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 phpmyadmin 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 selfsigned certificates using openssl

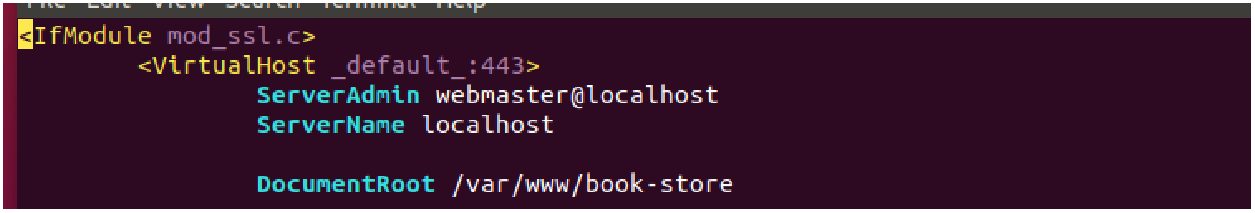
```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-
selfsigned.crt
```

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

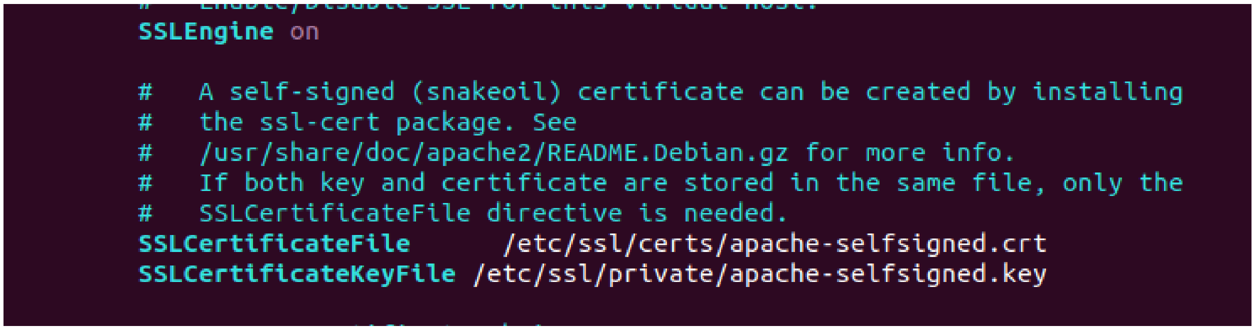
After SSL certificate has been created public and private certs.
These certs are used to validate that the website is trusted site.
Apache then needs few files changed so that websites can use https instead of http.

Create an Apache configuration snippet to define some SSL settings.

- /etc/apache2/sites-available/default-ssl.conf

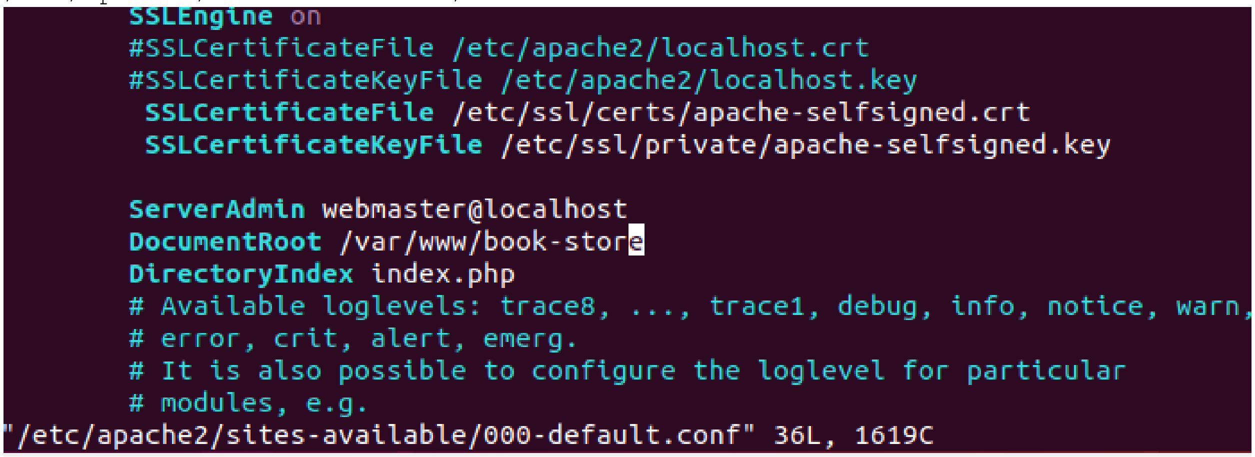


```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost
                ServerName localhost

                DocumentRoot /var/www/book-store
```



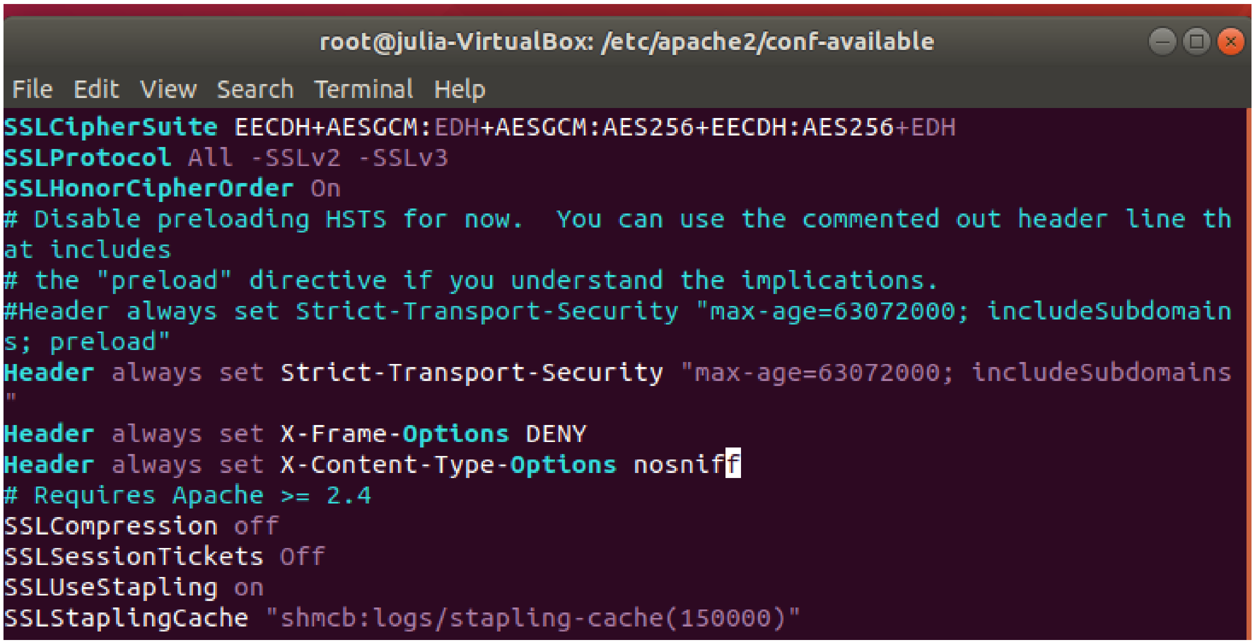
```
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on

        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2/README.Debian.gz for more info.
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
        SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
```

- /etc/apache2/sites-available/000-default.conf



```
        SSLEngine on
        #SSLCertificateFile /etc/apache2/localhost.crt
        #SSLCertificateKeyFile /etc/apache2/localhost.key
         SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
         SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/book-store
        DirectoryIndex index.php
        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
"/etc/apache2/sites-available/000-default.conf" 36L, 1619C
```

- /etc/apache2/conf-available/ssl-params.conf



root@julia-VirtualBox: /etc/apache2/conf-available
File Edit View Search Terminal Help

```
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
SSLProtocol All -SSLv2 -SSLv3
SSLHonorCipherOrder On
# Disable preloading HSTS for now.  You can use the commented out header line th
at includes
# the "preload" directive if you understand the implications.
#Header always set Strict-Transport-Security "max-age=63072000; includeSubdomain
s; preload"
Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains
"
Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff
# Requires Apache >= 2.4
SSLCompression off
SSLSessionTickets Off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
```

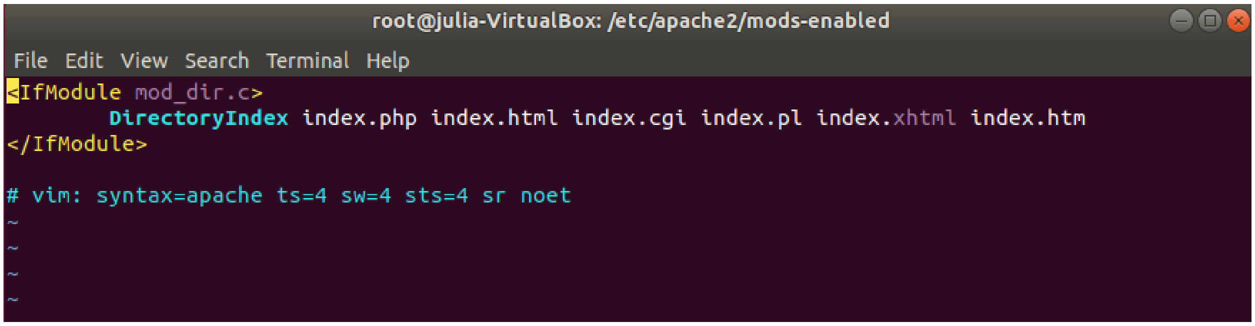
Below commands enables the SSL module and adds a symlink in the /etc/apache2/sites-enabled folder to the file /etc/apache2/sites-available/default-ssl.conf to include it into the active apache configuration. Then restart apache to enable the new configuration:

```
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

Change permanent redirect by changing this file:
vim /etc/apache2/sites-available/000-default.conf
```

Look for redirect and get screen shot of this file:
`/etc/apache2/sites-available/000-default.conf`

