PostgreSQL Database encryption (Link) 5
- Firewall Setup and Port Blocking 5
- HTTPS Certification with Let's Encrypt 6
- Running Server backups (not enough space) 6
  - Ran backup of life_coach db (11/21) 6

Troubleshooting 7
- Cannot connect with pgAdmin 7
- Changing users from ‘postgres’ to ‘developer’ issue 8
- Postgres connection Issue (11/10 Resolved) 8
- Undoing unneeded pgAdmin troubleshooting steps 9
- Changing Apache2 Config file 9

Load Testing/Balancing 9
- Load Testing 9
- Load Balancing 9

1. About This Document

1.1. **Author:** Patrick Zambri

1.2. **Purpose:** This document will contain all information about changes made to the final production instance Ubuntu 16.04 server set up by Professor Rivas. Everything I install/configure will be recorded here with links to guides. I will also record any additional steps or info, including passwords, under the section to which they apply.

2. Getting Started

2.1. Open WinSCP, download the key from Pablo's email, or Google Docs Key folder.

- 2.1.1. File Protocol: SFTP
- 2.1.2. Hostname: Abslifecoach.reev.us
- 2.1.3. Port number: 22
- 2.1.4. Username: developer
- 2.1.5. Password: N/A

2.2. Under 'Advanced', select 'Authentication' on the left sidebar, and for 'Private key file:' browse to wherever you downloaded the key to. Select it.

2.3. Log in.

2.4. On the top bar, look for the icon that looks like two monitors with a lightning bolt in front of them. 'Open session in PuTTy' (Ctrl+P).

2.5. PuTTy should open up and you are now logged into the Ubuntu linux server as developer@coach-pbmcj

3. Services currently running

3.1. SSH: 'service ssh status' (active 11/9)

3.2. Ufw Firewall: sudo ufw status (active 11/9)

3.3. Apache2: sudo systemctl status apache2 (active 11/9)

3.4. PHP 7.0: (Active 11/9)

- 3.4.1. Mcrypt (enabled 11/9)

3.5. PostgreSQL (Active 11/9)

4. Tasks In Progress

4.1. Looking into load balancing and update with specific documentation

4.2. Read through and edit this document to ready it for submission

- 4.2.1. Ensure consistency throughout the document.

5. Installations and Related Service Info

5.1. Apache

- 5.1.1. (https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-16-04) to setup and manage Apache2
- 5.1.2. Installation
  - 5.1.2.1. `sudo apt-get install apache2`
  - 5.1.2.2. Next step is to modify the permissions of Apache in the Firewall.

5.1.2.2.1. Apache uses port 80 for normal, unencrypted web traffic.

5.1.2.2.2. Apache Full opens both port 80, and port 443 (TLS/SSL encrypted traffic)

5.1.2.2.3. Apache secure only opens port 443

5.1.2.3. Allowed Apache Full Access: Full sudo ufw allow 'Apache Full'

5.1.3. Apache .htaccess setup

5.1.3.1. (https://www.digitalocean.com/community/tutorials/how-to-use-the-htaccess-file) with info about .htaccess. I did NOT use this install guide

5.1.3.2. Things to be aware of (from Digital Ocean link above):

5.1.3.2.1. One: **Speed**—the .htaccess page may slow down your server somewhat; for most servers this will probably be an imperceptible change. This is because of the location of the page: the .htaccess file affects the pages in its directory and all of the directories under it. Each time a page loads, the server scans its directory, and any above it until it reaches the highest directory or an .htaccess file. This process will occur as long as the AllowOverride allows the use of .htaccess files, whether or not the file the .htaccess files actually exists.

5.1.3.2.2. Two: **Security—**the .htaccess file is much more accessible than standard apache configuration and the changes are made live instantly (without the need to restart the server). Granting users permission to make alterations in the .htaccess file gives them a lot of control over the server itself. Any directive placed in the .htaccess file, has the same effect as it would in the apache configuration itself.

