Install SimpleRisk on RHEL 8

Introduction

SimpleRisk is a simple and free tool to perform risk management activities. Based entirely on open source technologies and sporting a Mozilla Public License 2.0, a SimpleRisk instance can be stood up in minutes and instantly provides the security professional with the ability to submit risks, plan mitigations, facilitate management reviews, prioritize for project planning, and track regular reviews. It is highly configurable and includes dynamic reporting and the ability to tweak risk formulas on the fly. It is under active development with new features being added all the time and can be downloaded for free or demoed at https://www.simplerisk.it/.

Disclaimer

The lucky security professionals work for companies who can afford expensive GRC tools to aide in managing risk. The unlucky majority out there usually end up spending countless hours managing risk via spreadsheets. It's cumbersome, time consuming, and just plain sucks. When Josh Sokol started writing SimpleRisk, it was out of pure frustration with the other options out there. What he's put together is undoubtedly better than spreadsheets and gets you most of the way towards the "R" in GRC without breaking the bank. That said, humans can make mistakes, and therefore the SimpleRisk software is provided to you with no warranties expressed or implied. If you get stuck, you can always try sending an e-mail to support@simplerisk.it and we'll do our best to help you out. Also, while SimpleRisk was written by a security practitioner with security in mind, there is no way to promise that it is 100% secure. You accept that as a risk when using the software, but if you do find any issues, please report them to us so that we can fix them ASAP.

Installing Lamp

Once you have your RHEL 8 environment you must first decide if your going to do this with SELinux on or off and if you wish for it to be on follow this guide to set it up (https://simplerisk.freshdesk.com/solution/articles/6000053609-how-do-i-get-simplerisk-to-work-with-selinux-) if you wish to turn it off we need to edit the RHEL config by typing “vi /etc/selinux/config” and change the line “SELINUX=Enforcing” and make it ”SELINUX=Permissive” reboot RHEL and verify with “getenforce” if it returns permissive you can continue. We now start by installing a lamp server this is comprised of Apache, MySQL, and PHP. Open terminal and begin the following process to install LAMP

1) Type “sudo bash”. This will a require the root password made at installation
2) Type “yum update”. This will update your Linux environment.
3) Type “yum install httpd”. This installs apache
4) Type “systemctl enable httpd” and “systemctl start httpd” this will enable and start the apache process.
5) Next we install MariaDB a MySQL manager. Type “yum install mariadb-server” and then “systemctl enable mariadb” and finally “systemctl start mariadb” this will install, enable, and start MariaDB.
6) Finally type “mysql_secure_installation” to better secure your sql databases and environment.

