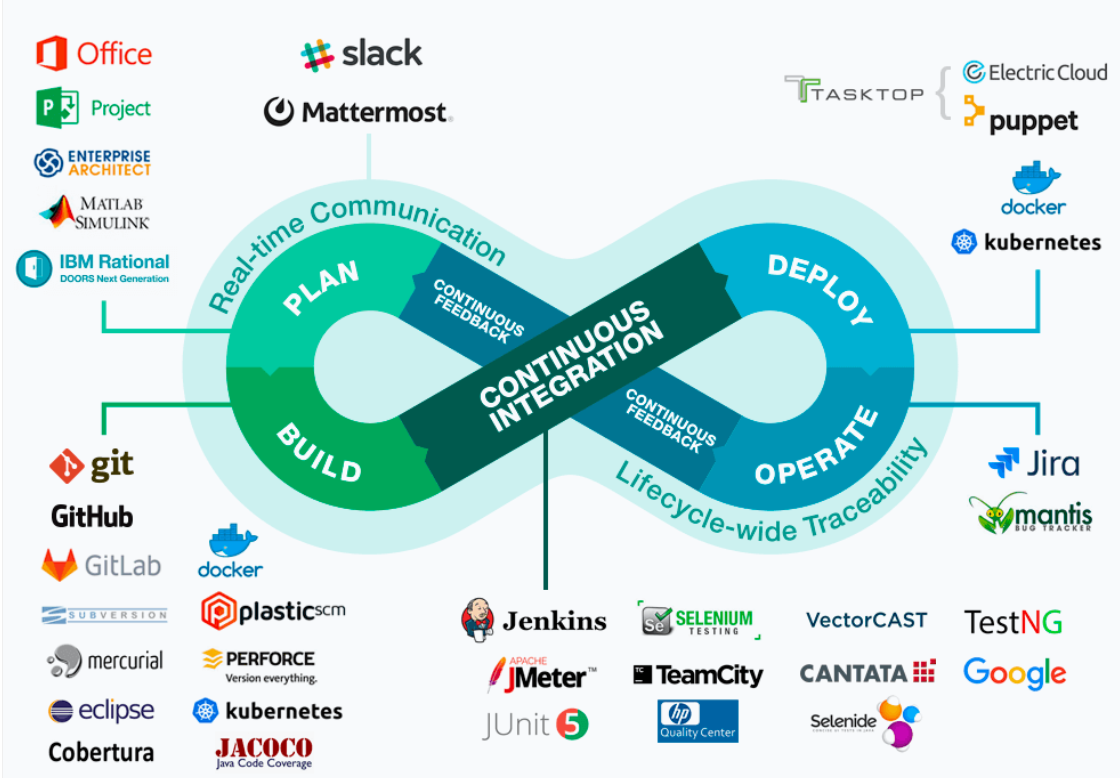
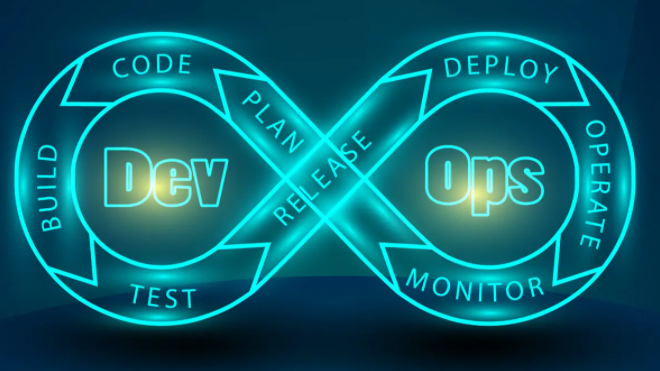
# **GITLAB-CI**

Qu’est-ce le **DevOps** ? Collaborer entre le Dev et le Ops pour livrer plus souvent en production

* Une **Philosophie** (!= métier)
* Un but : **réduire le time to market**
* Une extension de l’**agilité**
* Mise en avant de la **collaboration**
* Mise en avant des principes **C.A.L.M.S**
* **Détruire les silos** (groupe de travail)
* **DevOps, DataOps, DevSecOps** etc…

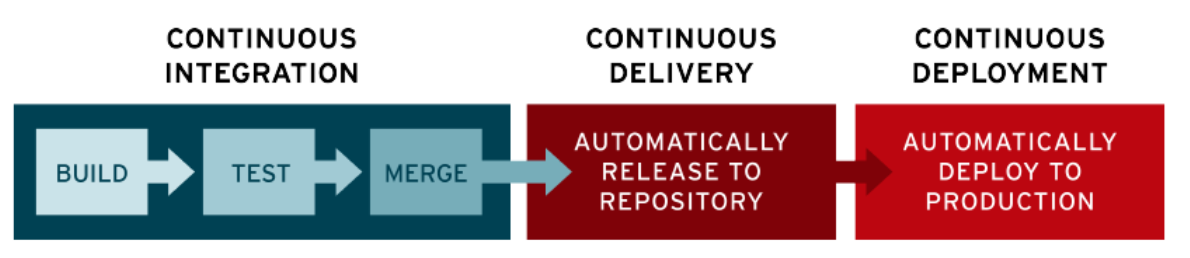




**Historique**

* 2003 : Google créé le **SRE (Site Relability Engeneer)**
* 2000 : Accélération du web
* 2009 : Patrick Debois parle de DevOps
* 2014 : Forte accélération du DevOps

La **forge logicielle** est l’endroit où on met le code en commun. Elle va traiter le code à livrer.



Merge request Image docker

## GitOps

GitOps est aujourd’hui une philosophie de plus en plus populaire, proche du DevOps elle pousse le concept de forge collaborative avant toute la stratégie autour de celle-ci.

Git devient la « source de vérité unique »

**Git** Versionning

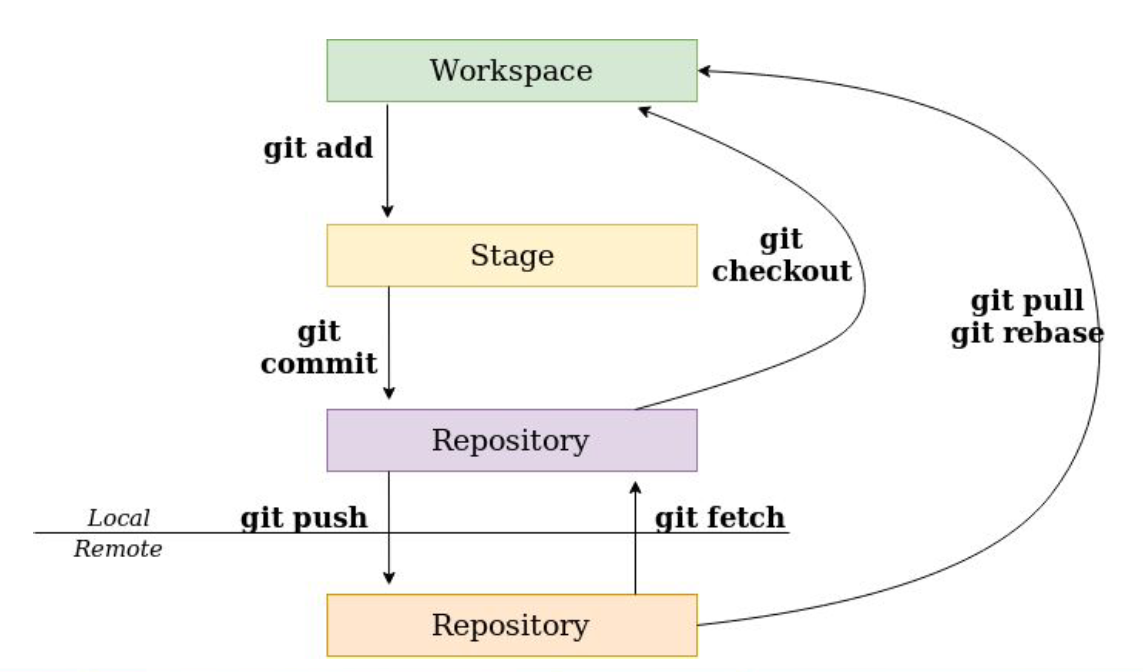
* Travailler à plusieurs sur le code efficacement
* Le cas de Kernel Linux
* Git, la descendance de SVN ?
* Base de tous les métiers IT
* Beaucoup de solutions de centralisation (Gitlab, Github, Bitbucket…)
* Système de branche
* Connexions à des serveurs distant SSH et HTTPS
* Le tout est géré dans un fichier **.git** à la base du projet
* Référence par empreinte SHA-1

Configuration

* Git config –system #/etc/gitconfig
* Git config
* Git config

Annuler son travail

* Git reset [HASH]
* Git reset [HASH]
* Git checkout [FICHIER]



### **TP : Installation & découverte de Git**

1. Installation de Git
2. Création d’un projet git
3. Créer et ajouter des fichiers dans le staging du commit
4. Commit en local sur une branch « dev »
5. Afficher l’historique des commits (CLI)
6. Effectuer des modifications
7. Afficher les différences avec le HEAD
8. Commit les modifications
9. Les annuler sans garder celle-ci

