GitLab as an Alternative Development Platform for Github.com

LinuxCon Europe 2014

October 13, 2014

Ralf Lang
Linux Consultant / Developer
B1 Systems GmbH
lang@b1-systems.de

Introducing B1 Systems

- founded in 2004
- operating both nationally and internationally
- more than 60 employees; low employee turnover
- Provider for IBM, SUSE, Oracle & HP
- vendor-independent (hardware and software)
- Focus:
  - Consulting
  - Support
  - Development
  - Training
  - Operations
  - Solutions

Areas of Expertise

- Virtualization (XEN, KVM & RHEV)
- Systems management (Spacewalk, Red Hat Satellite, SUSE Manager)
- Configuration management (Puppet & Chef)
- Monitoring (Nagios & Icinga)
- IaaS Cloud (OpenStack & SUSE Cloud)
- High availability (Pacemaker)
- Shared Storage (GPFS, OCFS2, DRBD & CEPH)
- File Sharing (ownCloud)
- Packaging (Open Build Service)
- Providing on-site systems administration and/or development

Partners

SUSE PartnerNet Platinum CERTIFIED SUSE PARTNER

redhat PREMIER BUSINESS PARTNER

ubuntu Advantage reseller

puppet labs®

pfSense

Nagios® Authorized Reseller

Hortonworks

ARISTA

rackspace® the open cloud company

ownCloud YOUR CLOUD, YOUR DATA, YOUR WAY! Gold Partner

GitLab – An Open Source Software to Collaborate on Code

Why Managed Version Control?

- easy management of privileges
- inline feedback options
- enforce reviews
- spend time coding, not managing tools and users

Why Should I run my own VCS?

- firm control over source code access
- no external parties involved
- dedicated resources
- easily integrates with custom tools and reports
- keep sensitive information in-house

What is GitLab?

- GitLab is an Open source software to collaborate on code.
- GitLab is based on Git, the most widely adopted version control system for software development.
- GitLab helps to ensure software quality by providing a feature-rich review system.
- GitLab simplifies distributed working on projects with a centralized server.

Some GitLab Features

- code review
- bug tracking
- personal and private branches
- management of numerous Git repositories
- 25,000 users on a single server
- highly available active/active cluster possible
- code snippets
- access control
- issue tracking
- Web hooks
- Wiki

GitLab is Collaborative

- unlimited number of public and private repositories
- unlimited number of public and private collaborators
- integrates with *LDAP*
- integrates with external ticket systems e.a. *Redmine*
- Omnibus package supports configuring an external database (*PostgreSQL* or *MySQL*)
- works with *JIRA* for issue tracking
- displays merge request status for builds on *Jenkins CI* (only Enterprise Edition)

Who Else Uses GitLab?

- More than 100,000 organizations, amongst others:
  - AT&T
  - Bell
  - CERN
  - Fraunhofer
  - Interpol
  - NASA
  - Red Hat

Who Works on GitLab?

- since September 2011
- an active community with hundreds of contributors
- managed by GitLab.com
- Enterprise support by GitLab B.V.

Traditional Git Workflow

1. Clone the repository.
2. Create a branch.
3. Modify source code.
4. Check in.
5. Create a patch or push changes to upstream.

Disadvantages & Drawbacks

- Write access:
  Every committer needs write access on projects.
  $\Rightarrow$ intended workflows could be omitted
- Format patch:
  Every committer submits his patches and has to wait for the maintaining of a reviewer.
  $\Rightarrow$ still a labor-intensive and error-prone process

The Gitlab Workflow

1. Fork repository into own name space.
2. Full access to own forked copy.
3. Edit online in browser or in local checkouts.
4. Create Merge Request.
5. Reviewer comments on diffs on the platform.
6. Automated process for pulling forks back into the mainstream repository.

