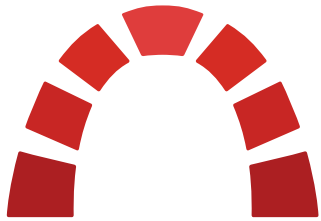


Redmine - Ein Handbuch zum Nachschlagen



REDMINE

flexible project management

Informationen zu Installation und Betrieb

Jan Niggemann, <jn@hz6.de>

Redmine - Ein Handbuch zum Nachschlagen

von Jan Niggemann

Veröffentlicht 28.08.2015

Copyright © 2013-2015 Jan Niggemann

Zusammenfassung

Dieses Buch soll Benutzern und Administratoren von Redmine-Systemen als Nachschlagewerk dienen.

Lizenz

Für den Moment ist dieses Buch unter der CC-BY-NC-SA 3.0 DE Lizenz lizenziert.

Sie haben folgende Rechte

Das Werk bzw. den Inhalt zu vervielfältigen, zu verbreiten und öffentlich zugänglich zu machen Abwandlungen oder Bearbeitungen des Werkes oder Inhaltes anzufertigen

Bedingungen

Namensnennung — Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.

Keine kommerzielle Nutzung — Dieses Werk bzw. dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden.

Weitergabe unter gleichen Bedingungen — Wenn Sie das lizenzierte Werk bzw. den lizenzierten Inhalt bearbeiten oder in anderer Weise erkennbar als Grundlage für eigenes Schaffen verwenden, dürfen Sie die daraufhin neu entstandenen Werke bzw. Inhalte nur unter Verwendung von Lizenzbedingungen weitergeben, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind.

Rechtlicher Hinweis

Auch wenn dieses Handbuch mit Sorgfalt geschrieben wurde, so kann der Autor doch keine Garantie für die Richtigkeit oder Fehlerfreiheit übernehmen.

Lizenzhinweise

Logo auf der Titelseite © 2009 Martin Herr

Widmung

Für Marie, Noemi, Sylviane und meine Eltern.

Vorwort	iii
1. Installation	1
Ruby, Rails, Gems und Passenger	1
Vorbereiten des Systems	1
Ruby herunterladen, bauen und installieren	1
Prüfen ob Ruby funktioniert	1
OpenSSL für Ruby bauen	1
Ist der Befehl Gem jetzt installiert?	2
Rails installieren	2
Passenger installieren (Applicationsserver)	2
Apache konfigurieren	2
Das Modul Passenger aktivieren	2
Redmine installieren	2
Redmine herunterladen	3
Das System weiter vorbereiten	3
Datenbank anlegen	3
Datenbankverbindung einrichten	3
Secret Token erstellen	3
Datenbank füllen (Tabellen anlegen)	3
Zugriffsrechte auf File-Ebene setzen	3
Redmine Installation prüfen	4
Apache und Passenger konfigurieren	4
Apache konfigurieren (subURI deployment)	4
2. Backup	5
Warum, was, wie und wie oft?	5
Backup der Datenbank	5
Backup der Daten	5
Backup-Intervall	5
3. Redmine Updates	6
Updates einspielen	6
Vorraussetzungen prüfen	6
Backup	6
Update durchführen	6
Datenbank aktualisieren	7
Aufräumarbeiten	7
Bekannte Probleme	7
A. Appendix A	8

Vorwort

Work in progress

Dieses Buch wird sich nur langsam mit Inhalt füllen. Ich habe eine Familie die ich sehr liebe und ich kann und möchte nicht jede freie Minute meines Lebens diesem Buch widmen. Trotzdem freue ich mich wenn Sie regelmäßig schauen, ob ein neues Kapitel vorliegt.

Hinweis in eigener Sache

Ich warte dieses Handbuch in meiner Freizeit und freue mich über eine kleine Aufmerksamkeit von

meinem ¹ oder Spenden via ².

Warum dieses Buch

Dieses Buch soll einerseits ein Leitfaden für Leser sein, die sich mit Redmine noch nicht (oder nicht so gut) auskennen. Andererseits hoffe ich das es, wenn es sich der Fertigstellung nähert, erfahrenen Anwender auch als Nachschlagewerk dienen kann.