Mkdir TP1

Cd TP1/

Git init

Touch test

Git status (0 fichier commits – branche principale)

Vim README.md

Txt -> Sup de vinci

Git status (git ne gere pas le fichier)

Git checkout -b SUP (créer une branche SUP il faut jms bosser sur la branche master(principale))

(un master est un commit qui pointe sur un HEAD. Un merge prend 2 branches et les assemble avec le master)

Git add README.md (mettre le ReadMe dans le staging)

Git commit -m « Add README.md » (fait un commit avec le message)

Git log (voit les messages laissé afficher l’historique des commit)

Git branch (afficher les branches)

Modifier le fichier README.md avec vim README.md

Git diff HEAD^-1 (fait la différence entre le dernier commit et l’actuel)

Git add README.md

Git reset README.md (annuler la dernière modification mais va laisser dans le stage)

Git reset README.md –hard va tout supprimer

### **GITLAB**

Les produits :

* Gitlab CE
  + Open source
  + Hebergeable
  + Basique
  + Gratuit
* Gitlab EE
  + Propriétaire
  + Hébergeable
  + Fonctions supplémentaires
  + Licence payante
* Gitlab.com
  + Site web

Les fonctionnalités :

* Manage (Suivi de contribution, tickets…)
* Plan (planification de tâche, tableau de gestion de projet)
* Create (création de code, des merges, des requests)
* Verify (Analyse du code)
* Package (Création de stockage)
* Secure
* Release
* Monitor

### TP2 : Découverte de Gitlab.com

1. Création d’un compte sur Gitlab.com

Créer un compte education

1. Configuration du compte (clé SSH : ssh-keygen -t ed25519)

Copier **ssh-keygen -t ed25519** en ligne de commande pour générer une clé ssh.

Faire un cat ssh.pub pour avoir la clé publique et la copier dans le SSH public sur Gitlab

1. Création d’un projet

Créer TP2

1. Créer un dashboard et ajoutez des « issues »
2. Commit et push le fichier (TP1) sur une branche « dev »
3. Merge Request pour mettre sur Master lié à une issue
4. Afficher l’historique des commits (GUI+CLI)
5. Accepter la MR sur gitlab.com

**Curl** : interroger des pages web

Blocage Installation Gitlab :

Voir le répertoire /var/log/gitlab

**Gitlab-ctl reconfigure** lorsque l’installation n’est pas allée jusqu’au bout

Voir doc API **REST** et **GraphQL**

**TP : administration de GitLab CE Omnibus**

1. Ajouter un message général pour avertir d’une maintenance
2. Vérifier avec la partie monitoring que tout fonctionne (EN CLI)
3. Redémarrer le composant Nginx
4. Vérifier le status (EN CLI)
5. Redémarrer tout le GitLab (avec **systemd** ou **gitlab-ctl**)

systemctl **status** gitlab-runsvdir.service ou **restart**

1. Exécuter 2 appels APIs

**curl --header "PRIVATE-TOKEN: HqpQY5uUEyZmaC1htuxD" "http://192.168.130.242/api/v4/projects" | python -m json.tool**

**HqpQY5uUEyZmaC1htuxD** est la token-access private avec des droits api et sudo

1. Créer un backup de GitLab

**Vim /etc/gitlab/gitlab.rb** -> voir l’url et les backup & restore

/var/opt/gitlab/backups

**Backup**

Les backups se créé dans **/var/opt/gitlab/backups**

**gitlab-backup create**

**Restauration**

**voir la doc** [**https://docs.gitlab.com/ee/raketasks/backup\_restore.html**](https://docs.gitlab.com/ee/raketasks/backup_restore.html)

les services a arrêter avant de faire la restauration

sudo gitlab-ctl stop unicorn

sudo gitlab-ctl stop puma

sudo gitlab-ctl stop sidekiq

Créer un

**https://docs.gitlab.com/ee/api/api\_ressources.html**

Résumé video « Les bases de GIT » <https://www.youtube.com/watch?v=gp_k0UVOYMw>

1. Présentation de la méthode de travail avant le sans le git avec 2 collègues qui travaillent sur des documents sans collaboration réelle, chacun à son tour et ayant la possibilité de perdre le document ou des bouts de code.
2. Présentation de la méthode de travail avec git de façon structurée avec des versions et collaboratif
3. Création d’un compte GitHub

### **Mini-TP : Docker**

1. Installer Docker (voir site https://docs.docker.com/engine/install/debian/)

$ sudo apt-get update

$ sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

gnupg-agent \

software-properties-common

Installation de Docker Engine

sudo apt-get update

$ sudo apt-get install docker-ce docker-ce-cli containerd.io

Le contenu /var/lib/docker/, y compris les images, les conteneurs, les volumes et les réseaux, est conservé.

1. Créer un conteneur (CMS Ghost, port par défaut :2368)
   1. **docker run -d --name monPremierDocker ghost**
2. Créer un Dockerfile et construire une image (cf screen)
   1. **docker build -t hello . (**le point pour dire le répertoire actuel)
   2. **docker run --rm hello**
   3. **docker ps -all**
3. Voir les images locales
   1. **docker images**

### **Fonctionnement de la CD²**

le fichier **.gitlab-ci.yml**

* root du projet
* syntaxe yml
* système de succession de stages qui contiennent des jobs parallèlisables

**les runners**

* multiOS(GNU/Linux, MACOS, BSD, Windows)
* Communication par **HTTP**
* Paramètrable par **config.toml**
* Différents **executors**
  + Shell
  + Docker
  + VirtualBox
  + Kubernetes

**Installtaion d’un runner**

* Ajout des dépôts GitLab
  + **Apt install gitlab-runner**
* Enregistrement
  + **Gitlab-runner register**

**TP : Ma première pipeline**

1. Installer un runner « shell »
2. Donner un tag « cli » au runner
3. Créer un .gitlab-ci.yml qui va juste faire un : (https://docs.gitlab.com/ee/ci/quick\_start/)
   1. Echo « Mon premier job ! » dans un premier job
   2. Echo « Mon deuxième job ! » dans un second job
4. Créer un second runner « Docker »
5. Lui donner le tag « conteneur »
6. Ajouter le dans le gitlab-ci.yml
   1. Un troisième job qui est dépendant des deux premiers qui s’exécute uniquement sur Master

INSTALLATION DU RUNNER

Cd /tmp

Wget <https://s3.amazonaws.com/gitlab-runner-downloads/master/deb/gitlab-runner_amd64.deb>

Dpkg -i gitlab-runner\_amd64.deb

**PATH="/sbin:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games » (vérifier le PATH)**

**Faire un copier-coller dans le vim /root/.bashrc pour ne plus avoir à le refaire**

Gitlab-runner register

(suivre instruction sur runner gitlab **set up and share manually**)

En faisant des copier-coller de l’IP et du token nom du tag **toto**

Sélectionner shell

Vérifier dans l’onglet runner qu’il a été activé. Voir s’il est activé et partagé.

On veut Créer un pipeline

Créer un nouveau projet nom :**demoCD**

Suivre les étapes du repositorie avec un **git clone**

**Mkdir demoCD**

**Cd demoCD**

**Git clone (copier l’adresse du projet)**

**Mettre la clé ssh**

**Créer un README.md -> touch README.md**

**Créer vim .gitlab-ci.yml**

Dans le fichier voir demo https://docs.gitlab.com/ee/ci/yaml/

{

Stage : (attend une liste)

- demoshell

Hello1 :

Stage : demoshell

Tags :

- toto

Script :

- echo « hello job !»

Hello2 :

Stage : demoshell

Script :

- echo « hello job2 !»

}

**Git add**

**Git commit**

**Git push** (le push va exécuter le pipeline et lancer les job)

Faire un F5 sur le navigateur sur le projet dans l’onglet Pipeline CI/CD

**Faire les 3 dernières commandes à chaque modification du document**

Créer nouveau runner Docker

Gitlab-runner register -> description container -> tags container docker test -> sélectionner docker -> image bash

Vim .gitlab.yml (modifier fichier)

{

Stage : (attend une liste)

stages:

- demoshell

- demoshell2

- democontainer

Job1:

stage: demoshell

tags:

- test-runner

script:

- echo "Hello Job1 !"