The following should display and should answer accordingly this will also be where you set your MySQL database root password so store it and keep it handy for later

```
[root@localhost ~]# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!


By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
```

```
... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

8) Now we install PHP 7.2 and some of the most common PHP modules:
sudo yum install php php-common php-cli php-gd php-curl php-mysqlnd php-mbstring php-ldap php-dom php-curl php-zip php-gd

9) Next we need to edit /etc/httpd/conf.modules.d/00-mpm.conf. Find the following lines. In the config you will see the quoted lines below. Update according to description.

- Comment out "LoadModule mpm_event_module modules/mod_mpm_event.so"
- Uncomment "LoadModule mpm_prefork_module modules/mod_mpm_prefork.so"

10) The last PHP configuration we will need to make is to update the memory_limit set in the php.ini. Use the following steps to update accordingly.
Vi /etc/php.ini
Use “/memory_limit” to find the value.
Set “memory_limit =” to the value to “256Mb”.
Next we need to edit “max_input_vars” and set this to 3000. If there is a preceding “;” remove it.
Lastly, hit escape and do “/wq” to write out the file.

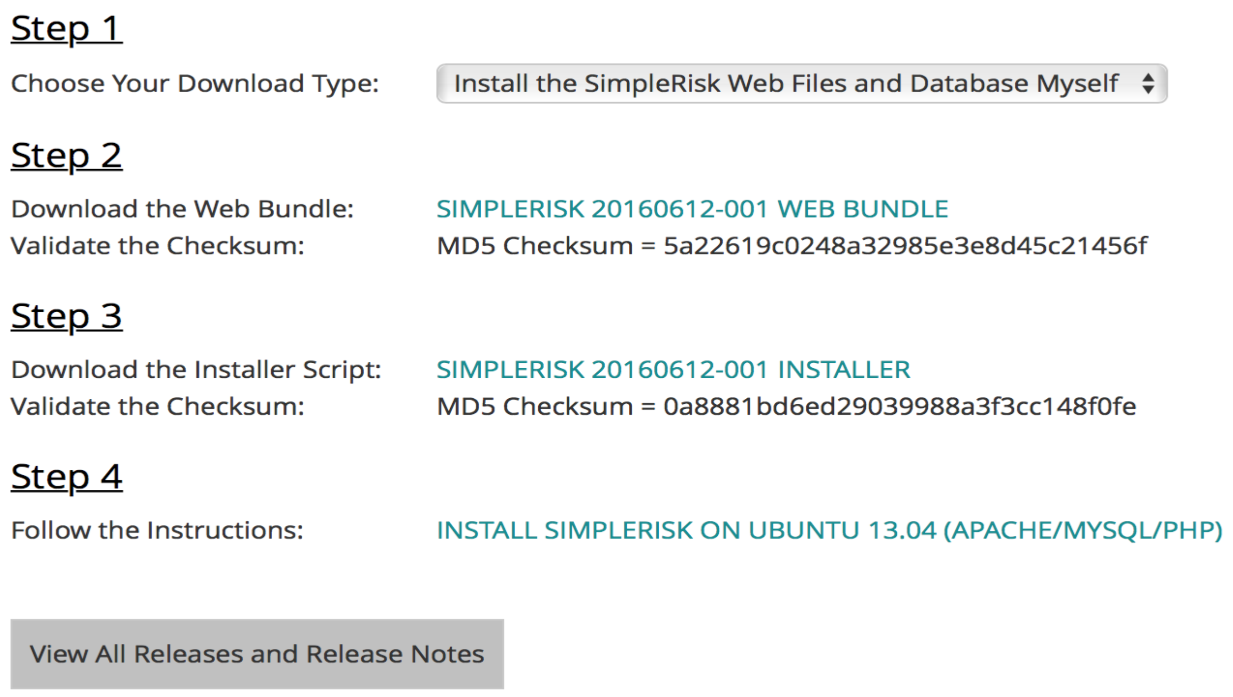
11) Verify the PHP installation, by typing the following command which will print the PHP version:
php -v

Installing SimpleRisk

This section entails downloading the software placing it properly setting the config files for the virtualhost as well as installation of the database. The following steps should result in a functional SimpleRisk environment to be spawned.

1) Obtaining the SimpleRisk Files

Open FireFox by clicking applications at the top left it should be listed under favorites it should also be listed under the Internet tab. Once FireFox loads, enter https://www.simplerisk.com/ into the URL bar to go to the SimpleRisk site. Click on the "Download" link at the top.



Step 1

Choose Your Download Type: Install the SimpleRisk Web Files and Database Myself

Step 2

Download the Web Bundle: SIMPLERISK 20160612-001 WEB BUNDLE
Validate the Checksum: MD5 Checksum = 5a22619c0248a32985e3e8d45c21456f

Step 3

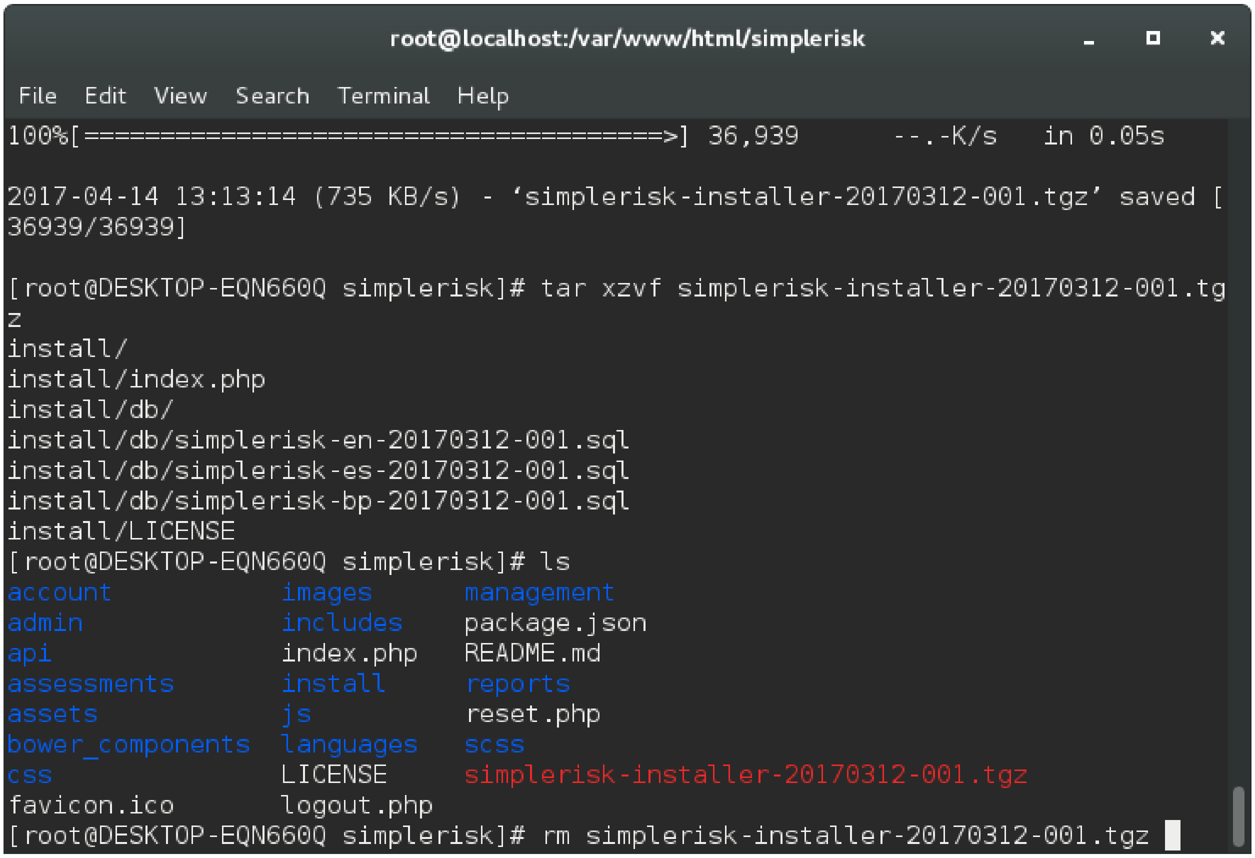
Download the Installer Script: SIMPLERISK 20160612-001 INSTALLER
Validate the Checksum: MD5 Checksum = 0a8881bd6ed29039988a3f3cc148f0fe

Step 4

Follow the Instructions: INSTALL SIMPLERISK ON UBUNTU 13.04 (APACHE/MYSQL/PHP)

Click to download and save both the Web Bundle and the Installer Script. Once you have the files downloaded, you can close the browser.

2) Now back in terminal type “cd /var/www/html/”
3) Next we extract the files with “tar xzvf ~/downloads/simplerisk-20210713-001.tgz”
4) Now remove leftover archive with “rm ~/downloads/simplerisk-20210713-001.tgz
5) Now we change directories to where we need the install script enter “cd /var/www/html/simplerisk”
6) Same as before we now extract the install files using “tar xzvf ~/downloads/simplerisk-installer-20170312-001.tgz”. Your SimpleRisk directory should now look like the following verify with “ls” and then remove the installer archive using “rm ~/downloads/simplerisk-installer-20170312-001.tgz”. After this the directory should look something like this.



root@localhost:/var/www/html/simplerisk
File Edit View Search Terminal Help

```
100%[======================================>] 36,939      --.-K/s   in 0.05s

2017-04-14 13:13:14 (735 KB/s) - ‘simplerisk-installer-20170312-001.tgz’ saved [
36939/36939]