Fehler und Hinweise

Ich gebe mir große Mühe, dieses Buch fehlerfrei zu halten. Allerdings wissen wir alle: "Wo gearbeitet wird, werden Fehler gemacht". Auch wenn ich sorgfältig prüfe was ich schreibe, so ist es doch wahrscheinlich das mir etwas "durchrutscht".

Bitte geben Sie mir einen kurzen Hinweis per eMail, wenn Sie einen Fehler finden. Vielen Dank!

Versionen

In der schnelllebigen EDV-Welt ändern sich besonders Versionsnummern laufend. Bitte beachten Sie, das sich auch Dateinamen und Pfade in dieser Anleitung können. Ich tue mein Bestes um die Daten aktuell zu halten, aber bitte prüfen Sie vor Eingaben, was Sie tun.

Textauszeichnungen in diesem Dokument

Text type	Bedeutung
monotype font	Benutzerein- bzw. -ausgaben und Konfigurationsdateien
↵	Zeile geht unten noch weiter

¹ <http://www.amazon.de/registry/wishlist/GAOFFPW3WLH2>

² https://www.paypal.com/cgi-bin/webscr?cmd=_s-xclick&hosted_button_id=GLW5CMTZFBP8A

Kapitel 1. Installation

Vorwort

Nach dem Lesen dieses Kapitels und dem Ausführen aller Schritte sollten Sie eine funktionierende Redmine-Instanz haben. Wenn etwas nicht funktioniert oder ein Schritt fehlschlägt, erstellen Sie bitte einen Post mit Details im Forum (Englisch)¹ und beziehen Sie sich auf diese Anleitung und die Schritt-Nummer. Ihr Post sollte auch die Ausgabe von `gem env` und /oder `RAILS_ENV=production /opt/redmine/script/about` enthalten.

Wenn Sie einen Fehler entdecken, informieren Sie mich bitte per email.

Über diese Anleitung

Dies ist eine Schritt-für-Schritt Anleitung um Redmine unter "Debian stable" (im Moment "Wheezy" genannt) zu installieren. Sie ist für Menschen geschrieben die sich mit Debian, der Shell, MySQL, Apache und Linux generell schon etwas auskennen. Um diese Anleitung kurz zu halten wird nicht jeder einzelne Schritt im Detail erklärt. Bitte beachten Sie: Ich gehe davon aus das Sie wissen wie Sie Ihr Werkzeug benutzen müssen. Sollte das nicht der Fall sein (wenn Sie z.B. nicht wissen wie man den MySQL Client benutzt oder Apache neu startet) benutzen Sie bitte die Suchmaschine Ihrer Wahl, informieren Sie sich, und lesen Sie erst dann weiter.

Ruby, Rails, Gems und Passenger

Vorbereiten des Systems

```
apt-get install gcc build-essential zlib1g zlib1g-dev libc \
libzlib-ruby libssl-dev libyaml-dev libcurl4-openssl-dev \
apache2-mpm-prefork apache2-prefork-dev libapr1-dev checkinstall ENTER
```

Ruby herunterladen, bauen und installieren

Bitte nach jeder Zeile **ENTER** drücken

```
cd ~
wget ftp://ftp.ruby-lang.org/pub/ruby/stable/ruby-1.9.3-p547.tar.gz
tar xvfz ruby-1.9.3-p547.tar.gz
cd ruby-1.9.3-p547
./configure --enable-pthread --prefix=/usr/local
make && checkinstall --type=debian --install=yes --fstrans=no --pakdir='~'
```

Prüfen ob Ruby funktioniert

```
ruby -v ENTER
```

Die Ausgabe sollte ungefähr so aussehen:

```
ruby 1.9.3p547 (2014-05-14 revision 45962) [i686-linux]
```

OpenSSL für Ruby bauen

Bitte nach jeder Zeile **ENTER** drücken

¹ <http://www.redmine.org/projects/redmine/boards/2>

```
cd ext/openssl/  
ruby extconf.rb  
make && checkinstall --type=debian --install=yes --fstrans=no --pakdir='~'
```

Ist der Befehl Gem jetzt installiert?

