Install SimpleRisk on Ubuntu 20.04

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.com/.

Disclaimer

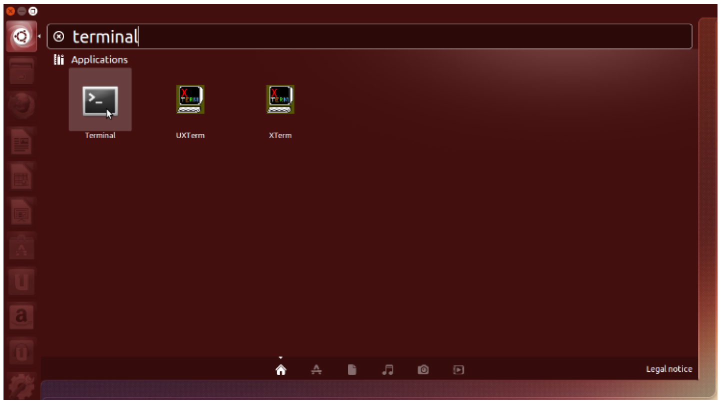
The lucky security professionals work for companies who can afford expensive GRC tools to aide in managing risk. The unlucky majority out there usually end up spending countless hours managing risk via spreadsheets. It's cumbersome, time consuming, and just plain sucks. When Josh Sokol started writing 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.com 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.

Install Ubuntu

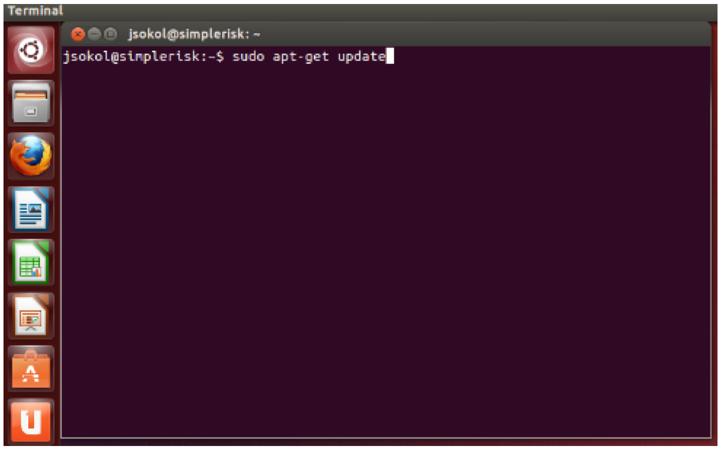
SimpleRisk should be able to work on just about any operating system that is capable of running PHP and MySQL. Since the purpose of this guide is to get you up and running with SimpleRisk as quickly as possible, we assume that you are using Ubuntu, a FREE and easy to use Linux-based operating system. Download the latest version of Ubuntu 20.04 and install it. See the Ubuntu documentation if you are having any issues there. Once you have a working installation, you can move on to the next installation steps.

Get the Latest Ubuntu Updates

Log in to your Ubuntu installation using the username and password you defined at setup. Select the Unity menu (the one at the very top of the bar on the left) and type "terminal" in the field that pops up. This should show you a shortcut to the terminal application. You can click it to launch the terminal, but it may be a good idea to drag it to the Unity bar on the left first so that you can easily start it in the future.

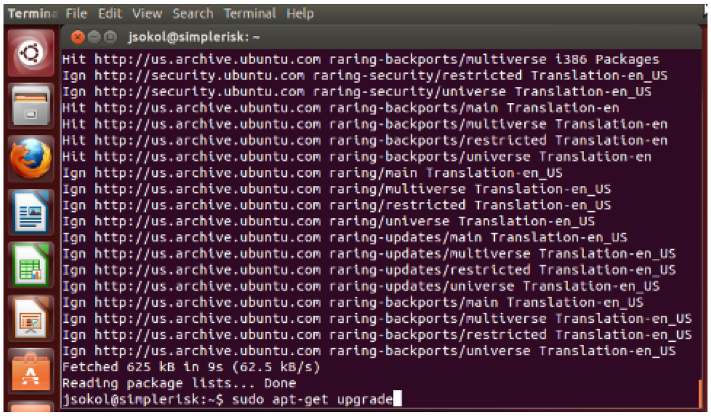


Once the terminal is launched, you will want to update the OS to the latest software versions available. To do this run “sudo apt-get update” and enter your password when prompted.



```
jsokol@simplerisk:~$ sudo apt-get update
```

This will pull down the latest version information for all of the installed operating system files. Now run “sudo apt-get dist-upgrade” and answer “y” when it asks if you would like to continue.