[root@DESKTOP-EQN660Q simplerisk]# tar xzvf simplerisk-installer-20170312-001.tg
z
install/
install/index.php
install/db/
install/db/simplerisk-en-20170312-001.sql
install/db/simplerisk-es-20170312-001.sql
install/db/simplerisk-bp-20170312-001.sql
install/LICENSE
[root@DESKTOP-EQN660Q simplerisk]# ls
account           images     management
admin             includes   package.json
api               index.php  README.md
assessments       install    reports
assets            js         reset.php
bower_components  languages  scss
css               LICENSE    simplerisk-installer-20170312-001.tgz
favicon.ico       logout.php
[root@DESKTOP-EQN660Q simplerisk]# rm simplerisk-installer-20170312-001.tgz
```

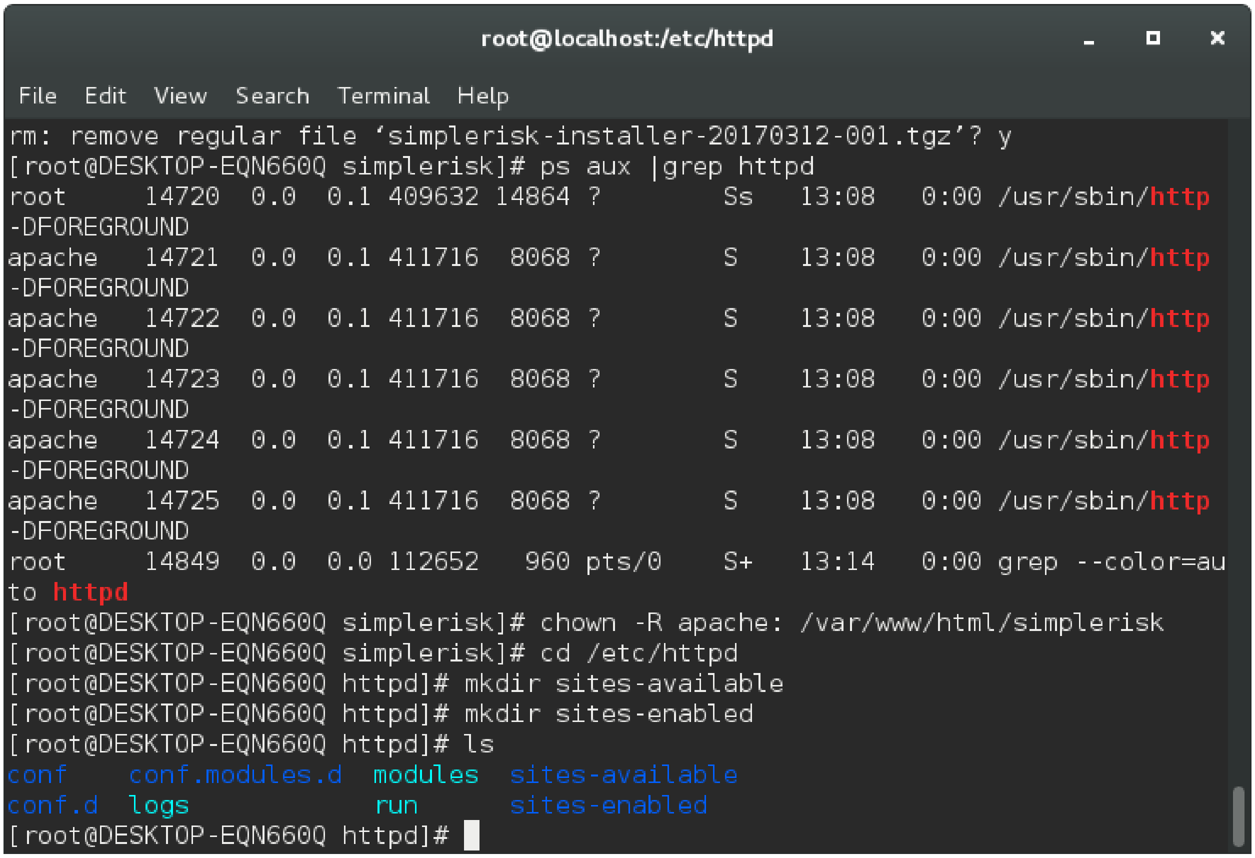
After these commands have gone through you should have all the necessary files for SimpleRisk. Next we need to set permissions and setup our virtualhost. We now set

ownership of the necessary folders to the user running httpd, normally this is defined as the user "apache". You can check with the following command "ps aux |grep httpd".
The first word in the 3rd line of output should be your httpd user.

Next change the owner of the SimpleRisk directory to apache using "chown -R apache: /var/www/html/simplerisk/"

Next we need to set httpd to set up our config for the virtualhost there are a few steps here and I will list them below.

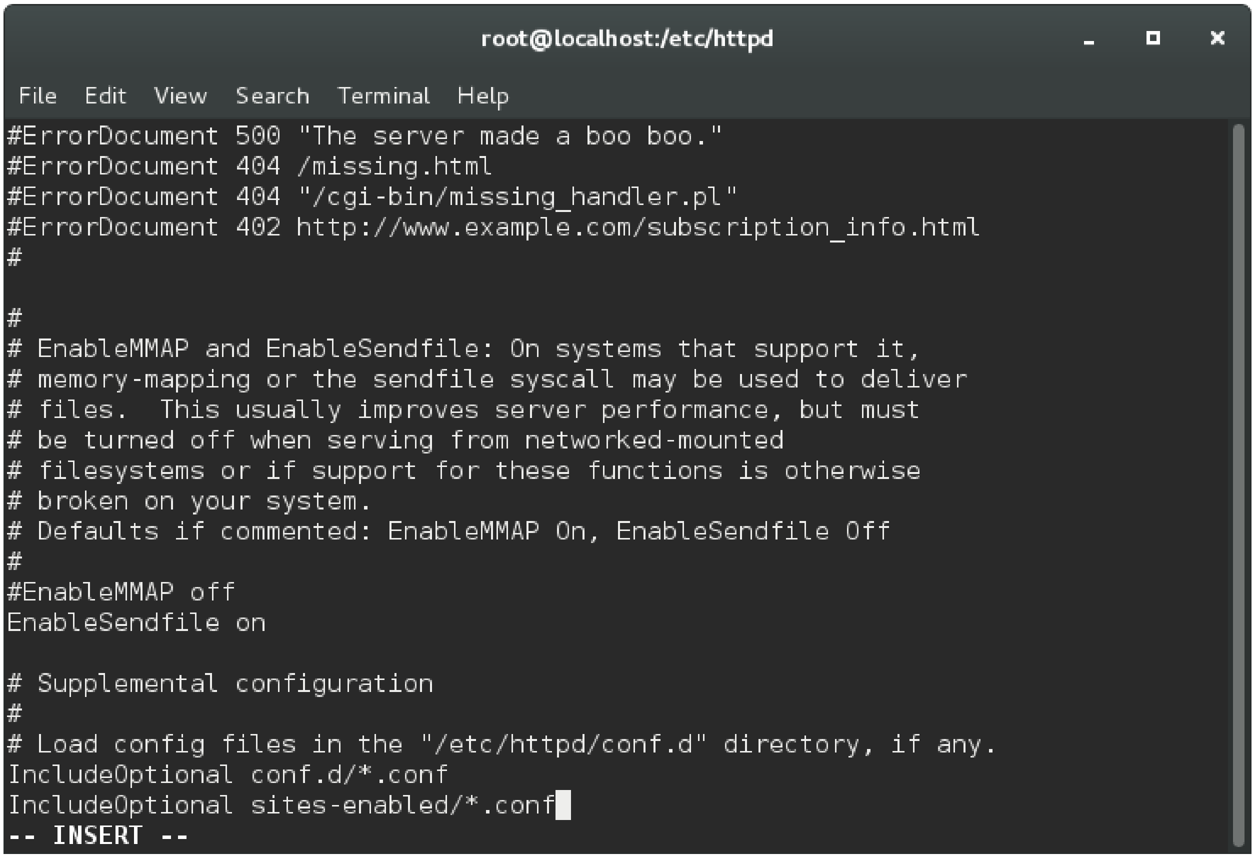
1) First we need to create a folder. Type "cd /etc/httpd"
2) Create the folder using "mkdir sites-enabled"


root@localhost:/etc/httpd
File Edit View Search Terminal Help