Enter this link into web browser and see if it works http://localhost/

Change the below file to look like the picture below:
`vim /etc/apache2/mods-enabled/dir.conf`


root@julia-VirtualBox: /etc/apache2/mods-enabled
File Edit View Search Terminal Help

```
<IfModule mod_dir.c>
        DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
~
~
~
~
```

```
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 on 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

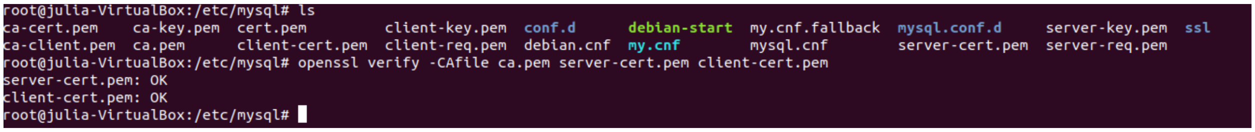
Set up ssl on the mysql…

Commands to create certificates using openssl:

```
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
```

Verifiy the certificates are valid:

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



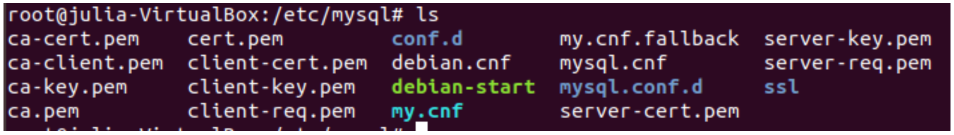
```
root@julia-VirtualBox:/etc/mysql# ls
ca-cert.pem    ca-key.pem  cert.pem         client-key.pem  conf.d      debian-start  my.cnf.fallback  mysql.conf.d     server-key.pem  ssl
ca-client.pem  ca.pem      client-cert.pem  client-req.pem  debian.cnf  my.cnf        mysql.cnf        server-cert.pem  server-req.pem
root@julia-VirtualBox:/etc/mysql# openssl verify -CAfile ca.pem server-cert.pem client-cert.pem
server-cert.pem: OK
client-cert.pem: OK
root@julia-VirtualBox:/etc/mysql#
```

Now that the certificates have been made 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/
```



```
root@julia-VirtualBox:/etc/mysql# ls
ca-cert.pem    cert.pem         conf.d        my.cnf.fallback  server-key.pem
ca-client.pem  client-cert.pem  debian.cnf    mysql.cnf        server-req.pem
ca-key.pem     client-key.pem   debian-start  mysql.conf.d     ssl
ca.pem         client-req.pem   my.cnf        server-cert.pem
```

Change directory to /etc/cert/

```
cd /etc/cert/
```

Change permission on the certificate files:

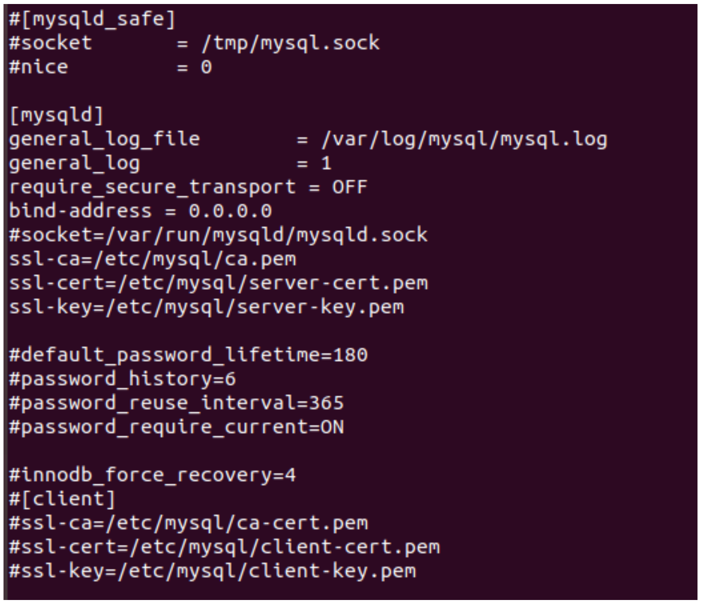
```
chown mysql:mysql cert.pem
chown mysql:mysql ca-cert.pem
chown mysql:mysql client-cert.pem
```

Configure mysql to use the certificates just created:

```
vim /etc/my.cnf
```

update value for bind-address = 0.0.0.0
# Type your own certificates directory
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 local host and not remote server to connect)
Example of what the file should look like is below:



```
#[mysqld_safe]
#socket         = /tmp/mysql.sock
#nice           = 0

[mysqld]
general_log_file        = /var/log/mysql/mysql.log
general_log             = 1
require_secure_transport = OFF
bind-address = 0.0.0.0
#socket=/var/run/mysqld/mysqld.sock
ssl-ca=/etc/mysql/ca.pem
ssl-cert=/etc/mysql/server-cert.pem
ssl-key=/etc/mysql/server-key.pem

#default_password_lifetime=180
#password_history=6
#password_reuse_interval=365
#password_require_current=ON

#innodb_force_recovery=4
#[client]
#ssl-ca=/etc/mysql/ca-cert.pem
#ssl-cert=/etc/mysql/client-cert.pem
#ssl-key=/etc/mysql/client-key.pem
```

Save and close
Restart mysql

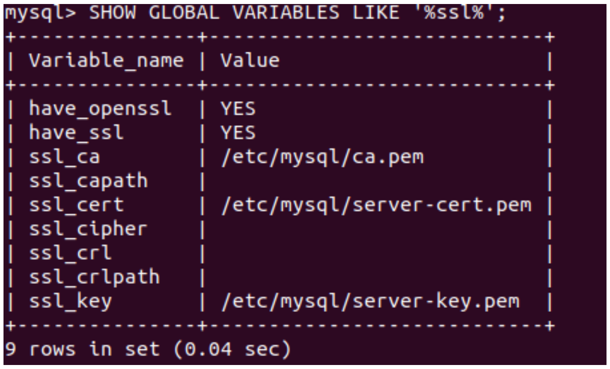
```
systemctl restart mysqld
```

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;
```



```
mysql> SHOW GLOBAL VARIABLES LIKE '%ssl%';
+---------------+----------------------------+
| Variable_name | Value                      |
+---------------+----------------------------+
| have_openssl  | YES                        |
| have_ssl      | YES                        |
| ssl_ca        | /etc/mysql/ca.pem          |
| ssl_capath    |                            |
| ssl_cert      | /etc/mysql/server-cert.pem |
| ssl_cipher    |                            |
| ssl_crl       |                            |
| ssl_crlpath   |                            |
| ssl_key       | /etc/mysql/server-key.pem  |
+---------------+----------------------------+
9 rows in set (0.04 sec)
```

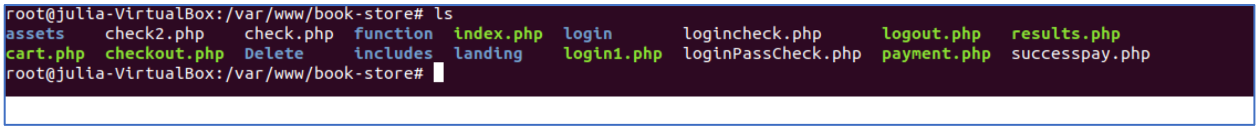
Now MySLQ is using the certs we generated with openSSL.

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/

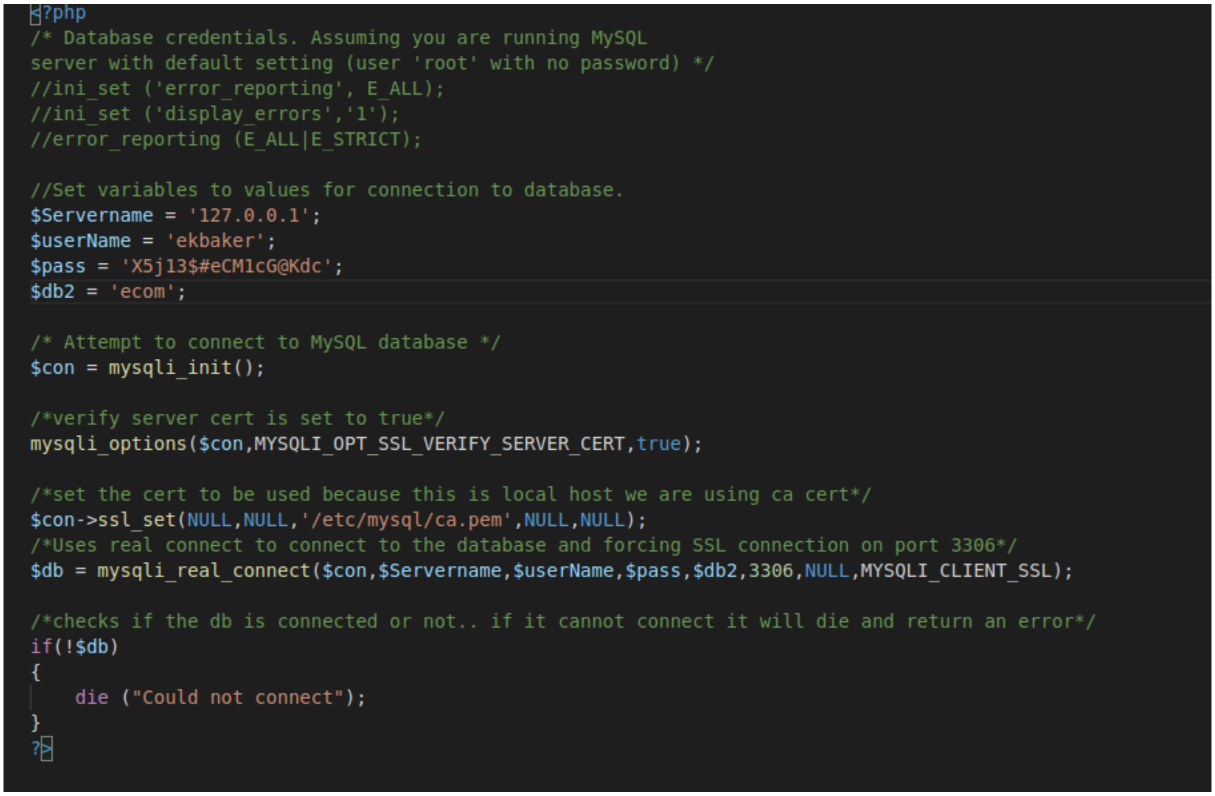
Copy the backup of the webwsite to directory giving in the path below in the picture.
/var/www/book-store/



```
root@julia-VirtualBox:/var/www/book-store# ls
assets    check2.php    check.php  function  index.php  login       logincheck.php      logout.php   results.php
cart.php  checkout.php  Delete     includes  landing    login1.php  loginPassCheck.php  payment.php  successpay.php
root@julia-VirtualBox:/var/www/book-store# 
```

One file will need to be changed in order to connect to mysql if a different cert name was used other than the one on original website.
This is the conf.php file which holds the database connection information and the path to the cert to be used for authentication. Update path and/or cert name. Since this is localhost we could not set it up as we would a remote server. Since this is running on same server php and mysql doesn't support localhost configurations. So a workround 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 help all the server.key, client.key and ca,pem file in one.