Seit Ruby 1.9 ist RubyGems per default mit dabei, daher sollte jetzt der Befehl "gem" verfügbar sein. Wenn die Installation fehlerfrei war wird der folgende Befehl eine Versionsnummer ausgeben (z.B. 1.8.2x):

```
gem -v 
```

Jetzt können wir rdoc installieren:

```
gem install rdoc 
```

Rails installieren

```
gem install rails --no-ri --no-rdoc 
```

Anmerkung

Eventuell erhalten Sie die Fehlermeldung "*no such file to load --zlib (LoadError)*". In diesem Fall müssen Sie zuerst zlib installieren:

```
cd ruby-1.9.3-p547/ext/zlib/  
ruby extconf.rb  
make  
make install
```

Passenger installieren (Applicationserver)

```
gem install passenger   
passenger-install-apache2-module 
```

Apache konfigurieren

Folgendes in /etc/apache/mods-available/passenger.load einfügen (wenn nötig Pfade anpassen).
Das folgende ist eine einzelne Zeile!

```
LoadModule passenger_module ↵  
/usr/local/lib/ruby/gems/1.9.1/gems/passenger-4.0.46/ext/ ↵  
apache2/mod_passenger.so
```

Und das hier kommt nach /etc/apache/mods-available/passenger.conf (wenn nötig Pfade anpassen).

```
PassengerRoot /usr/local/lib/ruby/gems/1.9.1/gems/passenger-4.0.46  
PassengerRuby /usr/local/bin/ruby  
PassengerDefaultUser www-data
```

Das Modul Passenger aktivieren

```
a2enmod passenger 
```

Redmine installieren

Redmine herunterladen

Holen Sie das neueste zip von redmine.org² und extrahieren sie es nach `/opt/redmine`

Das System weiter vorbereiten

Hinweis: Die Installation von `libmagick9-dev` installiert eine Menge Pakete (`depends / recommends`)

Bitte nach jeder Zeile `[ENTER]` drücken

```
gem install bundler
apt-get install libmagick9-dev
cd /opt/redmine
bundle install --without postgresql
```

Datenbank anlegen

Vorraussetzung für diesen Schritt ist, das Sie MySQL bereits installiert haben. Sollte das nicht der Fall sein, installieren Sie bitte MySQL mit dem Paketmanager Ihrer Distribution. Diese Befehle müssen Sie in Ihrem MySQL client ausführen, nicht auf der Shell! Wenn Sie nicht wissen wie Sie Ihren MySQL Client starten, durchsuchen Sie bitte das Internet und machen Sie Sich vertraut mit Basisaufgaben (Anmeldung, Userverwaltung, Datenbankverwaltung,...)

Bitte nach jeder Zeile `[ENTER]` drücken

```
create database redmine character set utf8;
create user 'redmine'@'localhost' identified by 'XXX';
grant all privileges on redmine.* to 'redmine'@'localhost';
```

Datenbankverbindung einrichten

Folgendes in `/opt/redmine/config/database.yml` schreiben

```
production:
  adapter: mysql2
  database: redmine
  host: localhost
  username: redmine
  password: XXX
```

Secret Token erstellen

`rake generate_secret_token [ENTER]`

Datenbank füllen (Tabellen anlegen)

`RAILS_ENV=production rake db:migrate [ENTER]`

Zugriffsrechte auf File-Ebene setzen

Bitte nach jeder Zeile `[ENTER]` drücken

```
cd /opt/redmine
mkdir tmp tmp/pdf public/plugin_assets
chown -R www-data:www-data files log tmp public/plugin_assets
```

² <http://www.redmine.org/projects/redmine/wiki/Download>


```
chmod -R 755 files log tmp public/plugin_assets
```

Redmine Installation prüfen

```
ruby script/rails server webrick -e production 
```

Öffnen Sie jetzt <http://localhost:3000> und sehen Sie Redmine "in Aktion".

Apache und Passenger konfigurieren

In dieser Anleitung konfigurieren wir den Einsatz in einer sub-URI. Lesen Sie andere Anleitungen auf redmine.org wenn Sie lieber einen name-based virtual host konfigurieren möchten.

