

How to Install a Redmine Web Application

Overview

Install Redmine

- Pre-installation settings.
- Install the bundler gem.
- Extract the redmine archive file.
- Create the database and database users.
- Install your Ruby Gems.
- Create a secret key.
- Create the database structure.
- Create the data configuration.
- Set the file system permissions.
- Register the application.
- Restart apache.

Additional documentation

Overview

This document describes how to install the Redmine web application. The Redmine application is a project management web application that functions on the Ruby platform. You can register the Redmine application in cPanel's [Application Manager](#) interface (*cPanel >> Home >> Software >> Application Manager*).

Warnings:

- cPanel Technical Support does **not** provide assistance with the steps in this document.
- We recommend that **only** experienced system administrators perform these steps.
- We are **not** responsible for any data loss.

Notes:

- We recommend that you perform the steps in this document via the command line as a cPanel user unless otherwise specified.
- You can also use cPanel's [Terminal](#) interface (*cPanel >> Home >> Advanced >> Terminal*) to perform these steps in cPanel & WHM version 72 and later.
- In this document, `user` represents the cPanel user, and `greenmine` represents the Redmine web application's name.

For more information, read [Redmine's](#) documentation.

Install Redmine



Pre-installation settings.

Before you begin, ensure that you remove the operating system-provided Ruby installation. Then, install Ruby via the following RPMs:

- `ea-ruby24-mod_passenger`

Note:

If you enable this module, the system will disable Apache's `mod_userdir` module.

- `ea-ruby24-ruby-devel`
- `ImageMagick-devel`
- `ea-apache24-mod_env`

Note:

The `ea-apache24-mod_env` module allows you to add environment variables when you register your application. For more information about environment variables, read our [Application Manager](#) documentation.

To install these RPMs, run the following command as the `root` user:

```
yum install ea-ruby24-mod_passenger ea-ruby24-ruby-devel ImageMagick-devel  
ea-apache24-mod_env
```

1

Install the `bundler` gem.

To install the `bundler` Ruby gem, run the following command as a cPanel user:

```
scl enable ea-ruby24 'gem install bundler --user-install'
```

If you do not wish to use the `scl` utility, perform the following steps to install the `bundler` Ruby gem:

1. Log in to the server as a cPanel user.
2. Add the following line to your shell's `rc` file (for example, `/home/user/.bashrc` file):

```
source /opt/cpanel/ea-ruby24/enable
```

3. Log out of your terminal session, and log back in again.

Note:
You can also create an additional terminal session.

4. Run the `gem install bundler --user-install` command.

2

Extract the `redmine` archive file.

After you install the `bundler` and `rake` Ruby gems, download the application and extract the archive file. To do this, perform the following steps:

Note:
In the following examples, `redmine-X.X.X.tar.gz` represents the most recent `redmine` version.

1. Access the [Redmine](#) website in a web browser.
2. Download the most recent `.tar.gz` file from the *Stable Releases* section. To do this, run the following command:

Note:
Make certain that you install `wget` before you run this command.

```
wget url http://www.redmine.org/releases/redmine-X.X.X.tar.gz
```

3. Extract the `.tar.gz` file's contents. To do this, run the following command:

```
tar xvzf redmine-X.X.X.tar.gz
```

4. After you extract the `.tar.gz` file, rename the file to your Ruby application's name. To do this, run the following command:

```
mv /home/$user/redmine-X.X.X /home/user/greenmine
```

Remember:

`greenmine` represents your web application's name.

3

Create the database and database users.

After you extract the `redmine` archive file, create the application's database and configure the database users. To do this, perform the following steps:

1. Create production and development databases and database users in cPanel's [MySQL Database Wizard](#) interface (*cPanel >> Home >> Databases >> MySQL Database Wizard*).
2. Copy the contents of the `/home/user/greenmine/config/database.yml.example` file to the `/home/user/greenmine/config/database.yml` file.
3. Open the `/config/database.yml` file with a text editor.
4. Add the following information to the file:
 - The databases' names.
 - The databases' usernames.
 - The databases' passwords.

The file will resemble the following example:

✓ [Click to view...](#)

```
# Default setup is given for MySQL with ruby1.9.
# Examples for PostgreSQL, SQLite3 and SQL Server can be found at
the end.
# Line indentation must be 2 spaces (no tabs).

production:
  adapter: mysql2
  database: kermit_greenminetest
  host: localhost
  username: kermit
  password: "misspiggy1"
  encoding: utf8

development:
  adapter: mysql2
  database: kermit_greenmine_development
  host: localhost
  username: kermit_dev
  password: "fozzyl"
  encoding: utf8
```

Install your Ruby Gems.

To install the Ruby Gems that reside in the `/home/user/greenmine/Gemfile` file, change to the `/home/user/greenmine` directory and run the following command.

```
scl enable ea-ruby24 'bundle install'
```

Remember:

`greenmine` represents your web application's name.

5

Create a secret key.