```php
<?php
/* Database credentials. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
//ini_set ('error_reporting', E_ALL);
//ini_set ('display_errors','1');
//error_reporting (E_ALL|E_STRICT);

//Set variables to values for connection to database.
$Servername = '127.0.0.1';
$userName = 'ekbaker';
$pass = 'X5j13$#eCM1cG@Kdc';
$db2 = 'ecom';

/* Attempt to connect to MySQL database */
$con = mysqli_init();

/*verify server cert is set to true*/
mysqli_options($con,MYSQLI_OPT_SSL_VERIFY_SERVER_CERT,true);

/*set the cert to be used because this is local host we are using ca cert*/
$con->ssl_set(NULL,NULL,'/etc/mysql/ca.pem',NULL,NULL);
/*Uses real connect to connect to the database and forcing SSL connection on port 3306*/
$db = mysqli_real_connect($con,$Servername,$userName,$pass,$db2,3306,NULL,MYSQLI_CLIENT_SSL);

/*checks if the db is connected or not.. if it cannot connect it will die and return an error*/
if(!$db)
{
    die ("Could not connect");
}
?>
```

Install tools to help with monitoring to help catch security insidents:

Sendmail:

Install Sendmail:

```
apt-get install Sendmail
```

Configure Sendmail files to work with gmail:

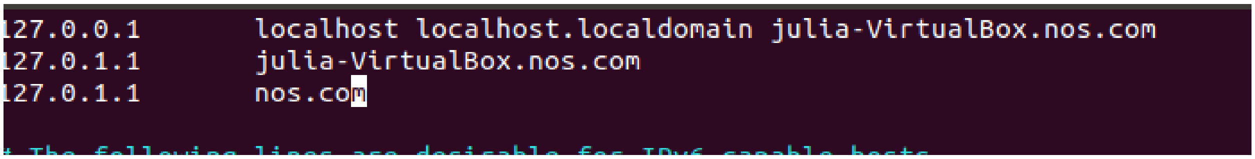
[Configure Sendmail to Relay Emails through Gmail SMTP – TecAdmin](#)
[Configuring Gmail as Sendmail email relay - LinuxConfig.org](#)
https://linuxconfig.org/configuring-gmail-as-Sendmail-email-relayhttps://linuxconfig.org/configuring-gmail-as-Sendmail-email-relay

The /etc/hosts/ file needs to be updated to resolve the error message while sending emails via send mail 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 made up domain .



```
127.0.0.1       localhost localhost.localdomain julia-VirtualBox.nos.com
127.0.1.1       julia-VirtualBox.nos.com
127.0.1.1       nos.com

# The following lines are desirable for IPv6 capable hosts
```

Save and Close file

Change server name to fully qualified name. So send mail doesn’t give error that the server isn’t in domain.

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

vim /etc/hostname

Add the domain name that we made up to the end of the server name in this file:

Example:

ServerName.nos.com

Restart the server..

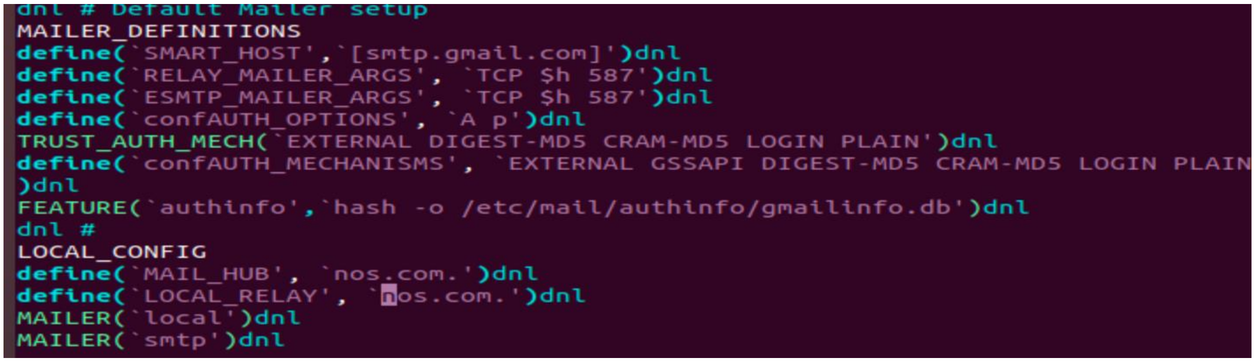
Now check your host name by typing the command below:

hostname

The files that will need modified is the Sendmail.mc

Add lines from example below to LOCAL_CONFIG to the /etc/mail/Sendmail.mc at the bottom. In this file we also need to update FEATURE path to /etc/mail/authinfo/gmailinfo.db.. to get crediential for Sendmail to work with gmail.

```
vim /etc/mail/Sendmail.mc
```

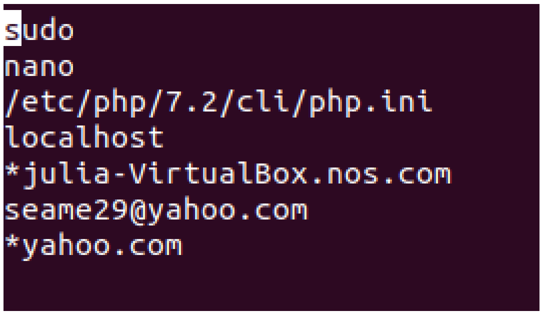


```
dnl # Default Mailer setup
MAILER_DEFINITIONS
define(`SMART_HOST',`[smtp.gmail.com]')dnl
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl
define(`ESMTP_MAILER_ARGS', `TCP $h 587')dnl
define(`confAUTH_OPTIONS', `A p')dnl
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN
)dnl
FEATURE(`authinfo',`hash -o /etc/mail/authinfo/gmailinfo.db')dnl
dnl #
LOCAL_CONFIG
define(`MAIL_HUB', `nos.com.')dnl
define(`LOCAL_RELAY', `nos.com.')dnl
MAILER(`local')dnl
MAILER(`smtp')dnl
```

The /etc/mail/local-host-names file will need to be modified to allow php, emails,localhost, and server name used on this server to be allowed to send emails from mail. This will mitigate the errors received while sending emails with Sendmail that whatever 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
```



```
sudo
nano
/etc/php/7.2/cli/php.ini
localhost
*julia-VirtualBox.nos.com
seame29@yahoo.com
*yahoo.com
```

Create a folder named authinfo

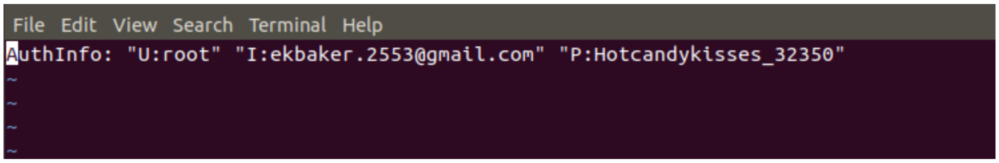
```
mkdir /etc/mail/authinfo
```

```
cd /etc/mail/authinfo/
```

Inside this folder I created a file named gmailinfo..

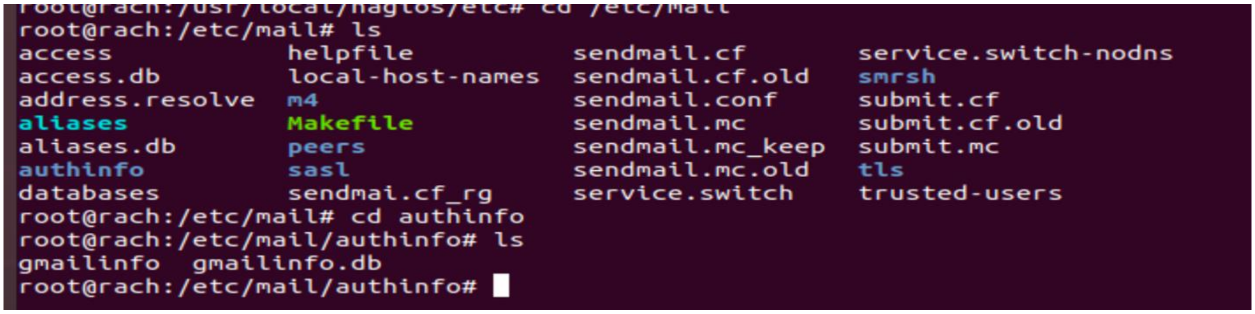
```
vim gmailinfo
```

Add the following information to the file:



```
AuthInfo: "U:root" "I:ekbaker.2553@gmail.com" "P:Hotcandykisses_32350"
~
~
~
~
```

Save and cloes file.



```
root@rach:/usr/local/nagios/etc# cd /etc/mail
root@rach:/etc/mail# ls
access           helpfile          sendmail.cf       service.switch-nodns
access.db        local-host-names  sendmail.cf.old   smrsh
address.resolve  m4                sendmail.conf     submit.cf
aliases          Makefile          sendmail.mc       submit.cf.old
aliases.db       peers             sendmail.mc_keep  submit.mc
authinfo         sasl              sendmail.mc.old   tls
databases        sendmai.cf_rg     service.switch    trusted-users
root@rach:/etc/mail# cd authinfo
root@rach:/etc/mail/authinfo# ls
gmailinfo  gmailinfo.db
root@rach:/etc/mail/authinfo# 
```

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]()
[How To Install Nagios 4 and Monitor Your Servers on Ubuntu 18.04 | DigitalOcean]()

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 issues it will send us emails and alert us of any events…