5.1.3.3. Steps I followed to Enable htaccess

5.1.3.3.1. Open file as 'sudo vim /etc/apache2/apache2.conf'

5.1.3.3.2. Remove comment sign (#) if you find it before this line (line number 187 approx.) AccessFileName .htaccess

5.1.3.3.3. Find the line (line 164) which looks like this:

5.1.3.3.3.1.
```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

5.1.3.3.4. And change to this:

5.1.3.3.4.1.
```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

5.2. PHP

5.2.1. PHP Installation

5.2.1.1. (https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04) for installation/setup. Skip to step 3 for PHP setup

5.2.1.2. `sudo apt-get install php libapache2-mod-php php-mcrypt`

5.2.1.2.1. Followed steps on site (except I did not include 'php-mysql' at the end of my install command)

5.2.2. Mcrypt Installation

5.2.2.1. `sudo apt-get install mcrypt php7.0-mcrypt`

5.2.2.2. `sudo service apache2 restart`

5.2.3. Test that php is configured properly

5.2.3.1. `sudo nano /var/www/html/info.php`

5.2.3.1.1. Edit info.php file to show info.

5.2.3.1.2. Info.php shows that php is running.

5.2.3.2. `sudo rm /var/www/html/info.php`

5.2.3.2.1. Removed the file because random people should not be able to access it by simply typing in the address.

5.2.4. UPDATE

5.2.4.1. PHP Mcrypt is no longer used because it is not compatible with PHP 7

5.2.4.2. (https://blog.craftblue.com/2017/06/php7-backwards-compatibility-mcrypt-key-sizes/) How to set up backwards compatibility for Mcrypt in PHP7

5.3. PostgreSQL

5.3.1. Installation

5.3.1.1. (https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-ubuntu-16-04) to setup and Using PostgreSQL Roles and Databases

5.3.1.2. Install Postgres

5.3.1.2.1. `sudo apt-get install postgresql postgresql-contrib`

5.3.1.3. General Usage Commands

5.3.1.3.1. To change to the postgres user: `sudo -u postgres psql postgres`

5.3.1.3.2. Exit the PostgreSQL session by typing: \q

5.3.1.4. Create user 'webuser'

5.3.1.4.1. `sudo -u postgres createuser --interactive`

5.3.1.4.2. Superuser? No.

5.3.1.4.3. Allow user to create DBs? No.

5.3.1.4.4. Allow user to create more new roles? No.

5.3.1.5. Created DB 'life_coach'

5.3.1.5.1. `sudo -u postgres createdb life_coach`

5.3.1.6. Set user passwords (https://www.liquidweb.com/kb/change-a-password-for-postgresql-on-linux-via-command-line/)

5.3.1.6.1. Set Password for user ‘Postgres’ to ‘abroadsquad’

5.3.1.6.2. Set Password for user ‘webuser’ to ‘Cappingteampablo2’

5.3.2. Database Backup

5.3.2.1. As user ‘postgres’ (use ‘sudo su - postgres’) I ran:

5.3.2.1.1. `pg_dump life_coach > /var/www/nov21_bkp.sql`

5.3.2.1.1.1. The file successfully appeared in the appropriate location. (/var/www/html)

5.3.2.2. General Information on backups

5.3.2.2.1. How to backup and restore DBs (https://www.vultr.com/docs/how-to-backup-and-restore-postgresql-databases-on-ubuntu-16-04)

5.3.2.2.1.1. Another backup info (https://www.commandprompt.com/blog/a_better_backup_with_postgresql_using_pg_dump/)

5.4. Set up phpPgadmin? (https://www.howtoforge.com/tutorial/ubuntu-postgresql-installation/)

5.4.1. No.

6. Security

6.1. PHP Mcrypt

6.1.1. See Mcrypt installation under PHP

6.1.1.1. Complete.

6.2. PostgreSQL Database encryption (http://www.cybertec.at/postgresql-instance-level-encryption/)

6.2.1. No longer required. Does not work and not enough time to complete.

6.3. Firewall Setup and Port Blocking

6.3.1. Firewall info and setup (https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-14-04)

6.3.1.1. `sudo ufw allow OpenSSH`

6.3.1.2. `sudo ufw enable`

6.3.1.3. `sudo ufw status`

6.3.2. Firewall Port Access Setup

6.3.2.1. Common firewall commands (https://www.digitalocean.com/community/tutorials/ufw-essentials-common-firewall-rules-and-commands)

6.3.2.2. Block an IP address: sudo ufw deny from (ip address)

6.3.2.3. Allowed Ports: Port 22 (ssh), Port 80, 433 (Apache, http, https), Port 5432 (Postgres), Outgoing on 25 (SMTP Mail).

6.3.2.4. Denied Ports: set default to deny all incoming connections.

6.4. HTTPS Certification with Let's Encrypt

6.4.1. Setup guide (https://certbot.eff.org/#ubuntuxenial-apache)

6.4.1.1. Followed steps from the guide.

6.4.1.2. Your key file has been saved at: /etc/letsencrypt/live/abslifecoach.reev.us/privkey.pem Your cert will expire on 2018-02-07.