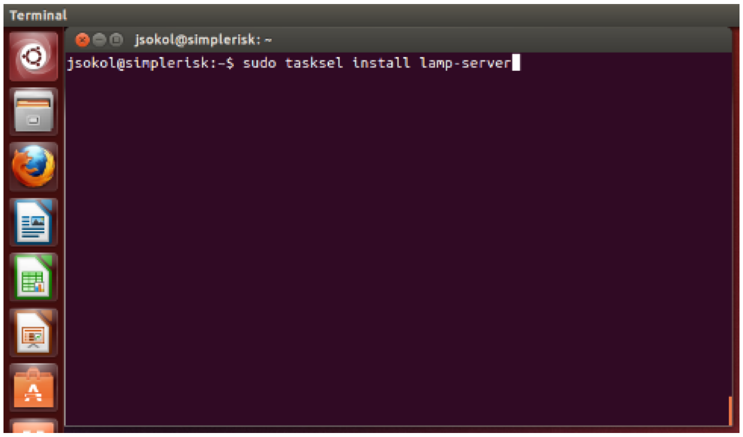
```
Hit http://us.archive.ubuntu.com raring-backports/multiverse i386 Packages
Ign http://security.ubuntu.com raring-security/restricted Translation-en_US
Ign http://security.ubuntu.com raring-security/universe Translation-en_US
Hit http://us.archive.ubuntu.com raring-backports/main Translation-en
Hit http://us.archive.ubuntu.com raring-backports/multiverse Translation-en
Hit http://us.archive.ubuntu.com raring-backports/restricted Translation-en
Hit http://us.archive.ubuntu.com raring-backports/universe Translation-en
Ign http://us.archive.ubuntu.com raring/main Translation-en_US
Ign http://us.archive.ubuntu.com raring/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com raring/restricted Translation-en_US
Ign http://us.archive.ubuntu.com raring/universe Translation-en_US
Ign http://us.archive.ubuntu.com raring-updates/main Translation-en_US
Ign http://us.archive.ubuntu.com raring-updates/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com raring-updates/restricted Translation-en_US
Ign http://us.archive.ubuntu.com raring-updates/universe Translation-en_US
Ign http://us.archive.ubuntu.com raring-backports/main Translation-en_US
Ign http://us.archive.ubuntu.com raring-backports/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com raring-backports/restricted Translation-en_US
Ign http://us.archive.ubuntu.com raring-backports/universe Translation-en_US
Fetched 625 kB in 9s (62.5 kB/s)
Reading package lists... Done
jsokol@simplerisk:~$ sudo apt-get upgrade
```

Installing Apache, PHP, and MySQL

The next step is to install the necessary files in order to run Apache with PHP and MySQL on this system.

To do this, first run the command “sudo apt-get install tasksel”.

Next, tell the server to install a LAMP stack by running the command “sudo tasksel install lamp-server”.



```
jsokol@simplerisk:~$ sudo tasksel install lamp-server
```

You should now see the terminal change into a package configuration application that downloads and installs the applications necessary in order to run a LAMP stack on the server. You will know that this installation process is complete when the package configuration screen goes away and you are back at the terminal shell.

Next we are going to update system dependencies

sudo apt update

sudo apt upgrade

Once the system are updated, we are adding PHP PPA

sudo apt install software-properties-common

sudo add-apt-repository ppa:ondrej/php

sudo apt update

The following step is to install PHP 8.x for apache

sudo apt install php8.1

To confirm the installation using the following command

php -v

Now we are going to install PHP 8.x extensions

```
sudo apt install php8.1-extension_name
```

Now, install some commonly used php-extensions with the following command.

```
sudo apt install php8.1-common php8.1-mysql php8.1-xml php8.1-xmlrpc php8.1-curl php8.1-gd
php8.1-imagick php8.1-cli php8.1-dev php8.1-imap php8.1-mbstring php8.1-opcache php8.1-soap
php8.1-zip php8.1-redis php8.1-intl -y
```

Last step is to configure PHP 8.x. Execute the following command on command prompt to edit php.ini

```
sudo nano /etc/php/8.1/apache2/php.ini
```

Hit F6 for search inside the editor and update the following values for better performance.

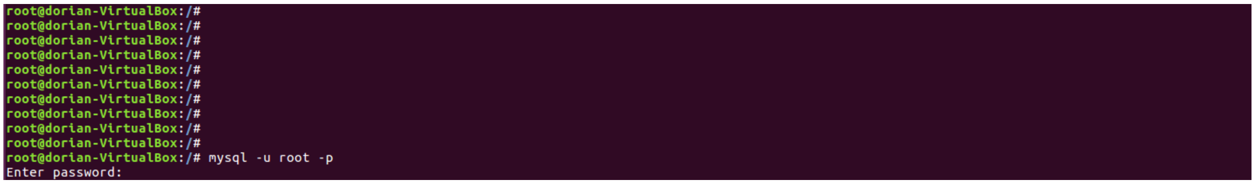
```
upload_max_filesize = 32M

post_max_size = 48M

memory_limit = 256M

max_execution_time = 600

max_input_vars = 3000

max_input_time = 1000
```

Then execute the following command on command prompt to restart apache web server.

```
sudo service apache2 restart
```

Now we need to do a little setup of MySQL for the install to go smoothly, this will change the ROOT MySQL user password and we suggest making this password strong and recording it elsewhere just in case.