Configure Nagios To Use Sendmail – Brandon Wamboldt

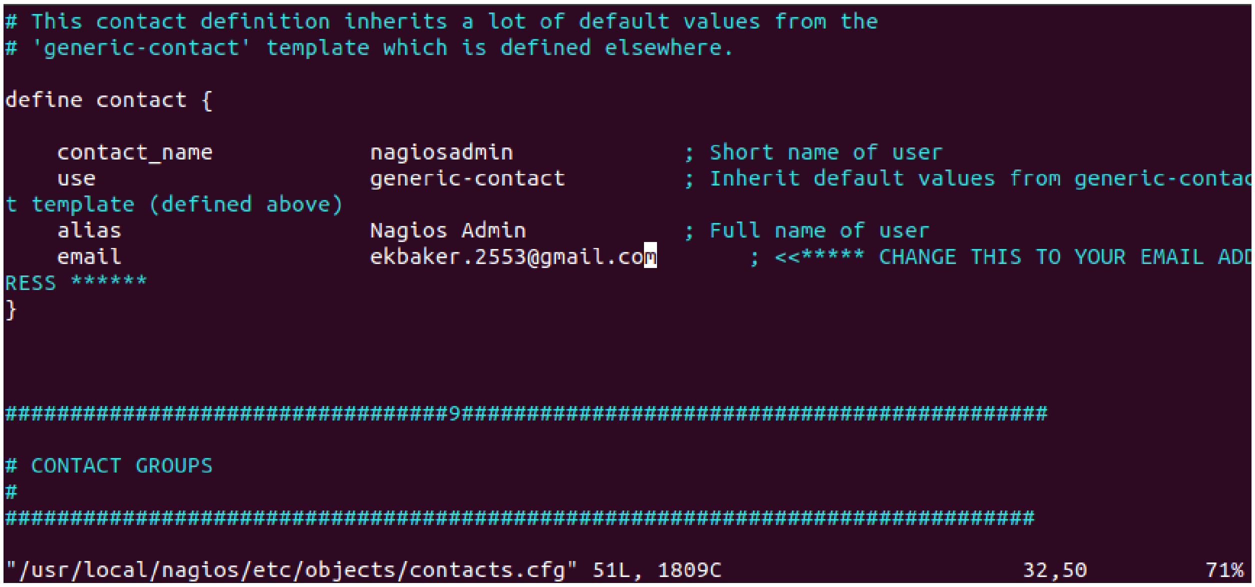
Nagios alerts via email

9.2. Configuring Nagios Server to Send Mail Notifications Red Hat Gluster Storage 3 | Red Hat Customer Portal

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:

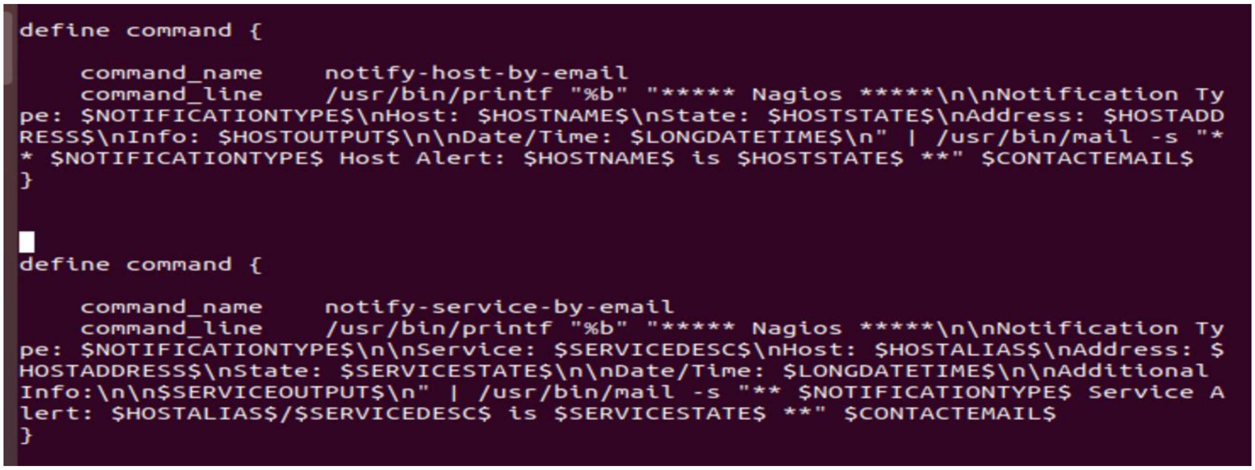


```
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {

    contact_name            nagiosadmin             ; Short name of user
    use                     generic-contact         ; Inherit default values from generic-contac
t template (defined above)
    alias                   Nagios Admin            ; Full name of user
    email                   ekbaker.2553@gmail.com       ; <<***** CHANGE THIS TO YOUR EMAIL ADD
RESS ******
}



##################################9#############################################

# CONTACT GROUPS
#
###############################################################################

"/usr/local/nagios/etc/objects/contacts.cfg" 51L, 1809C                       32,50         71%
```

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

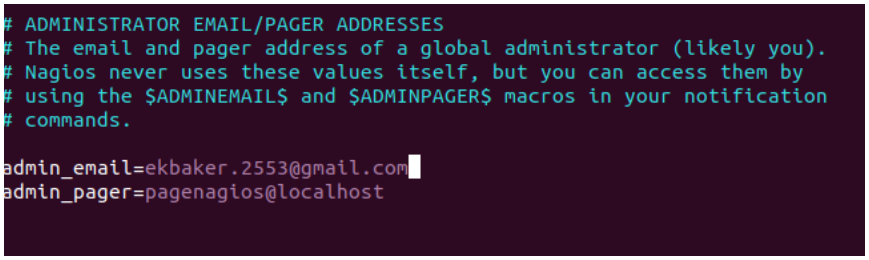
Vim /usr/local/nagios/etc/objects/command.cfg



```
define command {

    command_name    notify-host-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Ty
pe: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADD
RESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /usr/bin/mail -s "*
* $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ **" $CONTACTEMAIL$
}


define command {

    command_name    notify-service-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Ty
pe: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $
HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional
Info:\n\n$SERVICEOUTPUT$\n" | /usr/bin/mail -s "** $NOTIFICATIONTYPE$ Service A
lert: $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ **" $CONTACTEMAIL$
}
```

Save and exit the editor

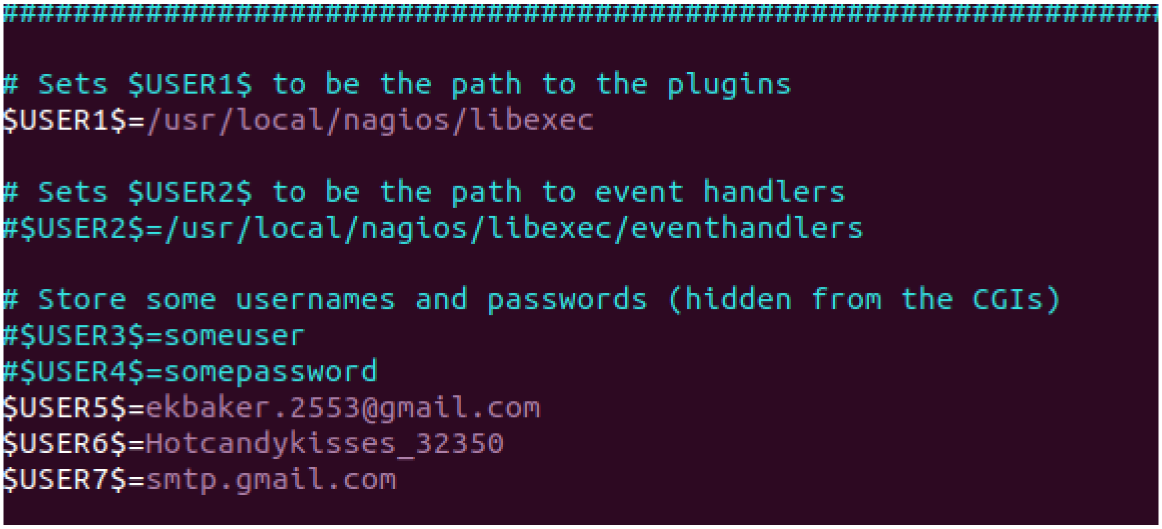
```
Localhost# vim /usr/local/nagios/etc/nagios.cfg
```



```
# ADMINISTRATOR EMAIL/PAGER ADDRESSES
# The email and pager address of a global administrator (likely you).
# Nagios never uses these values itself, but you can access them by
# using the $ADMINEMAIL$ and $ADMINPAGER$ macros in your notification
# commands.

admin_email=ekbaker.2553@gmail.com
admin_pager=pagenagios@localhost
```

```
Save and Exit
```

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



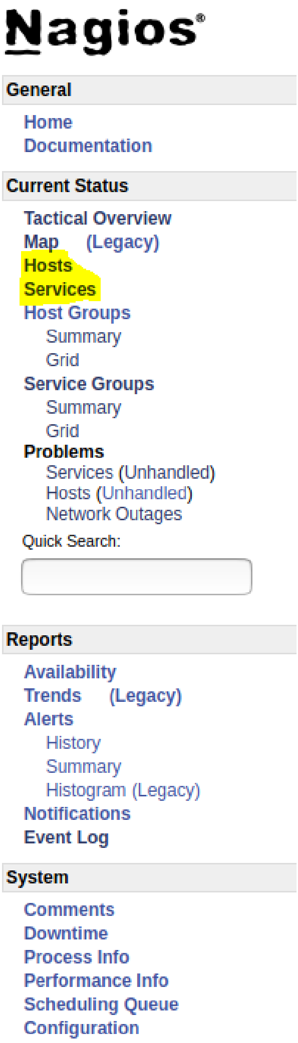
```
################################################################

# Sets $USER1$ to be the path to the plugins
$USER1$=/usr/local/nagios/libexec

# Sets $USER2$ to be the path to event handlers
#$USER2$=/usr/local/nagios/libexec/eventhandlers

# Store some usernames and passwords (hidden from the CGIs)
#$USER3$=someuser
#$USER4$=somepassword
$USER5$=ekbaker.2553@gmail.com
$USER6$=Hotcandykisses_32350
$USER7$=smtp.gmail.com
```