Job2:

stage: demoshell2

tags:

- test-runner

script:

- echo "Hello Job2 !"

dependencies:

- Job1

DemoJobDocker:

stage: democontainer

image: bash

tags:

- container

script:

- echo "Hello from Docker"

only:

- master

dependencies:

- Job1

}

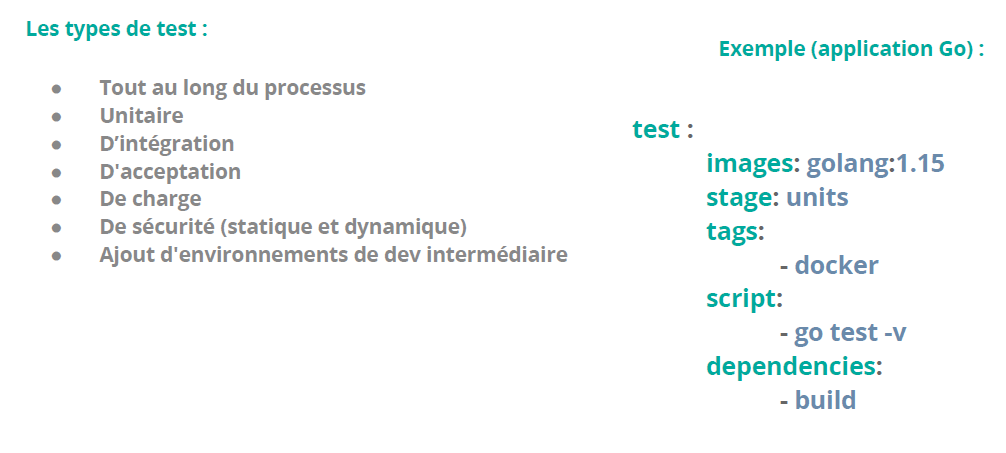
**Git add .gitlab.yml**

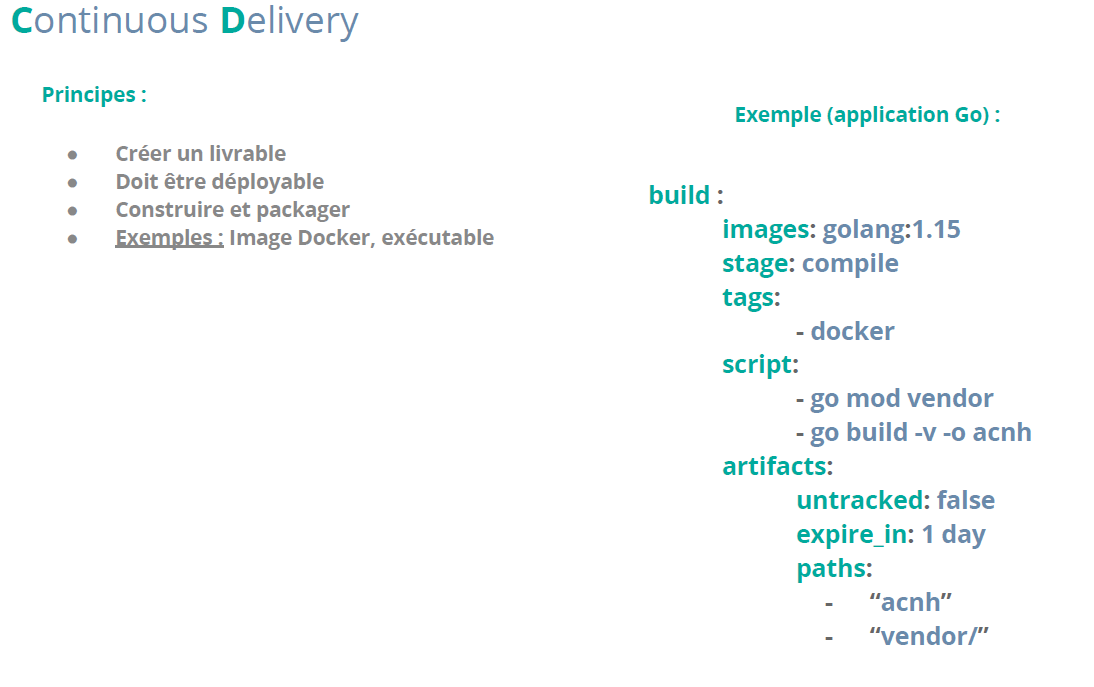
**Git commit -m « modification du fichier .yml »**

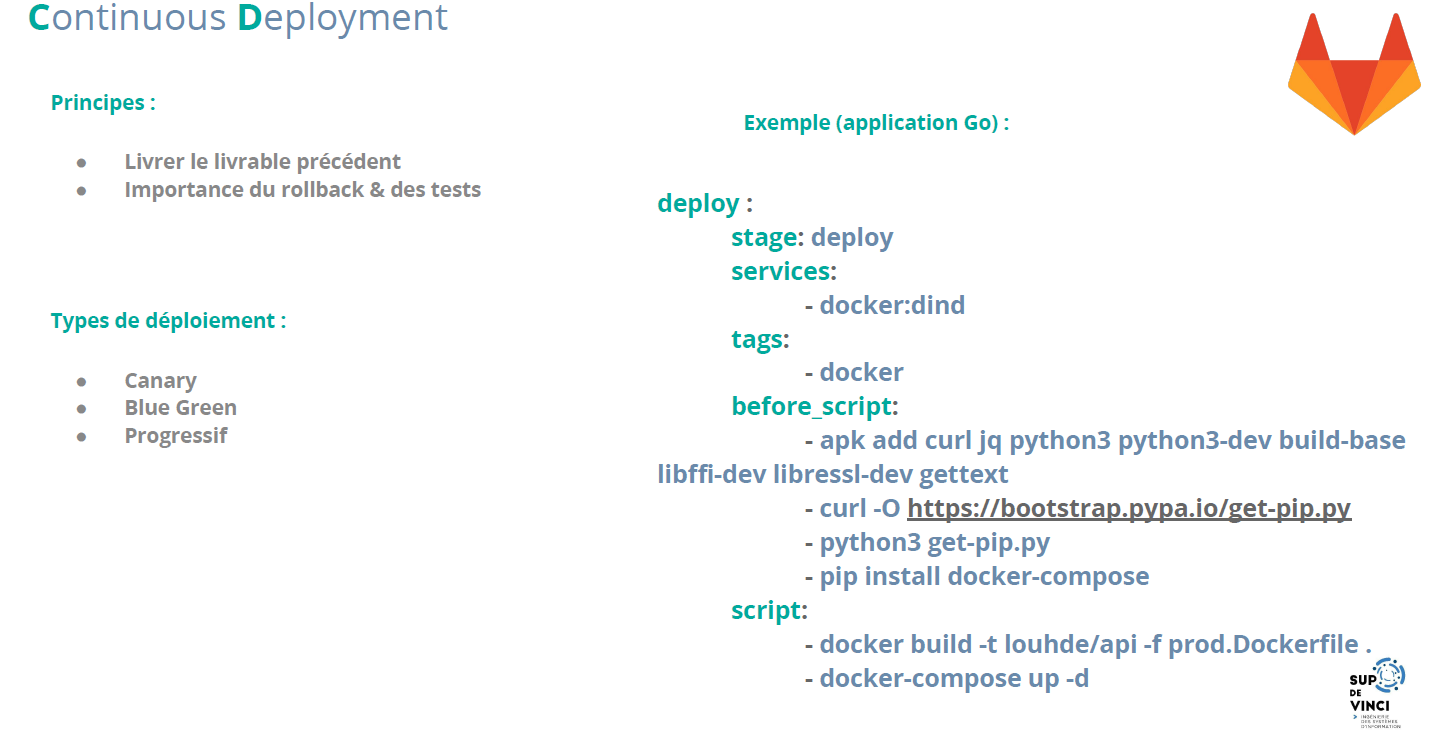
**Git push**

**Tous les runners sont configuré sur le /etc/gitlab/gitlab-runner/config.toml**

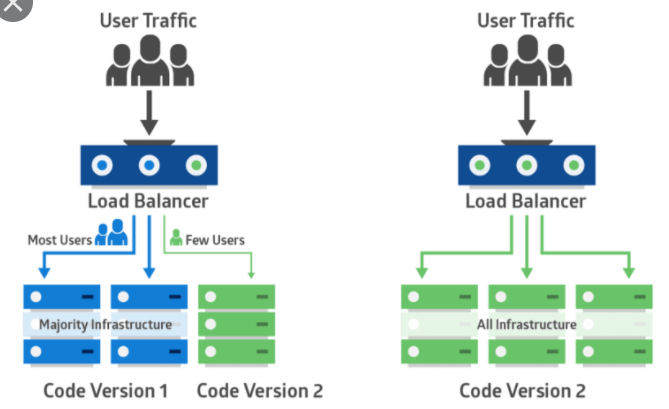
Le test ­| la QA (14h50)