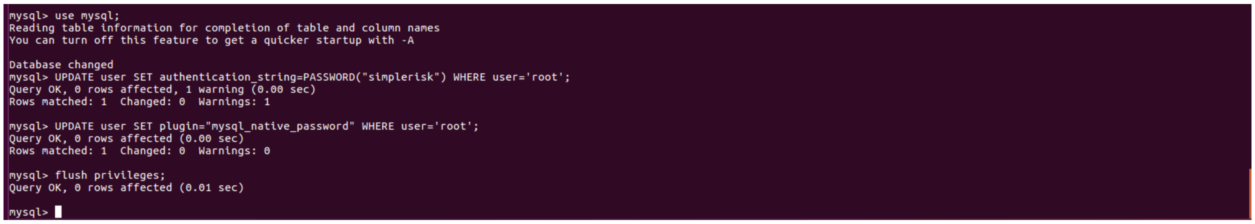
First login to the MySQL console using “sudo mysql -u root -p”. The terminal will then ask for a password which should be blank on a fresh install, so just hit enter/return.



```
root@dorian-VirtualBox:/#
root@dorian-VirtualBox:/#
root@dorian-VirtualBox:/#
root@dorian-VirtualBox:/#
root@dorian-VirtualBox:/#
root@dorian-VirtualBox:/#
root@dorian-VirtualBox:/#
root@dorian-VirtualBox:/#
root@dorian-VirtualBox:/#
root@dorian-VirtualBox:/#
root@dorian-VirtualBox:/# mysql -u root -p
Enter password:
```

Now in the console use the following to set the root password and confirm the plugin / change the plugin used for password authentication.

use mysql;
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'MyNewMySQLPassword';
flush privileges;



```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> UPDATE user SET authentication_string=PASSWORD("simplerisk") WHERE user='root';
Query OK, 0 rows affected, 1 warning (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 1

mysql> UPDATE user SET plugin="mysql_native_password" WHERE user='root';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> 
```

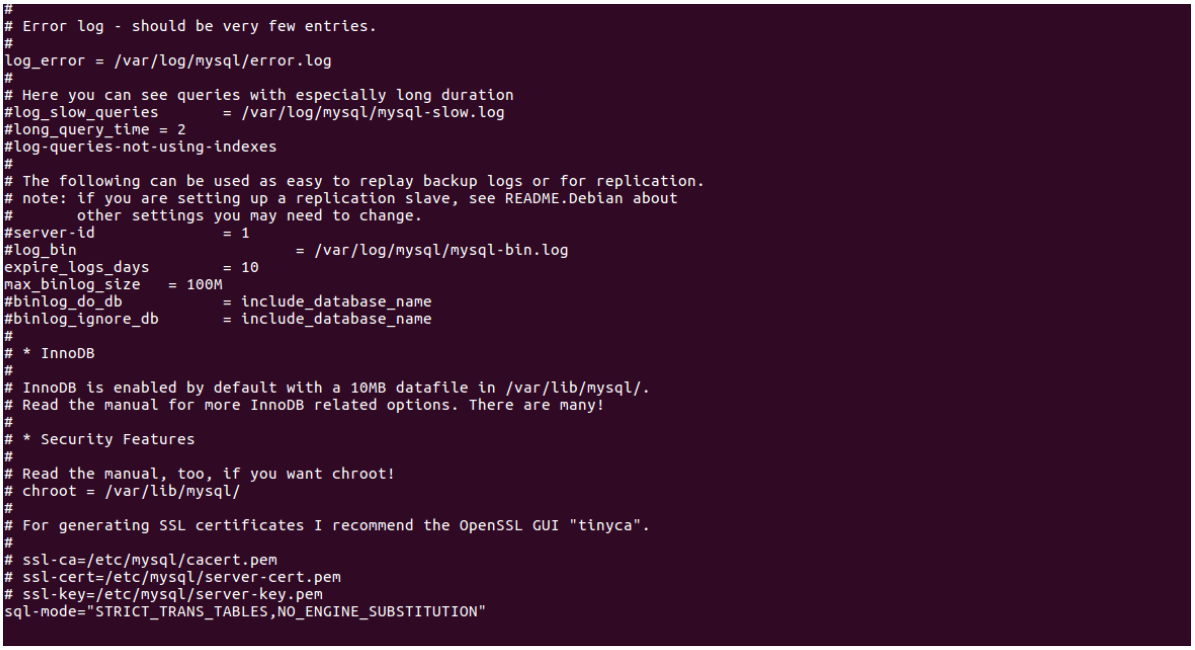
note: in this screenshot this has already been done hence 0 rows affected your should show 1 row affected.