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

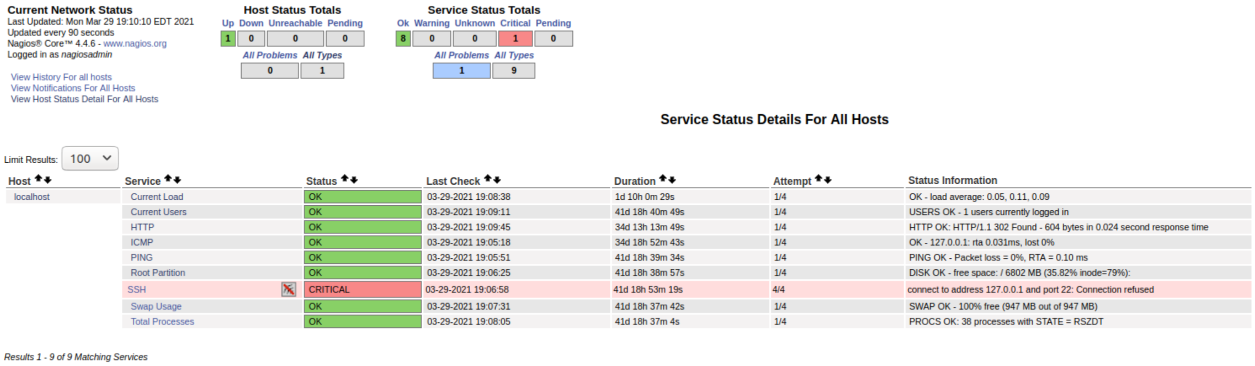
Test if email is working through nagios choose Services:

[Test Email Alert - View topic • Nagios Support Forum]

Home page:



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



Current Network Status
Last Updated: Mon Mar 29 19:10:10 EDT 2021
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as *nagiosadmin*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

All Problems	*All Types*
0	1

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
8	0	0	1	0

All Problems	*All Types*
1	9

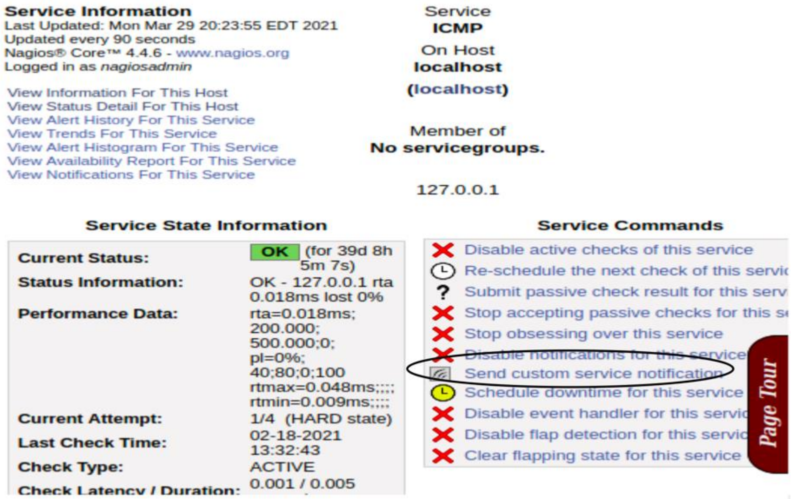
Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	03-29-2021 19:08:38	1d 10h 0m 29s	1/4	OK - load average: 0.05, 0.11, 0.09
	Current Users	OK	03-29-2021 19:09:11	41d 18h 40m 49s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	03-29-2021 19:09:45	34d 13h 13m 49s	1/4	HTTP OK: HTTP/1.1 302 Found - 604 bytes in 0.024 second response time
	ICMP	OK	03-29-2021 19:05:18	34d 18h 52m 43s	1/4	OK - 127.0.0.1: rta 0.031ms, lost 0%
	PING	OK	03-29-2021 19:05:51	41d 18h 39m 34s	1/4	PING OK - Packet loss = 0%, RTA = 0.10 ms
	Root Partition	OK	03-29-2021 19:06:25	41d 18h 38m 57s	1/4	DISK OK - free space: / 6802 MB (35.82% inode=79%):
	SSH	CRITICAL	03-29-2021 19:06:58	41d 18h 53m 19s	4/4	connect to address 127.0.0.1 and port 22: Connection refused
	Swap Usage	OK	03-29-2021 19:07:31	41d 18h 37m 42s	1/4	SWAP OK - 100% free (947 MB out of 947 MB)
	Total Processes	OK	03-29-2021 19:08:05	41d 18h 37m 4s	1/4	PROCS OK: 38 processes with STATE = RSZDT

Results 1 - 9 of 9 Matching Services

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



Service Information
Last Updated: Mon Mar 29 20:23:55 EDT 2021
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as *nagiosadmin*

View Information For This Host
View Status Detail For This Host
View Alert History For This Service
View Trends For This Service
View Alert Histogram For This Service
View Availability Report For This Service
View Notifications For This Service

Service
ICMP
On Host
localhost
(localhost)

Member of
No servicegroups.

127.0.0.1

Service State Information

Current Status:	OK (for 39d 8h 5m 7s)
Status Information:	OK - 127.0.0.1 rta 0.018ms lost 0%

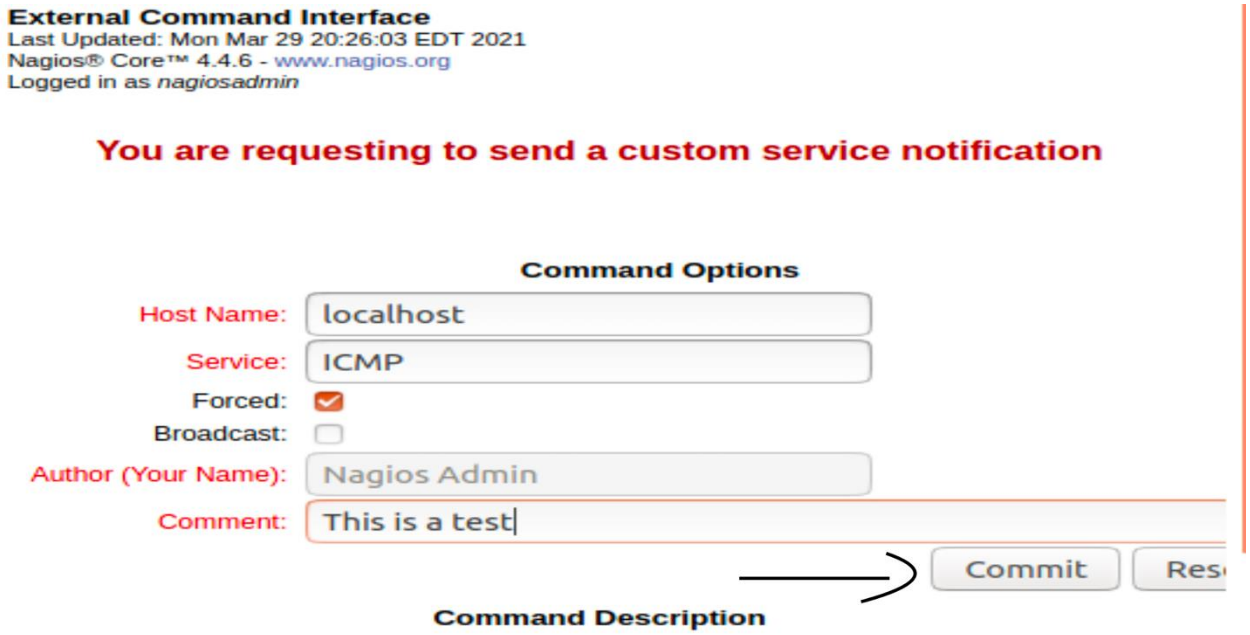
Performance Data:	rta=0.018ms;
	200.000;
	500.000;0;
	pl=0%;
	40;80;0;100
	rtmax=0.048ms;;;;
	rtmin=0.009ms;;;;

Current Attempt:	1/4 (HARD state)
Last Check Time:	02-18-2021 13:32:43
Check Type:	ACTIVE
Check Latency / Duration:	0.001 / 0.005

Service Commands

- ✗ Disable active checks of this service
- Re-schedule the next check of this servic
- ? Submit passive check result for this serv
- ✗ Stop accepting passive checks for this se
- ✗ Stop obsessing over this service
- ✗ Disable notifications for this service
- Send custom service notification
- Schedule downtime for this service
- ✗ Disable event handler for this servic
- ✗ Disable flap detection for this servic
- ✗ Clear flapping state for this service

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



External Command Interface
Last Updated: Mon Mar 29 20:26:03 EDT 2021
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as *nagiosadmin*

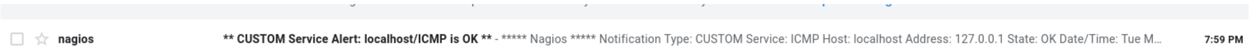
You are requesting to send a custom service notification

Command Options

Host Name:	localhost
Service:	ICMP
Forced:	☑
Broadcast:	☐
Author (Your Name):	Nagios Admin
Comment:	This is a test

Command Description

Shortly an email should appear lke the one below:



☐ ☆ **nagios** **** CUSTOM Service Alert: localhost/ICMP is OK **** - ***** Nagios ***** Notification Type: CUSTOM Service: ICMP Host: localhost Address: 127.0.0.1 State: OK Date/Time: Tue M... **7:59 PM**

Keepass2
https://linuxhint.com/install_keepass_ubuntu/

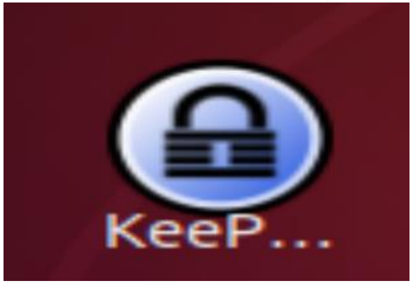
Install keepass2

```
apt-get install keepass2 -y
```

