

Procedimiento de instalación de Redmine 2.5.1.stable utilizando debian wheezy 7.0 (Versión 1.0)

El software a instalar es:

```
Redmine version      2.5.1.stable
Ruby version         2.0.0-p247 (2013-06-27) [i686-linux]
Rails version        3.2.17
Environment          production
Database adapter     Mysql2
SCM:
Git                  1.7.10.4
```

1) Actualice su debian:
aptitude update

2) Cree el usuario redmine:
adduser --disabled-password redmine

3) Instale el siguiente software:
aptitude install curl mysql-server mysql-client libmysqlclient-dev ruby-rmagick \
librmagick-ruby imagemagick ruby1.9.1-dev libmagickwand-dev sudo wget

4) Ingrese a su base de datos MySQL y cree la BD en blanco:
mysql -U -p

Dentro de MySQL:

```
# CREATE DATABASE redmine CHARACTER SET utf8;
# GRANT ALL PRIVILEGES ON redmine.* TO 'redmine'@'localhost' IDENTIFIED BY
'my_password';
```

Aquí se crea el usuario redmine con la clave de acceso “my_password” (Ponga el de su preferencia)

Salga de MySQL:

```
# quit;
```

5) Cree el usuario “redmine” e ingrese al sistema como este:
adduser redmine
su - redmine

6) Instale RVM (Ruby Version Manager)
curl -L https://get.rvm.io | bash

7) Asegúrese de tener en el /home/redmine/.profile lo siguiente:

```
# set PATH so it includes rvm bin if it exists
if [ -d "$HOME/.rvm/bin" ] ; then
  PATH="$PATH:$HOME/.rvm/bin"
fi
```

```
if [ -s "$HOME/.rvm/scripts/rvm" ] ; then
  source "$HOME/.rvm/scripts/rvm"
fi
```

```
if [ -s "$HOME/.rvm/scripts/completion" ] ; then
  source "$HOME/.rvm/scripts/completion"
fi
```

- 8) Descargue e instale Redmine:

```
# wget -c http://www.redmine.org/releases/redmine-2.5.1.tar.gz
# tar xvzf redmine-2.5.1.tar.gz
# mv redmine-2.5.1 redmine
# cd redmine
```

- 9) Configure el acceso a la Base de Datos:

```
# cp config/database.yml.example config/database.yml
```

- 10) Edite el archivo config/database.yml y coloque sus credenciales de acceso a la BD MySQL:

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

- 11) Configurar SUDO para permitir al usuario que ejecuta el REDMINE a ejecutar comandos como root sin password. **NOTA:** debe ser usuario root para realizar esta configuración:

Cree el archivo /etc/sudoers.d/redmine y agregue la siguiente línea:

```
redmine    ALL=(root)  NOPASSWD:ALL
```

NOTA: En nuestro caso el usuario que ejecuta el REDMINE es el usuario “redmine”

Reinicie el sudo

```
#!/etc/init.d/sudo restart
```

- 12) Redmine usa el “bundler” para manejar las dependencias de las “gems” que usa ruby:

```
# sudo aptitude install bundle
```

NOTA: Asegúrese de estar en el directorio /home/redmine/redmine y ejecute:

```
# bundle install --without=test development sqlite3 postgres
```

- 13) Generación del almacén secreto de tokens para redmine:

```
# bundle exec rake generate_secret_token
```

- 14) Cree la estructura de la Base de Datos de Redmine:

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

15) Llenamos los datos “por omisión” del Redmine (en este caso aprovechamos de instalarlo en español):

```
# RAILS_ENV=production REDMINE_LANG=es rake redmine:load_default_data
```

16) Creamos los directorios de trabajo de redmine y les damos la propiedad y permisos correspondientes:

```
# mkdir -p tmp tmp/pdf public/plugin_assets  
# sudo chown -R redmine:redmine files log tmp public/plugin_assets  
# sudo chmod -R 755 files log tmp public/plugin_assets
```

17) Aquí puedes ejecutar el servidor webrick para probar la nueva instancia de Redmine instalada:

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

Debe accederla colocando la IP de su servidor con el puerto 3000, ejemplo:

<http://localhost:3000> ó <http://192.168.1.100:3000>

El usuario y password por omisión es: admin/admin

Instalación de repositorio GIT para trabajar con Redmine:

GIT para poder servir un repositorio a través de Redmine utiliza un directorio para el repositorio “bare” y un directorio para ser utilizado como directorio de trabajo del SCM.

En este caso utilizaremos el directorio: /var/cache/git/<nombre repositorio>.git para crear los repositorios “bare”

Y utilizaremos el directorio /srv/repos/git/<identificador del repositorio> como los directorios de trabajo del GIT.

