**DR Plan**

Install the following packages on the Ubuntu VM Server to setup the basis for php website with backend of mysql database:

Apache:

The below link goes through process that is needed to install apache, mysql, and php. Not all the settings were taken from this link so only do the ones outlined in this document.

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04>

Install apache

sudo apt-get install apache2

Next, we will add a single line to the /etc/apache2/apache2.conf file to suppress a warning message

sudo vim /etc/apache2/apache2.conf

Line to be added at bottom:

ServerName 127.0.0.1

Restart apache server:

systemctl restart apache2

Allow web traffic

ufw app list

ufw app info "Apache Full"

ufw allow in "Apache Full"

MySQL:

<https://devanswers.co/install-apache-mysql-php-lamp-stack-ubuntu-20-04/>

Install mysql:

apt-get install mysql-server

When the installation is complete, we want to run a simple security script that will remove some dangerous defaults and lock down access to our database system a little bit. Start the interactive script by running:

mysql\_secure\_installation

You will be asked to enter the password you set for the MySQL root account. Next, you will be asked if you want to configure the VALIDATE PASSWORD PLUGIN.

Answer **y** for yes, or anything else to continue without enabling.

You’ll be asked to select a level of password validation. Keep in mind that if you enter **2**, for the strongest level, you will receive errors when attempting to set any password which does not contain numbers, upper and lowercase letters, and special characters, or which is based on common dictionary words.

There are three levels of password validation policy:

LOW Length >= 8

MEDIUM Length >= 8, numeric, mixed case, and special characters

STRONG Length >= 8, numeric, mixed case, special characters and dictionary file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1

If you enabled password validation, you’ll be shown a password strength for the existing root password, and asked you if you want to change that password. If you are happy with your current password, enter **n** for “no” at the prompt:

Using existing password for root.

Estimated strength of the password: 100

Change the password for root ? ((Press y|Y for Yes, any other key for No) : n

For the rest of the questions, you should press **Y** and hit the **Enter** key at each prompt. This will remove some anonymous users and the test database, disable remote root logins, and load these new rules so that MySQL immediately respects the changes we have made.

Php:

Install PHP

apt-get install php libapache2-mod-php php-mcrypt php-mysql

PhpMyAdmin:

The link below was used as general guide to installing and setting up phpMyAdmin on ubuntu server.

<https://www.linuxbabe.com/ubuntu/install-phpmyadmin-apache-lamp-ubuntu-20-04>

Install phpMyAdmin packages along with a few PHP extensions. These will help enable certain functionalities and improve performance.

apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl

For server selection choose apache2

Select Yes when asked whether to use dbconfig-common setup the database

It will then ask to choose a phpMyAdmin password and to confirm it.

Now run the following command to check if the /etc/apache2/conf-enabled/phpmyadmin.conf file exists.

file /etc/apache2/conf-enabled/phpmyadmin.conf

If there’s no error in the installation process, you should see the following command output.

/etc/apache2/conf-enabled/phpmyadmin.conf: symbolic link to ../conf-available/phpmyadmin.conf

If this file doesn’t exist on your server, it’s likely that you didn’t select Apache web server in the phpMyAdmin setup wizard. You can fix it with the following commands.

sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf

sudo a2enconf phpmyadmin

sudo systemctl reload apache2

Open ports for phpMyAdmin

sudo ufw allow 80,443/tcp

Now try the link below to see if you can access phpMyAdmin.

<http://127.0.0.1/phpMyAdmin>

Log in with the root and password created during setup.

If you are able to access the above link and log in then the installation was success.. if not please review the link below the heading phpMyAdmin towards the bottom for some good troubleshooting instructions.

Certificate creation:

<https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-18-04>

Certificate creation and setup on apache server:

Create self-signed certificates using openssl

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt

Save the certs into a directory that has restricted permission on the directory

SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt

SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

After the certificates have been created with public cert and private key.

These certificates are used to validate that the website is a trusted site.

