Salsa and Gitlab
cool and not so well known features
Alexander Wirt <formorer@debian.org>
08.06.2019
Overview

Introduction
Overview

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!
Gitlab

- Ruby on Rails based
- Scalable
- started as a Github clone
- Open Core
salsa.debian.org

- 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
Basics

API Token
Start with creating your API token
https://salsa.debian.org/profile/personal_access_tokens
Authenticated Requests

```bash
$ 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://metacpan.org/release/GitLab-API-v4
- https://github.com/xanzy/go-gitlab
- https://github.com/NARKOZ/gitlab
The salsa tool

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/
gitlabci config

before_script:
  - apt-get install rubygems ruby-dev -y

run-test:
  script:
    - ruby --version
Webfrontend
gitlab-ci linter

```
> '*.map'
expire_in: 1 week
```

**Status:** syntax is correct

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
</table>
| `bundle install --path vendor/bundle
nsed -i -e \'s#etc/apache2/##\' aliothrewriter.toml
ncat aliothrewriter.toml
find .\test -f anonscm.map` |
| `bundle exec ruby
bin/AliothRewriter
find .\test -f anonscm.map` |

**Test Job - build definition**

- **Tag list:** docker
- **Only policy:** refs, branches, tags
- **Except policy:**
- **Environment:**
- **When:** on success
salsa-ci

- 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
salsa-ci - Example

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

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

Create a merge request from git

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

Update target branch name for request
Even more manipulation

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

Do both in one step

```bash
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
issue boards

You can use issue boards to manage your todos and other tasks of you and your team(s)
There are several options to manipulate stuff by committing code and special commit messages.
Close Bugs per commit

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.
Contribute webhooks

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
Unsorted goodies

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.
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/