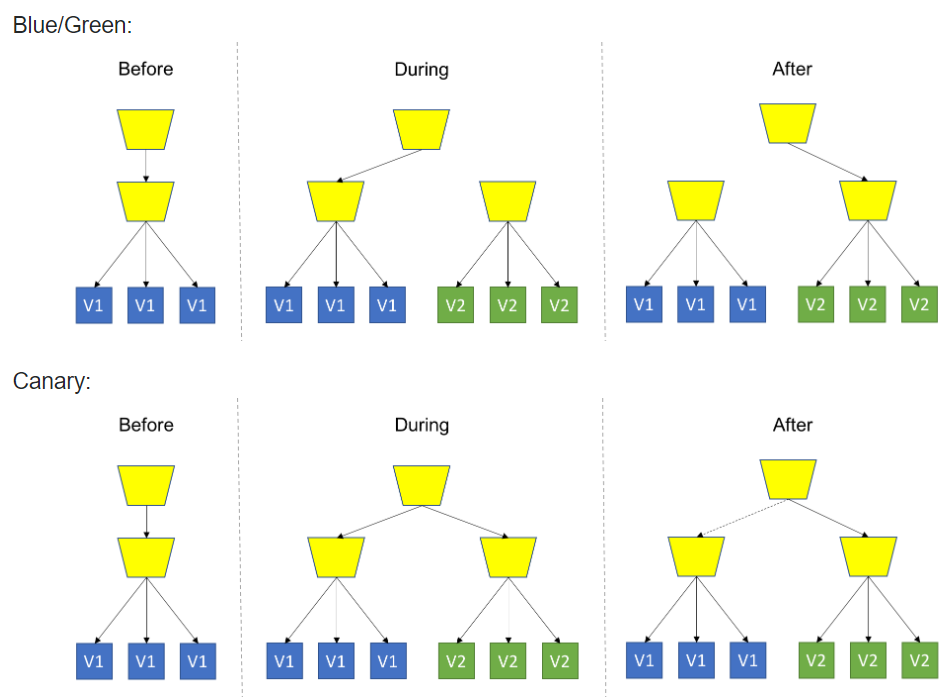




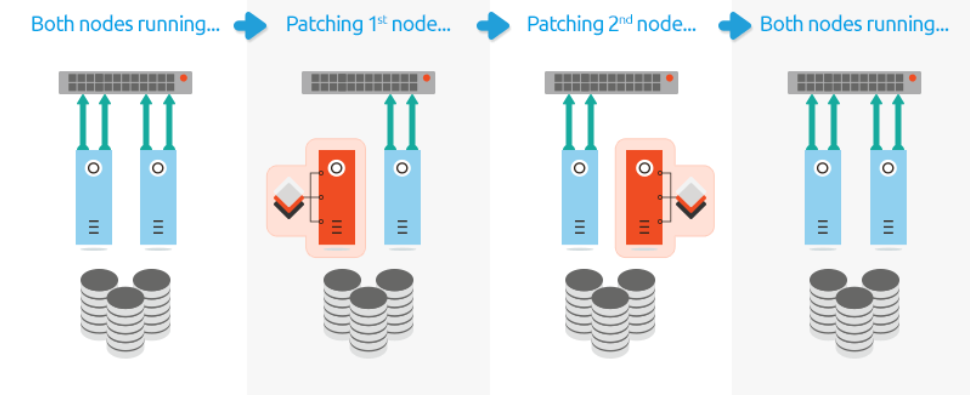


Les types de déploiement :





Deployment progressif



**TP : Ma première chaine CD**

1. Push votre projet « shell » du TP précédent
2. Créer un CD qui va :
   1. Tester les scripts que vous avez push
      1. Files=(git diff-tree --no-commit-id --name-only -t $CI\_COMMIT\_SHA)
   2. Vérifier la présence des scripts ajoutés dans le README
      1. Utiliser grep
   3. Créer une archive de vos scripts
      1. Tar -cvf archive.tar script/
      2. Uniquement sur Master
      3. L’artefact doit être disponible 1 semaine

Créer un nouveau projet

Créer un nouveau script script.sh

Dans le .gitlab-ci.yml ecrire

Stages

-QA

Shellexectest :

Stage : QA

Tags :

-shell

Script :

* + - Files= « git diff-tree --no-commit-id ---name-only -t $CI\_COMMIT\_SHA »
    - For i in $files ; do echo $i ; done (pour exécuter les scripts changer echo par bas)\*

Le script va lister tous les fichiers non commit se trouvant dans le répertoire

stages:

- QA

- Documentation

- Delivery

shellexectest:

stage: QA

tags:

- shell

script:

- files=`git diff-tree --no-commit-id --name-only -r $CI\_COMMIT\_SHA scripts/`

- for i in $files; do echo $i; done

- for i in $files; do bash $i; done

Readme:

stage: Documentation

tags:

- shell

script:

- files=`git diff-tree --no-commit-id --name-only -r $CI\_COMMIT\_SHA scripts/`

# - for i in $files; grep $i README.md; done

Archive:

stage: Delivery

tags:

- shell

script:

- tar -cvf archive.tar scripts/

only:

- master

artifacts:

name: "$CI\_JOB\_NAME-$CI\_COMMIT\_REF\_NAME"

paths:

- archive.tar

expire\_in: 1 week

Projet : Quand la CD/CD est aux services des Devs & des Ops

<https://gitlab.com/thger_formation/examen1>

1. Gestion des tickets (issues)
2. Bonnes pratiques sur le « flowgit » (branches, merge request et commits) et la collaboration
3. L’intégration de la CI/CD, l’utilisation des fonctionnalités Gitlab (variables, stages etc… )
4. Bien sûr respect code

Les commandes fait :

root@debian:/usr/local/bin# history

1 shutdown -0

2 shutdown -now

3 shutdown

4 ip a"

5 ip a

6 EXTERNAL\_URL = "https://192.168.130.242"

7 exit

8 apt-get install -y curl openssh-server ca-certificates

9 apt-get install -y postfix

10 curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh

11 bash

12 EXTERNAL\_URL = "https://192.168.130.242"

13 sudo EXTERNAL\_URL = "https://192.168.130.242"

14 EXTERNAL\_URL = "http://192.168.130.242"

15 apt-get install EXTERNAL\_URL

16 EXTERNAL\_URL="http://192.168.130.242"

17 external\_url = "http://192.168.130.242"

18 EXTERNAL\_URL="http://192.168.130.242"

19 echo $EXTERNAL\_URL

20 apt-get install gitlab-ce

21 apt-get install gitlab-ce

22 nano /etc/apt/sources.list.d/gitlab\_gitlab-ce.list.save

23 exit

24 su

25 echo $EXTERNAL\_URL

26 echo $EXTERNAL\_URL

27 EXTERNAL\_URL="http://192.168.130.242"

28 echo $EXTERNAL\_URL

29 systemctl gitlab status

30 gitlab-ctl status

31 gitlab-ctl

32 gitlab-ctl status

33 gitlab-ctl restart

34 gitlab-ctl

35 ls /etc/apt/

36 gitlab-ctl status

37 gitlab-ctl status

38 gitlab-ctl

39 echo $EXTERNAL\_URL

40 export GITLAB\_URL="//192.168.130.242"

41 gitlab-ctl status

42 echo $EXTERNAL\_URL

43 gitlab-ctl

44 gitlab-ctl stop

45 gitlab-ctl restart

46 gitlab

47 gitlab-ctl status

48 gitlab-ctl status

49 systemctl status gitlab

50 systemctl status gitlab.service

51 ls /etc/gitlab/

52 restart

53 shutdown -r now

54 apt-get update

55 apt upgrade

56 ls

57 su debian

58 su user

59 echo $EXTERNAL\_URL

60 EXTERNAL\_URL="http://192.168.130.242"

61 echo $EXTERNAL\_URL

62 apt-get install gitlab-ce

63 ls /etc/apt/sources.list

64 ls -a /etc/apt/sources.list

65 nano /etc/apt/sources.list

66 apt-get update

67 nano /etc/apt/sources.list

68 apt-get update

69 apt-get upgrade

70 apt-get install gitlab-ce

71 apt-get install gitlab-ee

72 apt-get install gitlab-ce

73 apt install gitlab-ce

74 curl -s https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh | sudo bash

75 curl -s https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh | bash

76 echo $EXTERNAL\_URL

77 apt install gitlab-ce

78 echo $EXTERNAL\_URL

79 cat /etc/gitlab/gitlab.rb

80 ip a

81 gitlab status

82 git status

83 su

84 gitlab-ctl status

85 gitlab-ctl restart nginx

86 gitlab-ctl status nginx

87 gitlab-ctl restart

88 ls /lib/api.rb

89 ls /lib/

90 ls /api

91 curl "https://gitlab.example.com/api/v4/projects"

92 curl "http///192.168.130.242/api/v4/projects"

93 curl "http://192.168.130.242/api/v4/projects"

94 curl --header "PRIVATE-TOKEN: <zo3CTw8yDWnQ6QnN9b6y>" "http://192.168.130.242/api/v4/projects"

95 APPEL API

96 systemd status gitlab

97 systemd status

98 curl "https://gitlab.example.com/api/v4/projects?private\_token=<your\_access\_token>"

99 curl "http://192.168.130.242/api/v4/projects?private\_token=<zo3CTw8yDWnQ6QnN9b6y>"

100 curl "http://192.168.130.242/api/v4/projects?private\_token=<PVKLtWgYuzey6oc4DmyR>"

101 curl --header "PRIVATE-TOKEN: <PVKLtWgYuzey6oc4DmyR>" "http://192.168.130.242/api/v4/projects"

102 curl --header "PRIVATE-TOKEN: <HqpQY5uUEyZmaC1htuxD>" "http://192.168.130.242/api/v4/projects"

103 systemcl status gitlab-runsvdir.service

104 systemctl status gitlab-runsvdir.service

105 curl --header "PRIVATE-TOKEN: <your\_access\_token>" "https://gitlab.example.com/api/v4/issues"

106 curl --header "PRIVATE-TOKEN: <HqpQY5uUEyZmaC1htuxD>" "http://192.168.130.242/api/v4/issues"

107 curl --header "PRIVATE-TOKEN: <HqpQY5uUEyZmaC1htuxD>" "http://192.168.130.242/api/v4/issues"

108 curl --header "PRIVATE-TOKEN: <HqpQY5uUEyZmaC1htuxD>" "http://192.168.130.242/api/v4/project"