Apache konfigurieren (subURI deployment)

```
ln -s /opt/redmine/public /var/www/redmine 
```

Fügen Sie dies in `/etc/apache2/sites-available/redmine` ein:

```
<Location /redmine>
  RailsEnv production
  RailsBaseURI /redmine
  Options -MultiViews
</Location>
```

Die neue Konfiguration wird wie folgt aktiviert:

```
a2ensite redmine 
```

Starten Sie Apache neu, prüfen Sie ob <http://yourhost.com/redmine> funktioniert und freuen Sie Sich wenn ja :-) Wenn Sie etwas unerwartetes sehen, posten Sie bitte Details im Forum³ (Englisch) und nehmen Sie Bezug auf diese Anleitung und die Kapitel-Nummer. Bitte fügen Sie die Ausgabe von `gem env` und/oder `RAILS_ENV=production /opt/redmine/script/about` bei.

³ <http://www.redmine.org/projects/redmine/boards/2>

Kapitel 2. Backup

Warum, was, wie und wie oft?

Niemand ist vor Hardwareschäden sicher, es kann jederzeit passieren. Festplatten fallen aus, Blitze schlagen ein... Und bitte kommen Sie mir jetzt nicht mit "Ich brauche kein Backup, ich habe ein RAID". Auch bei vermeintlichen Experten ist dieser Unsinn leider viel zu oft zu hören: Was nützt es Ihnen denn, wenn ein Dateisystemfehler auf mehrere Festplatten verteilt wird?

Ihr Backup sollte 2 Dinge umfassen, einmal die Datenbank und einmal die Daten selber. Mit welchen Tools Sie das genau anstellen bleibt Ihnen überlassen, aber eines dürfen Sie auf keinen Fall tun: Das Backup einfach bleiben lassen.

Backup der Datenbank

Jetzt könnte ich, ähnlich wie ich es in Kapitel 1 gemacht habe, einfach darauf verweisen das Google schon Lösungen präsentieren wird, wenn Sie denn nur danach suchen. Für Sie als Einsteiger mache ich es aber einfach: Benutzen Sie die Tools, die Sie schon kennen. Wenn Ihnen jetzt nichts einfällt, dann nehmen Sie einfach `automysqlbackup`¹. Wie Sie das einrichten erfahren Sie auf der Homepage.

Backup der Daten

Es ist ausreichend die Anhänge unter `/opt/redmine/files` zu sichern. Natürlich kann es auch Sinn machen, das Sie Ihren gesamten Redmine-Ordner sichern. Wenn Sie z.B. Konfigurationsoptionen verändern ist es schön und nützlich, ein Backup zur Hand zu haben. Auch hier gilt: Nehmen Sie einfach ein Tool, das Sie schon kennen. Das kann `rsync` sein, ein selbstgeschriebenes Shell-Skript oder spezialisierte Tools wie `rdiff-backup`, `duplicity`, `attic` oder `obnam`. Ich benutzte letzteres.

Backup-Intervall

Die Fragen nach dem Warum, was und wie sind geklärt (falls nicht, bitte das Kapitel von vorn beginnen). Bleibt noch die Frage nach dem Intervall. Wie oft Sie sichern ist letztlich von der Aktivität Ihrer Redmine-Installation abhängig. Die Leitfrage die Sie Sich stellen sollten ist folgende: Wie viele Daten kann ich mir leisten zu verlieren? Wenn Sie z.B. einmal täglich sichern, verlieren Sie im schlimmsten Fall die Daten eines ganzen Tages (und damit Arbeit und Geld). Wenn Sie 3 User in Redmine haben, sind das natürlich weniger Daten als 150 User produzieren, es kommt also "drauf an". Wenn Sie jetzt unbedingt einen Richtwert hören wollen: Machen Sie 1x Mittags und 1x Nachts ein Backup der Daten.

¹ <http://sourceforge.net/projects/automysqlbackup/>

Kapitel 3. Redmine Updates

Alle paar Wochen wird eine neue Version von Redmine veröffentlicht, die Fehler behebt und neue Features und Übersetzungen hinzufügt. Da die Redmine-Entwickler automatische Tests laufen lassen und darauf achten das auch für neue Features Tests erzeugt werden, kann davon ausgegangen werden, das diese Releases gefahrlos eingesetzt werden können.

Trotzdem scheinen viele Administratoren vor dem Update einer Produktiv-Instanz zurück zu schrecken. Das hier kein Grund zur Sorge besteht, soll dieses Kapitel zeigen.

