Gitit

Christian Külker

2024-07-14

Contents

1 Introduction

1.1 Introduction to Gitit Usage and Configuration

Gitit (darcsit) is a wiki system implemented in Haskell, utilizing git among other version control systems as its storage back-end. This document delineates the procedures and configurations necessary for installing, customizing, and managing a Gitit instance. The guide covers installation steps for Gitit on various Debian releases, emphasizing the necessity of source code installation on older versions for enhanced security. Detailed commands for setting up a new Gitit site, including initialization commands and configuration adjustments, are presented to facilitate local setup and usage.

The guide also discusses the deployment of themes and hints to the use of Docker containers as alternatives for Gitit installation, although the Docker method is not verified. For extending Gitit's functionality, methods for enabling PDF exports and syntax highlighting are outlined, demonstrating the configuration adjustments required in the Gitit setup files.

Version control integration is a significant aspect covered, highlighting how Gitit can manage changes via local and remote repositories. The guide suggests solutions for common issues such as merge conflicts and update visibility, which are inherent to collaborative environments using version control.

In summary, this guide serves as a quick approach for users seeking to deploy and maintain Gitit as a personal or collaborative wiki.

It is very easy to install and use.

- Source: https://github.com/jgm/gitit
- Wikipedia: https://en.wikipedia.org/wiki/Gitit_(software)

2 Installation

```
aptitude install gitit
```

3 New Site

```
mkdir ~/scratch/gitit-new-site
cd ~/scratch/gitit-new-site
git init
Initialized empty Git repository in /home/c/scratch/gitit-new-site/.git/
gitit --print-default-config > config.cfg
vim config.cfg
    address: 0.0.0.0
    address: 127.0.0.1
    pdf-export: no
    pdf-export: yes
gitit --debug -p 8080 -f gitit.cfg
Created repository in wikidata
Added Front Page.page to repository
Added Help.page to repository
Added Gitit User's Guide.page to repository
Created static/css/custom.css
Created static/img/logo.png
Created templates/footer.st
```

Point the browser to http://127.0.0.1:8080

4 Themes

- Wiki: https://github.com/jgm/gitit/wiki/Themes

4.1 Install Theme

- Github: https://github.com/t13a/gitit-bootstrap-theme

```
cd path/to/your-gitit-dir
rm -rf cache static templates
cd /tmp
git clone https://github.com/t13a/gitit-bootstrap-theme
gitit-bootstrap-theme/build.sh
...
cp -r gitit-bootstrap-theme/out/{static,templates} path/to/your-gitit-dir
```

Point browser to http://localhost.

5 URLs

- Home: http://gitit.net
- Github: https://github.com/jgm/gitit
- bootstrap theme: https://github.com/t13a/gitit-bootstrap-theme
- docker support: https://github.com/t13a/dockerfile-gitit-experimental
- Wikipedia: https://en.wikipedia.org/wiki/Gitit_(software)
- Demo: http://gitit.net/
- Clone: https://github.com/jgm/gitit.git
- Hacks: http://wiki.wcaleb.rice.edu/Gitit%20Hacks
- Systemd Start Script: https://floatingoctothorpe.uk/2017/setting-up-a-wiki-with-gitit.html
- User Guide: http://i40.iosb.fraunhofer.de/Gitit%20User%E2%80%99s%20Guide

6 Install Gitit on Debian Wheezy 7.8 {0.1.0}

The Debian release of Gitit is 0.10.0.1 from 07 Jun 2012 (consider source code install for security)

```
aptitude install gitit
```

7 Install Gitit on Debian Squeeze 9.9 {0.1.1}

The Debian release for Gitit is 0.12.1.1.

```
    aptitude install gitit
```

8 Install Gitit on Debian Buster 10.0 {0.1.1}

The Debian release for Gitit is 0.12.3.1.

```
aptitude install gitit
```

9 Start and use Gitit {0.1.0}

To see the default configuration (to understand the port number of the build in web server) use this line

```
gitit --print-default-config > config
```

You will then understand that it will use port 5001

Then just start it (that is all):

```
mkdir gitit
cd gitit
gitit
```

It will create the file structure and print some lines of information (not the port number!)

```
Created repository in wikidata
Added Front Page.page to repository
Added Help.page to repository
Added Gitit User's Guide.page to repository
Created static/css/custom.css
Created static/img/logo.png
Created templates/footer.st
```

Point your browser to: http://127.0.0.1:5001

Create yourself an account

```
Name:
password
```

No mail address needed. Make sure you use a better password. That’s it!

