

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 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:

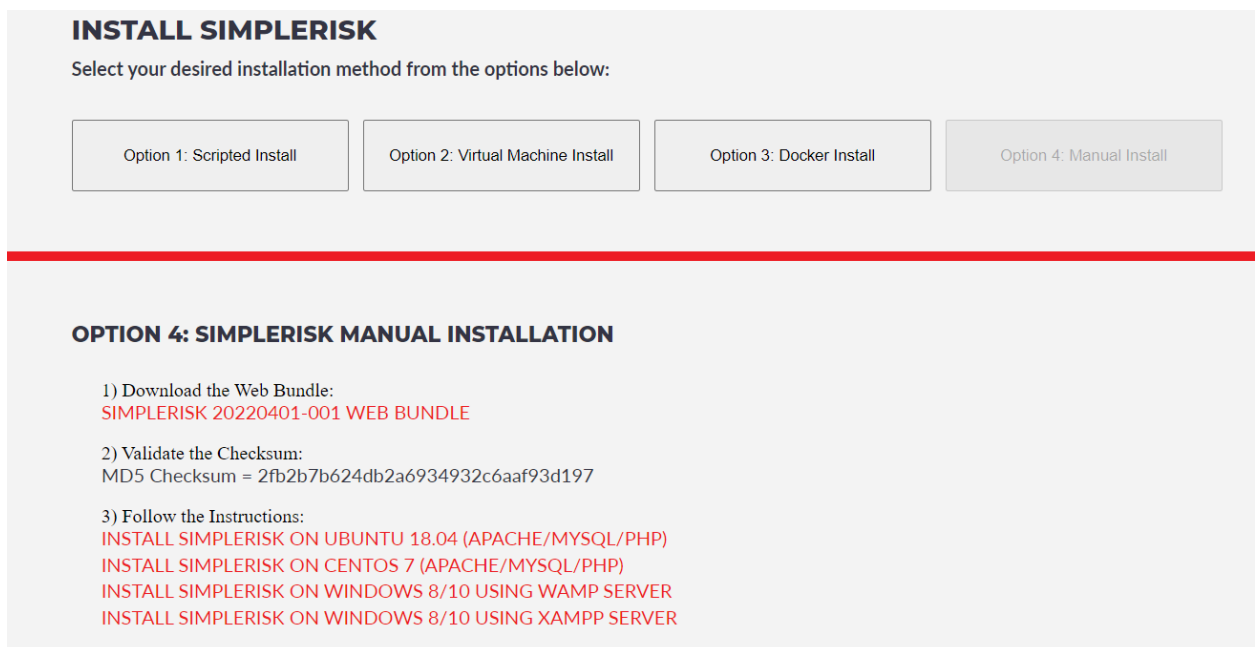
```
php -v
```

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. Then click the Manual installation button near the top right as shown below:



INSTALL SIMPLERISK

Select your desired installation method from the options below:

Option 1: Scripted Install Option 2: Virtual Machine Install Option 3: Docker Install Option 4: Manual Install

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 the link to Download the newest Web Bundle and continue on.

- 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`

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.