```
rm: remove regular file ‘simplerisk-installer-20170312-001.tgz’? y
[root@DESKTOP-EQN660Q simplerisk]# ps aux |grep httpd
root     14720  0.0  0.1 409632 14864 ?        Ss   13:08   0:00 /usr/sbin/http
-DFOREGROUND
apache   14721  0.0  0.1 411716  8068 ?        S    13:08   0:00 /usr/sbin/http
-DFOREGROUND
apache   14722  0.0  0.1 411716  8068 ?        S    13:08   0:00 /usr/sbin/http
-DFOREGROUND
apache   14723  0.0  0.1 411716  8068 ?        S    13:08   0:00 /usr/sbin/http
-DFOREGROUND
apache   14724  0.0  0.1 411716  8068 ?        S    13:08   0:00 /usr/sbin/http
-DFOREGROUND
apache   14725  0.0  0.1 411716  8068 ?        S    13:08   0:00 /usr/sbin/http
-DFOREGROUND
root     14849  0.0  0.0 112652   960 pts/0    S+   13:14   0:00 grep --color=au
to httpd
[root@DESKTOP-EQN660Q simplerisk]# chown -R apache: /var/www/html/simplerisk
[root@DESKTOP-EQN660Q simplerisk]# cd /etc/httpd
[root@DESKTOP-EQN660Q httpd]# mkdir sites-available
[root@DESKTOP-EQN660Q httpd]# mkdir sites-enabled
[root@DESKTOP-EQN660Q httpd]# ls
conf    conf.modules.d  modules  sites-available
conf.d  logs            run      sites-enabled
[root@DESKTOP-EQN660Q httpd]# 
```

3) Next we need to add "IncludeOptional sites-enabled/*.conf" to the end of the config located at "/etc/httpd/conf/httpd.conf" I did this with vim using "vi

/etc/httpd/conf/httpd.conf”. The result should look like this

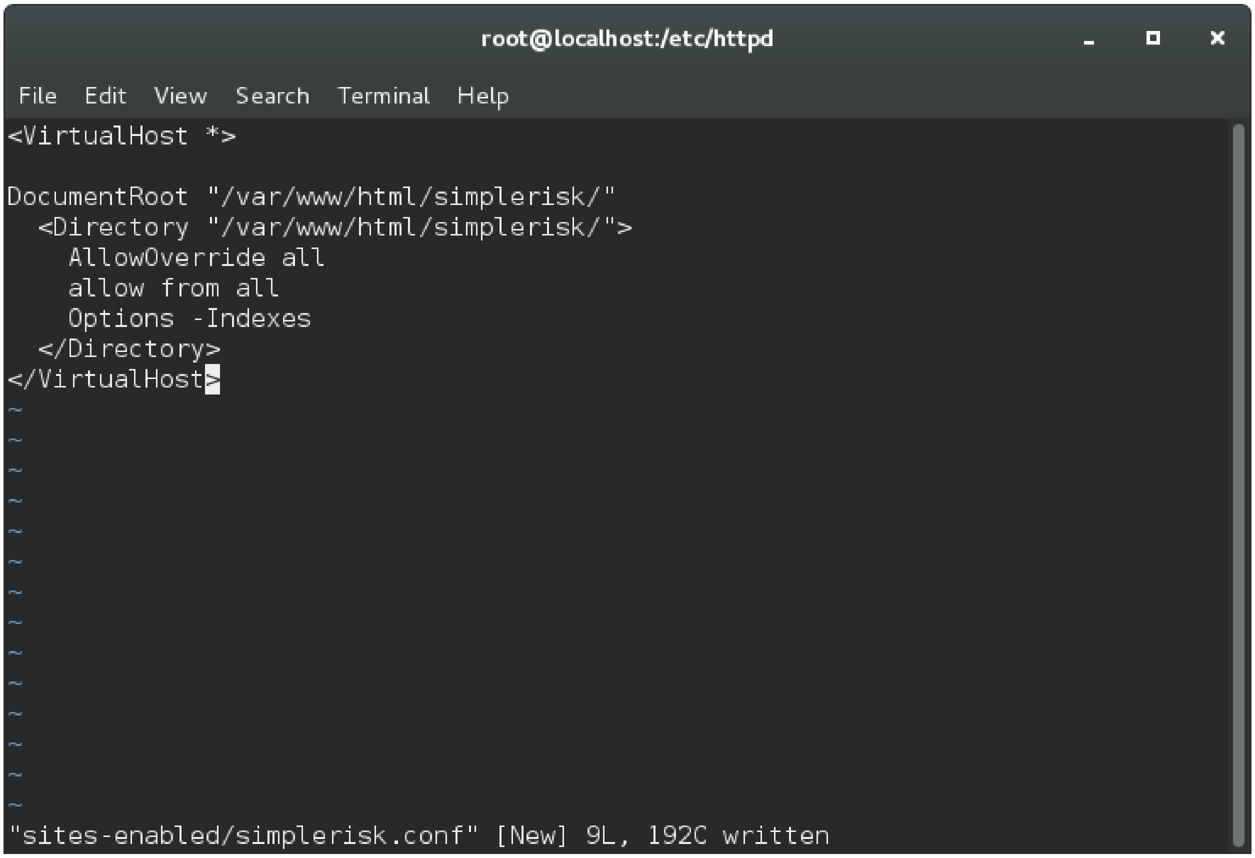

root@localhost:/etc/httpd
File Edit View Search Terminal Help