Procedimiento para crear repositorio GIT para proyectos REDMINE:

1) Instale el GIT en debian:

```
# sudo aptitude install git
```

2) Cree el directorio del repositorio “bare” del proyecto: (para efectos del ejemplo se llama “demo”):

```
# sudo mkdir -p /var/cache/git/demo.git
```

3) Ingrese al directorio e inicialice el repositorio “bare”

```
# cd /var/cache/git/demo.git  
# sudo git init --bare --share
```

4) Ingrese al directorio de trabajo de los repositorios y clone el directorio “bare” en un repositorio local:

```
# sudo mkdir -p /srv/repos/git  
# cd /srv/repos/git  
# sudo git clone /var/cache/git/demo.git demo
```

Agreguemos un archivo LEEME, hagamos un primer commit y realicemos el primer push del

```
repositorio:  
# cd demo  
# sudo touch LEEME  
# sudo git add LEEME  
# sudo git commit -am "Primer Commit"  
# sudo git push -u origin master
```

- 5) Levantemos de nuevo el servidor webrick, ingresemos a REDMINE: Creemos un nuevo proyecto "Demo"

```
# cd /home/redmine/  
# ruby script/rails server webrick -e production
```

Debe accederlo colocando la IP de su servidor con el puerto 3000, ejemplo:
<http://localhost:3000> ó <http://192.168.1.100:3000>

El usuario y password por omisión es: admin/admin

Creemos un nuevo proyecto "Demo"

Dentro del nuevo proyecto "Demo" vaya al menú Configuración -> Repositorios
Cree un nuevo repositorio y coloque los siguientes datos (adáptelo a su caso particular):

SCM = Git

Identificador = demo

URL = /var/cache/git/demo.git (aquí siempre se apuntará al directorio "bare" del repositorio, nunca al directorio de trabajo de SCM).

Codificación = UTF-8

- 6) Ya puede visualizar el repositorio dentro de la Opción "Repositorio" de REDMINE.

Configurar el servidor puma con nginx como proxy web para servir redmine en debian wheezy:

- 1) Instale el servidor puma:

```
# su - redmine
# cd redmine ; debe estar parado en /home/redmine/redmine
# sudo apt-get install libssl-dev
# sudo gem install puma
```

- 2) Cree el archivo de configuración para puma

```
# vi /home/redmine/redmine/config/puma.rb
```

Con el siguiente contenido:

```
directory '/home/redmine/redmine'
environment 'production'

threads 0,8
workers 1
preload_app!

daemonize true

bind 'unix:///home/redmine/redmine/tmp/sockets/redmine.sock'
pidfile '/home/redmine/redmine/tmp/pids/puma.pid'
stdout_redirect '/home/redmine/redmine/log/puma.stderr.log',
'/home/redmine/redmine/log/puma.stdout.log'

on_worker_boot do
  ActiveSupport.on_load(:active_record) do
    ActiveRecord::Base.establish_connection
  end
end
```

- 3) Cree el vínculo simbólico del archivo de configuración puma.rb al directorio /home/redmine/config:

```
# mkdir /home/redmine/config
# ln -s /home/redmine/redmine/config/puma.rb /home/redmine/config/
```

- 4) Cree el directorio, el script y configure los permisos para ejecutar el servidor puma:

```
# mkdir /home/redmine/bin
# vi /home/redmine/bin/service_puma.sh
Con el siguiente contenido:
#!/bin/bash

function start () {
  echo "Start Puma Server..."
  puma -C "$HOME/config/puma.rb"
  echo "Done"
}

function stop () {
  echo "Stop Puma Server..."
  kill $(cat $HOME/redmine/tmp/pids/puma.pid)
  echo "Done"
}

case "$1" in
  start)
    start
```

```

;;
stop)
    stop
;;
restart)
    stop
    start
;;
*)
    echo "Usage : service_puma.sh {start|stop|restart}"
;;
esac

```

```
# sudo chmod 755 /home/redmine/bin/service_puma.sh
```

- 5) Verifique que en el archivo `/home/redmine/.profile` esté incluido el directorio `$HOME/bin`

Debe contener:

```

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi

```

- 6) Instale el servidor proxy web nginx

```
# sudo aptitude install nginx
```

- 7) Elimine el vínculo simbólico para desactivar el virtualhost “default” de nginx:

```
# sudo rm /etc/nginx/sites-enabled/default
```

- 8) Cree el archivo de virtualhost para redmine:

```
# sudo vi /etc/nginx/sites-available/redmine
```

