Установка и настройка Debian + Redmine, или Путь Джедая.

“Да прибудет с тобой сила!”

Оглавление

Обозначения

Для обозначения вводимых команд используется курсив.
Например: # *nano /etc/hostname*
означает что команда выполняется с правами суперпользователя.
$ означает что для выполнения команды права суперпользователя не требуются.
Текст изменяемого файла указан без курсива и выделяется зеленым цветом.
Выходные данные в консоли выделяются цветом.

Базовая настройка сервера на Debian.

Сначала определимся с версией Debian, которую будем устанавливать.

Debian Squeeze 6.0.6 - версия ядра Linux 2.6.32 i386.

Для установки можно использовать образ debian-6.0.6-i386-netinst.iso. При установке снять все галочки с дополнительных пакетов. Назначаем пароль для root пользователя. Если пароль для root не задавался, то далее в тексте нужно будет использовать команду sudo, для запуска программа с правами суперпользователя.

Таким образом, установив базовую систему, приступим к ее настройке.

Настройку будем проводить под root'ом. При старте системы логинимся, используя пользователя root.

Настройка hostname

Назначаем имя компьютера (hostname):

```
# nano /etc/hostname
redmine

# hostname -F /etc/hostname
# exit
```

Настройка сети

Заходим под root. Проверим видит ли система наши сетевые карты (интерфейсы). Для этого смотрим сообщения ядра, которые выдавались при загрузке:

```
# dmesg | grep eth
```

В Linux ядро определяет сетевые интерфейсы как eth0, eth1 и так далее. Убеждаемся что, система видит сетевые интерфейсы.

Настройка статического IP-адреса

```
# nano /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1

auto eth0
```

Задаем DNS-сервер:

```
# nano /etc/resolv.conf
nameserver 192.168.1.1
```

Более детальная информация по настройке сети в Debian:
http://www.aitishnik.ru/linux02.html

Создание пользователей

Подключение к серверу с полномочиями root небезопасно, так как он имеет неограниченные права и это единственный пользователь, имя которого известно заранее.
Создадим нового пользователя alexandr и назначим ему группу users. А также создадим для него домашнюю папку:

```
# useradd -g users -s /bin/bash -d /home/alexandr -m alexandr
```

Назначим пароль новому пользователю.

```
# passwd alexandr
```

Для переключения между пользователями используется команда ***su <username>***. Для переключения на пользователя root достаточно команды ***su*** без параметров.

Настройка SSH соединения

Устанавливаем SSH-сервер для удаленного доступа к серверу и fail2ban для блокирования подбора паролей для ssh, http и др.

```
# aptitude install ssh fail2ban
```

Теперь можно подключиться к серверу удаленно:

```
$ ssh root@192.168.1.200
The authenticity of host '192.168.1.200 (192.168.1.200)' can't be established.
RSA key fingerprint is 4b:1a:4b:cb:d5:58:fa:30:66:9f:ee:b5:a3:a6:55:9b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.200' (RSA) to the list of known hosts.
```

Аутентификация по ключам

Настраиваем аутентификацию по ключам, позволяющую пользователю войти в систему на другом компьютере от своего имени без ввода пароля. Создаем ключи на ***локальном компьютере***:

```
$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/alexandr/.ssh/id_dsa):
```

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alexandr/.ssh/id_dsa.
Your public key has been saved in /home/alexandr/.ssh/id_dsa.pub.
The key fingerprint is:
ba:45:86:af:e0:cc:23:26:ae:60:8f:dd:7b:15:c1:e9 alexandr@alexandr-pc
```

Параметр -t задает тип ключа. Ответьте на вопросы, и команда создаст пару ключей в каталоге ~/.ssh; файлы будут названы в соответствии с типом, например, id_dsa и id_dsa.pub. Последний файл - публичный ключ, который копируется на удаленный компьютер, другой файл - приватный ключ, и им не следует делиться ни с кем.

Скопируем публичный ключ на удаленный компьютер командой:

```
$ ssh-copy-id -i ~/.ssh/id_dsa.pub alexandr@192.168.1.200
```

Команда запросит у вас пароль на удаленном компьютере и за тем скопирует туда публичный ключ. При следующей попытке подключиться к этом компьютеру под данным пользователем вы войдете в систему без запроса пароля.

Внимание! Если публичный ключ добавлен на удаленном компьютере, а при подключении выходит ошибка Agent admitted failure to sign using the key, на локальном компьютере выполните команду ***ssh-add***, которая добавит сгенерированный ключ в клиент SSH.

Использование Screen

Программа, на которую стоит обратить внимание всем пользователям SSH - *screen*. Если вы запускаете программу в удаленной оболочке и сетевое соединение разрывается, оболочка завершится со всеми программами, которые были в ней запущены. *Screen* называют мультиплексором терминалов, т.е. вы запускаете его в оболочке, и на первый взгляд ничего не меняется, но теперь вы находитесь в сессии *screen*. Устанавливаем на ***удаленном компьютере***:

```
# aptitude install screen
```

Соединитесь по SSH, запустите программу и нажмите <Ctrl>+<A>, затем <D>, и сессия терминала исчезнет.

```
$ ssh alexandr@192.168.1.200
$ screen
```

Выполните команду:

```
$ screen -r
```

и сессия терминала появится снова с запущенной в ней программой, как будто никуда не исчезала. Это особенно удобно с SSH, потому что можно войти в систему, запустить *screen* и выполнять все необходимые действия. Если подключение SSH будет прервано, можно переподключиться и выполнить команду *screen -r*. Это даже не обязательно делать с того же самого компьютера.

Чтобы завершить сессию *screen*, выполните команду:

```
$ exit
```

Установка и настройка Redmine

Версии пакетов

Установка Redmine может отличаться в зависимости от версии Ruby, Rails и др. Описанная далее установка проверена для следующих версий пакетов.

Пакет	*Команда для проверки версии*	*Версия пакета*
Ruby	ruby --version	1.8.7
Rubygems	gem --version	1.3.7
Rails		3.2.11
Redmine		2.2-stable
MySQL	mysql --version	5.1.66
Apache	apache2ctl -v	2.2.16
Bundler		1.2.3

Установка MySQL

Устанавливаем MySQL Server и Client. Также для установки Redmine понадобится установить dev-пакет mysqlclient:

```
# aptitude install mysql-server mysql-client libmysqlclient-dev
```

Улучшаем безопасность MySQL Server.

- Устанавливаем пароль для root аккаунта.
- Удаляем root аккаунты, которые доступны вне локального хоста.
- Удаляем анонимные пользовательские аккаунты.
- Удаляем тестовую базу данных (которая по умолчанию может быть доступна любому пользователю, в том числе и анонимному) и права, которые позволяют любому пользователю получить доступ к базе данных с именем, начинающимся с test_.

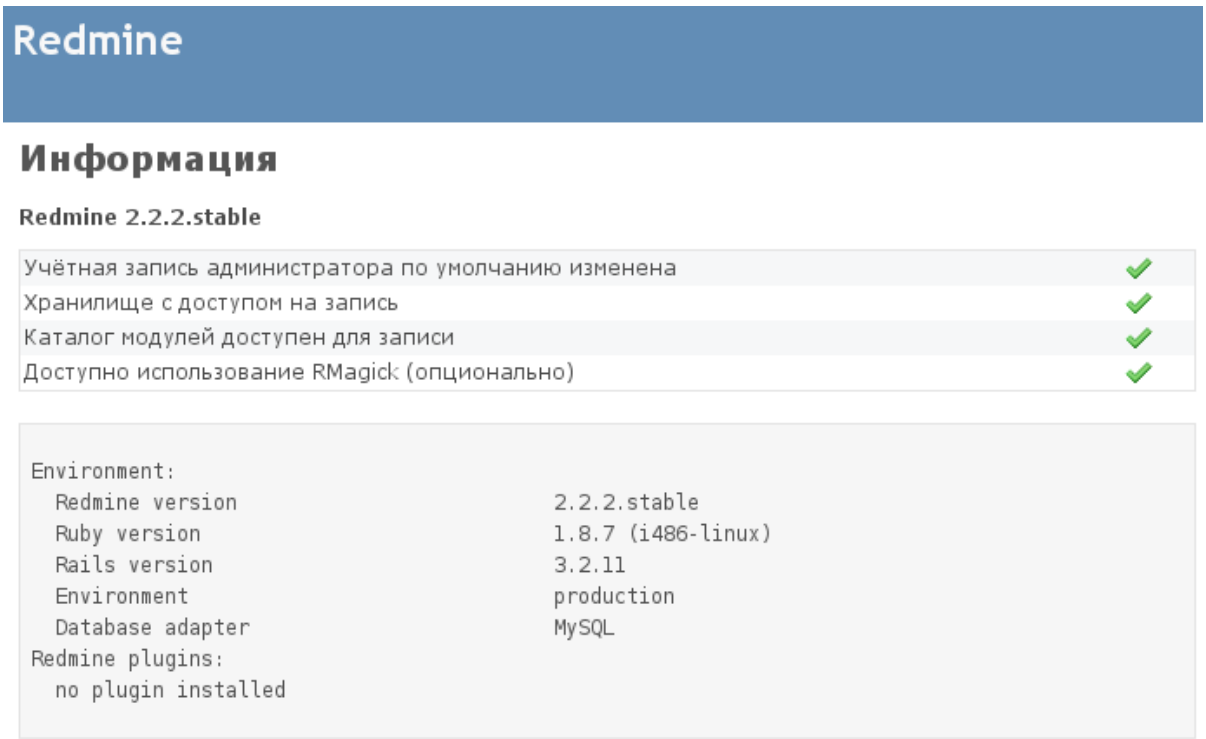
```
# mysql_secure_installation
На все вопросы отвечаем ‘Y’.
```

Создаем пользователя и базу данных для Redmine. Вместо my_password задаем пароль для пользователя redmine:

```
$ mysql -u root -p
mysql> CREATE DATABASE redmine CHARACTER SET utf8;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE USER 'redmine'@'localhost' IDENTIFIED BY 'my_password';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL privileges ON redmine.* TO 'redmine'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> quit
```

Установка Redmine

Теперь по адресу http://[your domain or ip]:3000/ доступен Redmine. Заходим используя
login: admin
pass: admin
Меняем сведения об учетной записи admin (Администрирование -> Пользователи -> admin). Меняем логин, имя, фамилию, e-mail и пароль.
На вкладке Администрирование ->Информация проверяем, что установка прошла успешно.



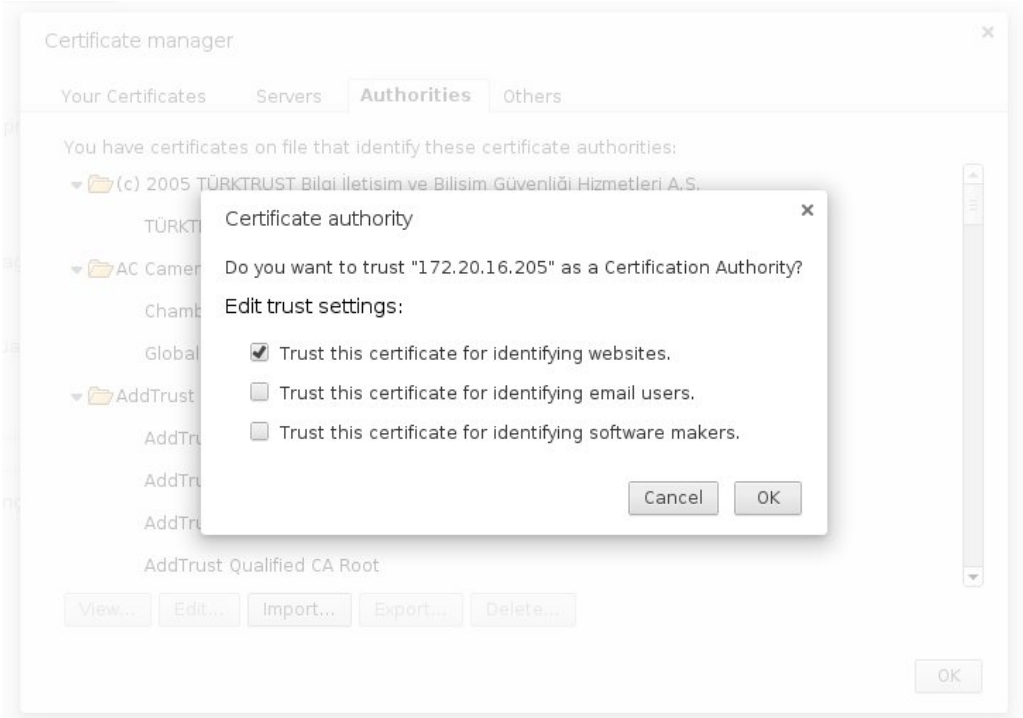
Настройка хоста Apache 2 для работы по https

Для того, чтобы организовать шифрованную передачу по протоколу SSL требуется специальный сертификат на сервере. Его подтверждают центры по сертификации (около $60-70 в год). Но можно сгенерировать сертификат и самостоятельно — такой сертификат называется самоподписным, потому что никакой центр аттестации его не подтверждает, а подтверждаете лично вы. Этим мы и займемся.

Добавление самоподписного сертификата в Chrome

При использовании самоподписного сертификата в браузере его необходимо добавить в исключения. В Firefox это делается просто со страницы блокировки при первом обращении к хосту. В **Google Chrome** придется немного повозиться. Чтобы добавить сертификат в Chrome, необходимо выполнить следующие действия:

1. Щелкаем по крестику рядом с https
2. Certificate Information -> Details -> Export...->Любой формат сохранения
3. Заходим в настройки Chrome: Settings->HTTPS/SSL->Manage Certificates...->Authorities->Import...
4. Перезагружаем браузер для принятия изменений



Настройка Redmine

Установка тем оформления

Список тем доступен по адресу
http://www.redmine.org/projects/redmine/wiki/Theme_List
Темы устанавливаются в папку public/themes и могут быть выбраны в панели

администратора. Пример установки темы A1 на сервере. Установите пакет unzip, если он не установлен:

```
# aptitude install unzip
# cd /var/www/redmine/public/themes/
# wget http://redminecrm.com/license_manager/3916/a1-1_1_0.zip
# unzip a1-1_1_0.zip
# rm a1-1_1_0.zip
```

Создание нового репозитория Mercurial и соединение его с Redmine

Внимание! Имя папки репозитория должно соответствовать **идентификатору проекта**. В противном случае репозиторий будет недоступен. Это происходит из-за того, что модуль аутентификации на Perl ищет репозиторий по идентификатору проекта и никак иначе. Для подключения дополнительных репозиториев к проекту можно использовать названия папки репозитория вида *<идентификатор проекта>.<имя репозитория>*.

Данные из локального репозитория Mercurial будут заноситься в базу данных Redmine при просмотре Хранилища в Redmine. Как автоматизировать обновление данных Redmine, при каждом push в репозиторий Mercurial, будет описано дальше.

Теперь репозиторий Mercurial будет доступен по адресу https://redmine.lan/hg/test, если используется SSL, или по адресу http://redmine.lan/hg/test, если используется не шифрованное соединение.

Особенности настроек проекта Redmine для доступа к хранилищу

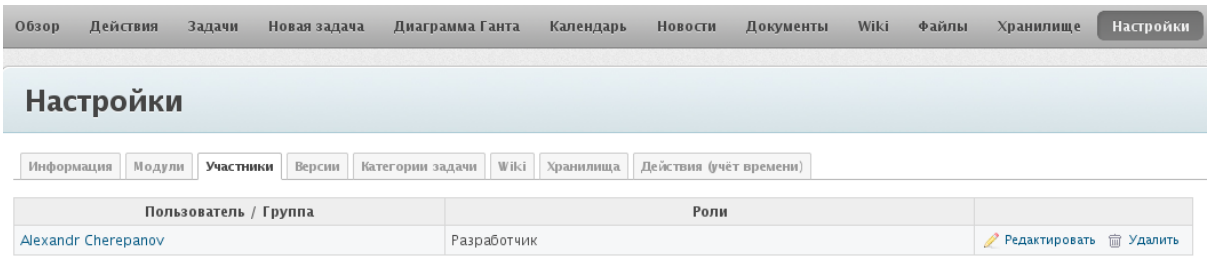
Доступ к хранилищу, привязанному к проекту, определяется на основе нескольких критериев:

1. Проект не общедоступный

Для доступа к репозиторию пользователь должен быть добавлен в проект в качестве участника, иначе при любой попытке доступа к репозиторию будет появляться ошибка HTTP Error: 500 (Internal Server Error). Далее доступ определяется на основе прав пользователя в этом проекте. Роли и права доступа настраиваются через Администрирование -> Роли права доступа (блок Хранилище):

- Для чтения репозитория необходимы права: **Просмотр хранилища** и **Просмотр изменений хранилища**
- Для возможности push своих изменений в репозиторий необходимо право: **Изменение файлов в хранилище**. Иначе при push появляется ошибка HTTP Error: 500 (Internal Server Error).

Пользователь добавляется в проект на вкладке Настройки->Участники:



2. Проект общедоступный

Если проект общедоступный, то любой пользователь Redmine (не участник проекта) может клонировать репозиторий, но для push необходимо быть участником проекта. Права для участников определяются также как для не общедоступного проекта.