

 EWU-CSCD488-W16 / StuView Private

 Unwatch ▼ 6  Star 0  Fork 0

 Code  Issues 0  Pull requests 0  Projects 0  Wiki  Pulse  Graphs  Settings

Branch: master ▼ StuView / README.md

Find file Copy path

 mikebean233 Fixed spaces in source snippets

caf0804 4 hours ago

7 contributors 

378 lines (264 sloc) | 12.1 KB

Raw Blame History   

Basic Ubuntu Server Setup

First install Ubuntu 16.04, only install **standard system utilities** when prompted

Update the System

1. `sudo apt-get update`
2. `sudo apt-get upgrade`

Install apache, mysql, and php

3. `sudo apt-get install apache2`

4. `sudo apt-get install mysql-server`

the installation will prompt for a root mysql user password

5. `sudo apt-get install php7.0 libapache2-mod-php7.0 php7.0-mysql php7.0-mcrypt php7.0-curl`

Enable the rewrite, ssl, and headers modules

6. `sudo a2enmod rewrite`

7. `sudo a2enmod ssl`

8. `sudo a2enmod headers`

Disable the deflate module

9. `sudo a2dismod deflate`

You will be prompted to confirm the action

VirtualHost setup

10. Replace the contents of the file `/etc/apache2/sites-available/000-default.conf` with the following (insert the host name of the server where appropriate):

```
<VirtualHost *:80>
    ServerName INSERT_HOST_HERE

    Redirect "/" "https://INSERT_HOST_HERE"

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/StuView/public

    <Directory /var/www/html/StuView/public>
```

```
        AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Enable session storage

11. In the file `/etc/php/7.0/apache2/php.ini` uncomment the following line:

```
session.save_path = "/var/lib/php/sessions"
```

12. Restart apache: `sudo service apache2 restart`

Install Node.js and npm

13. `curl -sL https://deb.nodesource.com/setup_7.x | sudo -E bash -`

14. `sudo apt-get install -y nodejs`

Install ffmpeg

15. `sudo apt-get install ffmpeg`

Install sendmail and configure

16. `sudo apt-get install sendmail mailutils sendmail-bin`

17. `sudo mkdir /etc/mail/authinfo/`

18. `cd /etc/mail/authinfo`
19. `sudo touch gmail-auth`
20. `sudo nano gmail-auth`
21. add line `AuthInfo: "U:root" "I:ewustuvview@gmail.com" "P:stuvview2017"` then write out and exit
22. `sudo makemap hash gmail-auth < gmail-auth`
23. `cd ..`
24. `sudo nano sendmail.mc`
25. Use `CTRL+W` to search the file for `MAILER_DEFINITIONS`
26. Directly underneath `MAILER_DEFINITIONS` add the following lines:

```
define(`SMART_HOST',`[smtp.gmail.com]')dn1

define(`RELAY_MAILER_ARGS', `TCP $h 587')dn1

define(`ESMTP_MAILER_ARGS', `TCP $h 587')dn1

define(`confAUTH_OPTIONS', `A p')dn1

TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dn1

define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dn1

FEATURE(`authinfo',`hash -o /etc/mail/authinfo/gmail-auth.db')dn1
```

27. `sudo make -C /etc/mail`
28. `sudo /etc/init.d/sendmail reload`

29. `sudo nano ../hosts`

30. Change line `127.0.0.1 localhost` to `127.0.0.1 localhost.localdomain ubuntu`

31. Use `echo "Testing stuvview email relay" | mail -s "Stuvview sendmail relay" YOUR-EMAIL-HERE@YOUR-DOMAIN.com` to test the configuration. Replace YOUR-EMAIL-HERE and YOUR-DOMAIN with the email and domain you want to send the email to.

Perform steps 32 through 38 only for production environments

Create a clone of the project in the folder `/var/www/html` :

32. `sudo rm -r /var/www/html`

33. `sudo chown -R www-data:www-data /var/www`

34. `sudo chmod 777 /var/www`

35. `sudo -Hu www-data git clone https://github.com/EWU-CSCD488-W16/StuView /var/www/html`

Populate the database

36. `sudo mysql -u root -p < /var/www/html/StuView/database/stuvview_normalized.sql`

37. `sudo mysql -u root -p < /var/www/html/StuView/database/user.sql`

Setup Production SSL

38. You will need your server on it's own domain to get a certitace from a CA (Certificarte Authority), once it is on a domain you can use [certbot](#) to maintain its certificate.

Setup a Vimeo account (this will take several days to process and is not crucial for starting the project for the first time)

39. Create a new [Vimeo](#) account

Note: Basic accounts are very limited, you will likely need a business account for practical operation of StuView

Create an api app for the Vimeo account

40. Create a new api [app](#) for the vimeo account

41. Your app should show up in the My Apps section now. Click on it.

42. Under the Upload Access Section click on **Request Upload Access** and fill out the form.

43. Click on **Request Upload Access**. It will take several (around 3) days for your access to be granted.

Create a new access token for the app

44. Click on your apps Authentication tab.

45. Under the heading **Generate an Access Token**, check the **Upload** and **Edit** boxes.

46. Click the **Generate Token** button.

Update the project with the Vimeo credentials

47. On the **Authentication** page you will need the information under headings **Client Identifier**, **Client Secrets**, and **You new Access Token**.

48. Copy the three previously mentioned items into the vimeo credentials file found at **PROJECT/StuView/scripts/js/video_upload/config.json** as follows:

```
{
  "client_id"      : "INSERT CLIENT ID HERE",
  "client_secret"  : "INSERT CLIENT SECRET HERE",
  "access_token"   : "INSERT ACCESS TOKEN HERE"
}
```

Development Setup Using VirtualBox

1. Create a new Ubuntu 16.04 Server LTS VirtualBox VM and perform the Basic Server Setup as described in the **Basic Ubuntu Server** setup section
2. Install openssh: `sudo apt-get install openssh-server`
3. Install Guest Additions: `sudo apt-get install virtualbox-guest-dkms`

Setup Networking

4. In the file `/etc/network/interfaces` replace the contents with the following:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 192.168.56.102
    netmask 255.255.255.0

auto enp0s8
iface enp0s8 inet dhcp
```

5. Shutdown the VM: `sudo shutdown now`
6. In Virtualbox create a host only network as follows:

- Virtual Box -> Preferences... a dialog window will appear
- Network -> Host-only Networks
- click the + to create a new network, the network name should be **vboxnet0**
- select **vboxnet0**
- click the screw driver icon and make sure the IPv4 Address is **192.168.56.1**

7. In Virtualbox, navigate to the network settings of the current VM, in the network tab set the following:

- Adapter 1
 - Enable Network Adapter: **Checked**
 - Attached to: **Host Only Adapter**
 - Name: **vboxnet0**
 - Advanced:
 - AdapterType: **Paravirtualized Network (virtio-net)**
 - Promiscuous Mode: **Deny**
 - Cable Connected: **Checked**
- Adapter 2:
 - Enabled Network Adapter: **Checked**
 - Attached to: **NAT**
 - Advanced:
 - Adapter Type: **Paravirtualized Network(virtue-net)**
 - Cable Connected: **Checked**

Setup the Shared Folder

8. In Virtualbox navigate to the shared folders section for the current VM.
9. Click on the Plus folder button
10. Under Folder Path, navigate to the root of the local clone of the project.

11. Set Folder Name to **StuView_Folder**

12. Check Auto-mount

13. Click ok, then ok again

14. Start the VM

Add Users to vboxsf Group

15. Add the user www-data to the group vboxsf: `sudo usermod -aG vboxsf www-data`

16. Add the current user to the group vboxsf: `sudo usermod -aG vboxsf $(whoami)`

17. Reboot the VM: `sudo reboot now`

Point Apache to the Shared Folder

18. In the file `/etc/apache2/sites-available/000-default.conf` make the following changes:

Old Entry	New Entry
DocumentRoot /var/www/html/StuView/public	DocumentRoot /media/sf_StuView_Folder/StuView/public
<Directory /var/www/html/StuView/public>	<Directory /media/sf_StuView_Folder/StuView/public>