109 curl --header "PRIVATE-TOKEN: <HqpQY5uUEyZmaC1htuxD>" "http://192.168.130.242/api/v4/projects"

110 curl --header "PRIVATE-TOKEN: <HqpQY5uUEyZmaC1htuxD>" "http://192.168.130.242/api/v4/projects" | python -m json.tool

111 vim /etc/gitlab/gitlab.rb

112 apt-get install vim

113 vim /etc/gitlab/gitlab.rb

114 cd /var/opt/gitlab/backups/

115 ls

116 gitlab -backup create

117 gitlab-backup create

118 ls

119 curl --header "PRIVATE-TOKEN: <HqpQY5uUEyZmaC1htuxD>" "http://192.168.130.242/api/v4/projects"

120 cd /home/user/

121 curl --header "PRIVATE-TOKEN: HqpQY5uUEyZmaC1htuxD" "http://192.168.130.242/api/v4/projects"

122 curl --header "PRIVATE-TOKEN: HqpQY5uUEyZmaC1htuxD" "http://192.168.130.242/api/v4/projects" | python -m json.tool

123 apt-get update

124 apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common

125 curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -

126 curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -

127 apt-key fingerprint 0EBFCD88

128 add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian \

129 $(lsb\_release -cs) \

130 stable"

131 apt-get update

132 apt-get install docker-ce docker-ce-cli containerd.io

133 docker run hello-world

134 docker --version

135 ls /var/lib/docker/

136 ls /var/lib/docker/containers/

137 docker run -d –name monPremierDocker ghost

138 docker run -d –-name monPremierDocker ghost

139 docker run --help

140 docker run -d –-name monPremierDocker ghost

141 docker ps --all

142 docker run -d ghost

143 docker build -t hello

144 docker run –rm hello

145 docker run –rm "hello"

146 docker run --help

147 docker run –-rm hello

148 ps --all

149 docker ps --all

150 docker build -t hello

151 docker build --help

152 docker build -t hello

153 docker build -t 'hello'

154 docker build -t HELLO

155 docker status

156 docker --help

157 docker build -t hello

158 docker ps --all

159 docker build -t test

160 docker build -t test /home/user/

161 ls /home/user/

162 ls

163 ls -a

164 docker images

165 docker start

166 systemctl docker

167 docker run -d --name "monPremierDocker" ghost

168 docker run -d --name "monPremierDocker" ghost

169 docker ps

170 docker ps

171 docker ps

172 docker build -t hello

173 echo 2368

174 docker run -d -p 8080:2368 ghost

175 ss -lntp |grep 8080

176 ss -lntp |grep 2368

177 ss -lntp |grep 8080

178 docker run -d -p 8080:2368 ghost

179 docker run -d -p 2323:2368 ghost

180 ss -lntp | grep 2323

181 mkdir TP4-DOCKERFILE

182 cd TP4-DOCKERFILE/

183 vim dockerfile

184 ls

185 docker build -t hellosup

186 docker build -t hello .

187 docker run --rm hello .

188 docker run --rm hello

189 docker ps

190 docker images

191 docker ps -a

192 docker run hello

193 docker ps -a

194 docker ps -all

195 docker ps --all

196 docker run --rm hello

197 docker ps -a

198 docker image

199 docker images

200 ls

201 ls dockerfile

202 ls /etc/gitlab/

203 cd ..

204 ls

205 curl -LJO https://gitlab-runner-downloads.s3.amazonaws.com/latest/deb/gitlab-runner\_<arch>.deb

206 curl -LJO http://gitlab-runner-downloads.s3.amazonaws.com/latest/deb/gitlab-runner\_<arch>.deb

207 curl -LJO https://gitlab-runner-downloads.s3.amazonaws.com/latest/deb/gitlab-runner\_<arch>.deb

208 curl -LJO https://gitlab-runner-downloads.s3.amazonaws.com/latest/deb/gitlab-runner-linux-386.deb

209 dpkg -i gitlab-runner-linux-386.deb

210 curl -LJO https://s3.amazonaws.com/gitlab-runner-downloads/master/binaries/gitlab-runner-linux-386

211 ls

212 dpkg -i gitlab-runner-linux-386.deb

213 apt-get install dpkg

214 dpkg --help

215 dpkg -i .gitlab-runner-linux-386.deb

216 dpkg -i ./gitlab-runner-linux-386.deb

217 dpkg -i /home/user/gitlab-runner-linux-386.deb

218 dpkg -i /home/user/gitlab-runner-linux-386

219 apt install gitlab-runner-linux-386

220 apt install gitlab-runner-linux-386.deb

221 dpkg -i gitlab-runner-linux-386

222 dpkg -i gitlab-runner-linux-386.deb

223 ls gitlab-runner-linux-386

224 ls /usr/local/bin/

225 chmod +x gitlab-runner

226 chmod +x gitlab-runner-linux-386

227 dpkg -i gitlab-runner-linux-386

228 curl -L --output /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-

229 chmod +x /usr/local/bin/gitlab-runner

230 ls /usr/local/bin/gitlab-runner

231 dpkg -i gitlab-runner-linux-386

232 dpkg -i /usr/local/bin/gitlab-runner

233 curl -L --output /usr/local/bin/gitlab-runner https://s3.amazonaws.com/gitlab-runner-downloads/master/deb/gitlab-runner\_amd64.deb

234 dpkg -i gitlab-runner\_386.deb

235 dpkg -i /usr/local/bin/gitlab-runner\_386.deb

236 ls

237 chmod +x gitlab-runner-linux-386

238 chmod +x gitlab-runner-linux-386.deb

239 dpkg -i gitlab-runner\_386.deb

240 ls

241 ls /usr/local/bin/

242 ls /usr/local/bin/gitlab-runner

243 cd /tmp

244 wget https://s3.amazonaws.com/gitlab-runner-downloads/master/deb/gitlab-runner\_amd64.deb

245 dpkg -i gitlab-runner\_amd64.deb

246 ls

247 ls -l

248 chmod +x gitlab-runner\_amd64.deb

249 dpkg -i gitlab-runner\_amd64.deb

250 ls -l

251 dpkg -i gitlab-runner\_amd64.deb

252 ls -l

253 apt clean

254 apt update

255 dpkg -i gitlab-runner\_amd64.deb

256 dpkg -i gitlab-runner\_amd64.deb

257 curl -L https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh | sudo bash

258 export GITLAB\_RUNNER\_DISABLE\_SKEL=true; sudo -E apt-get install gitlab-runner

259 apt-cache madison gitlab-runner

260 export GITLAB\_RUNNER\_DISABLE\_SKEL=true; sudo -E apt-get install gitlab-runner=10.0.0

261 export GITLAB\_RUNNER\_DISABLE\_SKEL=true; -E apt-get install gitlab-runner=10.0.0

262 export GITLAB\_RUNNER\_DISABLE\_SKEL=true;| -E apt-get install gitlab-runner=10.0.0

263 gitlab-runner status

264 gitlab-runner register

265 ls

266 ls -l

267 dpkg -l | grep gitlab

268 systemctl status gitlab-runner

269 gitlab-runner register

270 gitlab-runner register

271 sudo gitlab-runner register

272 ls

273 ls /tmp

274 ls -l /tmp

275 systemctl status gitlab-runner

276 gitlab-runner register

277 gitlab-runner register

278 systemctl restart gitlab-runner

279 systemctl status gitlab-runner

280 gitlab-runner register

281 ls /usr/local/bin/gitlab-runner

282 ls -a /usr/local/bin/gitlab-runner

283 cd /tmp

284 gitlab-runner register

285 ls /tmp/

286\*

287 clear

288 history

289 gitlab-runner register

290 gitlab-runner register .

