



Salsa and Gitlab

cool and not so well known features

Alexander Wirt <formorer@debian.org>

08.06.2019

Overview



Introduction

Introduction

Gitlab

salsa.debian.org

API

CI - Continues Integration

Merge Requests

issue boards

Manipulation of things per commit

Pages

Unsorted goodies

The end - your feedback

The end - your feedback

Who am I



- ▶ Debian Contributor since 1999
- ▶ Debian Developer since 2003
- ▶ Debian Listmaster
- ▶ Debian Backports ftpmaster
- ▶ salsa Admin

And otherwise



- ▶ Open Source Consultant @credativ
- ▶ Father of two wonderful girls
- ▶ RPG and tabletop nerd

We need you!



- ▶ Ruby on Rails based
- ▶ Scalable
- ▶ started as a Github clone
- ▶ Open Core

- ▶ Based on gitlab
- ▶ managed via Ansible
- ▶ 5.200 Users
- ▶ 31.800 Projects occupying 425 GB

Gitlab has a nice API to do things. Use it.
<https://docs.gitlab.com/ee/api/>

About using curl

Please use a library and please use paging

API Token

Start with creating your API token

https://salsa.debian.org/profile/personal_access_tokens

```
$ curl --header  
→ "PRIVATE-TOKEN:$GITLAB_API_PRIVATE_TOKEN"  
→ https://salsa.debian.org/api/v4/$resource
```

Check for a project



Hint: do not call /projects

```
PROJECT=security-tracker-team/security-tracker  
GET https://salsa.debian.org/$project
```

Check for a project



```
GROUP="debian-security"  
NAME="security-tracker"  
GET $API/projects/$GROUP%2F$NAME 2>/dev/null | jq  
  ↪ .description  
"Debian Security Tracker"
```

Subscribe to an issue



```
curl -XPOST $API/projects/5/issues/93/subscribe
```

Do not use curl

Please use a library!

- ▶ <https://python-gitlab.readthedocs.io/en/stable/>
- ▶ <https://metacpan.org/release/GitLab-API-v4>
- ▶ <https://github.com/xanzy/go-gitlab>
- ▶ <https://github.com/NARKOZ/gitlab>

```
salsa --help
usage: salsa <command> <parameters> <options>
```

Most used commands:

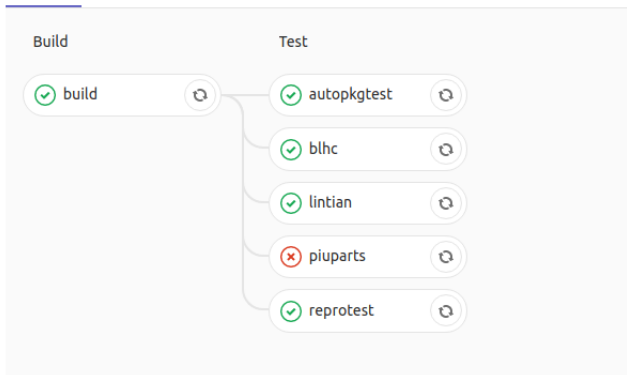
- whoami : gives information on the token owner
- checkout, co: clone repo in current dir
- fork : fork a project
- mr : create a merge request
- push_repo : push **local** git repo to upstream
↪ repository

Gitlab integrates a powerful CI tool which you can use to build and test your Code/Packages/Websites and so on.

<https://docs.gitlab.com/ce/ci/>

```
before_script:  
  - apt-get install rubygems ruby-dev -y  
  
run-test:  
  script:  
    - ruby --version
```

Pipeline Jobs 6 Failed Jobs 1



```
17 | - '*.map'  
18 |   expire_in: 1 week  
19 |
```

Validate

Status: syntax is correct

Parameter	Value
Test Job - build_definition	<pre>bundle install --path vendor/bundle\nsed -i -e 's#/etc/apache2/##' aliothrewriter.toml\ncat aliothrewriter.t bundle exec ruby bin/AliothRewriter\nfind .\ntest -f anonscm.map</pre> <p>Tag list: docker Only policy: refs, branches, tags Except policy: Environment: When: on_success</p>

- ▶ Pipeline provided by the salsa-ci team
- ▶ covers Debian package building
- ▶ can create apt repositories
- ▶ also covers several QA tools like:
 - ▶ Lintian
 - ▶ Autopkgtests
 - ▶ Piuparts
 - ▶ and probably a lot others

Check the Salsa CI Dokumentation

```
image: registry.salsa.debian.org/salsa-ci-team/ci-image
↳ e-git-buildpackage:latest

build:
  artifacts:
    paths:
      - "*.deb"
    expire_in: 1 day
  script:
    - gitlab-ci-git-buildpackage-all
```

Manipulate merge requests via git



```
git push -o merge_request.create
```

Create a merge request from git

```
git push -o merge_request.target=branch_name
```

Update target branch name for request

Even more manipulation

```
git push -o merge_request.create -o  
↪ merge_request.target=branch_name
```

Do both in one step

```
git push -o  
↪ merge_request.merge_when_pipeline_succeeds
```