10 PDF Support {0.1.0}

Export the default configuration

```
gitit --print-default-config > config
```

Already in the older version for Debian Wheezy PDF support is implemented. (It probably need PDFLaTeX installed - maybe XeLaTeX in later versions). It can be easily enabled in the configuration.

```
vim config

pdf-export: yes
```

Then start gitit with:

```
gitit -f config
```

11 Deploy Gitit to the Real Life {0.1.0}

By default everyone can create an account in Gitit. That is good, because Gitit can be uses as personalized wiki on once own laptop for example. However an installation on the internet might need some kind of restriction. Two restrictions can be easily enabled in Gitit: 1. a passphrase and 2. basic Authentication via Apache. The latter is not covered here.

To enable the pass phrase, edit the configuration

```
access-question: What is my favorite water?
access-question-answers: wet, Wet, WET
```

12 Syntax Highlighting {0.1.0}

```
aptitude install libghc-highlighting-kate-dev
```

Then test on a page

~~~
````{#example .bash .numberLines startFrom="100"}
~~~

~~~
#!/bin/bash
if [ -f '/bin/bash' ]; then
    echo "found bash"
fi
````
~~~

The word ‘example’ will get in HTML: id="example".

13 Gitit and Git {0.1.0}

There are at least 2 ways to use a git repository as storage back-end for Gitit:

1. Let Gitit create a repository (difficult for external access), see sub section “Share Gitit Git Repository”
2. Use a remote repository (difficult to keep in sync)

13.1 Share Gitit Git Repository

To add a new host to Gitit: add the public key to the `authorized_keys` file in the home directory of the Gitit user, for example in `/srv/gitit/.ssh` or `/home/gitit.ssh`.

Then you can clone:

```
mkdir -p ~/g/searn.pielots.de
cd ~g/searn.pielots.de
git clone gitit@searn.pielots.de:searn/wikidata
```

13.2 Use A Remote Repository

After the initial run of Gitit, a git repository is created. Delete this. Then create a new repository via `gitolite` for example and clone this repository at the location of the old one, most likely in `wikidata`. Then add all files to the new repository.

The advantage is that it s easy to manage a repository via gitolite for more than one user. However Gitit will not automatically push changes from its local copy to the remote branch. This can lead to race conditions.

One way to solve it would be to cron job execute one per minute to always push changes. However that can lead to merge conflicts that would not be seen by the author of the wiki, nor by the author who uses the repository on the command line remotely.

A solution is to add a pre and post commit hook:

```
echo "#!/bin/sh" > /srv/w/d/japan.tour.recipes/wikidata/.ssh/hooks/pre-commit
echo "#!/bin/sh" > /srv/w/d/japan.tour.recipes/wikidata/.ssh/hooks/post-commit
echo "git pull" >> /srv/w/d/japan.tour.recipes/wikidata/.ssh/hooks/pre-commit
echo "git push" >> /srv/w/d/japan.tour.recipes/wikidata/.ssh/hooks/post-commit
chmod +x /srv/w/d/japan.tour.recipes/wikidata/.ssh/hooks/pre-commit
chmod +x /srv/w/d/japan.tour.recipes/wikidata/.ssh/hooks/post-commit
```

14 Install Gitit via Docker {0.1.4} (Not tested)

- tomzo/gitit

15 Install Gitit from git {0.1.1}

There are several methods: `stack` and `cabal`

15.1 Stack build, test install {0.1.1}

The `stack` build, test install do NOT work.

```
git clone https://github.com/jgm/gitit.git
cd gitit
aptitude install haskell-stack
stack install
Downloaded lts-12.4 build plan.
AesonException "Error in $.packages.cassava.constraints.flags: failed to parse
field packages: failed to parse field constraints: failed to parse field flags:
Invalid flag name: \"bytestring--lt-0_10_4\""
```

15.2 Cabal Install

This will download and build a lot of stuff in ~/.cabal (1.2 GB) including `pandoc`! This either ignores the installed `pandoc` version or decides it wants a newer one. The `build` and `test` actions claim to work, but if one compares with the time and amount of stuff done during the `install` action, it is clear that the build (at least of the dependencies) are done during `install`.