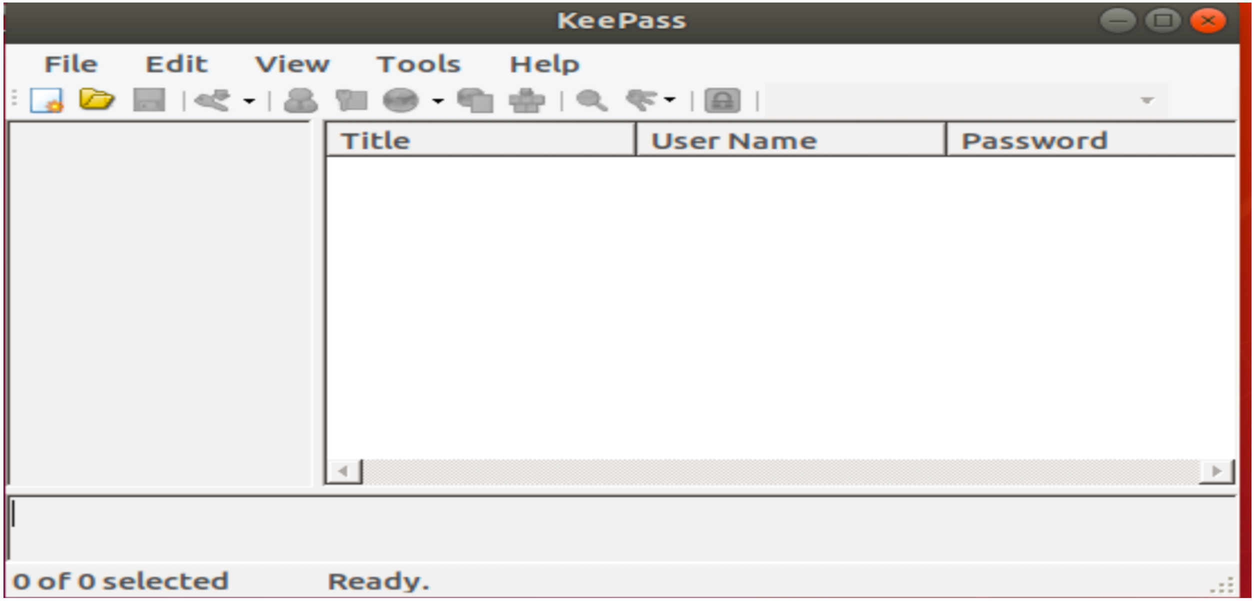
Setup Keepass2 for use:

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

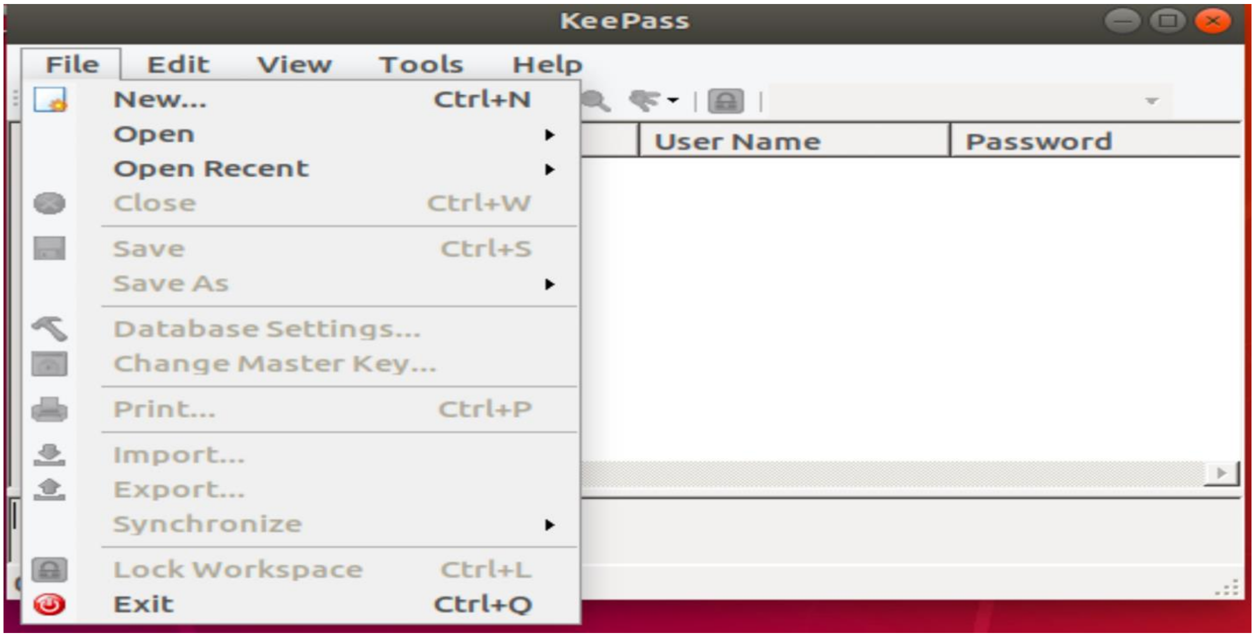
Left click the this icon in your applications folder:



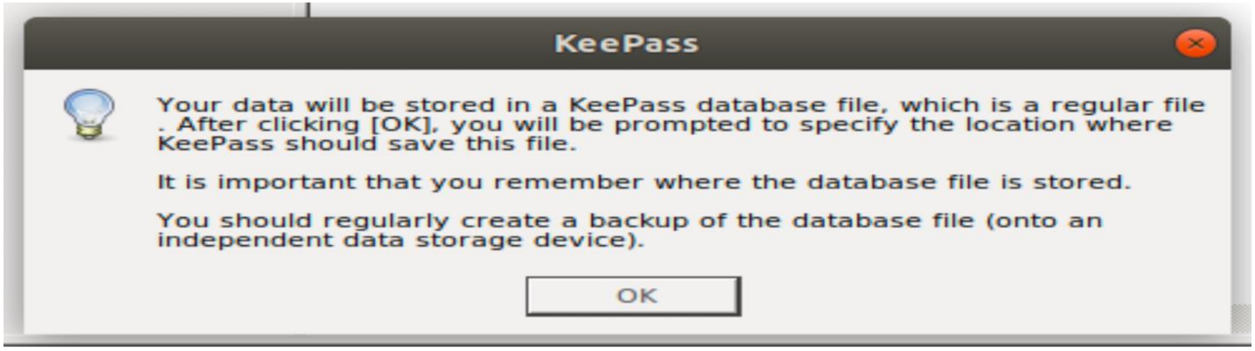
This will bring up a screen like below:



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:



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

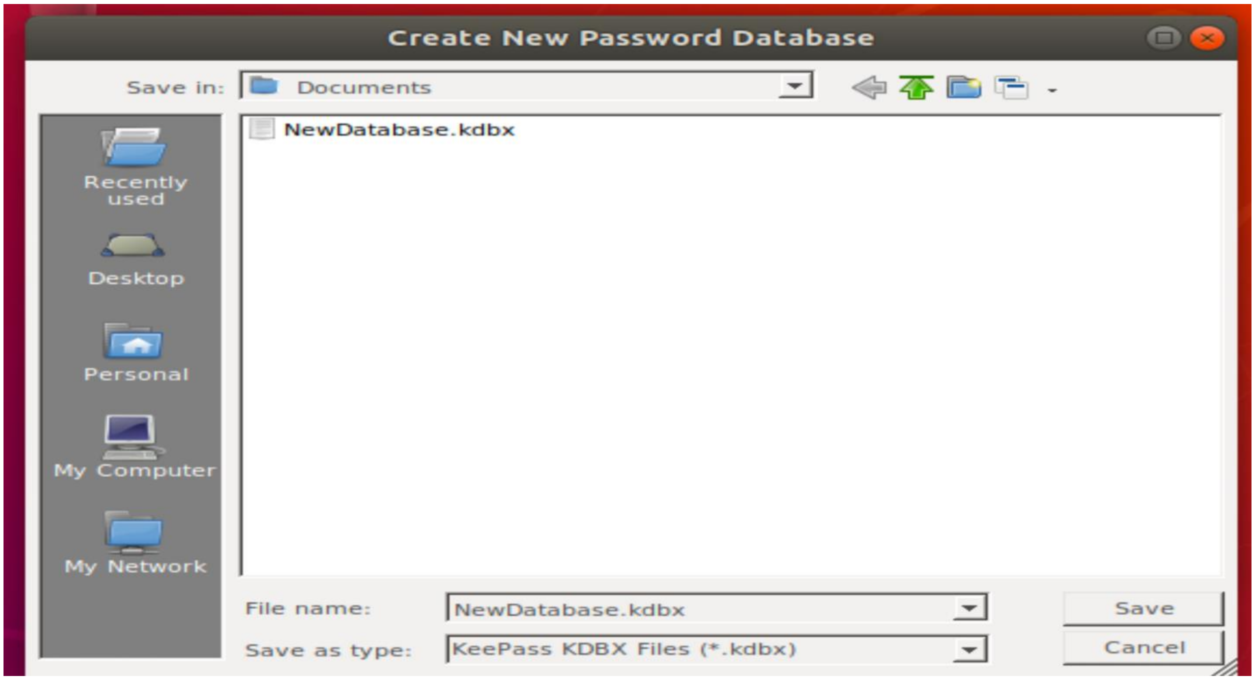


Your data will be stored in a KeePass database file, which is a regular file . After clicking [OK], you will be prompted to specify the location where KeePass should save this file.

It is important that you remember where the database file is stored.

You should regularly create a backup of the database file (onto an independent data storage device).

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.

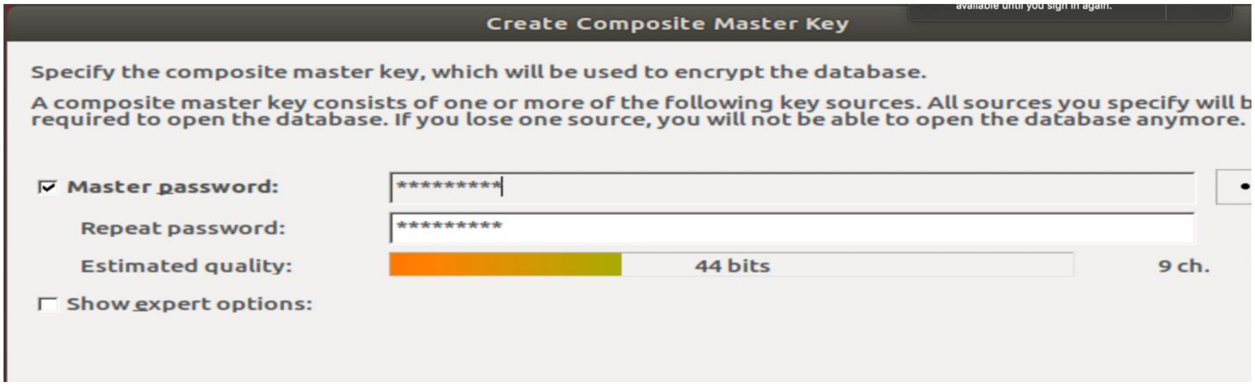


Save in: Documents

File name: NewDatabase.kdbx

Save as type: KeePass KDBX Files (*.kdbx)

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.



available until you sign in again.
Create Composite Master Key

Specify the composite master key, which will be used to encrypt the database.