After you install your Ruby Gems, create the "secret" key, which the system uses to encode cookies. To generate the "secret" key, run the following command:

```
scl enable ea-ruby24 'rake secret'
```

This command returns a key hash that resembles the following example:

```
0d3f00336c10ac419f9ed08b4448dc1e652179b86eb1c434d3e4cf68b15aa92fbac1dc1f24  
a19b150b63a133fe4e89b8fb79ca5e78026bcbcf8ce183f5c9ca8c
```

After you create the secret key, create the `/home/user/greenmine/config/secrets.yml` file and add the following content:

```
production:  
  secret_key_base: secretkey
```

Note:

Replace `secretkey` with the key that the `scl enable ea-ruby24 'rake secret'` command returned.

Remember:

`greenmine` represents your web application's name.

Create the database structure.

After you add the security key to the `/home/user/greenmine/config/secrets.yml` file, create the database structure. To do this, change to the `/home/user/greenmine` directory and run the following command:

```
RAILS_ENV=production scl enable ea-ruby24 'bundle exec rake db:migrate'
```

The output will resemble the following partial example:

✓ [Click to view...](#)

```

== 20160416072926 RemovePositionDefaults: migrating
=====
-- change_column("boards", :position, :integer, {:default=>nil})
  -> 0.0025s
-- change_column("custom_fields", :position, :integer, {:default=>nil})
  -> 0.0024s
-- change_column("enumerations", :position, :integer, {:default=>nil})
  -> 0.0023s
-- change_column("issue_statuses", :position, :integer, {:default=>nil})
  -> 0.0021s
-- change_column("roles", :position, :integer, {:default=>nil})
  -> 0.0022s
-- change_column("trackers", :position, :integer, {:default=>nil})
  -> 0.0020s
== 20160416072926 RemovePositionDefaults: migrated (0.0139s)
=====

== 20160529063352 AddRolesSettings: migrating
=====
-- add_column(:roles, :settings, :text)
  -> 0.0109s
== 20160529063352 AddRolesSettings: migrated (0.0109s)
=====

```

Create the data configuration.

After you create the database structure, create the database's default configuration data set. To do this, perform the following steps:

1. Run the following command:

```

RAILS_ENV=production scl enable ea-ruby24 'bundle exec rake
redmine:load_default_data'

```

2. When the system prompts you, select the database's locale. To do this, enter your chosen locale's two letter abbreviation in the command line, for example:

```

Select language: ar, az, bg, bs, ca, cs, da, de, el, en, en-GB, es,
es-PA, et, eu, fa, fi, fr, gl, he, hr, hu, id, it, ja, ko, lt, lv, mk,
mn, nl, no, pl, pt, pt-BR, ro, ru, sk, sl, sq, sr, sr-YU, sv, th, tr,
uk, vi, zh, zh-TW [en] en-GB

```

Note:

This example selects British English (`en-GB`) as the database's locale.

Set the file system permissions.

After you create the database's data configuration, set the file system permissions. To do this, change to the `/home/user/greenmine` directory and run the following series of commands:

```
$ mkdir -p tmp tmp/pdf public/plugin_assets
$ chown -R user: files log tmp public/plugin_assets
$ chmod -R 755 files log tmp public/plugin_assets
```



Register the application.

After you set the file permissions, register your web application. To do this, use UAPI's `PassengerApps::register_application` function, or cPanel's *Application Manager* interface (*cPanel >> Home >> Software >> Application Manager*).

Restart apache.

Important:

This step is **optional**. UAPI's `register_application` function performs an Apache restart, but may not execute immediately.

After you register the Apache configuration, restart Apache. To do this, run the following command as the root user:

```
/usr/local/cpanel/scripts/restartsrv_httpd
```

After Apache restarts, you can access the `greenmine` application in cPanel's *Application Manager* interface (*cPanel >> Home >> Software >> Application Manager*).

Additional documentation

Suggested documentation For cPanel users For WHM users For developers

- [How to Install a Redmine Web Application](#)
- [How to create a Sinatra Ruby Application](#)
- [How to Create Ruby Web Applications](#)
- [How to Create a Jekyll Ruby Application](#)
- [Guide to Ruby Installations - Requirements](#)

- [Ruby on Rails](#)
- [RubyGems](#)
- [Application Manager](#)
- [Apache Handlers](#)
- [Errors](#)

- [How to Install a Redmine Web Application](#)

- [How to create a Sinatra Ruby Application](#)
- [How to Create Ruby Web Applications](#)
- [How to Create a Jekyll Ruby Application](#)
- [Guide to Ruby Installations - Requirements](#)

- [UAPI Functions - PassengerApps::register_application](#)
- [UAPI Functions - PassengerApps::unregister_application](#)
- [UAPI Functions - PassengerApps::edit_application](#)
- [cPanel API 2 Functions - RoR::listapps](#)
- [cPanel API 2 Functions - RoR::listrewrites](#)