19. In the file `/etc/apache2/apache2.conf` make the following change:

Old Entry	New Entry
<Directory /var/www/>	<Directory /media/sf_StuView_Folder>

Setup SSL with a self signed certificate

20. Create the ssl certificate: `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt`

Enter the following at the prompts:

```
Country Name:      US
State:             WA
Locality:          Cheney
Organization:      Eastern Washington University
Organizational unit name: Computer Science Dept
Common Name:       192.168.56.102
```

21. In the file `/etc/apache2/sites-available/default-ssl.conf` replace the contents with the following:

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@localhost
        DocumentRoot /media/sf_StuView_Folder/StuView/public

        <Directory /media/sf_StuView_Folder/StuView/public>
            AllowOverride All
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        SSLEngine on
        SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
        SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
        <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
            SSLOptions +StdEnvVars
        </Directory>
    </VirtualHost>
</IfModule>
```

22. Enable the ssl virtualhost: `sudo a2ensite default-ssl.conf`

23. restart apache: `sudo service apache2 restart`

Populate the database

24. `sudo mysql -u root -p < /media/sf_StuView_Folder/database/stuvview_normalized.sql`

25. `sudo mysql -u root -p < /media/sf_StuView_Folder/database/user.sql`

Allow external database connections

26) In the file `/etc/mysql/mysql.conf.d/mysqld.cnf`

Replace the line

```
bind-address            = 127.0.0.1
```

with the following

```
#bind-address            = 127.0.0.1
#skip-networking
```

Setup phpmyadmin

27. Install phpmyadmin: `sudo apt-get install phpmyadmin`

Make the following selections:

```
Web server to configure automatically: apache2
Configure database for phpmyadmin with dbconfig-common?: No
```

28. Restart apache: `sudo service apache2 restart`

Setup xdebug (php debugging)

29. Install xdebug: `sudo apt-get install php-xdebug`

30. Add the following lines to end of the file `/etc/php/7.0/apache2/php.ini` :

```
; Added for xdebug
zend_extension="{PATH TO XDEBUG MODULE}/xdebug.so"
xdebug.remote_enable=1
xdebug.remote_handler=dbgp
xdebug.remote_mode=req
xdebug.remote_host=192.168.56.1
xdebug.remote_port=9000
xdebug.remote_log="/var/log/apache2/xdebug.log"
```

*make sure to find the actual path to **xdebug.so**, alternatively you can use the following commands (**note**: those are backticks, not single quotes around the find command):*

```
sudo chmod 222 /etc/php/7.0/apache2/php.ini
sudo echo "" >> /etc/php/7.0/apache2/php.ini
sudo echo "; Added for xdebug" >> /etc/php/7.0/apache2/php.ini
sudo echo "zend_extension=\"`find / -name xdebug.so 2> /dev/null`\"" >> /etc/php/7.0/apache2/php.ini
sudo echo "xdebug.remote_enable=1" >> /etc/php/7.0/apache2/php.ini
sudo echo "xdebug.remote_handler=dbgp" >> /etc/php/7.0/apache2/php.ini
sudo echo "xdebug.remote_mode=req" >> /etc/php/7.0/apache2/php.ini
sudo echo "xdebug.remote_host=192.168.56.1" >> /etc/php/7.0/apache2/php.ini
sudo echo "xdebug.remote_port=9000" >> /etc/php/7.0/apache2/php.ini
sudo echo "xdebug.remote_log=\"/var/log/apache2/xdebug.log\"" >> /etc/php/7.0/apache2/php.ini
sudo chmod 644 /etc/php/7.0/apache2/php.ini
```

31. Restart apache: `sudo service apache2 restart`

Your development server should be all setup now

A couple of notes:

- View the project by navigating to <http://192.168.56.102> in a browser.
- The first time you go to the project the browser will complain about a self signed ssl certificate, this is normal for the development setup.
- Access phpmyadmin by navigating to <http://192.168.56.102/phpmyadmin>
- You can debug php with the IDE of your choosing by using xdebug on port 9000
- You can ssh into the server with a terminal of your liking with the command `ssh {USERNAME}@192.168.56.102`
- You can connect to the MySQL server on 192.168.56.102 by using the credentials defined in the file `PROJECT/database/user.sql`