6.4.1.3. To obtain a new or tweaked version of this certificate in the future, simply run certbot again with the "certonly" option.

6.4.1.4. To non-interactively renew *all* of your certificates, run "certbot renew"

6.4.1.5. Your account credentials have been saved in your Certbot configuration directory at /etc/letsencrypt.

6.4.1.6. You should make a secure backup of this folder now.* This configuration directory will also contain certificates and private keys obtained by Certbot so making regular backups of this folder is ideal.

6.4.1.6.1. *I did not make a secure backup of this folder.

6.4.1.7. Tested configuration at: (https://www.ssllabs.com/ssltest/analyze.html?d=abslifecoach.reev.us&latest)

6.4.1.7.1. Overall Rating: A

6.4.2. Auto-Renew

6.4.2.1. I did NOT enable auto-renew

6.5. Running Server backups (not enough space)

6.5.1. (http://www.thegeekstuff.com/2010/10/dd-command-examples/) to site with backup image info briefly. It has multiple ways to back up.

6.5.2. (https://www.youtube.com/watch?v=yuou5OiNnvA) to a video of someone doing it. Maybe not relevant as he is using a removable flash drive.

6.5.2.1. ‘Df’ to see info about the drives on the server

6.5.2.2. `Sudo dd if=/dev/sda1 of=~sda1bkp.img`

6.5.2.2.1. (‘if’ stands for ‘input file’, and of stands for ‘output file’

6.5.3. Output:

6.5.3.1.
```
dd: writing to '/home/developer/sda1bkp.img': No space left on device
16693489+0 records in
16693488+0 records out
8547065856 bytes (8.5 GB, 8.0 GiB) copied, 228.635 s, 37.4 MB/s
```

6.5.4. Ran backup of life_coach db (11/21)

6.5.4.1. See “Database Backup” in section 5.3.2

6.5.4.2. Consider implementing a regular backup schedule to follow.

7. Troubleshooting

7.1. Cannot connect with pgAdmin

7.1.1. Issue:

7.1.1.1. Not able to login and getting the “FATAL: Peer authentication failed for user “life_coach”.

7.1.2. Reset password for user ‘postgres’ to see if that resolved the issue.

7.1.2.1.1. Sudo -u postgres psql postgres

7.1.2.1.2. \password postgres

7.1.2.1.3. ‘abroadsquad’

7.1.2.1.4. \q (this brings you back to developer user now)

7.1.2.1.5. (This did not resolve the issue.)

7.1.3. Attempting to connect to server in pgAdmin 4.

7.1.3.1. Description of interface and results:

7.1.3.1.1. Connection Tab:

7.1.3.1.1.1. Hostname/address: abslifecoach.reev.us, Port: 5432, Maintenance database: postgres, Username: postgres or webuser, Password: pass1 or pass2, Role: N/A

7.1.3.1.1.2. SSL Tab:

7.1.3.1.1.2.1. SSL Mode: Prefer, nothing else

7.1.3.1.1.2.2. When I browse to and select the private key and it inputs that for the SSL tab, I still cannot connect.

7.1.3.1.2. This is the error I get: “Unable to connect to server: could not connect to server: Connection timed out (0x0000274C/10060) Is the server running on host "abslifecoach.reev.us" and accepting TCP/IP connections on port 5432?”

7.1.4. Attempting to allow access to pgAdmin following instructions from (http://suite.opengeo.org/docs/latest/dataadmin/pgGettingStarted/firstconnect.html)

7.1.4.1. Note: See section 7.4 where I undo most of these steps.

7.1.4.1.1. Attempt 1 (This step may have been unnecessary. Might reduce security?)

7.1.4.1.1.1. I believe it allows you to do ‘sudo psql -u postgres’ command in linux command line.

7.1.4.1.1.2. Edited the local unix domain socket connections method from ‘peer’ to ‘md5’

7.1.4.1.1.3. Unsuccessful

7.1.4.1.2. Attempt 2 (To allow pgAdmin connections)

7.1.4.1.2.1. The line that i needed to edit is already updated to what I should be.

7.1.4.1.2.2. Still can't connect using pgAdmin

7.1.4.1.3. Attempt 3 - **POSSIBLE SECURITY RISK**

7.1.4.1.3.1. From link, follow steps for "Allowing remote connections".

7.1.4.1.3.2. Unsuccessful.

7.1.4.1.4. Attempt 4

7.1.4.1.4.1. What about trying that pgAdmin web thing? phpPgAdmin in section 5.4?

7.1.4.1.4.2. Did not try this yet

7.1.5. <u>Resolution</u>:

7.1.5.1. Directly connect using SSH with terminal or other Bash environment.

7.2. Changing users from 'postgres' to 'developer' issue

7.2.1. <u>Issue</u>:

7.2.1.1. (11/10) once you log into postgres account on the command line using 'sudo su - postgres', you cannot swap back to the developer user.