- Debian `pandoc` 1.17.2
- Cabal `pandoc` 2.2.1

(see `pandoc` chapter for difference in syntax highlighting)

```
aptitude install cabal-install
git clone https://github.com/jgm/gitit.git
cd gitit
cabal build
cabal test
cabal update
cabal install
```

Test it:

```
export PATH=$HOME/.cabal/bin:$PATH
which gitit
$HOME/.cabal/bin/gitit
gitit --version
gitit version 0.12.3.1 -plugins
pandoc --version
pandoc 2.2.1
which pandoc
$HOME/.cabal/bin/pandoc
```

16 Pandoc Installed from Cabal {0.1.1}

This is about the difference of Debian 9 squeeze installed `pandoc` versus `cabal` installed `pandoc`:

Debian 9.9 squeeze:

```
/usr/bin/pandoc --version
pandoc 1.17.2
Compiled with texmath 0.8.6.7, highlighting-kate 0.6.3.
Syntax highlighting is supported for the following languages:
    abc, actionscript, ada, agda, apache, asn1, asp, awk, bash, bibtex, boo, c,
    changelog, clojure, cmake, coffee, coldfusion, commonlisp, cpp, cs, css,
    curry, d, diff, djangotemplate, dockerfile, dot, doxygen, doxygenlua, dtd,
    eiffel, elixir, email, erlang, fasm, fortran, fsharp, gcc, glsl,
    gnuassembler, go, hamlet, haskell, haxe, html, idris, ini, isocpp, java,
    javadoc, javascript, json, jsp, julia, kotlin, latex, lex, lilypond,
    literatecurry, literatehaskell, llvm, lua, m4, makefile, mandoc, markdown,
    mathematica, matlab, maxima, mediawiki, metafont, mips, modelines, modula2,
    modula3, monobasic, nasm, noweb, objectivec, objectivecpp, ocaml, octave,
    opencl, pascal, perl, php, pike, postscript, prolog, pure, python, r,
    relaxng, relaxngcompact, rest, rhtml, roff, ruby, rust, scala, scheme, sci,
    sed, sgml, sql, sqlmysql, sqlpostgresql, tcl, tcsh, texinfo, verilog, vhdl,
    xml, xorg, xslt, xul, yacc, yaml, zsh
Default user data directory: $HOME/.pandoc
Copyright (C) 2006-2016 John MacFarlane
Web:  http://pandoc.org
This is free software; see the source for copying conditions.
There is no warranty, not even for merchantability or fitness
```

Cabal:

```
~/.cabal/bin/pandoc --version
pandoc 2.2.1
Compiled with pandoc-types 1.17.5.4, texmath 0.11.2.2, skylighting 0.7.7
Default user data directory: $HOME/.pandoc
Copyright (C) 2006-2018 John MacFarlane
Web:  http://pandoc.org
This is free software; see the source for copying conditions.
There is no warranty, not even for merchantability or fitness
for a particular purpose.
```

Debian 10 Buster:

```
pandoc --version
pandoc 2.2.1
Compiled with pandoc-types 1.17.5.1, texmath 0.11.1, skylighting 0.7.5
Default user data directory: /home/c/.pandoc
Copyright (C) 2006-2018 John MacFarlane
Web:  http://pandoc.org
This is free software; see the source for copying conditions.
There is no warranty, not even for merchantability or fitness
for a particular purpose.
```

17 Common Problems

17.1 Wikdata Update Invisible

It is possible to change the back end via git and via web front end. If the wiki git repository is the master repository the wiki will be updated automatically. However, if the repository is inside a gitolite repository the git repository of gitit and the user are just clones of the master repository. Making a change to the repository will **not** be visible via the wiki.

17.2 Merge conflicts

Unlike other wikis, like MoinMoin, there is no edit warning if one user edits a file via wiki and an other user is editing the same file via git. The first commit wins and the other will be left alone with a merge conflict. In case of the wiki front end, there is no one to solve this. Maybe this can be somewhat mitigated with a pre-commit hook that updates the repository, so that the commit will overwrite the new changed file. This needs a prove of concept.

17.3 Template Updates Invisible

Usually the gitit templates are not part of the wikidata git repository. Changes are needed to add for example navigation bars. However, even if the `*.ts` files can be edited via git, changes to a file will not be visible by the wiki. For the wiki to see the changed template, it needs to restart. Here a post-commit hook could be used, to restart the wiki. However, this is difficult, as the commit is a user commit, but restarting the Gitit wiki needs a different user.

18 History

Version	Date	Notes
0.1.6	2024-07-14	Formatting, typos
0.1.5	2020-09-10	Add 3 problems of using Gitit
0.1.4	2020-09-02	Change version description
0.1.3	2020-08-20	Gitit installation Debian 10 Buster
0.1.2	2019-12-08	Overhaul
0.1.1	2019-07-26	Gitit installation from git, Debian Squeeze
0.1.0	2019-07-25	Initial release

19 Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOCUMENT OR THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

20 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.