⇒ no need to grant or revoke access
⇒ no hassle with long threads of patch e-mails
⇒ enforces review paradigms
⇒ little setup costs for additional team members

Internal Issue/Review System

- APIs for external ticketing
    - access Redmine tickets through commit messages in Git
    - or use internal ticketing
- API for Gitlab CI
    - continuous Integration: Automated builds and test suite runs on commit
    - improved software quality
    - use dead code detectors or code coverage tools

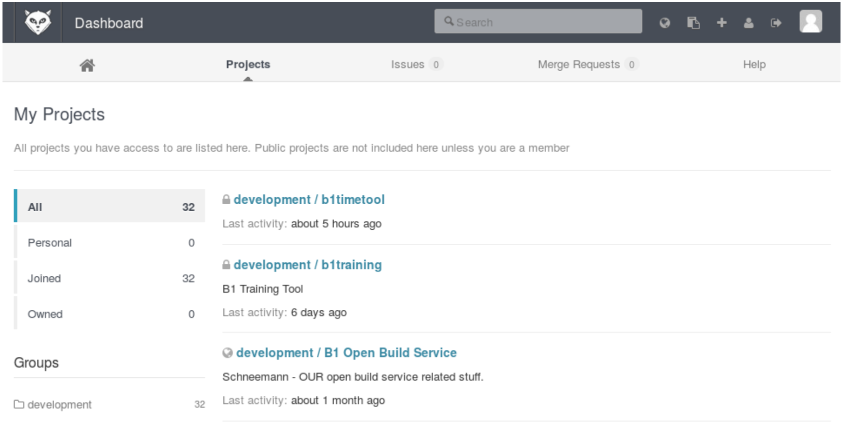
Access Control

- GitLab provides an access control for user and groups based on permission levels.
- Users' abilities depend on their access level on a particular project or group.
- If a user is both in a project group and in the project itself the highest permission level is used.
- The GitLab administrator receives all permissions.

GitLab Continuous Integration

- integrates with the GitLab installation to run tests for projects
- login with GitLab account
- Simply add projects with one click
- on-premises software: can be installed on arbitrary (Linux) server(s)

The GitLab Dashboard – Project Overview


Dashboard

Search

Projects | Issues 0 | Merge Requests 0 | Help

My Projects

All projects you have access to are listed here. Public projects are not included here unless you are a member

All	32
Personal	0
Joined	32
Owned	0

Groups

development 32

development / b1timetool

Last activity: about 5 hours ago

development / b1training

B1 Training Tool

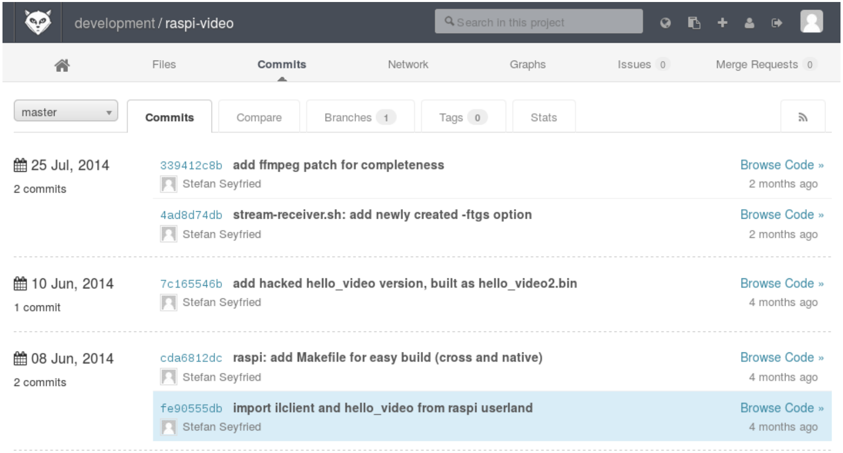
Last activity: 6 days ago

development / B1 Open Build Service

Schneemann - OUR open build service related stuff.