The next step of setting up MySQL for a SimpleRisk install will be to set the sql-mode. To do this use the following steps:

1) use “nano /etc/mysql/mysql.conf.d/mysqld.cnf”

2) At the bottom of the config file add the following to set the sql-mode.

sql-mode="NO_ENGINE_SUBSTITUTION"



```
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#log_slow_queries       = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
#server-id              = 1
#log_bin                        = /var/log/mysql/mysql-bin.log
expire_logs_days        = 10
max_binlog_size   = 100M
#binlog_do_db           = include_database_name
#binlog_ignore_db       = include_database_name
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# For generating SSL certificates I recommend the OpenSSL GUI "tinyca".
#
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem
sql-mode="STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION"
```

3) Now write the file out (ctrl X, Y, enter).

Now we will configure Apache for the SimpleRisk API

1) Run the command “sudo a2enmod rewrite ssl” to enable mod_rewrite & ssl for Apache.

2) Run the command “sudo rm /etc/apache2/sites-enabled/000-default.conf” to delete the default apache2 configuration fileI thin.

3) Now we will create the virtualhost configuration for SimpleRisk create new text file by using:

```
sudo nano /etc/apache2/sites-enabled/simplerisk.conf
```

4) Now using the following as an example you can copy edit the virtualhost configuration like so (**Note, spacing of different lines matters)**:

```
<VirtualHost  *:80>

        ServerName simplerisk

        DocumentRoot "/var/www/simplerisk"

        <Directory "/var/www/simplerisk">

                Options -Indexes

                AllowOverride All

                allow from all

        </Directory>

</VirtualHost>
```

3) Save the file and close your text editor.

Generating an SSL certificate

Enabling HTTPS is necessary to secure the connection between your web browser and the SimpleRisk instance you’re creating. If you already have a certificate from your organization you may skip this section but you will need to have it readily available to continue the installation as the virtualhost configurations we provide enforce HTTPS. You can always generate one for now and change it to another one later. Continue on in the terminal to generate your cert and key.

1) First, we need to choose a directory to store our cert and key pair, in this example we will create the directory “/root/certs” and work from there.

```
sudo bash
```

```
mkdir /root/certs && cd /root/certs
```

2) Create the certificate using the following command:

```
sudo openssl req -new -newkey rsa:4096 -x509 -sha256 -days 9999 -nodes -out simplerisk.crt -keyout simpleriskkey.key
```

3) You'll be prompted to enter some certificate details and when you have entered all of the required prompts we will change the file permissions of the key to be accessible only by root. This is an important step and should not be skipped.

```
sudo chmod 400 /root/certs/simpleriskkey.key
```

4) Backup your certificate and key to an external storage that can be secured. This completes the process of generating a new self-signed key.

Configuring the Virtual Hosts for HTTPS

Next we will configure the virtual hosts to use HTTPS. In this section we will tell apache to use our SSL certificate and redirect any port 80 traffic to 443.

1) Open /etc/apache2/sites-enabled/simplerisk.conf using

```
sudo nano /etc/apache2/sites-enabled/simplerisk.conf
```

We'll need to add a few new lines to your virtualhost to enable the http to https redirect. Lines being added are in green If this was configured as shown in the ubuntu install guide it should look like the following. If you are using a cert from a CA then you will need to also add the following line just under the orange lines in the example below.

"SSLCertificateChainFile /path/to/insertcerthere.crt "

```
<VirtualHost *:80>

        ServerName simplerisk

        DocumentRoot "/var/www/simplerisk"

        <Directory "/var/www/simplerisk">

                Options -Indexes

                AllowOverride All

                allow from all

        </Directory>

        SSLEngine on
```

```
    SSLCertificateFile /root/certs/simplerisk.crt
    SSLCertificateKeyFile /root/certs/simpleriskkey.key
</VirtualHost>
```

2) Write the file out and close it. (ctrl X, Y, enter)

3) Create a new file using

```
sudo nano /etc/apache2/sites-enabled/ssl-simplerisk.conf
```

This file will contain the actual configuration for the https page.