A composite master key consists of one or more of the following key sources. All sources you specify will b required to open the database. If you lose one source, you will not be able to open the database anymore.

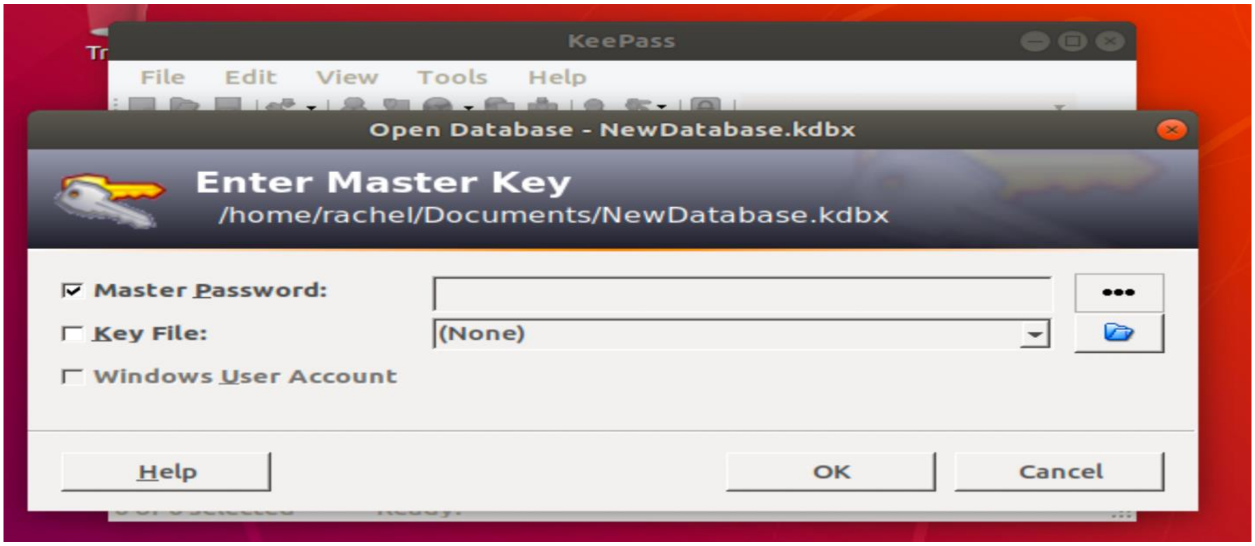
☑ Master password: *********

Repeat password: *********

Estimated quality: 44 bits 9 ch.

☐ Show expert options:

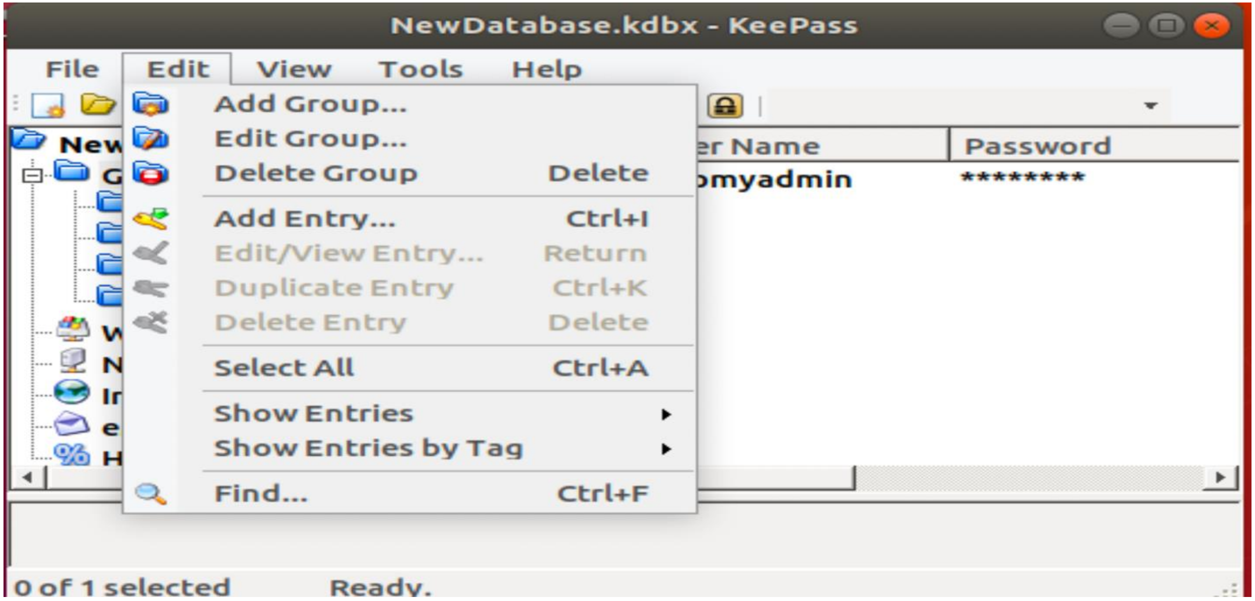
Now to log into keepass with the password just created. Then click ok at the bottom right hand side of the Enter Master Key screen.



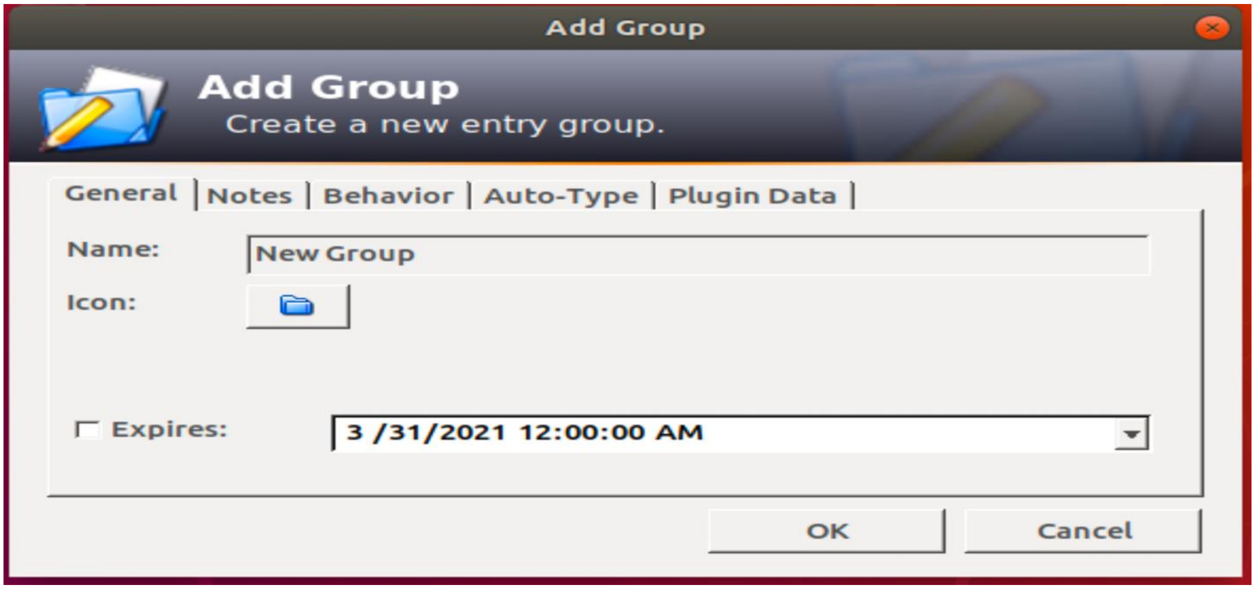
Enter Master Key
/home/rachel/Documents/NewDatabase.kdbx

☑ Master Password:
□ Key File: (None)
□ Windows User Account

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:



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.