```
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#

#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall may be used to deliver
# files.  This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
EnableSendfile on

# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
IncludeOptional sites-enabled/*.conf
-- INSERT --
```

4) Now we create our config so go ahead and create a text file using your text editor of choice using something like this “vi /etc/httpd/sites-enabled/simplerisk.conf”
5) In the SimpleRisk.conf we just created it must contain the following text


root@localhost:/etc/httpd
File Edit View Search Terminal Help

```
<VirtualHost *>

DocumentRoot "/var/www/html/simplerisk/"
  <Directory "/var/www/html/simplerisk/">
    AllowOverride all
    allow from all
    Options -Indexes
  </Directory>
</VirtualHost>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"sites-enabled/simplerisk.conf" [New] 9L, 192C written
```

An easy to copy example is pasted below

```
<VirtualHost *:80>
ServerName simplerisk
DocumentRoot "/var/www/html/simplerisk"
   <Directory "/var/www/html/simplerisk">
   Options -Indexes
   AllowOverride All
   allow from all
   </Directory>
</VirtualHost
```

6) Restart httpd using “systemctl restart httpd”

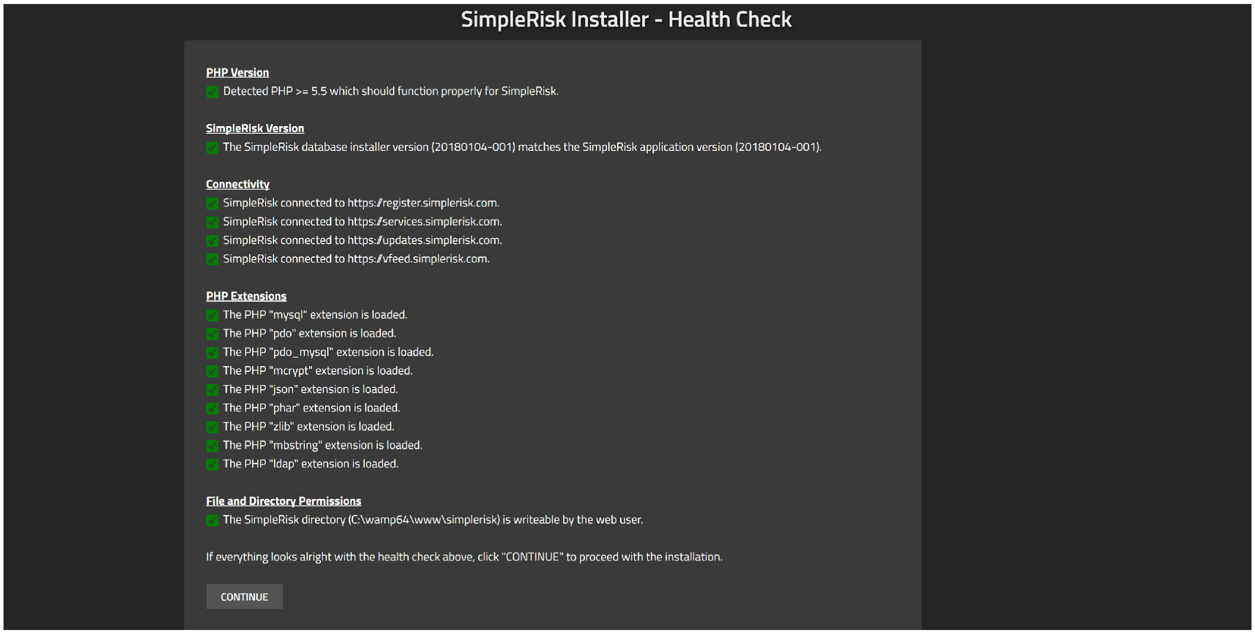
One more configuration change that will be necessary for deploying your instance. We will need to now set the SQL-Mode.

1) vi /etc/my.cnf
2) Scroll to the bottom and add the following line:
   sql-mode="NO_ENGINE_SUBSTITUTION"
3) systemctl restart mariadb

Last we must open the port for HTTP traffic so SimpleRisk may be accessed from other machines.

1) sudo firewall-cmd --add-service=http --permanent
2) sudo firewall-cmd --reload

Now your server has been configured for SimpleRisk, next we need to install the database using the install script that was downloaded earlier in the process. Now navigate to your instance's database installer by opening a browser and navigating to "http://IP.Address/install" and example of what this could be is "http://127.0.0.1/install/" if you did not change it it will be there by default. If everything works as expected, you will see an installer health check designed to help ensure the environment is ready for the database installation.



SimpleRisk Installer - Health Check

PHP Version

✓ Detected PHP >= 5.5 which should function properly for SimpleRisk.

SimpleRisk Version

✓ The SimpleRisk database installer version (20180104-001) matches the SimpleRisk application version (20180104-001).

