

sitaramc / gitolite

<> Code

Pull requests 3

Actions

Projects

Wiki

Security

Insights

Hosting git repositories -- Gitolite allows you to setup git hosting on a central server, with very fine-grained access control and many (many!) more powerful features.

[github.com/sitaramc/gitolite/wiki](#)

GPL-2.0 license

8.3k stars 1k forks 382 watching 4 Branches 73 Tags Activity

Public repository

master 4 Branches 73 Tags

Go to file

Go to file + Add file Code ...

sitaramc	this file should have been git-ignored and never committed at all	2 months ago
contrib	fixed up several broken URLs (minor bu...	6 months ago
src	fixed up several broken URLs (minor bu...	6 months ago
t	fixed up several broken URLs (minor bu...	6 months ago
CHANGELOG	v3.6.13	6 months ago
CONTRIBUTING	fixed up several broken URLs (minor bu...	6 months ago
COPYING	add COPYING file	12 years ago
ISSUE_TEMPLATE	Add support for Github's new TEMPLATE ...	8 years ago
PULL_REQUEST_TEMPLATE	Add support for Github's new TEMPLATE ...	8 years ago
README.markdown	minor URL fixup	6 years ago
check-g2-compat	(minor) change urls to fit new domain ...	11 years ago
convert-gitosis-conf	Properly migrate [gitosis] section	12 years ago
install	Install script can now modify shebangs...	4 years ago

README

GPL-2.0 license

Gitolite README

about this README

(Github-users: click the "wiki" link before sending me anything via github.)

This is a minimal README for gitolite, so you can quickly get started with:

- installing gitolite on a fresh userid on a Unix(-like) machine, using ssh
- learning enough to do some basic access control

For anything more, you need to look at the complete documentation, at: <http://gitolite.com/gitolite>. Please go there for what/why/how, concepts, background, troubleshooting, more details on what is covered here, advanced features not covered here, migration from older gitolite, running gitolite over http (rather than ssh), and many more topics.

Assumptions

- You are familiar with:
 - OS: at least one Unix-like OS
 - ssh: ssh, ssh keys, ssh authorized keys file
 - git: basic use of git, bare and non-bare remotes
- You are setting up a fresh, ssh-based, installation of gitolite on a Unix machine of some sort.
- You have root access, or someone has created a userid called "git" for you to use and given you a password for it. This is a brand new userid (or you have deleted everything but `.bashrc` and similar files to make it look like one!)
- If your server is not connected to the internet, you know how to clone the gitolite source code by using some in-between server or "git bundle".

Installation and setup

server requirements

- any unix system
- sh
- git 1.6.6 or later
- perl 5.8.8 or later
- openssh 5.0 or later
- a dedicated userid to host the repos (in this document, we assume it is "git", but it can be anything; substitute accordingly)
- this user id does NOT currently have any ssh pubkey-based access
 - ideally, this user id has shell access ONLY by "su - git" from some other userid on the same server (this ensure minimal confusion for ssh newbies!)

steps to install

First, prepare the ssh key:

- login to "git" on the server
- make sure `~/.ssh/authorized_keys` is empty or non-existent
- make sure your ssh public key from your workstation has been copied as `$HOME/YourName.pub`

Next, install gitolite by running these commands:

```
git clone https://github.com/sitaramc/gitolite
mkdir -p $HOME/bin
gitolite/install -to $HOME/bin
```



Finally, setup gitolite with yourself as the administrator:

```
gitolite setup -pk YourName.pub
```



If the last command doesn't run perhaps "bin" is not in your "PATH". You can either add it, or just run:

```
$HOME/bin/gitolite setup -pk YourName.pub
```



If you get any other errors please refer to the online documentation whose URL was given at the top of this file.

adding users and repos

Do NOT add new repos or users manually on the server. Gitolite users, repos, and access rules are maintained by making changes to a special repo called "gitolite-admin" and *pushing* those changes to the server.

To administer your gitolite installation, start by doing this on your workstation (if you have not already done so):

```
git clone git@host:gitolite-admin
```



NOTE: if you are asked for a password, something went wrong.. Go hit the link for the complete documentation earlier in this file.

Now if you "cd gitolite-admin", you will see two subdirectories in it: "conf" and "keydir".

To add new users alice, bob, and carol, obtain their public keys and add them to "keydir" as alice.pub, bob.pub, and carol.pub respectively.

To add a new repo "foo" and give different levels of access to these users, edit the file "conf/gitolite.conf" and add lines like this:

```
repo foo
  RW+    =  alice
  RW     =  bob
  R      =  carol
```



Once you have made these changes, do something like this:

```
git add conf
git add keydir
git commit -m "added foo, gave access to alice, bob, carol"
git push
```



When the push completes, gitolite will add the new users to ~/.ssh/authorized_keys on the server, as well as create a new, empty, repo called "foo".

help for your users

Once a user has sent you their public key and you have added them as specified above and given them access, you have to tell them what URL to access their repos at. This is usually "git clone git@host:reponame"; see man git-clone for other forms.

NOTE: again, if they are asked for a password, something is wrong.

If they need to know what repos they have access to, they just have to run "ssh git@host info".

access rule examples

Gitolite's access rules are very powerful. The simplest use was already shown above. Here is a slightly more detailed example:

```
repo foo
  RW+    =  alice
  -  master    =  bob
  -  refs/tags/v[0-9] =  bob
  RW     =  bob
  RW  refs/tags/v[0-9] =  carol
  R      =  dave
```



Here's what these example rules say:

- alice can do anything to any branch or tag -- create, push, delete, rewind/overwrite etc.
- bob can create or fast-forward push any branch whose name does not start with "master" and create any tag whose name does not start with "v"+digit.
- carol can create tags whose names start with "v"+digit.
- dave can clone/fetch.

Please see the main documentation linked above for all the gory details, as well as more features and examples.

groups

Gitolite allows you to group users or repos for convenience. Here's an example that creates two groups of users:

```
@staff      =  alice bob carol
@interns    =  ashok
```



```
repo secret
  RW      =  @staff

repo foss
  RW+     =  @staff
  RW      =  @interns
```

Group lists accumulate. The following two lines have the same effect as the earlier definition of @staff above:

```
@staff      =  alice bob
@staff      =  carol
```



You can also use group names in other group names:

```
@all-devs   =  @staff @interns
```



Finally, @all is a special group name that is often convenient to use if you really mean "all repos" or "all users".

commands

Users can run certain commands remotely, using ssh. Running

```
ssh git@host help
```



prints a list of available commands.

The most commonly used command is "info". All commands respond to a single argument of "-h" with suitable information.

If you have shell on the server, you have a lot more commands available to you; try running "gitolite help".

LICENSE

contact and support

Please see <http://gitolite.com/gitolite/#contactsupport> for mailing list and IRC info.

license

The gitolite software is copyright Sitaram Chamarty and is licensed under the GPL v2; please see the file called COPYING in the source distribution.

Please see <http://gitolite.com/gitolite/#license> for more.

▪

Releases

73 tags

Packages

No packages published

Contributors 38



+ 24 contributors

Languages

Perl 87.7% Shell 11.4% Vim Script 0.9%