Con el siguiente contenido:

```

upstream puma_redmine {
    server      unix:/home/redmine/redmine/tmp/sockets/redmine.sock fail_timeout=0;
    #server    127.0.0.1:3000; #dev mode
}

server {
    server_name  redmine.cenditel.gob.ve #aquí coloque su server_name
    listen      0.0.0.0:80;
    root        /home/redmine/redmine;

    access_log  /var/log/nginx/redmine.log;
    error_log   /var/log/nginx/redmine.log;

    location / {
        try_files $uri @ruby;
    }

    location @ruby
    {
        #proxy_set_header X-Forwarded-Proto https; # unquote if you are in HTTPS
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_read_timeout 300;
        proxy_pass http://puma_redmine;
    }
}

```

- 9) Cree el vínculo simbólico para activar el nuevo virtualhost:

```
# sudo ln -s /etc/nginx/sites-available/redmine /etc/nginx/sites-enabled/
```

10) Cree el script de arranque del redmine:

```
# sudo vi /etc/init.d/redmine
```

Con el siguiente contenido:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          redmine
# Required-Start:    $local_fs $network $remote_fs
# Required-Stop:     $local_fs $remote_fs
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: projects manager
# Description:       This file should be used to start and stop redmine.
### END INIT INFO

[ -f /etc/default/rcS ] && . /etc/default/rcS
. /lib/lsb/init-functions

case "$1" in
  start)
    su - redmine -c 'service_puma.sh start'
    ;;
  stop)
    su - redmine -c 'service_puma.sh stop'
    ;;
  restart)
    su - redmine -c 'service_puma.sh restart'
    ;;
  *)
    echo "Usage : /etc/init.d/redmine {start|stop|restart}"
    ;;
esac
```

11) Aplique permisos de ejecución al script de arranque:

```
# sudo chmod 755 /etc/init.d/redmine
```

12) Agreguemos el script de arranque automatizado al arranque del servidor:

```
# sudo insserv redmine
```

13) Como usuario “root” podemos arrancar el servidor de redmine y el nginx:

```
# /etc/init.d/redmine start
# /etc/init.d/nginx start
ó
# service redmine start
# service nginx start
```

Si todo va bien debería mostrarnos el siguiente mensaje:

```
Start Puma Server...
[9577] Puma starting in cluster mode...
[9577] * Version 2.8.2 (ruby 2.0.0-p247), codename: Sir Edmund Percival Hillary
[9577] * Min threads: 0, max threads: 8
```

```
[9577] * Environment: production
[9577] * Process workers: 1
[9577] * Preloading application
[9577] * Listening on unix:///home/redmine/redmine/tmp/sockets/redmine.sock
[9577] * Daemonizing...
Done
Starting nginx: nginx.
```


Instalación del plugin `redmine_git_hosting` con `gitolite` para automatizar la administración de repositorios `GIT` dentro de `REDMINE`

- 1) Detenga los servicios de `redmine` y `nginx` (como `root`)
`# /etc/init.d/redmine stop`
`# /etc/init.d/nginx stop`
- 2) Instale la librería `libicu-dev`
`# aptitude install libicu-dev`
- 3) Ingrese como usuario `redmine`:
`# su - redmine`
- 4) Vaya al directorio de `plugins` dentro del directorio donde está instalado `REDMINE`:
`# cd REDMINE_ROOT/plugins`
- 5) Clone los repositorios de los `plugins` a instalar, ingrese al directorio y actualice a la rama `0.7.2`:
`# git clone https://github.com/jbox-web/redmine_git_hosting.git`
`# cd redmine_git_hosting/`
`# git checkout v0.7.2`
- 6) Vaya al directorio raíz de la instalación de `REDMINE`, ejecute el “`bundle`” y luego haga la instalación de los nuevos `plugins`:
`# cd REDMINE_ROOT`
`# sudo bundle install`
`# sudo RAILS_ENV=production rake redmine:plugins:migrate`
- 7) Inicie de nuevo el `redmine` y el `nginx`: (como `root`)
`# /etc/init.d/redmine start`
`# /etc/init.d/nginx start`
- 8) Ingrese de nuevo como usuario `redmine` y cree la clave pública y privada del usuario `redmine` en el directorio “`ssh_keys`” dentro del `plugin redmine_git_hosting`, (sin clave, opción: `-N` ", note que son 2 apóstrofes, no comillas dobles.)
`# su - redmine`
`# ssh-keygen -N " -f \
/home/redmine/redmine/plugins/redmine_git_hosting/ssh_keys/redmine_gitolite_admin_id_rsa`
- 9) Creamos el usuario `git`, sin acceso `shell` (sólo `root` con “`su - git`”) y sin `password` y especificamos como directorio hogar del usuario el directorio que usaremos para albergar los repositorios “`bare`”:

`# adduser --system --shell /bin/bash --gecos 'git version control' --group --disabled-password --home /var/cache/git git`

Cambiamos la propiedad de los archivos del directorio `/var/cache/git` al nuevo usuario:

```
# chown -R git:git /var/cache/git
```

- 10) Siendo `root` haga login como usuario `git`:

```
# su - git
```

11) Verifique que no exista (o que esté vacío) el ~/.ssh/authorized_keys del usuario git

12) Asegúrese de que la clave pública del redmine esté disponible en el directorio raíz del usuario git, \$HOME/redmine_gitolite_admin_id_rsa.pub, si no es así, cree un vínculo simbólico que apunte a la misma.