To accomplish this Apache needs files modified so that websites can use https instead of http.

* Modify the /etc/apache2/sites-available/default-ssl.conf file entries ServerName with localhost, DocumentRoot with directory to website, SSLCertificateFile with full path and cert name and SLLCertificateKeyFile with full path and key name then save and close.
* /etc/apache2/sites-available/default-ssl.conf

Text

Description automatically generated

Text

Description automatically generated

* Modify /etc/apache2/sites-available/000-default.conf file entries DocumentRoot with directory website, DirectoryIndex with mainpage of website, SSLCertificateFile with full path self-signed cert and the SSLCertificateKeyFile with full path self-signed key file.
* /etc/apache2/sites-available/000-default.conf

Text

Description automatically generated

* Modify the /etc/apache2/conf-available/ssl-params.conf entires to look like the below example in the below image.
* /etc/apache2/conf-available/ssl-params.conf

Text

Description automatically generated

Below commands enables the SSL module and adds a symlink in the /etc/apache2/sites-enabled directory to the file /etc/apache2/sites-available/default-ssl.conf to include it into the active apache configuration. Then restart apache to enable the new configuration:

restart apache

stystemctl restart apache2

Enable mod\_ssl, the Apache SSL module and mod\_headers

a2enmod ssl

a2enmod headers

Enable the SSL Virtual Hot

a2ensite default-ssl

Enable our ssl-params.conf file  
a2enconf ssl-params

Enter this link into web browser and see if it works <https://localhost/>

Change the below file to look like the image below the command so that the apache set to home page of website by adding index.php file as shown below.

vim /etc/apache2/mods-enabled/dir.conf

Text

Description automatically generated

Restart apache

**Certificate creation and setup on the mysql instance:**

The two links below are used for the setup of the mysql self signed certificate setup for MySQL.

<https://websiteforstudents.com/how-to-setup-self-signed-ssl-tls-on-mysql/>

<https://www.digitalocean.com/community/tutorials/how-to-configure-ssl-tls-for-mysql-on-ubuntu-18-04>

Set up ssl on the mysql…

Commands to create certificates using openssl are to be ran on terminal in the order below:

openssl genrsa 2048 > ca-key.pem

openssl req -new -x509 -nodes -days 3600 -key ca-key.pem -out ca.pem

openssl req -newkey rsa:2048 -days 3600 -nodes -keyout server-key.pem -out server-req.pem

openssl rsa -in server-key.pem -out server-key.pem

openssl x509 -req -in server-req.pem -days 3600 -CA ca.pem -CAkey ca-key.pem -set\_serial 01 -out server-cert.pem

openssl req -newkey rsa:2048 -days 3600 -nodes -keyout client-key.pem -out client-req.pem

openssl rsa -in client-key.pem -out client-key.pem

openssl x509 -req -in client-req.pem -days 3600 -CA ca.pem -CAkey ca-key.pem -set\_serial 01 -out client-cert.pem

openssl verify -CAfile ca.pem server-cert.pem client-cert.pem

chown -R mysql:mysql /etc/mysql/

chmod 600 client-key.pem server-key.pem ca-key.pem

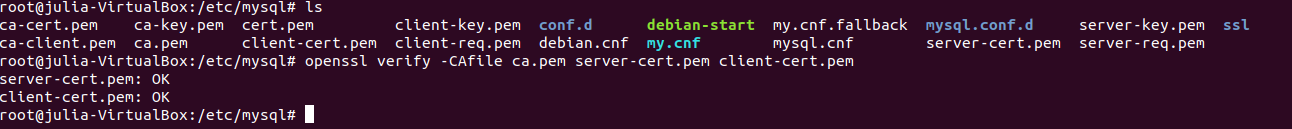
cat server-cert.pem client-cert.pem > ca-client.pem

cp ca-client.pem ca-cert.pem

cat client-cert.pem client-key.pem > cert.pem

Run the opensll below to verify that the certificates are valid:

openssl verify -CAfile ca.pem server-cert.pem client-cert.pem



Now that the certificates have been generated they need to be copied to certificates directory