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/simplerisk
        SSLEngine on
        SSLCertificateFile /root/certs/simplerisk.crt
        SSLCertificateKeyFile /root/certs/simpleriskkey.key
        <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
```

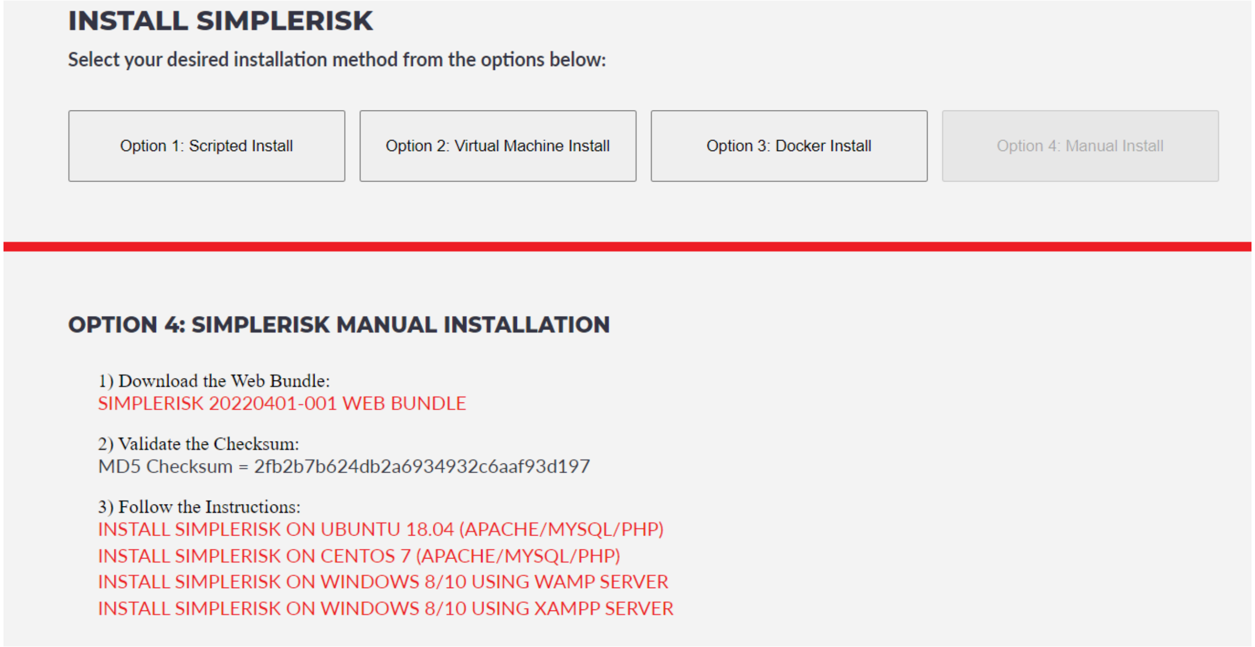
```
                        SSLOptions +StdEnvVars

                </Directory>

                <Directory "/var/www/simplerisk">

                        AllowOverride all

                        allow from all

                        Options -Indexes

                </Directory>

        </VirtualHost>

</IfModule>
```

7) The final step is to save your apache configuration, close the file and restart apache using:

```
sudo systemctl restart apache2..
```

Obtaining the SimpleRisk Files

Click on the FireFox logo in the Unity bar on the left. 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.



INSTALL SIMPLERISK

Select your desired installation method from the options below:

OPTION 4: SIMPLERISK MANUAL INSTALLATION

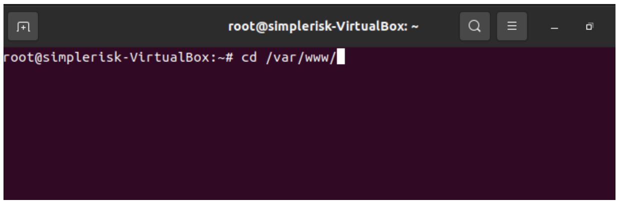
1) Download the Web Bundle:
SIMPLERISK 20220401-001 WEB BUNDLE

2) Validate the Checksum:
MD5 Checksum = 2fb2b7b624db2a6934932c6aaf93d197

3) Follow the Instructions:
INSTALL SIMPLERISK ON UBUNTU 18.04 (APACHE/MYSQL/PHP)
INSTALL SIMPLERISK ON CENTOS 7 (APACHE/MYSQL/PHP)
INSTALL SIMPLERISK ON WINDOWS 8/10 USING WAMP SERVER
INSTALL SIMPLERISK ON WINDOWS 8/10 USING XAMPP SERVER

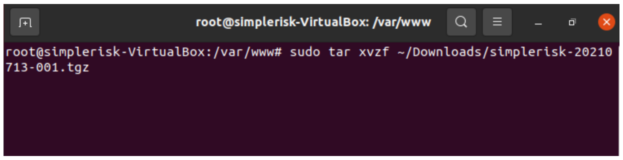
Click to download and save the Web Bundle there is no longer a separate database installer script to be downloaded and is now included with the core download. Once you have the files downloaded, you can close the browser.

Change to the new Apache web root by running the command “cd /var/www/”.