291 ls

292 ls -l

293 ls /home/usr

294 ls /home/user

295 cd /home/user/

296 remove gitlab-runner-linux-386

297 rm gitlab-runner-linux-386

298 rm gitlab-runner-linux-386.deb

299 ls

300 ls /usr/local/bin/

301 cd /usr/local/bin/gitlab-runner

302 cd /usr/local/bin/

303 ls -a

304 gitlab-runner gitlab-runner

305 gitlab-runner register

306 dpkg --purge gitlab-runner

307 dpkg --remove gitlab-runner

308 dpkg --reconfigure gitlab-runner

309 dpkg --reconfigure

310 history

hamidou@DESKTOP-UIQRHPC:~$ sudo apt-get update [sudo] password for hamidou: Sorry, try again. [sudo] password for hamidou: Hit:1 http://archive.ubuntu.com/ubuntu bionic InRelease Get:2 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB] Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB] Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB] Get:5 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [1089 kB] Get:6 http://archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [360 kB] Get:7 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [868 kB]Get:8 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [107 kB] Get:9 http://archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [23.0 kB] Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1113 kB] Get:11 http://archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [348 kB] Get:12 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [269 kB] Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [21.7 kB] Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6920 B] Get:15 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [7516 B] Get:16 http://archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [4764 B] Get:17 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [7736 B] Get:18 http://archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [4588 B] Get:19 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [95.5 kB] Get:20 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [20.4 kB] Get:21 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [708 kB] Get:22 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [236 kB] Get:23 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [8512 B] Get:24 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2908 B] Fetched 5555 kB in 12s (445 kB/s) Reading package lists... Done hamidou@DESKTOP-UIQRHPC:~$ sudo apt-get install git Reading package lists... Done Building dependency tree Reading state information... Done The following package was automatically installed and is no longer required: libdumbnet1 Use 'sudo apt autoremove' to remove it. Suggested packages: git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn The following packages will be upgraded: git 1 upgraded, 0 newly installed, 0 to remove and 145 not upgraded. Need to get 3915 kB of archives. After this operation, 16.4 kB of additional disk space will be used. Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:2.17.1-1ubuntu0.7 [3915 kB] Fetched 3915 kB in 27s (148 kB/s) (Reading database ... 30168 files and directories currently installed.) Preparing to unpack .../git\_1%3a2.17.1-1ubuntu0.7\_amd64.deb ... Unpacking git (1:2.17.1-1ubuntu0.7) over (1:2.17.1-1ubuntu0.5) ... Setting up git (1:2.17.1-1ubuntu0.7) ... hamidou@DESKTOP-UIQRHPC:~$ git --version git version 2.17.1 hamidou@DESKTOP-UIQRHPC:~$ git config --global user.name "Hamidou" hamidou@DESKTOP-UIQRHPC:~$ git config --global user.email "hamidou.mary@supdevinci-edu.fr" hamidou@DESKTOP-UIQRHPC:~$ ls fichier.txt note.py pays.txt test.py hamidou@DESKTOP-UIQRHPC:~$ git init Initialized empty Git repository in /home/hamidou/.git/ hamidou@DESKTOP-UIQRHPC:~$ mkdir TP\_GIT hamidou@DESKTOP-UIQRHPC:~$ mv .git TP\_GIT/ hamidou@DESKTOP-UIQRHPC:~$ ls TP\_GIT fichier.txt note.py pays.txt test.py hamidou@DESKTOP-UIQRHPC:~$ cd TP\_GIT/   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ cd hamidou@DESKTOP-UIQRHPC:~$ ls TP\_GIT fichier.txt note.py pays.txt test.py hamidou@DESKTOP-UIQRHPC:~$ cd /home/hamidou/ hamidou@DESKTOP-UIQRHPC:~$ ls TP\_GIT fichier.txt note.py pays.txt test.py hamidou@DESKTOP-UIQRHPC:~$ cd TP\_GIT/   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls -a . .. .git   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git status s On branch master No commits yet nothing to commit (create/copy files and use "git add" to track)   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git status On branch master No commits yet nothing to commit (create/copy files and use "git add" to track)   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git add Nothing specified, nothing added. Maybe you wanted to say 'git add .'?   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git add -help usage: git add [<options>] [--] <pathspec>... -n, --dry-run dry run -v, --verbose be verbose -i, --interactive interactive picking -p, --patch select hunks interactively -e, --edit edit current diff and apply -f, --force allow adding otherwise ignored files -u, --update update tracked files --renormalize renormalize EOL of tracked files (implies -u) -N, --intent-to-add record only the fact that the path will be added later -A, --all add changes from all tracked and untracked files --ignore-removal ignore paths removed in the working tree (same as --no-all) --refresh don't add, only refresh the index --ignore-errors just skip files which cannot be added because of errors --ignore-missing check if - even missing - files are ignored in dry run --chmod <(+/-)x> override the executable bit of the listed files   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ touch test-tp   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls test-tp   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git add test-tp   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git status On branch master No commits yet Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: test-tp   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ vim test   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git status On branch master No commits yet Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: test-tp Untracked files: (use "git add <file>..." to include in what will be committed) test   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git branch   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git branch -r   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ vim README.md   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git status On branch master No commits yet Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: test-tp Untracked files: (use "git add <file>..." to include in what will be committed) README.md test   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git checkout -b SUP Switched to a new branch 'SUP'   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git branch   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git add README.md   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git sta stage stash status   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git status On branch SUP No commits yet Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: README.md new file: test-tp Untracked files: (use "git add <file>..." to include in what will be committed) test   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git commit -m "Add README.md" [SUP (root-commit) df45418] Add README.md 2 files changed, 1 insertion(+) create mode 100644 README.md create mode 100644 test-tp   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git log commit df454184be8c46792cb0d03eec5215cd87289bba (HEAD -> SUP) Author: Hamidou <hamidou.mary@supdevinci-edu.fr> Date: Mon Sep 28 15:12:42 2020 +0200 Add README.md   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git branch \* SUP   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ vim README.md   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git diff HEAD diff --git a/README.md b/README.md index 09cb374..c9fa77f 100644 --- a/README.md +++ b/README.md @@ -1 +1,3 @@ SUP DE VINCI + +MODIFICATION 1   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git add README.md   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git reset README.md Unstaged changes after reset: M README.md   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git diff HEADd diff --git a/README.md b/README.md index 09cb374..c9fa77f 100644 --- a/README.md +++ b/README.md @@ -1 +1,3 @@ SUP DE VINCI + +MODIFICATION 1   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git diff HEAD diff --git a/README.md b/README.md index 09cb374..c9fa77f 100644 --- a/README.md +++ b/README.md @@ -1 +1,3 @@ SUP DE VINCI + +MODIFICATION 1   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git reset README.md -harde usage: git reset [--mixed | --soft | --hard | --merge | --keep] [-q] [<commit>] or: git reset [-q] [<tree-ish>] [--] <paths>... or: git reset --patch [<tree-ish>] [--] [<paths>...] -q, --quiet be quiet, only report errors --mixed reset HEAD and index --soft reset only HEAD --hard reset HEAD, index and working tree --merge reset HEAD, index and working tree --keep reset HEAD but keep local changes --recurse-submodules[=<reset>] control recursive updating of submodules -p, --patch select hunks interactively -N, --intent-to-add record only the fact that removed paths will be added later   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git reset README.md -hard error: did you mean `--hard` (with two dashes ?)   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git reset README.md --hard fatal: Cannot do hard reset with paths.   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git reset --hard README.md fatal: Cannot do hard reset with paths.   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ssh-keygen -t ed25519 Generating public/private ed25519 key pair. Enter file in which to save the key (/home/hamidou/.ssh/id\_ed25519): ssh Enter passphrase (empty for no passphrase): Enter same passphrase again: Your identification has been saved in ssh. Your public key has been saved in ssh.pub. The key fingerprint is: SHA256:zmKUOizzjEx4e05mZ72QSoAM+UQzh4MIBC4GQWk4edQ hamidou@DESKTOP-UIQRHPC The key's randomart image is: +--[ED25519 256]--+ |%B\*o. | |O=++E | |==.. | |=+ . | |..o o S | | . o o = | |. = O B + | | + %.\* o . | | +o= . | +----[SHA256]-----+   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls README.md ssh ssh.pub test test-tp   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ pwd /home/hamidou/TP\_GIT   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ cat ssh -----BEGIN OPENSSH PRIVATE KEY----- b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW QyNTUxOQAAACCpHoRyxT9aGtLMS6Fy3eEFBlTerbjbFNaETYmzrbjIeQAAAKC9sYpDvbGK QwAAAAtzc2gtZWQyNTUxOQAAACCpHoRyxT9aGtLMS6Fy3eEFBlTerbjbFNaETYmzrbjIeQ AAAEDb3DnvFVM6JxFabAjhwXLvM4fUdetMatidH55D35PeF6kehHLFP1oa0sxLoXLd4QUG VN6tuNsU1oRNibOtuMh5AAAAF2hhbWlkb3VAREVTS1RPUC1VSVFSSFBDAQIDBAUG -----END OPENSSH PRIVATE KEY-----   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ cat ssh.pub ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKkehHLFP1oa0sxLoXLd4QUGVN6tuNsU1oRNibOtuMh5 hamidou@DESKTOP-UIQRHPC   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git config --global user.name "Hamidou MARY"   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git config --global user.email "hamidou.mary@supdevinci-edu.fr"   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push -u origin master error: src refspec master does not match any. error: failed to push some refs to 'origin'   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git remote add origin git@gitlab.com:hamidou.mary/tp2.git   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git add .   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git commit -m "Initial commit" [SUP 0852a81] Initial commit 4 files changed, 11 insertions(+) create mode 100644 ssh create mode 100644 ssh.pub create mode 100644 test   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push -u origin master error: src refspec master does not match any. error: failed to push some refs to 'git@gitlab.com:hamidou.mary/tp2.git'   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git remote origin   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git remote -v origin git@gitlab.com:hamidou.mary/tp2.git (fetch) origin git@gitlab.com:hamidou.mary/tp2.git (push)   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push fatal: The current branch SUP has no upstream branch. To push the current branch and set the remote as upstream, use git push --set-upstream origin SUP   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push --set-upstream origin SUP The authenticity of host 'gitlab.com (172.65.251.78)' can't be established. ECDSA key fingerprint is SHA256:HbW3g8zUjNSksFbqTiUWPWg2Bq1x8xdGUrliXFzSnUw. Are you sure you want to continue connecting (yes/no)? yes Warning: Permanently added 'gitlab.com,172.65.251.78' (ECDSA) to the list of known hosts. git@gitlab.com: Permission denied (publickey). fatal: Could not read from remote repository. Please make sure you have the correct access rights and the repository exists.   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ Git checkout -b DEV Command 'Git' not found, did you mean: command 'vit' from deb vit command 'nit' from deb python-nevow command 'wit' from deb wit command 'git' from deb git Try: sudo apt install <deb name>   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git checkout -b DEV Switched to a new branch 'DEV'   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls README.md ssh ssh.pub test test-tp   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git branch \* DEV SUP   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git checkout -b SUP fatal: A branch named 'SUP' already exists.   
   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git branch -b SUP git: 'branch-b' is not a git command. See 'git --help'.   
   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ vim README.md   