If the directory is not created create it:

mkdir /etc/cert

Copy the files to /etc/cert

cp\*.pem /etc/cert/

Text

Description automatically generated

Change directory to /etc/cert/

cd /etc/cert/

Change permission on the certificate files:

chown mysql:mysql cert.pem

chown mysql:mysql ca-cert.pem

chown mysql:mysql client-cert.pem

Update mysql to use the certificates just created:

vim /etc/my.cnf

Modify these values below in the my.conf file:

bind-address = 0.0.0.0

**# Change the certificates entries to point to the certificates**  
ssl-ca=/etc/mysql/ca.pem  
ssl-cert=/etc/mysql/server-cert.pem  
ssl-key=/etc/mysql/server-key.pem

We are not setting up require\_secure\_transport = on (because we are only using localhost and not remote server to connect)

Example of what the file should look like is in the below image:

Text

Description automatically generated

Save and close

Restart mysql

systemctl restart mysqld

Now login to the MySQL and check the SSL.

mysql -u root -p  
Type password to root at password prompt:

Run query below to make sure SSL section value is '**YES**'.

SHOW VARIABLES LIKE '%ssl%';  
STATUS;

Text

Description automatically generated

Now MySLQ is using the certs that was generated with openSSL.

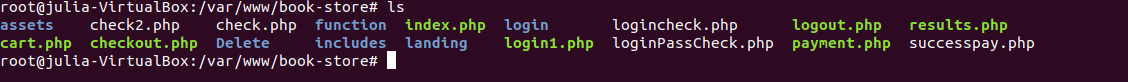
Configure Website

<https://www.linode.com/docs/guides/hosting-a-website-ubuntu-18-04/>

<https://websiteforstudents.com/setup-apahce2-with-php-support-on-ubuntu-servers/>

<https://www.cloudbooklet.com/how-to-install-lamp-apache-mysql-php-in-ubuntu-20-04/>

Copy the backup of the webwsite to directory giving in the path below in the image. /var/www/book-store/



A file will need to be modified in order to connect to mysql if a different certificate name was used other than the one on original website.

This is the conf.php file which has the database connection information. Update path and/or cert name in the ssl\_set line. Since this is setup on localhost you could not configure this as we would a remote server. A workaround was implemented that I found on this site:

<https://serverfault.com/questions/399487/cant-connect-to-mysql-using-self-signed-ssl-certificate>

I used one file named ca.pem that merged all the server.key, client.key and ca,pem file in one.

Text

Description automatically generated

Install the different tools below to help with monitoring and alerting if any incidents on the server:

Sendmail:

**Install Sendmail:**

apt-get install Sendmail

**Configure Sendmail files to work with gmail:**

[Configure Sendmail to Relay Emails through Gmail SMTP – TecAdmin](https://tecadmin.net/sendmail-to-relay-emails-through-gmail-stmp/)

[Configuring Gmail as Sendmail email relay - LinuxConfig.org](https://linuxconfig.org/configuring-gmail-as-sendmail-email-relay)

[https://linuxconfig.org/configuring-gmail-as-Sendmail-email-relayhttps://linuxconfig.org/configuring-gmail-as-Sendmail-email-relay](https://linuxconfig.org/configuring-gmail-as-sendmail-email-relay)

The /etc/hosts file needs to be updated to resolve the error message while sending emails via sendmail “That this server is not part of a domain”.

### vim /etc/hosts

### On the line that has localhost add localhost.localdomain and servername.domainname

### Add two more lines with loop back IP with server name and a domain name, then the next line is factious domain.

### Text Description automatically generated

### Save and Close file

### We also need to change server name to include the domain making it a fully qualified name in the file /etc/hostname. So sendmail doesn’t give error “Server isn’t in domain”.