```
root@simplerisk-VirtualBox:~# cd /var/www/
```

Remove the html folder using the command “sudo rm -r html”. Extract the web bundle into the web directory using the command “sudo tar xvzf ~/Downloads/simplerisk-20220527-001.tgz” (or the newest available version).



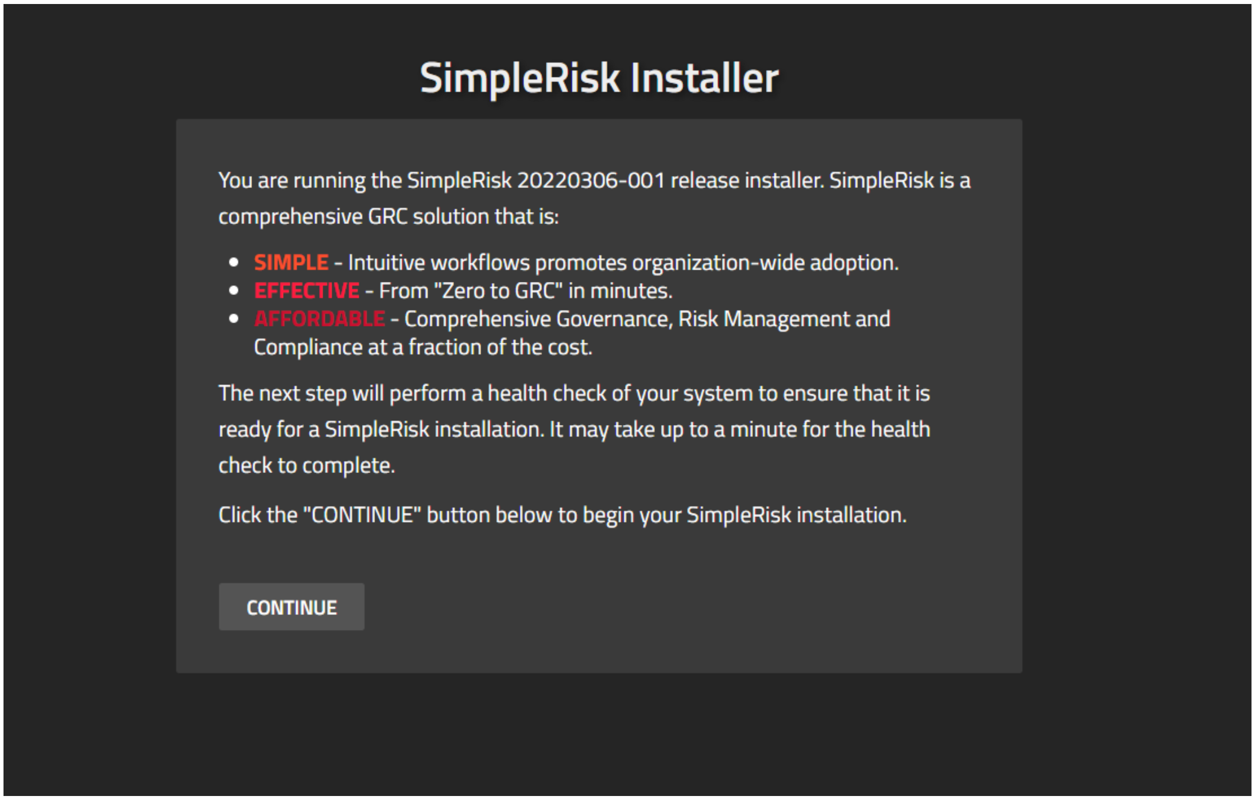
```
root@simplerisk-VirtualBox:/var/www# sudo tar xvzf ~/Downloads/simplerisk-20210
713-001.tgz
```

This will extract the files into a directory into the directory /var/www/.

Now we change the ownership permissions of the “simplerisk” directory and all its sub-directories to be owned by the www-data user (or whatever user Apache is running as) using the command “sudo chown –R www-data: /var/www/*”.

Installing the Database

As of SimpleRisk release 20220401-001 the database installer is now included with the core download and no separate download is required. To access the installer to complete the installation navigate to http://localhost in your browser. If you have followed all steps up to this point correctly you should see the following.



SimpleRisk Installer

You are running the SimpleRisk 20220306-001 release installer. SimpleRisk is a comprehensive GRC solution that is:

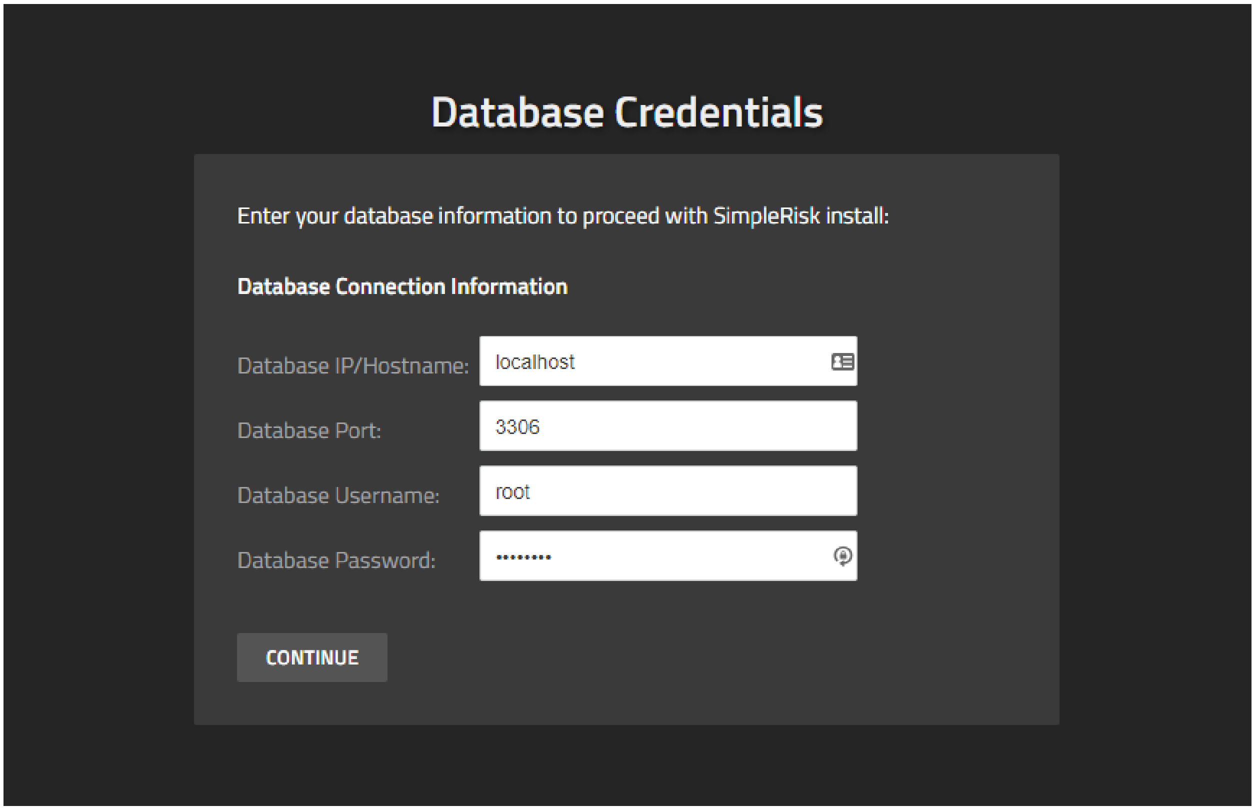
- SIMPLE - Intuitive workflows promotes organization-wide adoption.
- EFFECTIVE - From "Zero to GRC" in minutes.
- AFFORDABLE - Comprehensive Governance, Risk Management and Compliance at a fraction of the cost.