7.2.2. <u>Resolution</u>:

7.2.2.1. USE: 'sudo -u postgres psql postgres'

7.2.2.1.1. You can then use '\q' to exit from the postgres user to the developer user with no issues.

7.2.2.2. DO NOT USE 'sudo su - postgres' or you will be unable to get out of the postgres user.

7.2.2.2.1. Workaround if you do use 'sudo su - postgres': kill the connection (close the putty/terminal window) and reconnect as 'Developer'

7.3. Postgres connection Issue (11/10 Resolved)

7.3.1. <u>Issue</u>:

7.3.1.1. 'psql: could not connect to server: no such file or directory' issue

7.3.2. Attempt 1:

7.3.2.1. Tried editing some things that were suggested on an online form.

7.3.2.1.1. `ln -s /tmp/.s.PGSQL.5432 /var/run/postgresql/.s.PGSQL.5432`

7.3.2.2. Nothing I tried worked, so I moved onto the resolution, which undid everything that was done here.

7.3.3. <u>Resolution</u>:

7.3.3.1. Removed Postgres and reinstalled it (11/10)

7.3.3.1.1.1. `sudo apt-get purge postgresql-9.5`

7.3.3.1.1.2. `sudo apt-get purge --auto-remove postgresql-9.5`

7.3.3.1.1.3. `sudo apt-get install postgresql postgresql-contrib`

7.3.3.1.1.4. Followed original install directions, created user 'webuser', created db 'life_coach', set passwords for postgres and webuser users.

7.4. Undoing unneeded pgAdmin troubleshooting steps

7.4.1. Issue:

7.4.1.1. Previous steps to allow pgAdmin access were ineffective, and could pose a security risk.

7.4.2. Resolution:

7.4.2.1. (following same instructions from last time from http://suite.opengeo.org/docs/latest/dataadmin/pgGettingStarted/firstconnect.html)

7.4.2.1.1. Attempt 1 (allowing local connections) - This step was already undone.

7.4.2.1.2. Attempt 2 (to allow pgAdmin connections) - **Left this the same.**

7.4.2.1.3. Attempt 3 (allowing remote connections) - undid this step.

7.5. Changing Apache2 Config file

7.5.1. Change "upload_max_filesize" from '2M' to '8M' (http://www.miscdebris.net/blog/2008/04/14/changing-the-php-file-upload-limit-in-ubuntu-linux/)

7.5.1.1. Sudo nano /etc/php/7.0/apache2/php.ini

7.5.1.2. Then followed directions in the link to change max file size from 2M to 8M.

7.5.2. Side edit: changing timeout from 300 seconds to 60 seconds. In /etc/apache2/apache2.conf

8. Load Testing/Balancing

8.1. Load Testing

8.1.1. Attempt 1: LoadImpact

8.1.1.1. Load testing page (https://loadimpact.com/). Free version only allows 25 users, but the sidebar help tab has good information.

8.1.2. Attempt 2: Loader.io

8.1.2.1.1. Ran a test with test range of 0-1000 where a user joins and continually grows until 1000 while each user is continually requesting. Server crashed at 390 concurrent users.

8.2. Load Balancing

8.2.1. Use HAProxy

8.2.1.1. Installation and configuration of HAProxy. (https://devops.profitbricks.com/tutorials/install-and-configure-haproxy-load-balancer-on-ubuntu-1604/)

8.2.1.1.1. Follow this installation guide to setup load balancing.

8.2.1.1.2. Due to the fact that we only have one available server, we cannot set up load balancing in our actual environment. If it were set up, it would keep the server from being overwhelmed by high volumes of traffic through sharing the load of that traffic with another server.

- 8.2.2. Alternative Load Balancing Solutions
  - 8.2.2.1. ProfitBricks DCD
    - 8.2.2.1.1. A more simple load balancer component. (https://www.profitbricks.com/help/Load_Balancer)
  - 8.2.2.2. AWS Load Balancing
    - 8.2.2.2.1. (https://aws.amazon.com/elasticloadbalancing/)
  - 8.2.2.3. Google Load Balancing
    - 8.2.2.3.1. (https://cloud.google.com/compute/docs/load-balancing/)