### <https://gridscale.io/en/community/tutorials/hostname-fqdn-ubuntu/>

### vim /etc/hostname

### Add the factious domain name to the end of the server name in this file:

### Example:

### ServerName.nos.com

### Restart the server..

### After the server comes back online and you login. Run the blow command via terminal. Now check your host name by typing the command below:

### hostname

### Next we will modify the Sendmail.mc file. To configure sendmail to use gmail relay server to mass emails through.

### Add lines from example below to LOCAL\_CONFIG section of the /etc/mail/Sendmail.mc file towards the bottom. In this file we also need to update FEATURE path to /etc/mail/authinfo/gmailinfo.db.

### vim /etc/mail/Sendmail.mc

### Text Description automatically generated

### The /etc/mail/local-host-names file will need to be modified by adding php, emails, localhost, and server name entries into the file. This will mitigate the errors received while sending emails with Sendmail “that server or email you are using is not authorized to send emails”.

### An example is given in the photo below:

### vi /etc/mail/local-host-names

### Create a directory named authinfo that will hold files that Sendmail uses to allow gmail as your email relay provider.

### mkdir /etc/mail/authinfo

### cd /etc/mail/authinfo/

### Inside this directory I created a file named gmailinfo..

### vim gmailinfo

### Add the following information to the file:

### Text Description automatically generated

### Save and cloes file.

### Text Description automatically generated

### Then compiled the file into a db that Sendmail can use.

makemap hash /etc/mail/authinfo/gmailinfo < /etc/mail/authinfo/gmailinfo

### cd /etc/mail

### Compile Sendmail with new settings:

### sendmailconfig

### Say yes to all the questions.

Now test send mail to see If you are able to receive and email from local Sendmail.

### Echo “This is a test email” | mail -s “this is a test” ekbaker.2553@gmail.com

Nagios Core:

[Nagios Core - Installing Nagios Core From Source](https://support.nagios.com/kb/article/nagios-core-installing-nagios-core-from-source-96.html#Ubuntu)

[How To Install Nagios 4 and Monitor Your Servers on Ubuntu 18.04 | DigitalOcean](https://www.digitalocean.com/community/tutorials/how-to-install-nagios-4-and-monitor-your-servers-on-ubuntu-18-04)

Install nagios:

Prerequisites need installed:

sudo apt-get update  
sudo apt-get install -y autoconf gcc libc6 make wget unzip libapache2-mod-php7.2 libgd-dev

**Downloading the *Source***

cd /tmp  
wget -O nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.5.tar.gz  
tar xzf nagioscore.tar.gz

Compile

cd /tmp/nagioscore-nagios-4.4.5/  
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled  
sudo make all

### Create User And Group

This creates the nagios user and group. The www-data user is also added to the nagios group.

sudo make install-groups-users  
sudo usermod -a -G nagios www-data

### Install Binaries

This step installs the binary files, CGIs, and HTML files.

sudo make install

### Install Service / Daemon

This installs the service or daemon files and also configures them to start on boot.

sudo make install-daemoninit

### Install Command Mode

This installs and configures the external command file.

sudo make install-commandmode

### Install Configuration Files

This installs the \*SAMPLE\* configuration files. These are required as Nagios needs some configuration files to allow it to start.

sudo make install-config

### Install Apache Config Files

This installs the Apache web server configuration files and configures Apache settings.

sudo make install-webconf  
sudo a2enmod rewrite  
sudo a2enmod cgi

### Configure Firewall

You need to allow port 80 inbound traffic on the local firewall so you can reach the Nagios Core web interface.

sudo ufw allow Apache  
sudo ufw reload

### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

### Restart apache web service

Need to restart it because it is already running.

sudo systemctl restart apache2.service

### Start Service / Daemon

This command starts Nagios Core.

sudo systemctl start nagios.service

### For more functionality within nagios we will need now to install nagios-plugins

### Prerequisites

Make sure that you have the following packages installed.