The next step will perform a health check of your system to ensure that it is ready for a SimpleRisk installation. It may take up to a minute for the health check to complete.

Click the "CONTINUE" button below to begin your SimpleRisk installation.

Click “continue” to and you will be met with a healthcheck page. Please note we should see all green all the way down the list in Ubuntu installations. As long as everything is green you may proceed and click continue.

You should now see the database configuration page. You will be required to enter the credentials for the MySQL root user which was configured in an earlier step. An example of this page is shown below.



Database Credentials

Enter your database information to proceed with SimpleRisk install:

Database Connection Information

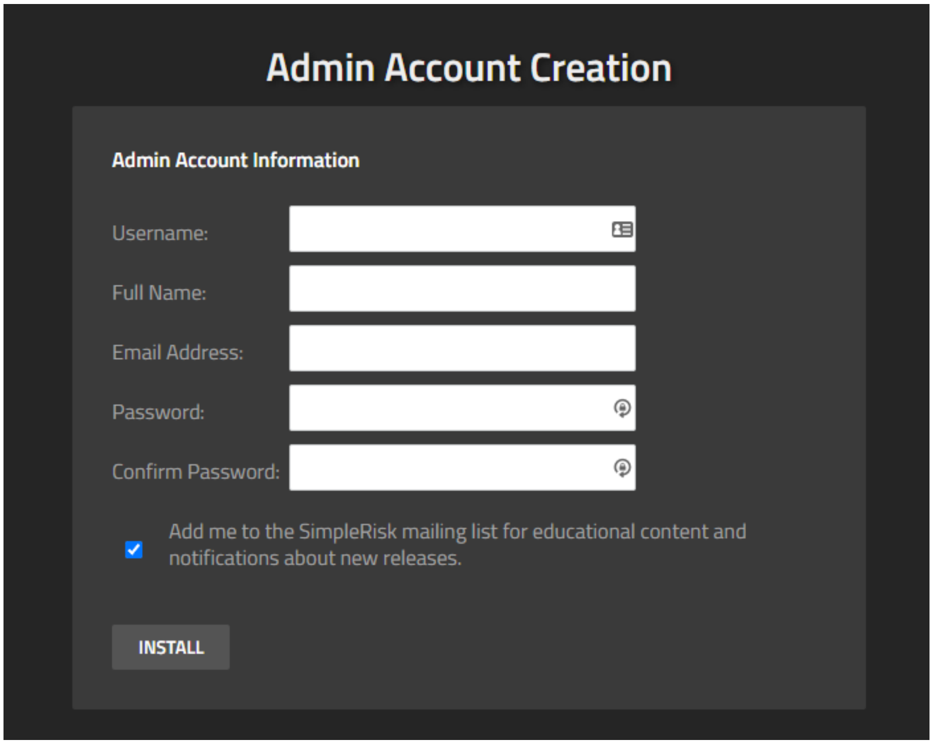
Database IP/Hostname: localhost

Database Port: 3306

Database Username: root

Database Password: ••••••••

Please note before continuing if you are setting up with an MySQL instance that is not local to the web server you will need to configure the Database IP/Hostname and port. In basic installations this step is not required and these values can be left as their defaults. Click “continue” once the credentials to access the MySQL server have been configured. You will now be able to configure the details of the SimpleRisk database. In a general installation all of these can be left default.