Updates einspielen

Vorraussetzungen prüfen

Bevor Sie beginnen sollten Sie prüfen ob Ihr Server die Installationsvoraussetzungen¹ für die neue Version erfüllt.

Backup

Bevor Sie beginnen sollten Sie überprüfen ob Ihr Backup sauber funktioniert. Wenn etwas schief geht, können Sie ein Restore machen und haben nur wenige Daten verloren. Außerdem ist es immer eine gute Idee, Backups mal testweise wiederherzustellen.

Sichern Sie nun den Inhalt Ihrer Datenbank in eine Datei, der folgende Befehl tut's für MySQL:

```
mysqldump -u BENUTZERNAME -p PASSWORT DATENBANK | gzip > ~/redmine_db.gz
```

Update durchführen

Zunächst einmal müssen Sie das aktuelle Redmine-Paket herunterladen (Sie erinnern sich von wo?) und nach `/opt/redmine_neu` entpacken.

Kopieren Sie jetzt das Konfigurationsfile für die Datenbank (`config/database.yml`) vom alten System zum Neuen. Wenn Sie Redmine mit MySQL und Ruby 1.9 einsetzen, ändern Sie bitte den Adapter zu `mysql2`.

Kopieren Sie das Basis-Konfigurationsfile `config/configuration.yml` vom alten System zum Neuen.

Kopieren Sie den Inhalt des Ordners `files` zum neuen System. In diesem Ordner befinden sich alle hochgeladenen bzw. angehängten Dateien.

Jetzt müssen wir die Plugins kopieren. Was Sie tun müssen hängt davon ab, welche Version Sie aktualisieren. Falls Sie von Redmine <2.0 kommen, kopieren Sie die Ordner Ihrer *selbst installierten Plugins* von `vendor/plugins`, falls Sie von anderen Versionen kommen, die aus dem Ordner `plugins` (ohne `vendor` davor) in den entsprechenden Ordner des neuen Systems. Stellen Sie dabei sicher das Sie nur Plugins kopieren, die nicht Teil Ihres alten Redmines waren.

Erzeugen Sie einen neuen Schlüssel, führen Sie folgendes in `/opt/redmine_neu` aus:

```
rake generate_secret_token
```

Prüfen Sie, ob für das von Ihnen eingesetzte Theme eine neue Version verfügbar ist. Wenn nicht, kopieren Sie das Theme einfach zum neuen System.

¹ <http://www.redmine.org/projects/redmine/wiki/RedmineInstall#Requirements>

Achtung

Überschreiben Sie `config/settings.yml` nicht mit der alten Version!

Datenbank aktualisieren

Aktualisieren Sie die Datenbank, der folgende Befehl fügt z.B. neu hinzugekommene Tabellen hinzu:

```
rake db:migrate RAILS_ENV=production
```

Und das Selbe nochmal für Plugins:

```
rake redmine:plugins:migrate RAILS_ENV=production
```

Aufräumarbeiten

Fertig, jetzt sollten Sie noch den Cache leeren und existierende Sessions beenden:

```
rake tmp:cache:clear  
rake tmp:sessions:clear
```

Starten Sie die Redmine-Instanz neu und prüfen Sie falls nötig die Rechte für neu hinzu gekommene Funktionen (Administration -> Rollen).

Bekannte Probleme

Wenn Sie bei der Migration eine Fehlermeldung bezüglich "member_roles" bekommen, dann kann das an einer früheren Migration liegen, die korrekt durchgelaufen ist. In diesen Fall könnten die Tabellen `member_roles` und / oder `group_users` schon existieren und die Migration schlägt fehl. Löschen Sie in diesem Fall einfach die Tabellen, oder benennen Sie sie um:

```
mysql> rename table member_roles to member_roles_saved  
mysql> rename table groups_users to groups_users_saved
```

Ein weiterer Fehler der manchmal auftritt lautet "undefined method 'add_frozen_gem_path'". Wenn Sie diesen Fehler sehen prüfen Sie bitte, ob der Ordner `vendor/rails` existiert. Falls ja, entfernen Sie ihn bitte oder benennen Sie ihn um, er könnte eine alte RoR-Version enthalten.

Sehen Sie, so schlimm war das doch garnicht.

Anhang A. Appendix A

...