sudo apt-get install -y autoconf gcc libc6 libmcrypt-dev make libssl-dev wget bc gawk dc build-essential snmp libnet-snmp-perl gettext

### Compile + Install

cd /tmp/nagios-plugins-release-2.2.1/  
sudo ./tools/setup  
sudo ./configure  
sudo make  
sudo make install

### For nagios to monitor localhost which we are using at this time.. We need to configure some internal files to nagios so that if we have any alerts nagios will send emails to the nagios admin email account…

### [Configure Nagios To Use Sendmail – Brandon Wamboldt](https://brandonwamboldt.ca/configure-nagios-to-use-sendmail-1188/)

### [Nagios alerts via email](https://community.spiceworks.com/topic/117801-nagios-alerts-via-email)

### [9.2. Configuring Nagios Server to Send Mail Notifications Red Hat Gluster Storage 3 | Red Hat Customer Portal](https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3/html/console_administration_guide/configuring_nagios_to_send_mail_notifications)

### Since this is localhost we do not need to setup nagios to monitor an external server but we do need to supply nagios with the email we would like our alerts to be sent to.

vim /usr/local/nagios/etc/objects/contacts.cfg

Find the email directive and replace its value with your own email address:

### Timeline Description automatically generated with medium confidence

### Save and exit the editor

### We need to modify the commands file we need to change where the mail is being launched from on both define command - command\_name - notify-service-by-email since we are using Sendmail with gmail setup to send out emails if any alerts arises

### Vim /usr/local/nagios/etc/objects/command.cfg

### Text Description automatically generated

### Save and exit the editor

Localhost# vim /usr/local/nagios/etc/nagios.cfg

Text

Description automatically generated

Save and exit

vim /usr/local/nagios/etc/resource.cfg

### Text Description automatically generated

### Add fields with values for gmail setup in this file.. username, password, and smtp for gmail

### Test if email is working through nagios choose Services:

### [Test Email Alert - View topic • Nagios Support Forum](https://support.nagios.com/forum/viewtopic.php?f=7&t=26154)

### Home page:

### Graphical user interface, text, application Description automatically generated

### Under Services choose a service to test by left clicking the Service name:

### Table Description automatically generated

### Left click on the right handside “Send customer service notification”

### Text, timeline Description automatically generated with medium confidence

### Another window will appear and chicke forced.. fill out a comment and hit commit button.

### Graphical user interface, text, application, email Description automatically generated

### Shortly an email should appear lke the one below:

### 

Keepass2

<https://linuxhint.com/install_keepass_ubuntu/>

Install keepass2

apt-get install keepass2 -y

Setup Keepass2 for use:

First after the install we need to open up keepass2 and create a database to store our groups and users in:

Left click the this icon in your applications directory:

A logo on a red background

Description automatically generated

This will bring up a screen like below:

Graphical user interface, text, application, email

Description automatically generated

At the top of this window on the farthest left hand side is an option named “File” left click this it will drop down a box like below left click new:

Graphical user interface, text, application

Description automatically generated

A new window will pop-up over the top of the above window. Click Ok on the pop-up Box.

Graphical user interface, text

Description automatically generated

Another window will pop-up. Here I just chose default name and location. Left click Save button on the right handside bottom side of the pop-up screen.

Graphical user interface, text, application

Description automatically generated

In the next window please choose a secure password that will be used to access the keepass2 each time you open the window to access it. Enter the password in both fields Master password and Repeat password. The password needs to be the same or an error will pop-up. Next click ok at the bottom of this screen.

Graphical user interface, application

Description automatically generated

Now to log into keepass with the password just created. Then click ok at the bottom right hand side of the Enter Master Key screen.

Graphical user interface

Description automatically generated

Now we will create a new group to hold user names and passwords under. On main page left click edit and scroll down to Edit Group and left click:

Graphical user interface, text, application

Description automatically generated

Another page will pop-up that we can give our new Group a meaningful name to the users that will be stored inside it. Click OK to create the group.

