



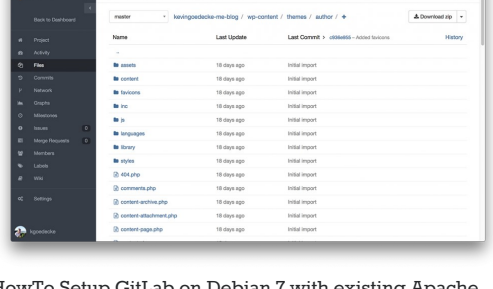
# Setup GitLab on Debian 7 with existing Apache WebServer

Published by [kgoedecke](#) on September 17, 2015

For the last couple of years I've been using [gitolite](#), with gitweb to host my private git repositories. I guess everybody that used gitolite for a while will agree that its by far not as user friendly as using [GitLab](#) is. The web interface looks like its been designed decades ago and on top its missing several major features like code review, merge requests, issue tracking and much more.

Two days ago I came across [GitLab](#), a complete git repository management suite written in Ruby. The community edition is completely free and totally fine for anything less than 100 users. The web interface looks sick, pure eye-candy 🍬

The only problem was, by default GitLab uses an Nginx web server, but I wanted to use it with my existing Apache WebServer, so here's a little HowTo. The documentation of GitLab is quite good and very helpful, never the less I decided to write this tutorial.



## HowTo Setup GitLab on Debian 7 with existing Apache Server

### Step 1: Installing GitLab

Install required package dependencies:

```
kgoedecke@my-vps:~$ sudo apt-get install curl openssl-server ca-certificates postfix
```

Add the GitLab package servers to your servers sources. The guys from GitLab provide a simple script that does the work for you. Simply get it via curl and execute it.

```
kgoedecke@my-vps:~$ curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-deb/scripts/get-deb.sh | sudo bash
```

Afterwards you can use `apt-get` to simply install GitLab:

```
kgoedecke@my-vps:~$ sudo apt-get install gitlab-ce
```

Now run the configuration and start GitLab:

```
kgoedecke@my-vps:~$ sudo gitlab-ctl reconfigure
```

GitLab is now installed and running. All we have to do now, is let our Apache know how to access it.

### Step 2: Edit GitLab Config File

Open the GitLab config file with nano in `/etc/gitlab/gitlab.rb`

```
kgoedecke@my-vps:~$ sudo nano /etc/gitlab/gitlab.rb
```

Set the external URL to the URL you want your GitLab to be reached with:

```
external_url 'http://gitlab.example.com'
```

Also find the following two commands, uncomment them and set them to the following values:

```
nginx['enable'] = false
web_server['external_users'] = ['www-data']
```

Now save the file. Thats pretty much all you need to configure on the GitLab side.

### Step 3: Add Apache user to GitLab user group

In order to add the Apache user to the GitLab user group run the following command:

```
kgoedecke@my-vps:~$ sudo useradd -G gitlab-www www-data
```

### Step 4: Configure Apache to use GitLab

#### 4.1 Create New VirtualHosts file

Go to `/etc/apache2/sites-available/` and create a new VirtualHost.

```
kgoedecke@my-vps:~$ cd /etc/apache2/sites-available/
kgoedecke@my-vps:~$ sudo touch gitlab
kgoedecke@my-vps:~$ sudo nano gitlab
```

Insert the following code (source: <http://stackoverflow.com/questions/25785903/gitlab-7-2-1-with-apache-server-instead-of-nginx>)

```
<VirtualHost *:80>
  ServerName gitlab.example.com
  ServerSignature Off

  ProxyPreserveHost On

  <Location />
    Order deny,allow
    Allow from all

    ProxyPassReverse http://127.0.0.1:8080
    ProxyPassReverse http://gitlab.example.com/
  </Location>

  RewriteEngine On
  RewriteCond %{DOCUMENT_ROOT}/%{REQUEST_FILENAME} !-f
  RewriteRule .* "http://127.0.0.1:8080/%{REQUEST_URI}" [P,QSA]

  # needed for downloading attachments
  DocumentRoot /opt/gitlab/embedded/service/gitlab-rails/public
</VirtualHost>
```

and replace `http://gitlab.example.com/` with the desired domain/subdomain you want your GitLab to be reached by. This configuration tells Apache to redirects all requests to 127.0.0.1:8080, on which the  [Unicorn](#)  server of GitLab is listening to.

Close this file with CTRL + X and Press "Y" to save.

#### 4.2 Enable new VirtualHost

To enable the virtualhosts file run the following command:

```
kgoedecke@my-vps:~$ sudo a2ensite gitlab
```

#### 4.3 Enable required Apache mods

Enable all necessary apache mods:

```
kgoedecke@my-vps:~$ sudo a2enmod proxy
kgoedecke@my-vps:~$ sudo a2enmod proxy_http
```

Restart your Apache.

```
kgoedecke@my-vps:~$ sudo service apache2 restart
```

Now you should be able to access your GitLab web interface.

The default login credentials are:

Username: root  
Password: 5ive!ife

Published in [Uncategorized](#)

- [Apache](#)
- [Debian](#)
- [Git](#)
- [GitLab](#)
- [Nginx](#)
- [Private Git](#)

[Previous Post](#) [Next Post](#)  
[Backup Mac Photos Library with Rsync over SSH](#) [Using LoopBack IOS SDK with CocosPods in Swift](#)

## 8 Comments

**Asanka**

Thanks bro! This helped to find the key things hidden in that massive config file 🍬

March 29, 2016 | [Reply](#)

**Dejv**

Thanks!

March 31, 2016 | [Reply](#)

**Eddz**

Thx for this tuto.

You can also advice to run: "a2enmod rewrite"

April 21, 2016 | [Reply](#)

**Mike Street**

Thanks for the great tutorial - helped me get the last 1%!

A couple of changes I would suggest:

- On my VPS, I had to install apache to be able to get to the `/etc/apache2/sites-available/` folder
- In that folder, the file needs to be `gitlab.conf`, not just `gitlab`
- I had to enable the rewrite module
- `a2enmod rewrite`

June 17, 2016 | [Reply](#)

**Anonymous**

thank you man!

June 29, 2016 | [Reply](#)

**Simon D**

Recipe Compile Error in  
`/opt/gitlab/embedded/cookbooks/cache/cookbooks/gitlab/recipes/default.rb`

Error executing action 'run' on resource 'execute[clear the gitlab-rails cache]'

Why am I getting those errors? :S 🍬 I have followed your nice tutorial from top to bottom 🍬

August 14, 2016 | [Reply](#)

**Anonymous**

Thanks

August 14, 2016 | [Reply](#)

**Anonymous**

How would you setup ssl using this method?

September 23, 2016 | [Reply](#)

## Leave a Reply

Your email address will not be published.

Comment

Name optional

Email optional

Website