Connectivity

✓ SimpleRisk connected to https://register.simplerisk.com.
✓ SimpleRisk connected to https://services.simplerisk.com.
✓ SimpleRisk connected to https://updates.simplerisk.com.
✓ SimpleRisk connected to https://vfeed.simplerisk.com.

PHP Extensions

✓ The PHP "mysql" extension is loaded.
✓ The PHP "pdo" extension is loaded.
✓ The PHP "pdo_mysql" extension is loaded.
✓ The PHP "mcrypt" extension is loaded.
✓ The PHP "json" extension is loaded.
✓ The PHP "phar" extension is loaded.
✓ The PHP "zlib" extension is loaded.
✓ The PHP "mbstring" extension is loaded.
✓ The PHP "ldap" extension is loaded.

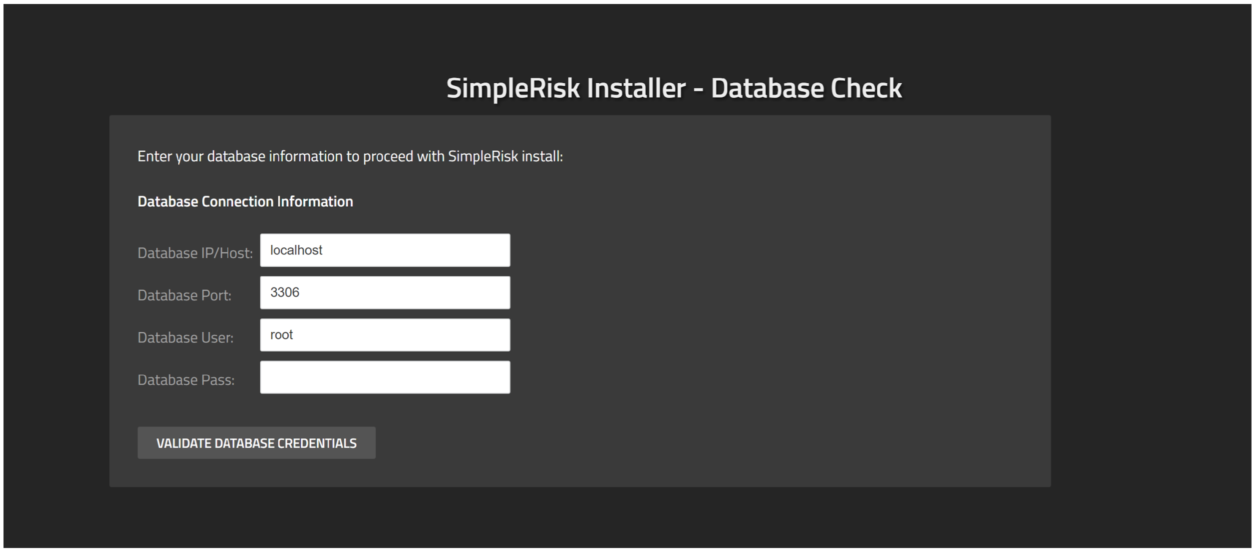
File and Directory Permissions

✓ The SimpleRisk directory (C:\wamp64\www\simplerisk) is writeable by the web user.

If everything looks alright with the health check above, click "CONTINUE" to proceed with the installation.

This page will show if any of the most common issues are present in your environment. If everything looks good here you may hit continue at the bottom.

You will now be taken to the database connection stage of the install. This is where you can edit how SimpleRisk connects to your database. For a simple setup and for our demonstration the only thing you should need to fill out on this page is the MySQL “Database Pass:” field, once entered you may hit the “Validate Database Credentials” button and it will ensure SimpleRisk is able to make a connection to the database to setup the SimpleRisk user with the least privileges required for the program to function. An example of this page is posted below.



SimpleRisk Installer - Database Check

Enter your database information to proceed with SimpleRisk install:

Database Connection Information

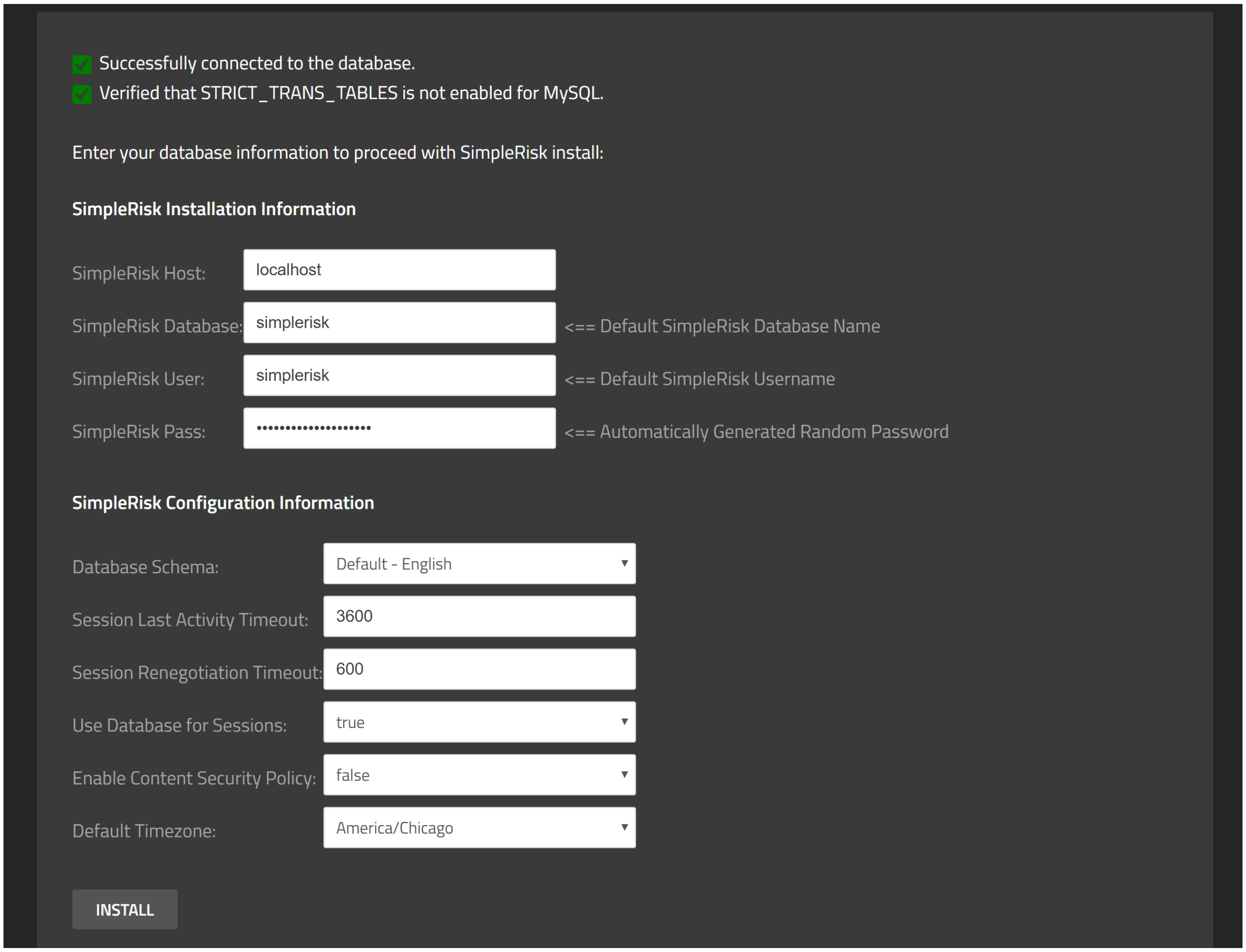
Database IP/Host: localhost

Database Port: 3306

Database User: root

Database Pass:

Now that you have setup the SimpleRisk database connection your last step is to setup the SimpleRisk install information and the SimpleRisk configuration information, you do not have to enter or change anything on this page for a standard install. The most important thing to note on this page is that you have 2 checks at the top notifying you that SimpleRisk was able to connect to the database and that STRICT_TRANS_TABLES is not enabled. If everything looks correct you may click the install button. An example of this page is posted below.



☑ Successfully connected to the database.