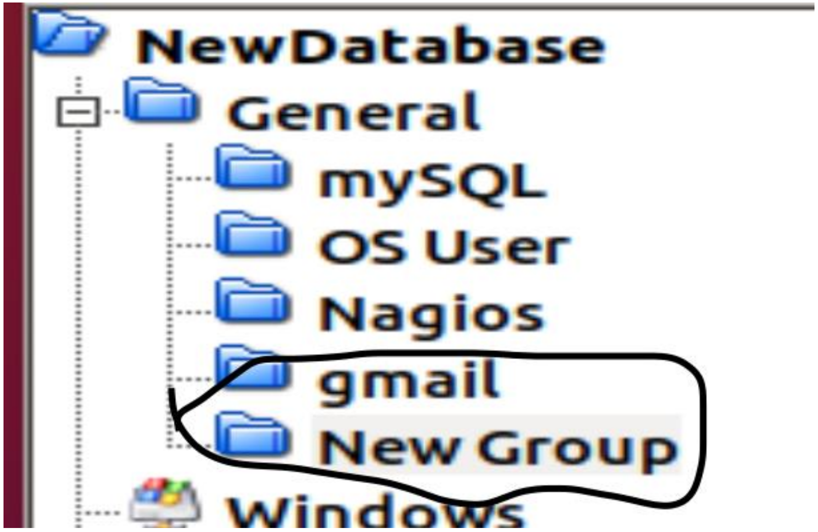


Name: New Group

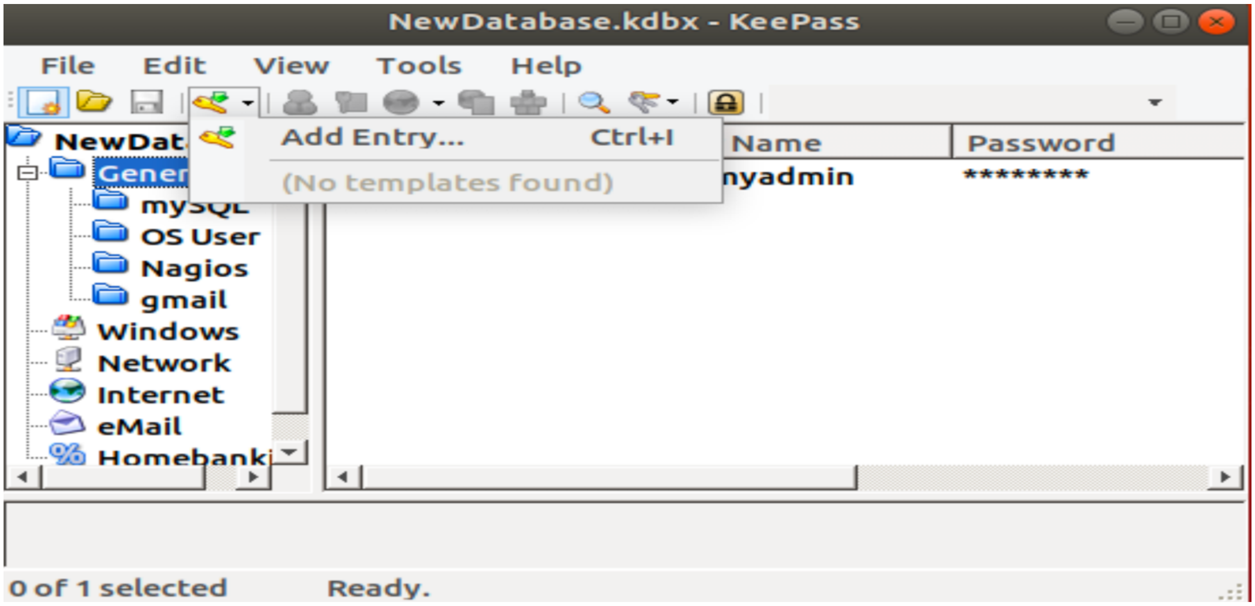
Icon:

□ Expires: 3 /31/2021 12:00:00 AM

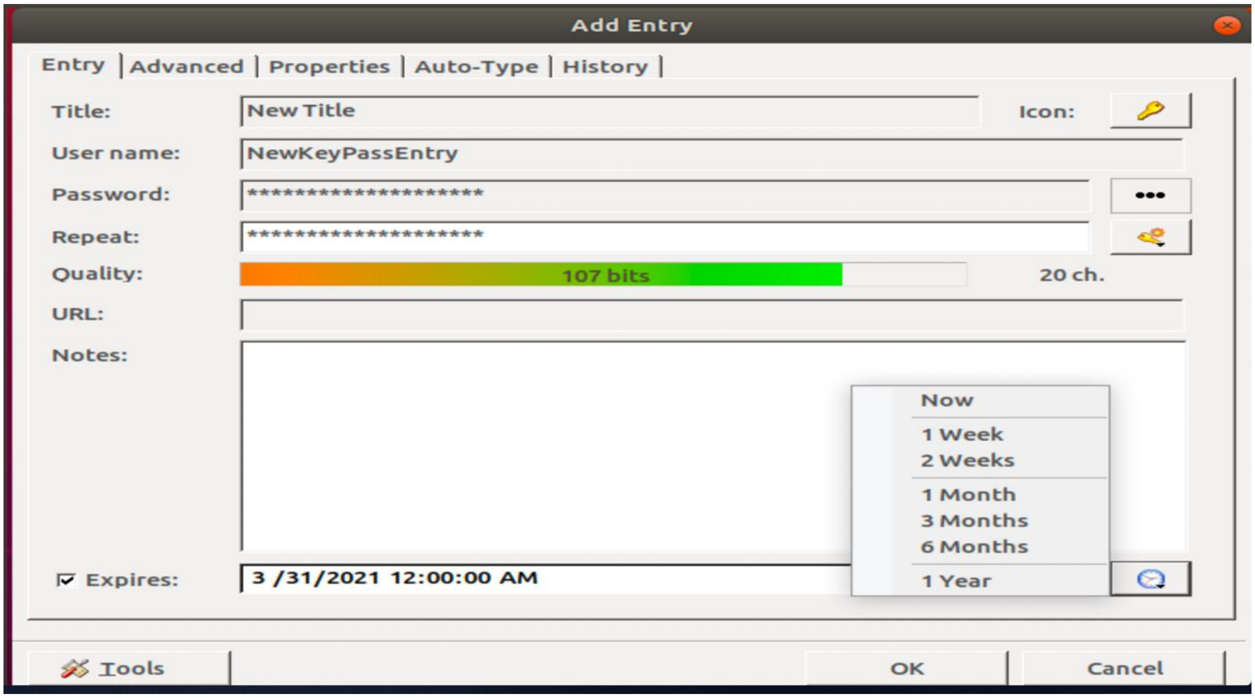
Now that our group is created New Group we want to add users with their passwords to this group. Highlight New Group under the General Group we just created.



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



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.


Add Entry

Title: New Title

User name: NewKeyPassEntry

Password: ********************

Repeat: ********************

Quality: 107 bits 20 ch.

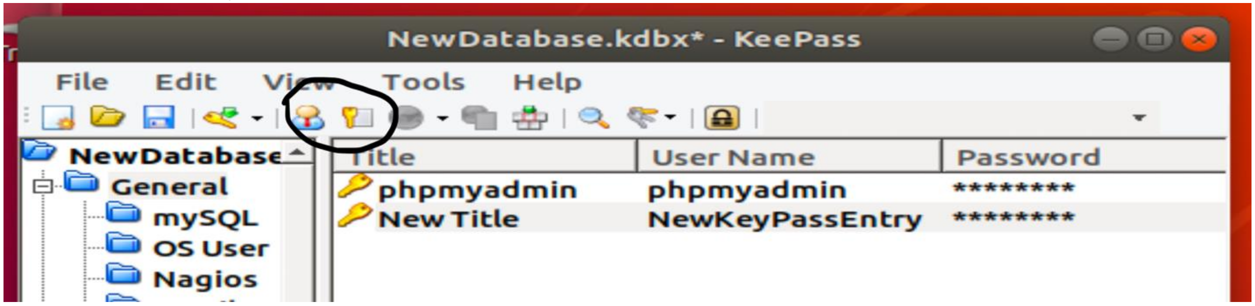
URL:

Notes:

☑ Expires: 3 /31/2021 12:00:00 AM

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.



Title	User Name	Password
phpmyadmin	phpmyadmin	********
New Title	NewKeyPassEntry	********

Now keepass is all setup and ready for use.

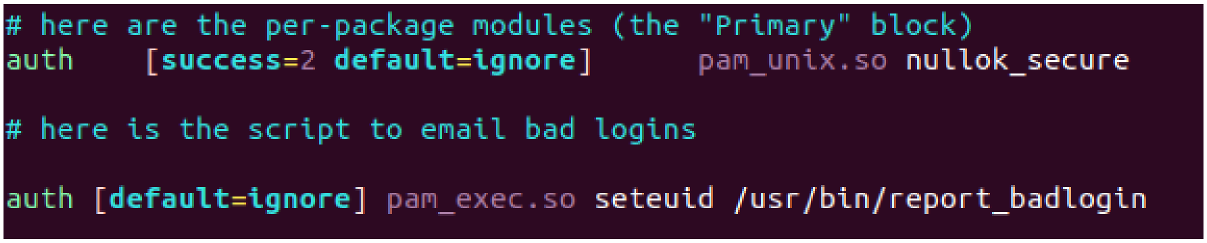
Setup bashfile to send emails if login failure occurs.
https://unix.stackexchange.com/questions/339417/running-a-shell-script-on-n-failed-login-attempts

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 picture.

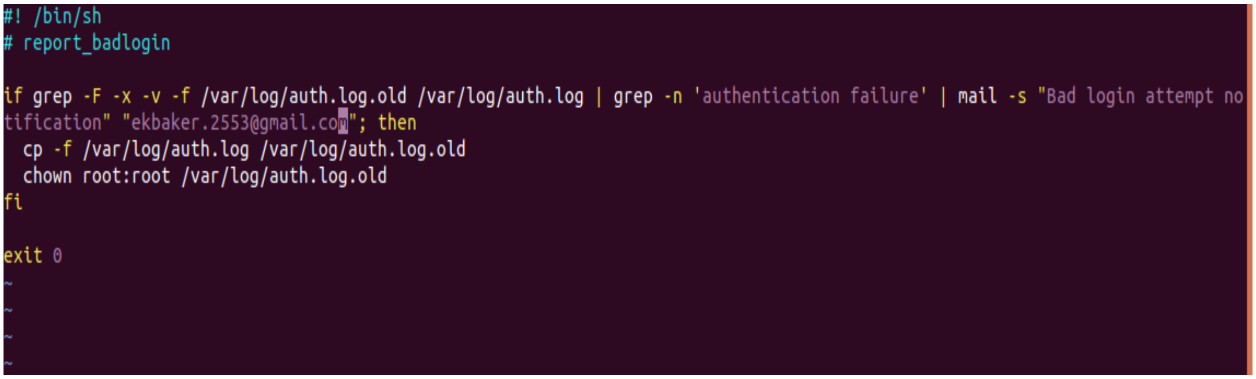
```
vim /etc/pam.d/sshd
```



```
# here are the per-package modules (the "Primary" block)
auth    [success=2 default=ignore]      pam_unix.so nullok_secure

# here is the script to email bad logins

auth [default=ignore] pam_exec.so seteuid /usr/bin/report_badlogin
```

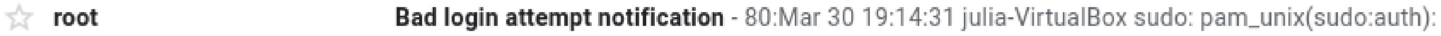
Create bash file named report_badlogin exactly as the example picture below. This file will check if there is a new failed login and then sends an email if there is.

```
vim /usr/bin/report_badlogin
```



```
#! /bin/sh
# report_badlogin

if grep -F -x -v -f /var/log/auth.log.old /var/log/auth.log | grep -n 'authentication failure' | mail -s "Bad login attempt no
tification" "ekbaker.2553@gmail.com"; then
  cp -f /var/log/auth.log /var/log/auth.log.old
  chown root:root /var/log/auth.log.old
fi

exit 0
~
~
~
~
```

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



☆ root **Bad login attempt notification** - 80:Mar 30 19:14:31 julia-VirtualBox sudo: pam_unix(sudo:auth):

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
```