

# Daniel Mühlbacher's Blog

*A blog about my projects*

- [Home](#)
- [Port Tools](#)
- [About](#)
- 
- 
- 
- 
- 
- 

## How-To: Install a private Debian git server using gitolite and GitLabHQ

[Jan8th](#)

[2012](#) [Leave a Comment](#) Written by [Daniel Mühlbacher](#)

[Git](#) is currently the most known and used subversion control system.

If you think about using a subversion system for your projects I would definitely recommend using git.

Furthermore, [GitLabHQ](#) is a great web interface for a git server with user authentication just like GitHub with one exception: it is private and no git repository can be accessed publicly.

This guide shows you how you can **install your own private git server** with gitolite on a Debian system.

*This has been tested with Debian squeeze and Ubuntu 10.04 as well as Ubuntu 11.04.*

First of all, you may want to know **what [gitolite](#) is**. Hence, it lets you set up many git repositories using one user account and provide access to many developers without given them real user ids on the system.

;

So, now let us come to the interesting part – the **gitolite installation**.

**Note:** I will perform everything as user root. So just login using your normal user account and do su to get into the root account.

Create git user account and the SSH keys.

?

```

1 adduser git          # this will be our git repository admin
2 account
3 su - git            # login as this user
4 ssh-keygen -t rsa   # create SSH keys for this account - no
  password!
```

Now we need to save the public key because we need it later for our gitolite installation:

?

```
1 cp .ssh/id_rsa.pub git.pub
```

Install git on the server:

?

```

1 # either do:
2   exit          # get into the root account again
3   apt-get install git-core      # install git
4   su - git      # login as the git user again
5 # or (even easier):
6   sudo apt-get install git-core # it may be possible that a our
  git user is not allowed to perform sudo!
```

Install gitolite:

?

```

01 git clone git://github.com/sitaramc/gitolite gitolite-
02 source          # download current gitolite source
03 cd gitolite-source
04 mkdir -p ~/bin ~/share/gitolite/conf ~/share/gitolite/hooks
05 # these directories are needed for the installation
06 src/gl-system-install ~/bin ~/share/gitolite/conf
07 ~/share/gitolite/hooks # install gitolite
08 cd ~
09
```

```

10exit
11chown git:git -R /home/git                # this makes
12sure that really everything in the home directory is owned by git
  su - git
  gl-setup
  git.pub
  # setup gitolite with our public key
    # During this step you need to hit enter and then a file opens
  with vi.
    # Here make sure that you change the value of $REPO_UMASK to
  0007!
    # Then hit :wq.

```

**Note:** It may be possible that your bash can't find the gl-setup command. Then just add the following line at the end of .profile:

```

?
1PATH="$HOME/bin:$PATH"

```

Now remove the public key because we don't need it anymore:

```

?
1rm git.pub

```

Finally, we check if our gitolite setup works:

```

?
1git clone git@localhost:gitolite-admin.git
2# if this works without any problems do
3rm -rf gitolite-admin

```

If this does not work please google for the error message. Sometimes the SSH key produces an error...

```
;
```

Now, we can go on with the **installation of GitLabHQ**.

Hence, GitLabHQ is a nice web interface for git repositories, which uses ruby on rails.

At first, we install some basic packages.

**Note:** if some of these packages are not found you can search for them with apt-cache search. For example if you want to find everything that contains python dev just type in apt-cache search python dev and look for the needed package.

[?](#)

```
apt-get install make gcc libxml2 libxml2-dev libxslt1.1 libxslt1-
1 dev python-dev wget binutils libsqlite3-dev libreadline6-dev
libncurses5-dev libgdbm3 libdb4.8-dev libicu-dev libicu44 apache2
-mpm-prefork curl
```

As the current ruby version does not fulfill the requirements of GitLabHQ we need to install it from scratch:

[?](#)

```
# please note! use sudo here although we are executing this as
root!!!
1 sudo bash -s stable < ; <;(curl -sk
2 https://raw.githubusercontent.com/wayneeseguin/rvm/master/binscripts/rvm-
3 installer)
4 source /etc/profile.d/rvm.sh
5 rvm install ruby-1.9.2
rvm use ruby-1.9.2 --default
```

Now, this will take some time but afterwards we can proceed with rails and the passenger module we need for Apache.

[?](#)

```
1gem update --system
2gem install rails
3gem install passenger
```

After the installation has been finished - if some errors occurred you mostly need to install some additional packages - we can go on with installing the passenger module.