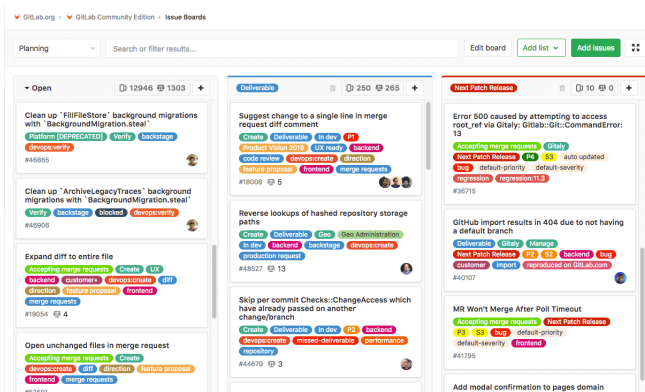
Update merge request to merge when pipeline succeeds

Even more manipulation

```
git push -o merge_request.create -o  
↔ merge_request.merge_when_pipeline_succeeds
```

Do both in one step

You can use issue boards to manage your todos and other tasks of you and your team(s)



The screenshot displays the GitLab Issue Board interface. At the top, it shows the organization 'GitLab.org' and the board name 'GitLab Community Edition - Issue Boards'. Below this, there are controls for 'Planning', a search bar, and buttons for 'Edit board', 'Add list', and 'Add issues'. The board is divided into three columns:

- Open:** Contains four issues with titles like 'Clean up `FiilFileStore` background migrations...' and 'Clean up `ArchiveLegacyTraces` background migrations...'. Each issue has a list of tags such as 'Platform (DEPRECATED)', 'Verify', 'backstage', 'devops:verify', 'backstage', 'blocked', 'devops:verify', 'Accepting merge requests', 'Create', 'UX', 'backlog', 'customers', 'devops:create', 'diff', 'direction', 'feature proposal', 'frontend', 'merge requests', 'Open unchanged files in merge request', 'Accepting merge requests', 'Create', 'dependencies', 'diff', 'direction', 'feature proposal', 'frontend', 'merge requests'.
- Deliverable:** Contains three issues with titles like 'Suggest change to a single line in merge request diff comment', 'Reverse lookups of hashed repository storage paths', and 'Skip per commit Checks: ChangeAccess which have already passed on another change/branch'. Each issue has tags such as 'Create', 'Deliverable', 'In dev', 'P1', 'Product Vision 2019', 'UX ready', 'backlog', 'code review', 'devops:create', 'direction', 'feature proposal', 'frontend', 'merge requests', 'Create', 'Deliverable', 'Geo', 'Geo Administration', 'In dev', 'backlog', 'backstage', 'devops:create', 'production request', 'Create', 'Deliverable', 'In dev', 'P2', 'backlog', 'devops:create', 'missed-deliverable', 'performance', 'repository'.
- Next Patch Release:** Contains three issues with titles like 'Error 500 caused by attempting to access root_ref via Gitaly: Gitlab::CommandError: 13', 'GitHub import results in 404 due to not having a default branch', and 'MR Won't Merge After Poll Timeout'. Each issue has tags such as 'Accepting merge requests', 'Gitaly', 'Next Patch Release', 'P4', 'S3', 'auto updated', 'bug', 'default-priority', 'default-severity', 'regression', 'regression:11.3', 'Accepting merge requests', 'Gitaly', 'Deliverable', 'Gitaly', 'Manage', 'Next Patch Release', 'P2', 'S2', 'backlog', 'bug', 'customer', 'import', 'reproduced on GitLab.com', 'Accepting merge requests', 'Next Patch Release', 'P1', 'S3', 'bug', 'default-priority', 'default-severity', 'frontend'.

There are several options to manipulate stuff by committing code and special commit messages.

To close bugs on commit by including our close webhook into your project.

```
https://webhook.salsa.debian.org/close/SOURCENAME
```

Where SOURCENAME is the name of the sourcepackage where you want to close bugs in. Other options are listed in the documentation.

Mark Bugs pendings per commit



To mark bugs pending on commit by including our tagpending webhook into your project.

`https://webhook.salsa.debian.org/tagpending/SOURCENAME`

Where SOURCENAME is the name of the sourcepackage where you want to mark bugs in. Other options are listed in the documentation.

If you want to have more functionality and don't want to host the hook on your own: please write a plugin for our webhook daemon!

You can host your documentation/pages/blog/... within pages.debian.net by creating gitlab ci job in your project and pushing it to salsa. If you need more complicated things, use gitlab-ci to build your pages.

```
image: registry.gitlab.com/pages/hugo:latest
variables:
  GIT_SUBMODULE_STRATEGY: recursive

test:
  script:
  - hugo
  except:
  - master

pages:
  script:
  - hugo
  artifacts:
  paths:
  - public
  only:
  - master
```


Reset your 2fa codes via ssh

```
ssh git@salsa.debian.org 2fa_recovery_codes
```

Create a repo by push

Achtung! They will be private after creation.

Feedback

Do you have anything to share about
gitlab/salsa?

Please check our docs and help us to improve them.

<https://salsa.debian.org/formorer/minidebconf2019talk/>