☑ Verified that STRICT_TRANS_TABLES is not enabled for MySQL.

Enter your database information to proceed with SimpleRisk install:

SimpleRisk Installation Information

SimpleRisk Host: localhost

SimpleRisk Database: simplerisk <== Default SimpleRisk Database Name

SimpleRisk User: simplerisk <== Default SimpleRisk Username

SimpleRisk Pass: •••••••••••••••••••• <== Automatically Generated Random Password

SimpleRisk Configuration Information

Database Schema: Default - English

Session Last Activity Timeout: 3600

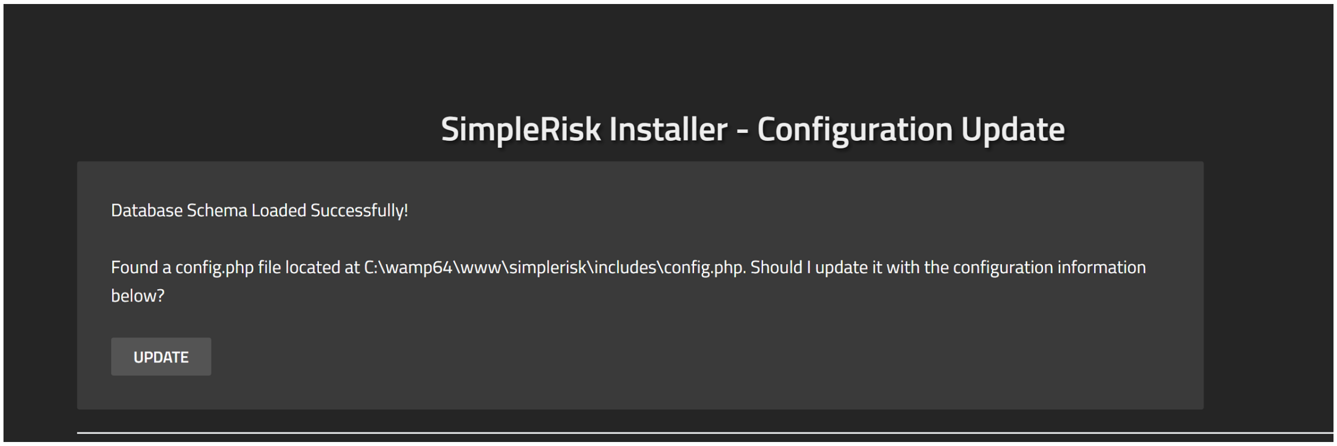
Session Renegotiation Timeout: 600

Use Database for Sessions: true

Enable Content Security Policy: false

Default Timezone: America/Chicago

Your final step for setting up the database will be to click the update button and you will have completed your SimpleRisk installation. It is always a good idea to delete the "install" directory once it is no longer needed using the command "sudo rm –r install" in terminal. A picture of the final page of the installer is attached below.



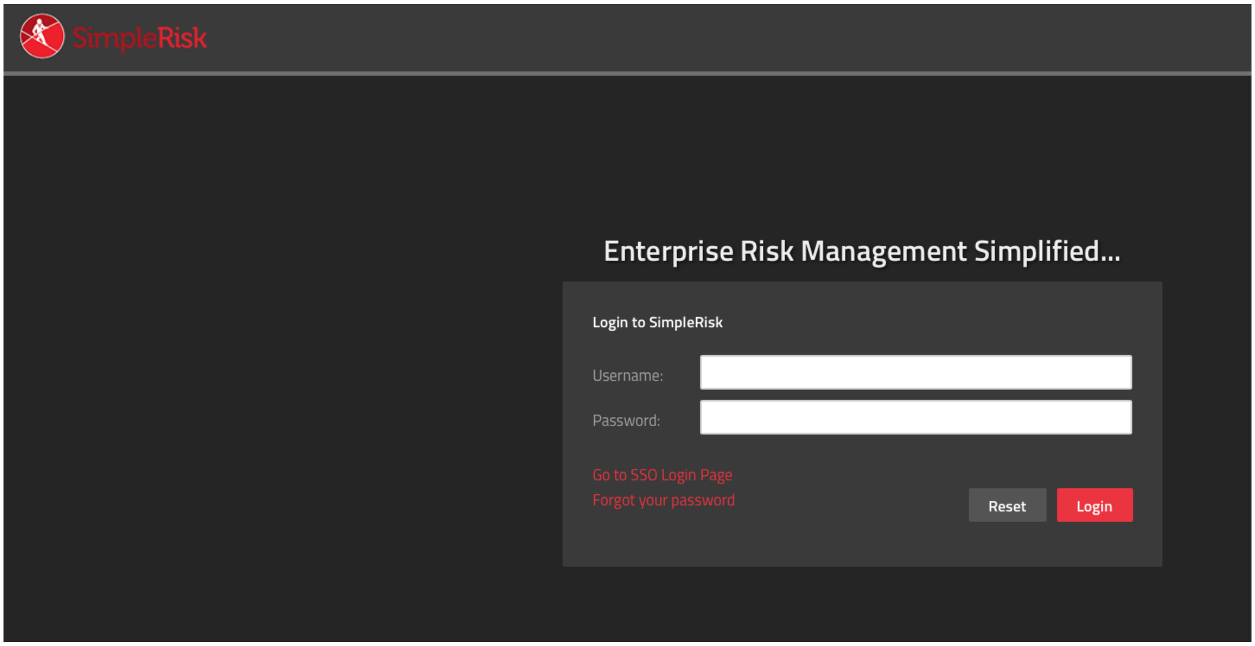
SimpleRisk Installer - Configuration Update

Database Schema Loaded Successfully!

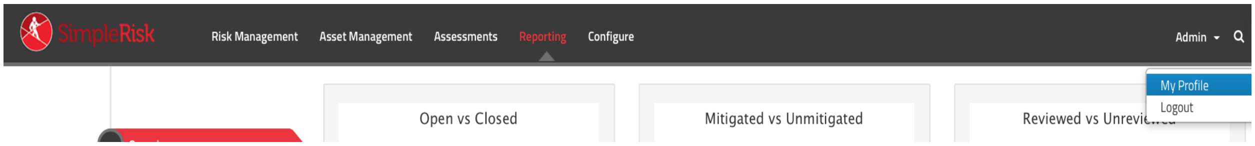
Found a config.php file located at C:\wamp64\www\simplerisk\includes\config.php. Should I update it with the configuration information below?

Logging in to SimpleRisk

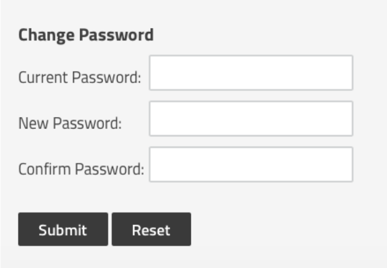
You should now have performed all of the steps you need to for SimpleRisk to be up and running. Now is the moment of truth where we hopefully get to see if all of your hard work paid off. You now need to point your web browser to the URL where SimpleRisk would be installed. If you followed the optional instructions, then it should be located at http://simplerisk/ if not you will need to guide your browser to http://yourserverip/. You will know that you've got the right page when you see something like this:



Enter username "admin" and password "admin" to get started. Then, select the "Admin" dropdown at the top right and click on "My Profile".



Enter your current password as "admin" and place a new long and randomly generated password into the "New Password" and "Confirm Password" fields. Then click "Submit".



Change Password

Current Password:

New Password:

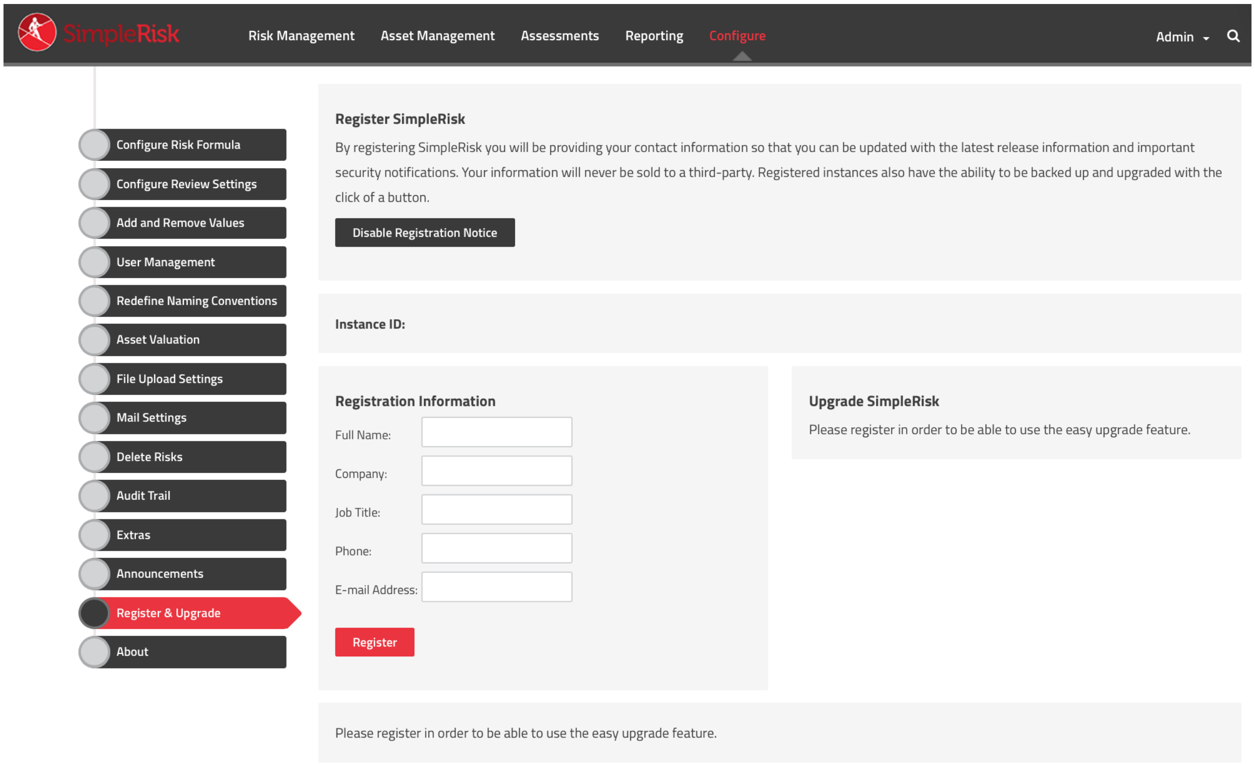
Confirm Password:

You should receive a message saying that your password was updated successfully. If so, then this is your new "admin" password for SimpleRisk. If you received a message saying that "The password entered does not adhere to the password policy", you can change the policy by selecting "Configure" from the menu at the top followed by "User Management" on the left side. You will see a "Password Policy" section at the bottom of the page where you can change the policy and try changing your password again.

Registering SimpleRisk

This step is completely optional, but without it upgrades of SimpleRisk will require manual downloads of the new version, backing up your configuration file, extracting the new files, restoring the configuration file, and a database upgrade. It sounds like more effort than it really is, but we've made the process far simpler if you're willing to tell us who you are. To register your SimpleRisk instance, select "Configure" from the menu at the top followed by

“Register & Upgrade” from the menu at the left.



Register SimpleRisk

By registering SimpleRisk you will be providing your contact information so that you can be updated with the latest release information and important security notifications. Your information will never be sold to a third-party. Registered instances also have the ability to be backed up and upgraded with the click of a button.

Instance ID:

Registration Information

Full Name:

Company:

Job Title:

Phone:

E-mail Address:

Upgrade SimpleRisk

Please register in order to be able to use the easy upgrade feature.

Please register in order to be able to use the easy upgrade feature.

Enter your information and select the “Register” button. This will create a unique Instance ID for your SimpleRisk instance and download the Upgrade Extra which enables functionality for one-click backups and upgrades. If you run into issues with the registration process, we recommend that you check to ensure that the “simplerisk” directory and its sub-directories are writeable by the www-data user (or whatever user Apache is running as).

**** This completes your installation of SimpleRisk ****

SimpleRisk Paid Support and Extras

Everything that you’ve seen up to this point is completely free for you to install and use, forever. That said, we offer a number of ways for you to enhance your SimpleRisk instance with even more functionality. If you like what you see, and find it useful, please consider purchasing one of our inexpensive Paid Support plans or Extra functionality so that we can continue to offer you the best open source risk management tool available. Thank you!