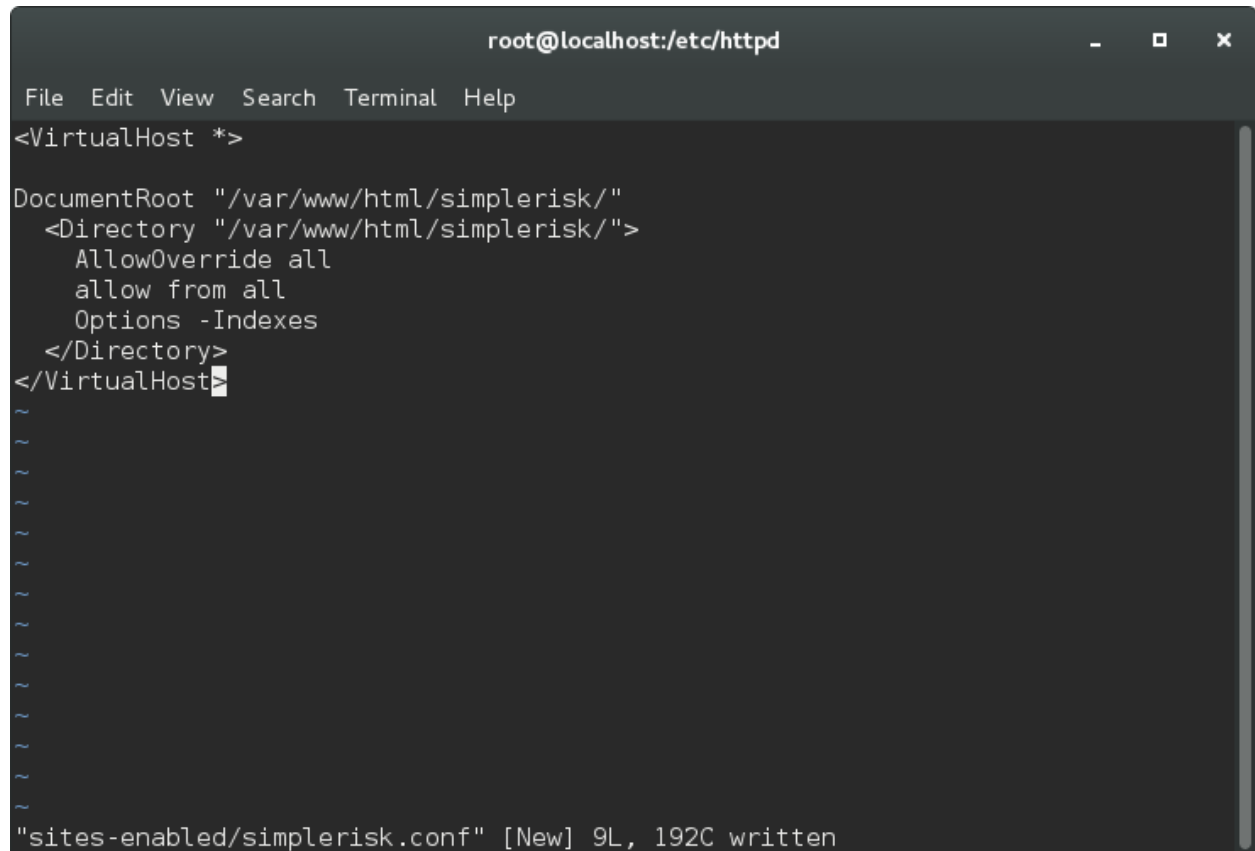
- 1) First we need to create a folder. Type `cd /etc/httpd`
- 2) Create the folder using `mkdir sites-enabled`

```
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



The screenshot shows a terminal window titled "root@localhost:/etc/httpd". The window contains the following XML configuration for a VirtualHost:

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

At the bottom of the terminal, a status message reads: "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

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.

CONTINUE

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:	<input type="text" value="localhost"/>
Database Port:	<input type="text" value="3306"/>
Database Username:	<input type="text" value="root"/>
Database Password:	<input type="password" value="....."/>

CONTINUE

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:

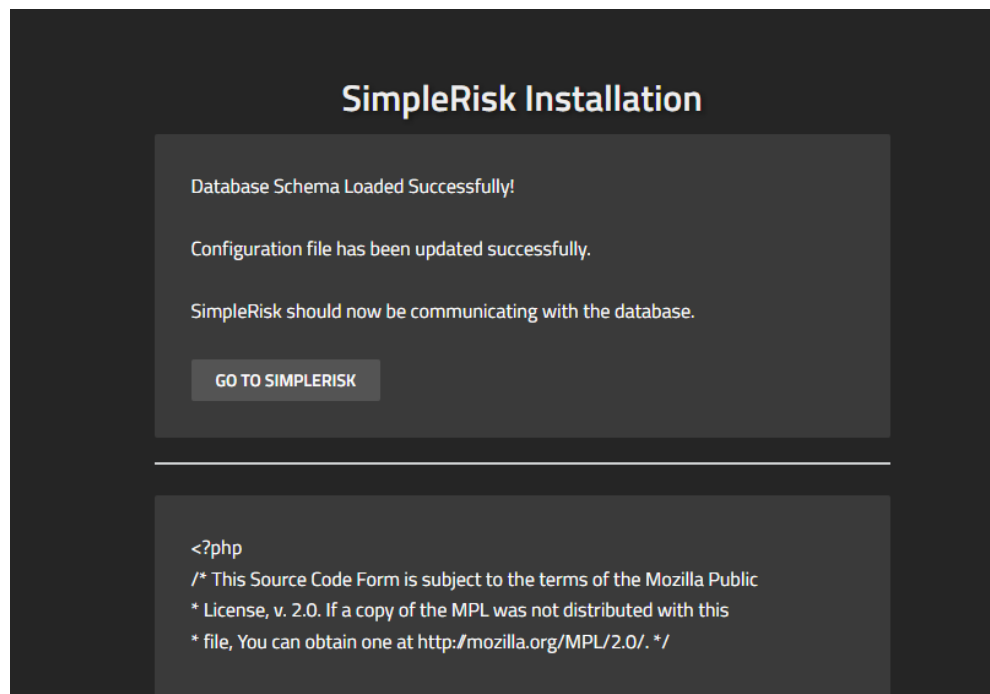
Password:

Confirm Password:

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


INSTALL

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.



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:

SimpleRisk

Enterprise Risk Management Simplified...

Login to SimpleRisk

Username:


Password:

[Go to SSO Login Page](#)
[Forgot your password](#)


Reset

Login

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

SimpleRisk

[Risk Management](#) [Asset Management](#) [Assessments](#) [Reporting](#) [Configure](#)

Admin 

Open vs Closed

Mitigated vs Unmitigated

Reviewed vs Unreviewed

[My Profile](#)
[Logout](#)

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:

Submit

Reset

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.

The screenshot displays the SimpleRisk web application interface. At the top, a dark navigation bar contains the SimpleRisk logo and a menu with items: Risk Management, Asset Management, Assessments, Reporting, and Configure (highlighted in red). On the left side, a vertical sidebar lists various configuration options, with 'Register & Upgrade' highlighted in red. The main content area is divided into several sections. The top section, titled 'Register SimpleRisk', explains the purpose of registration and includes a 'Disable Registration Notice' button. Below this is an 'Instance ID' field. The central part of the page features a 'Registration Information' form with input fields for Full Name, Company, Job Title, Phone, and E-mail Address, followed by a red 'Register' button. To the right of the form is a box titled 'Upgrade SimpleRisk' with the text 'Please register in order to be able to use the easy upgrade feature.' At the bottom of the page, a light gray banner repeats this message: '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!