zenika

<animés par la passion>

Gitlab Configuration Details

Version 1.18-SNAPSHOT

2018-08-24

Table of Contents

Table 1. History

Date	Author	Detail
2018-08-24	bcouetil	Reworked asciidoc index.html
2018-08-24	bcouetil	Icones added for download + favicon added for webpage
2018-08-23	bcouetil	Initial commit

1. Introduction

Gitlab in his free online plan offers some nice features :

- 2,000 CI pipeline minutes per group per month on our shared runners
- Unlimited private projects and collaborators
  - Built-in CI/CD
  - Cycle Analytics
  - Issue Boards
  - Time tracking
  - Preview your changes with Review Apps
  - Publish static websites for free with GitLab Pages
  - Git LFS 2.0 support

2. Server configuration

⚠ TODO

3. Project creation

- Create a project in web interface.
  - check **Create README.md** or else **master** branch is not created
- Import git project in Eclipse
- Change git config, adding :

```
[user]
  name = yourname
  email = youraccount@yourprovider.com
```

3.1. rebase without merge commit

- In the web interface, go to **Project** → **Settings** → **General** → **Merge request**, and configure as follow :

  ☑ Fast-forward merge

  ☑ Only allow merge requests to be merged if the pipeline succeeds

  □ Only allow merge requests to be merged if all discussions are resolved

3.2. Protect master branch

Gitlab workflow is not as straightforward as Gerrit's (see below). In Eclipse, it's too easy to push to master instead of our current Merge Request's branch.

Here is how to protect it :

- In Gitlab interface, navigate to your project → **Settings** → **Repository** → expend **Protected Branches** and configure :
  - Branch = **master**
  - Allowed to merge = **Maintainers**
  - Allowed to push = **No one**

3.3. Merge Requests workflow

We assume that we will be the only one to push on that branch, so we do not check-out the branch, and just push there. This helps do fewer interactions in Eclipse.

Initialize the change

- Go to **Issues** → **New issue**
- Set information and create
- Click **Create merge request**
- In Eclipse, right click on your repo → **Pull**
  - The branch should appear under **Branches** → **Remote Tracking**
- Update the files…
- Commit
- Right click on local/master → Push Branch…
  - Toggle
    - ☐ Configure Pull
    - ☐ Force
  - Next → Finish

On each new dev iteration

- Commit with amend, ignore the warning
- Right click on local/master → Push Branch…
- Toggle
  - ☐ Configure Pull
  - ☑ Force
- Next → Finish

Merge in Gitlab

Browse the Merge Request in Gitlab

☑ Remove source branch

Remove the Remote branch in your git repo → Branches → Remote Tracking

Right click on your repo → **Pull**

- If you get an error, you have obviously configure Pull for the dead branch, so use **Pull…** instead and select **master** branch

3.4. Pipeline creation

Pipeline content

```
image: maven:latest

stages:
  - build
  - test
  - deploy

variables:
  #MAVEN_CLI_OPTS: "-s cg-settings.xml --batch-mode"
  MAVEN_CLI_OPTS: "--batch-mode"
  MAVEN_OPTS: "-Dmaven.repo.local=.m2/repository"

cache:
  paths:
    - .m2/repository/

build:
  stage: build
  script:
    - apt-get update -qq && apt-get install -qq --assume-yes libc6-i386
    - mvn clean dependency:purge-local-repository
    - mvn install -DskipTests -Dassembly.skipAssembly=true


# the pipeline step must be named 'pages' for gitlab to deploy locally (in addition to github)
pages:
  stage: build
  script:
    #- ./src/scripts/asciidocOnlyModified.sh
    - ./src/scripts/asciidocHistory.sh .
    - apt-get update -qq && apt-get install -qq --assume-yes graphviz
    - git config --global user.email "gitlab@noreply.com"
    - git config --global user.name "GitLab"
    - mvn generate-resources -Dadoc.skip=false --non-recursive
    - cp src/docs/asciidoc/*.adoc target/generated-docs/
    - cp -R src/docs/asciidoc/subdocs target/generated-docs/
    - mvn scm-publish:publish-scm
    - mkdir public
    - mv target/generated-docs/* public/
  artifacts:
    paths:
      - public
  #only:
  #  - master


unit-test:
  stage: test
  script:
    - mvn clean
    - mvn test -Dcheckstyle.skip=true -pl cg-utils

integration-test:
  stage: test
  script:
    - mvn clean
    - mvn verify failsafe:verify -Dcg.ut.skip=true -Dcheckstyle.skip=true -pl cg-utils

assembly:
  stage: deploy
  script:
    - mvn clean
    - mvn install -DskipTests -Dcheckstyle.skip=true -pl cg-utils
  #only:
  #  - master
```

4. Appendix

4.1. Revision marks

Differences since last tag

```
```