Graphical user interface, text, application

Description automatically generated

Now that our group is created New Group we want to add users with their passwords to this group. Highlight New Group under the General Group we just created.

Graphical user interface, text, application, chat or text message

Description automatically generated

At the top of keepass there is a key with green plus sign. Left click and then left click Add Entry.

Graphical user interface, text, application

Description automatically generated

This will pop-up a new window that will allow us to give meaningful Title to our User Name. Add an entry for User name, Password. To help keep track of when the password needs to be changed we can set up Expires and for this project in our ACPs we have set it where the passwords need to change every 180 days. We would choose 6 Months from the options selection. Then click OK to create our user with password.

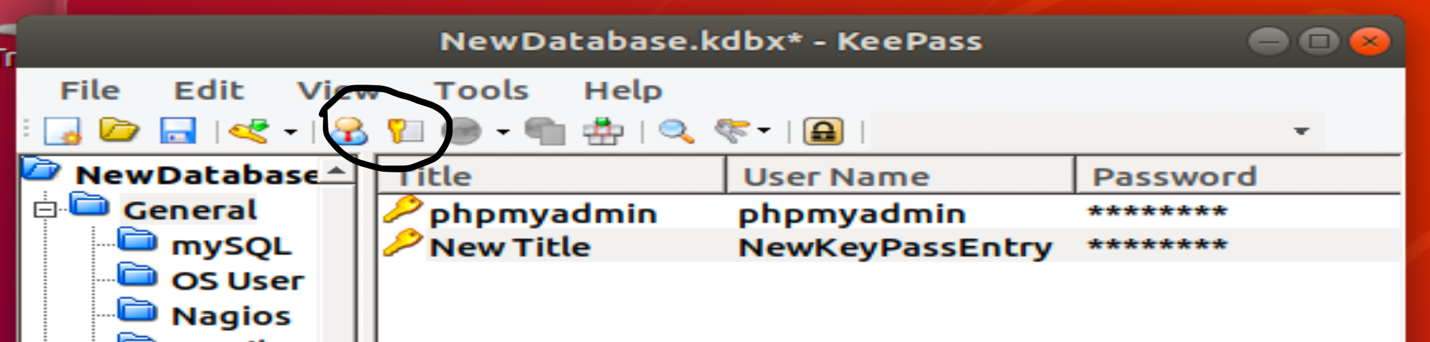
Graphical user interface, text, application

Description automatically generated

To get user name or password of the user there is two icons on the main page that can be used.

The person icon gets the user name and copies the user name to clipboard for use.. via paste function.

The key icon to the right of the person icon copies the password to the clipboard and can be retrieved via the paste function.



Now keepass is all setup and ready for use.

**Setup bashfile to send emails if login failure occurs.**

<https://unix.stackexchange.com/questions/339417/running-a-shell-script-on-n-failed-login-attempts>

<https://askubuntu.com/questions/727156/email-on-failed-login-attempt>

### To trigger the bash file put this line in sshd pam file

Alter a line of code in the /etc/pam.d/sshd file exactly as seen in below the below image.

vim /etc/pam.d/sshd

### Text Description automatically generated

### Create bash file named report\_badlogin exactly as the example image below. This file will check if there is a new failed login and then sends an email if there is.

vim /usr/bin/report\_badlogin

Text

Description automatically generated

Should receive and email like the example image below if the setup is working correctly.



Firewall Rules:

Here is all the firewall rules that will need to be setup if they have not been setup through this DR installation guide:

Firewall for ubuntu

Enable firewall

sudo ufw enable

open up SSH

sudo ufw allow ssh

open up port 80 for http

Sudo ufw allow Apache

nagios rules

sudo ufw allow ‘Apache Full’

allow Sendmail

ufw allow 25

sudo ufw logging on

mysql port allow

ufw allow 3306

reload the ufw settings

sudo ufw reload

check the ufw settings

Ufw status