Admin Account Creation

Admin Account Information

Username:

Full Name:

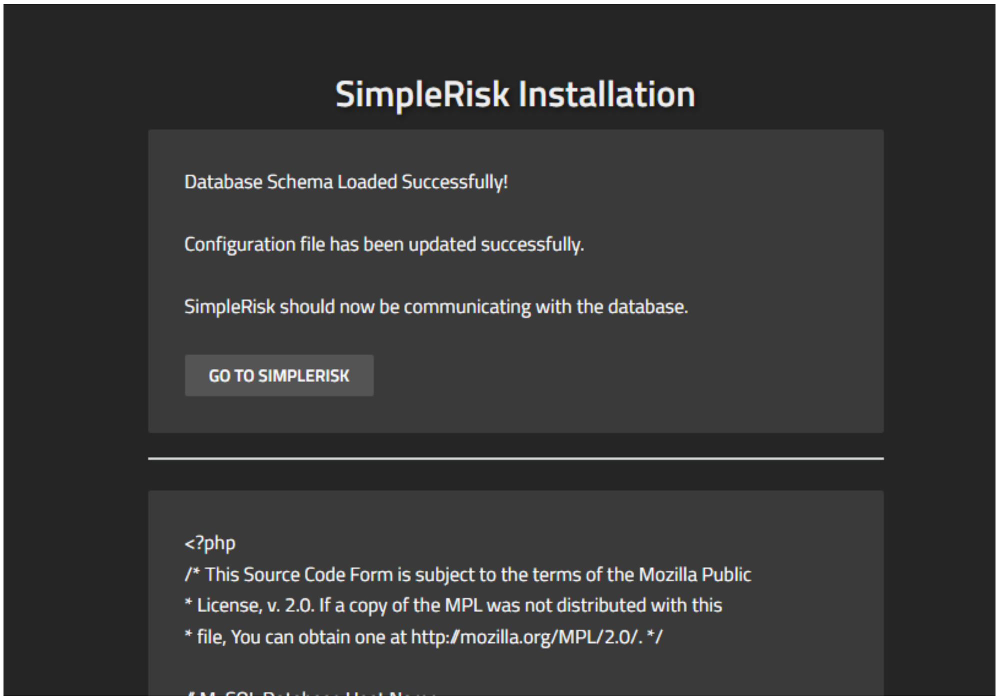
Email Address:

Password:

Confirm Password:

☑ Add me to the SimpleRisk mailing list for educational content and notifications about new releases.

We have now arrived at the final setup screen as seen above. Here you will configure your admin account and assign the details you will use to login to SimpleRisk with for that admin account. Please note that the use of a valid email address is highly recommended as this will be where password recovery emails will be sent. You are also given the opportunity to opt into our mailing list by checking the box at the bottom. In this you will receive product release updates and educational content related to the SimpleRisk platform. Once you see the image below in your browser you are finished and may click the “Go to SimpleRisk” button to login and begin using the application.



SimpleRisk Installation

Database Schema Loaded Successfully!

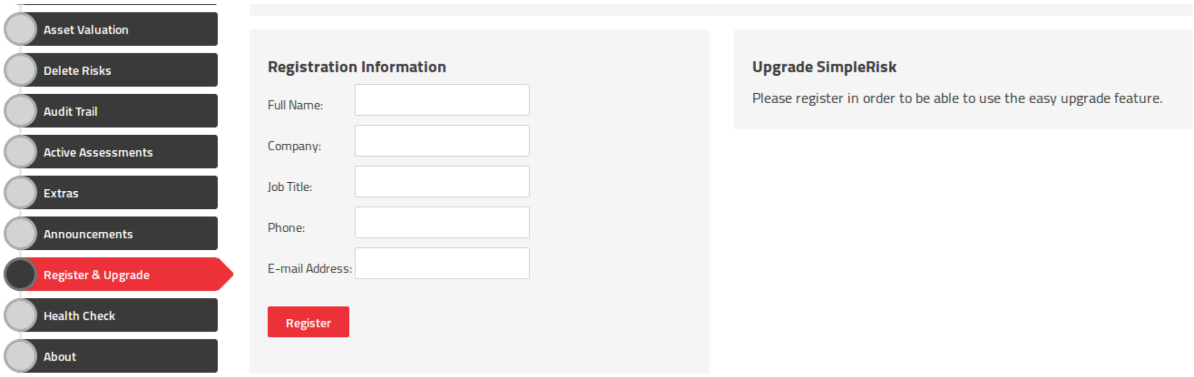
Configuration file has been updated successfully.

SimpleRisk should now be communicating with the database.

```
<?php
/* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
```

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.



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.

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!