```
# sudo ln -s \  
/home/redmine/redmine/plugins/redmine_git_hosting/ssh_keys/redmine_gitolite_admin_id_rsa.  
pub /var/cache/git/
```

13) Agregue el directorio “bin” al PATH en el .profile del usuario git:

```
# vi ~/.profile  
Debe contener el siguiente código:  
set PATH so it includes user private bin if it exists  
if [ -d "$HOME/bin" ] ; then  
    PATH="$PATH:$HOME/bin"  
fi
```

14) Ejecute los siguientes comandos:

```
root$ su - git  
git$ mkdir /var/cache/git/bin  
git$ source /var/cache/git/.profile  
git$ git clone http://github.com/sitaramc/gitolite  
git$ gitolite/install -to /var/cache/git/bin  
git$ gitolite setup -pk redmine_gitolite_admin_id_rsa.pub
```

15) Edite el archivo .gitolite.rc y asegúrese de que quede como se describe a continuación:

```
git$ vi .gitolite.rc  
## Look for GIT_CONFIG_KEYS and make it look like :  
GIT_CONFIG_KEYS          => '.*',  
## Then save and exit
```

16) Si planea que los repositorios se inicialicen automáticamente debe agregar las siguientes líneas en el .gitolite.conf

```
# vi .gitolite/conf/gitolite.conf  
Debe contener:  
repo    @all  
RW+          = redmine_gitolite_admin_id_rsa
```

17) Configure el sudo, agregando (como root) al archivo /etc/sudoers.d/redmine lo siguiente:

```
# vi /etc/sudoers.d/redmine  
  
redmine    ALL=(git)      NOPASSWD:ALL  
git        ALL=(redmine)  NOPASSWD:ALL
```

18) Agregue al /etc/sudoers (como root) las siguientes líneas:

```
# vi /etc/sudoers  
Defaults:git    !requiretty  
Defaults:redmine !requiretty
```

19) Reinicie el sudo:

```
# /etc/init.d/sudo restart
```

20) Asegúrese que la clave pública del usuario redmine este en el .known_hosts del usuario git, para

eso, ejecute los siguientes comandos: (en ningún momento le debe pedir contraseña):

```
root$ su - redmine
redmine$ssh -i \
/home/redmine/redmine/plugins/redmine_git_hosting/ssh_keys/redmine_gitolite_admin_id_rsa \
git@localhost info
```

Debe producir una salida parecida a la siguiente:

```
hello redmine_gitolite_admin_id_rsa, this is git@redmine2 running gitolite3 v3.5.3.1-25-ge492c03
on git 1.7.10.4
```

```
R W  gitolite-admin
R W  testing
```

21) Instale el intérprete de Ruby:

Dado que los comandos hooks de post-receive (comandos lanzados posteriormente al ejecutar un push que actualizan la información del repositorio dentro de redmine) son programados en Ruby, este debe estar instalado, esto no debe hacer ningún conflicto con el RVM (Ruby Version Manager):

```
# aptitude install ruby
```

22) Fin de la instalación:

El plugin ya está instalado y probablemente requiera algunos ajustes a través de la opción:

Administración -> Redmine Git Hosting

Se sugiere vaya a la pestaña “Config Test” para que compruebe que todas las configuraciones son correctas, para evitar problemas de permisos ejecuta al finalizar:

```
#chown -R redmine:redmine /home/redmine/redmine
```

Manual realizado por José Alejandro Vázquez (avazquez AT SPAMFREE cenditel DOT gob DOT ve) para la Fundación Centro Nacional de Desarrollo e Investigación en Tecnologías Libres (CENDITEL), adscrita al MPP para la Ciencia, Tecnología e Innovación.



Este documento se distribuye de forma libre bajo los criterios de la Licencia Creative Commons que permite: A otros, modificar y ampliar la obra para fines no comerciales, siempre y cuando se de crédito a su (s) autor (es) y la licencia de las nuevas obras creadas a partir de la original posean iguales términos y condiciones a la licencia de la obra original.

URL's Consultadas:

https://github.com/jbox-web/redmine_git_hosting/wiki/Redmine-Installation

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

https://github.com/jbox-web/redmine_git_hosting/wiki/Step-by-step-installation-instructions

Puede revisar también:

<http://ruby-journal.com/how-to-setup-rails-app-with-puma-and-nginx/>

<http://www.calculate-linux.org/main/en/gitolite>