[?](#)

```
1passenger-install-apache2-module
```

The problem here is that passenger needs some more packages but if you read the steps carefully the installation will give you the required commands. If the installation has been finished there should be an output like this:

[?](#)

```
LoadModule passenger_module /usr/local/rvm/gems/ruby-1.9.2-
1p290/gems/passenger-3.0.11/ext/apache2/mod_passenger.so
2PassengerRoot /usr/local/rvm/gems/ruby-1.9.2-p290/gems/passenger-
33.0.11
PassengerRuby /usr/local/rvm/wrappers/ruby-1.9.2-p290/ruby
```

Copy these lines and paste it into the Apache config files - we will configure the appropriate Apache host later.

OK, it's time for our last requirement: python Pip:

```
?
1curl http://python-distribute.org/distribute_setup.py | python
2easy_install pip
```

Now, the main event comes... we finally install GitLabHQ:

```
?
cd /home/git # we install everything related to git into our
1 user's home directory
2 git clone git://github.com/gitlabhq/gitlabhq.git
3 cd gitlabhq
4 pip install pygments
5 gem install bundler
6 bundle install
```

This needs some time but the problem is that Apache has some problems with that. In fact, we need a work-around for that. After this installation has been done open the Gemfile and in lines where ":git" appears remove everything after the first comma. Just like this:

```
?
1gem "grit", :git =>; "https://github.com/gitlabhq/grit.git"
```

now becomes:

```
?
1gem "grit"
```

Finally, you need to rerun our last command:

```
?
1bundle install
```

Great, we are nearly finished with installing GitLabHQ. One major part ist the database installation:

```
?
1RAILS_ENV=production rake db:setup
2RAILS_ENV=production rake db:seed_fu
```

It may be possible that you get some errors here. If so, check if you have installed 2 different versions of rake. If this is the case remove the older one with:

[?](#)

```
1gem uninstall rake
```

and choose the appropriate version. If you get the errors again just reinstall rake:

[?](#)

```
1gem install rake
```

OK, it's time for our last step: Apache.

Just create a new Apache virtual host file and use this configuration (you can try out other configuration parameters later):

[?](#)

```
1< ;VirtualHost *:80>;
2  DocumentRoot /home/git/gitlabhq/public
3  < ;Directory /home/git/gitlabhq/public/>;
4      AllowOverride all
5      Options -MultiViews
6  < ;/Directory>;
7< ;/VirtualHost>;
```

The problem is that our Apache user can't access all this things. So, we need to fix this before restarting Apache:

[?](#)

```
01chown git:git -R /home/git
02usermod -a -G git www-data          # add our Apache user to the
03git group - in fact, the user can access git's home directory
04chown www-data:www-data /var/www # this is important for the
05following steps!
06su - www-data
07ssh localhost                      # it's just important to say
08yes here. then you can cancel this command with CTRL + c.
09exit
10cp /home/git/.ssh/id_rsa /var/www/.ssh/
11chown www-data:www-data -R /var/www/.ssh
   su - www-data
   ssh git@localhost info           # if everything goes well you
   will get a list of some repos...
```



## Recent Posts

- [Facebook Hacker Cup 2013 – Qualification \(my solutions\)](#)
- [Statistics 2012](#)
- [“Port Tools 2.x” \(iPad\) – Update 2](#)
- [How-To: Install a private Debian git server using gitolite and GitLabHQ](#)
- [Statistics 2011](#)

## Archives

- [January 2013](#) (2)
- [June 2012](#) (1)
- [January 2012](#) (2)
- [August 2011](#) (2)
- [July 2011](#) (2)
- [June 2011](#) (1)

## Links

- [Donate / Support my work](#)
- [NiftySide – Daniel Mühlbacher](#)
- [iCloud calendar URLs](#)
- [FaveCon AddressBook](#)
- [Nagios Status Information \(NSI\)](#)
- [Nagios](#)

## Newsletter

Be up-to-date with new information!

Email Address\*

First Name\*

Last Name

\* = required field

## Navigation

- [Register](#)
- [Log in](#)
- [Entries RSS](#)
- [Comments RSS](#)

- [WordPress.org](http://WordPress.org)

Copyright © 2011-2013 by Daniel Mühlbacher

[EvoLve](#) theme by Theme4Press • Powered by [WordPress](#) [Daniel Mühlbacher's Blog](#)

A blog about my projects