Last activity: about 1 month ago

The GitLab Dashboard – Commits



25 Jul, 2014
2 commits

339412c8b **add ffmpeg patch for completeness**
Stefan Seyfried — Browse Code » — 2 months ago

4ad8d74db **stream-receiver.sh: add newly created -ftgs option**
Stefan Seyfried — Browse Code » — 2 months ago

10 Jun, 2014
1 commit

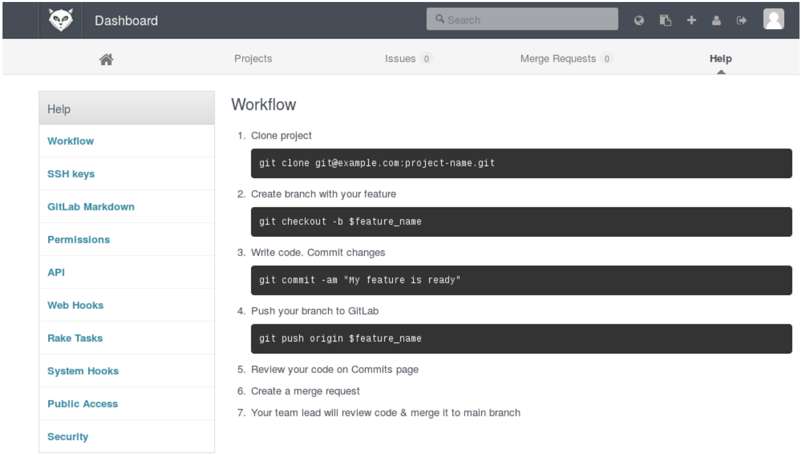
7c165546b **add hacked hello_video version, built as hello_video2.bin**
Stefan Seyfried — Browse Code » — 4 months ago

08 Jun, 2014
2 commits

cda6812dc **raspi: add Makefile for easy build (cross and native)**
Stefan Seyfried — Browse Code » — 4 months ago

fe90555db **import ilclient and hello_video from raspi userland**
Stefan Seyfried — Browse Code » — 4 months ago

The GitLab Dashboard – Workflow



Workflow

1. Clone project

```
git clone git@example.com:project-name.git
```

2. Create branch with your feature

```
git checkout -b $feature_name
```

3. Write code. Commit changes

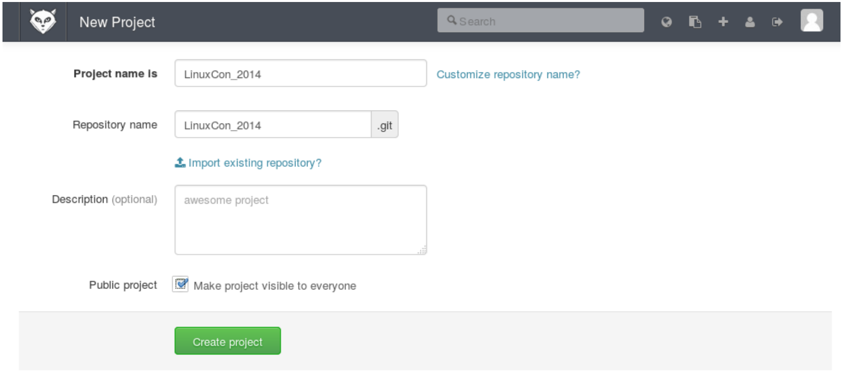
```
git commit -am "My feature is ready"
```

4. Push your branch to GitLab

```
git push origin $feature_name
```

5. Review your code on Commits page
6. Create a merge request
7. Your team lead will review code & merge it to main branch

The GitLab Dashboard – New Project



Project name is LinuxCon_2014 Customize repository name?

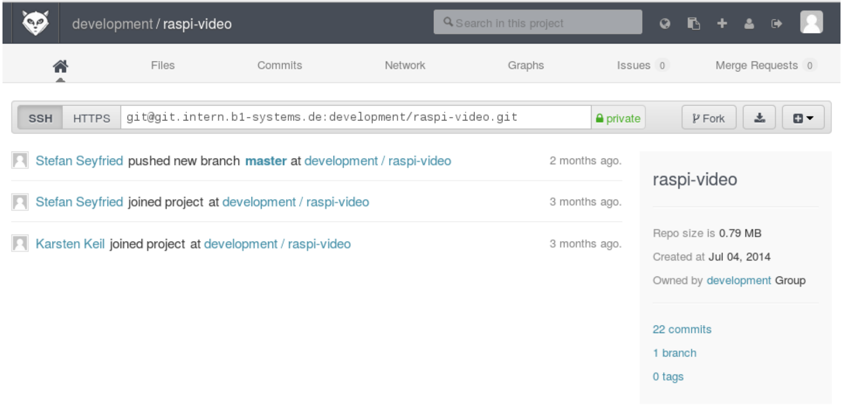
Repository name LinuxCon_2014 .git

Import existing repository?

Description (optional) awesome project

Public project [✓] Make project visible to everyone

The GitLab Dashboard – Projects



development / raspi-video

git@git.intern.b1-systems.de:development/raspi-video.git

Stefan Seyfried pushed new branch **master** at development / raspi-video — 2 months ago.

Stefan Seyfried joined project at development / raspi-video — 3 months ago.

Karsten Keil joined project at development / raspi-video — 3 months ago.

raspi-video

Repo size is 0.79 MB

Created at Jul 04, 2014

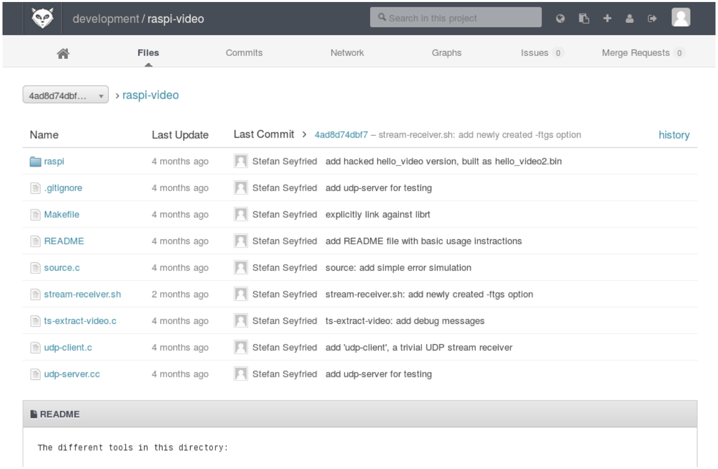
Owned by development Group

22 commits

1 branch

0 tags

The GitLab Dashboard – Project Files



Last Commit › 4ad8d74dbf7 – stream-receiver.sh: add newly created -ftgs option — history

Name	Last Update	Last Commit	
raspi	4 months ago	Stefan Seyfried	add hacked hello_video version, built as hello_video2.bin
.gitignore	4 months ago	Stefan Seyfried	add udp-server for testing
Makefile	4 months ago	Stefan Seyfried	explicitly link against librt
README	4 months ago	Stefan Seyfried	add README file with basic usage instractions
source.c	4 months ago	Stefan Seyfried	source: add simple error simulation
stream-receiver.sh	2 months ago	Stefan Seyfried	stream-receiver.sh: add newly created -ftgs option
ts-extract-video.c	4 months ago	Stefan Seyfried	ts-extract-video: add debug messages
udp-client.c	4 months ago	Stefan Seyfried	add 'udp-client', a trivial UDP stream receiver
udp-server.cc	4 months ago	Stefan Seyfried	add udp-server for testing

README

```
The different tools in this directory:
```

More Information on GitLab . . .

- GitLab.com:
  `GitLab.com`
- GitLab Continuous Integration (CI):
  `https://about.gitlab.com/gitlab-ci/`
- Official GitLab Documentation:
  `http://doc.gitlab.com/ce/`

Thank you for your attention!

For further information, please contact:
info@b1-systems.de or +49 (0)8457 - 931096