   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git remote add origin git@gitlab.com:hamidou.mary/tp2.git fatal: remote origin already exists.   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls README.md ssh ssh.pub test test-tp   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls /home/hamidou/ TP\_GIT fichier.txt note.py pays.txt test.py   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ mv ssh /home/hamidou/.ssh   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls README.md ssh.pub test test-tp   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ mv ssh.pub /home/hamidou/.ssh   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls /home/hamidou/.ssh/ known\_hosts ssh ssh.pub   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push -u origin master error: src refspec master does not match any. error: failed to push some refs to 'git@gitlab.com:hamidou.mary/tp2.git'   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git remote add origin git@gitlab.com:hamidou.mary/tp2.git fatal: remote origin already exists.   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git add .   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git commit -m "Initial commit" [DEV 06f3d08] Initial commit 2 files changed, 8 deletions(-) delete mode 100644 ssh delete mode 100644 ssh.pub   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push -u origin master error: src refspec master does not match any. error: failed to push some refs to 'git@gitlab.com:hamidou.mary/tp2.git'   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git checkout SUP Switched to branch 'SUP'   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push --set -upstream origin SUP error: unknown switch `p' usage: git push [<options>] [<repository> [<refspec>...]] -v, --verbose be more verbose -q, --quiet be more quiet --repo <repository> repository --all push all refs --mirror mirror all refs -d, --delete delete refs --tags push tags (can't be used with --all or --mirror) -n, --dry-run dry run --porcelain machine-readable output -f, --force force updates --force-with-lease[=<refname>:<expect>] require old value of ref to be at this value --recurse-submodules[=<check|on-demand|no>] control recursive pushing of submodules --thin use thin pack --receive-pack <receive-pack> receive pack program --exec <receive-pack> receive pack program -u, --set-upstream set upstream for git pull/status --progress force progress reporting --prune prune locally removed refs --no-verify bypass pre-push hook --follow-tags push missing but relevant tags --signed[=<yes|no|if-asked>] GPG sign the push --atomic request atomic transaction on remote side -o, --push-option <server-specific> option to transmit -4, --ipv4 use IPv4 addresses only -6, --ipv6 use IPv6 addresses only   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push --set-upstream origin SUP git@gitlab.com: Permission denied (publickey). fatal: Could not read from remote repository. Please make sure you have the correct access rights and the repository exists.   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push fatal: The current branch SUP has no upstream branch. To push the current branch and set the remote as upstream, use git push --set-upstream origin SUP   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push --set-upstream origin SUP git@gitlab.com: Permission denied (publickey). fatal: Could not read from remote repository. Please make sure you have the correct access rights and the repository exists.   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls /home/hamidou/ TP\_GIT fichier.txt note.py pays.txt test.py   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls /home/hamidou/.ssh/ known\_hosts ssh ssh.pub   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ cat /home/hamidou/.ssh/ssh.pub ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKkehHLFP1oa0sxLoXLd4QUGVN6tuNsU1oRNibOtuMh5 hamidou@DESKTOP-UIQRHPC   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ssh-add /home/hamidou/.ssh/ssh.pub Could not open a connection to your authentication agent.   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push -u origin master error: src refspec master does not match any. error: failed to push some refs to 'git@gitlab.com:hamidou.mary/tp2.git'   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git remote -v origin git@gitlab.com:hamidou.mary/tp2.git (fetch) origin git@gitlab.com:hamidou.mary/tp2.git (push)   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ssh-keygen -t ed25519 Generating public/private ed25519 key pair. Enter file in which to save the key (/home/hamidou/.ssh/id\_ed25519): Enter passphrase (empty for no passphrase): Enter same passphrase again: Your identification has been saved in /home/hamidou/.ssh/id\_ed25519. Your public key has been saved in /home/hamidou/.ssh/id\_ed25519.pub. The key fingerprint is: SHA256:H/hsMSWpUdC4xiN0nnf+jtBhdH9HLINwTYTeZVOgukU hamidou@DESKTOP-UIQRHPC The key's randomart image is: +--[ED25519 256]--+ | .+o .=+.o| | . o..+.o.oo| | . +.oo.E.+o+| | . B+.\*o..= | | oSo=o+ +| | + O.. o| | B .. | | . . .. | | ... | +----[SHA256]-----+   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ cat /home/hamidou/.ssh/ssh.pub ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKkehHLFP1oa0sxLoXLd4QUGVN6tuNsU1oRNibOtuMh5 hamidou@DESKTOP-UIQRHPC   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ cat /home/hamidou/.ssh/ cat: /home/hamidou/.ssh/: Is a directory   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ ls /home/hamidou/.ssh/ id\_ed25519 id\_ed25519.pub known\_hosts ssh ssh.pub   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ cat /home/hamidou/.ssh/id\_ed25519 -----BEGIN OPENSSH PRIVATE KEY----- b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW QyNTUxOQAAACDmHxZIGE7Xn5EvKTb/m4yg1U5jPWPVgOiBt2aNY9rjWgAAAKA9h924PYfd uAAAAAtzc2gtZWQyNTUxOQAAACDmHxZIGE7Xn5EvKTb/m4yg1U5jPWPVgOiBt2aNY9rjWg AAAECvKKdjDPTqWx8V7BAjNWyf18kOrzPQ2DOJUAKB7KqWTOYfFkgYTtefkS8pNv+bjKDV TmM9Y9WA6IG3Zo1j2uNaAAAAF2hhbWlkb3VAREVTS1RPUC1VSVFSSFBDAQIDBAUG -----END OPENSSH PRIVATE KEY-----   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ cat /home/hamidou/.ssh/id\_ed25519.pub ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOYfFkgYTtefkS8pNv+bjKDVTmM9Y9WA6IG3Zo1j2uNa hamidou@DESKTOP-UIQRHPC   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push -u origin master error: src refspec master does not match any. error: failed to push some refs to 'git@gitlab.com:hamidou.mary/tp2.git'   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push -u origin master error: src refspec master does not match any. error: failed to push some refs to 'git@gitlab.com:hamidou.mary/tp2.git'   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push fatal: The current branch SUP has no upstream branch. To push the current branch and set the remote as upstream, use git push --set-upstream origin SUP   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git push --set-upstream origin SUP Counting objects: 10, done. Delta compression using up to 4 threads. Compressing objects: 100% (6/6), done. Writing objects: 100% (10/10), 1.04 KiB | 213.00 KiB/s, done. Total 10 (delta 0), reused 0 (delta 0) To gitlab.com:hamidou.mary/tp2.git \* [new branch] SUP -> SUP Branch 'SUP' set up to track remote branch 'SUP' from 'origin'.   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git fetch remote: Enumerating objects: 5, done. remote: Counting objects: 100% (5/5), done. remote: Compressing objects: 100% (4/4), done. remote: Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 Unpacking objects: 100% (4/4), done. From gitlab.com:hamidou.mary/tp2 0852a81..9994cd7 SUP -> origin/SUP   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git branch DEV \* SUP   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git checkout Your branch is behind 'origin/SUP' by 2 commits, and can be fast-forwarded. (use "git pull" to update your local branch)   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git pull Updating 0852a81..9994cd7 Fast-forward ssh | 7 ------- ssh.pub | 1 - 2 files changed, 8 deletions(-) delete mode 100644 ssh delete mode 100644 ssh.pub   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git branch DEV \* SUP   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git branch -b master error: unknown switch `b' usage: git branch [<options>] [-r | -a] [--merged | --no-merged] or: git branch [<options>] [-l] [-f] <branch-name> [<start-point>] or: git branch [<options>] [-r] (-d | -D) <branch-name>... or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch> or: git branch [<options>] (-c | -C) [<old-branch>] <new-branch> or: git branch [<options>] [-r | -a] [--points-at] or: git branch [<options>] [-r | -a] [--format] Generic options -v, --verbose show hash and subject, give twice for upstream branch -q, --quiet suppress informational messages -t, --track set up tracking mode (see git-pull(1)) -u, --set-upstream-to <upstream> change the upstream info --unset-upstream Unset the upstream info --color[=<when>] use colored output -r, --remotes act on remote-tracking branches --contains <commit> print only branches that contain the commit --no-contains <commit> print only branches that don't contain the commit --abbrev[=<n>] use <n> digits to display SHA-1s Specific git-branch actions: -a, --all list both remote-tracking and local branches -d, --delete delete fully merged branch -D delete branch (even if not merged) -m, --move move/rename a branch and its reflog -M move/rename a branch, even if target exists -c, --copy copy a branch and its reflog -C copy a branch, even if target exists --list list branch names -l, --create-reflog create the branch's reflog --edit-description edit the description for the branch -f, --force force creation, move/rename, deletion --merged <commit> print only branches that are merged --no-merged <commit> print only branches that are not merged --column[=<style>] list branches in columns --sort <key> field name to sort on --points-at <object> print only branches of the object -i, --ignore-case sorting and filtering are case insensitive --format <format> format to use for the output   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ git branch DEV fatal: A branch named 'DEV' already exists.   
 hamidou@DESKTOP-UIQRHPC:~/TP\_GIT$ cd hamidou@DESKTOP-UIQRHPC:~$ ls TP\_GIT fichier.txt note.py pays.txt test.py hamidou@DESKTOP-UIQRHPC:~$ ls TP\_GIT/ README.md test test-tp hamidou@DESKTOP-UIQRHPC:~$ git clone 21442594 fatal: repository '21442594' does not exist hamidou@DESKTOP-UIQRHPC:~$ git clone https://gitlab.com/hamidou.mary/simple-bash-scripts Cloning into 'simple-bash-scripts'... Username for 'https://gitlab.com': hamidou.mary Password for 'https://hamidou.mary@gitlab.com': warning: redirecting to https://gitlab.com/hamidou.mary/simple-bash-scripts.git/ remote: Enumerating objects: 325, done. remote: Counting objects: 100% (325/325), done. remote: Compressing objects: 100% (219/219), done. remote: Total 325 (delta 91), reused 325 (delta 91), pack-reused 0 Receiving objects: 100% (325/325), 65.67 KiB | 2.43 MiB/s, done. Resolving deltas: 100% (91/91), done. hamidou@DESKTOP-UIQRHPC:~$ ls TP\_GIT fichier.txt note.py pays.txt simple-bash-scripts test.py hamidou@DESKTOP-UIQRHPC:~$ cd simple-bash-scripts/   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ ls README.md scripts   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ cd scripts/   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ ls addition.sh disk-space.sh multiplication.sh table.sh affect.sh division.sh pomodoro.sh test-file.sh archive-and-encrypt.sh encrypt.sh prime.sh thumbnail.sh armstrong.sh evenodd.sh process.sh traps.sh binary2decimal.sh factorial.sh random-emoji.sh up.sh collectnetworkinfo.sh fibonacci.sh randomfile.sh versioncompare.sh color.sh get-temperature.sh rang-random.sh weather.sh convertlowercase.sh hardware\_machine.sh read-menu.sh whereIP.sh count-lines.sh hello-world.sh remotebackup.sh while-menu.sh cpu.sh hextodec.sh server-health.sh while-read.sh dec2hex.sh interactive.sh simplecalc.sh decimal2binary.sh list-dir.sh special-pattern.sh directorysize.sh lowercase.sh substraction.sh   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ vim uptime&date.sh [1] 709 date.sh: command not found [1]+ Stopped vim uptime   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ vim uptime-date.sh   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ chmod +x uptime-date.sh   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ ./uptime-date.sh 10:16:55 up 22:37, 0 users, load average: 0.52, 0.58, 0.59 Tue Sep 29 10:16:55 CEST 2020   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ git checkout -b NEW\_SCRIPTS Switched to a new branch 'NEW\_SCRIPTS'   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ ls addition.sh disk-space.sh multiplication.sh table.sh affect.sh division.sh pomodoro.sh test-file.sh archive-and-encrypt.sh encrypt.sh prime.sh thumbnail.sh armstrong.sh evenodd.sh process.sh traps.sh binary2decimal.sh factorial.sh random-emoji.sh up.sh collectnetworkinfo.sh fibonacci.sh randomfile.sh uptime-date.sh color.sh get-temperature.sh rang-random.sh versioncompare.sh convertlowercase.sh hardware\_machine.sh read-menu.sh weather.sh count-lines.sh hello-world.sh remotebackup.sh whereIP.sh cpu.sh hextodec.sh server-health.sh while-menu.sh dec2hex.sh interactive.sh simplecalc.sh while-read.sh decimal2binary.sh list-dir.sh special-pattern.sh directorysize.sh lowercase.sh substraction.sh   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ git add uptime-date.sh   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ git status On branch NEW\_SCRIPTS Changes to be committed: (use "git reset HEAD <file>..." to unstage) new file: uptime-date.sh   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ git commit -m "Nouveau script pour la mise en place de la uptime et la date" [NEW\_SCRIPTS 0da9f64] Nouveau script pour la mise en place de la uptime et la date 1 file changed, 6 insertions(+) create mode 100755 scripts/uptime-date.sh   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ git branch \* NEW\_SCRIPTS master   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ git push fatal: The current branch NEW\_SCRIPTS has no upstream branch. To push the current branch and set the remote as upstream, use git push --set-upstream origin NEW\_SCRIPTS   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ git push --set-upstream origin NEW\_SCRIPTS Username for 'https://gitlab.com': hamidou.mary Password for 'https://hamidou.mary@gitlab.com': warning: redirecting to https://gitlab.com/hamidou.mary/simple-bash-scripts.git/ Counting objects: 4, done. Delta compression using up to 4 threads. Compressing objects: 100% (3/3), done. Writing objects: 100% (4/4), 425 bytes | 141.00 KiB/s, done. Total 4 (delta 1), reused 0 (delta 0) remote: remote: To create a merge request for NEW\_SCRIPTS, visit: remote: https://gitlab.com/hamidou.mary/simple-bash-scripts/-/merge\_requests/new?merge\_request%5Bsource\_branch%5D=NEW\_SCRIPTS remote: To https://gitlab.com/hamidou.mary/simple-bash-scripts \* [new branch] NEW\_SCRIPTS -> NEW\_SCRIPTS Branch 'NEW\_SCRIPTS' set up to track remote branch 'NEW\_SCRIPTS' from 'origin'.   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ git tag --help   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ ls addition.sh disk-space.sh multiplication.sh table.sh affect.sh division.sh pomodoro.sh test-file.sh archive-and-encrypt.sh encrypt.sh prime.sh thumbnail.sh armstrong.sh evenodd.sh process.sh traps.sh binary2decimal.sh factorial.sh random-emoji.sh up.sh collectnetworkinfo.sh fibonacci.sh randomfile.sh uptime-date.sh color.sh get-temperature.sh rang-random.sh versioncompare.sh convertlowercase.sh hardware\_machine.sh read-menu.sh weather.sh count-lines.sh hello-world.sh remotebackup.sh whereIP.sh cpu.sh hextodec.sh server-health.sh while-menu.sh dec2hex.sh interactive.sh simplecalc.sh while-read.sh decimal2binary.sh list-dir.sh special-pattern.sh directorysize.sh lowercase.sh substraction.sh   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ cd.. cd..: command not found

hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts/scripts$ cd ..

hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ ls README.md scripts   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ vim README.md   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ git add README.md   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ git commit -m "jout text script uptime-date.sh" [NEW\_SCRIPTS 1a95cdc] jout text script uptime-date.sh 1 file changed, 3 insertions(+), 2 deletions(-)   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ git push --set-upstream origin NEW\_SCRIPTS Username for 'https://gitlab.com': hamidou.mary Password for 'https://hamidou.mary@gitlab.com': warning: redirecting to https://gitlab.com/hamidou.mary/simple-bash-scripts.git/ Counting objects: 3, done. Delta compression using up to 4 threads. Compressing objects: 100% (3/3), done. Writing objects: 100% (3/3), 440 bytes | 220.00 KiB/s, done. Total 3 (delta 1), reused 0 (delta 0) remote: remote: To create a merge request for NEW\_SCRIPTS, visit: remote: https://gitlab.com/hamidou.mary/simple-bash-scripts/-/merge\_requests/new?merge\_request%5Bsource\_branch%5D=NEW\_SCRIPTS remote: To https://gitlab.com/hamidou.mary/simple-bash-scripts \* [new branch] NEW\_SCRIPTS -> NEW\_SCRIPTS Branch 'NEW\_SCRIPTS' set up to track remote branch 'NEW\_SCRIPTS' from 'origin'.   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ git pull Username for 'https://gitlab.com': hamidou.mary Password for 'https://hamidou.mary@gitlab.com': warning: redirecting to https://gitlab.com/hamidou.mary/simple-bash-scripts.git/ remote: Enumerating objects: 3, done. remote: Counting objects: 100% (3/3), done. remote: Compressing objects: 100% (3/3), done. remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 Unpacking objects: 100% (3/3), done. From https://gitlab.com/hamidou.mary/simple-bash-scripts 1dbd99a..22dee4c master -> origin/master \* [new tag] 1.1.0 -> 1.1.0 \* [new tag] New\_version\_updatime-date -> New\_version\_updatime-date Already up to date.   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ git pull Username for 'https://gitlab.com': hamidou.mary Password for 'https://hamidou.mary@gitlab.com': warning: redirecting to https://gitlab.com/hamidou.mary/simple-bash-scripts.git/ remote: Enumerating objects: 1, done. remote: Counting objects: 100% (1/1), done. remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 Unpacking objects: 100% (1/1), done. From https://gitlab.com/hamidou.mary/simple-bash-scripts 22dee4c..05067c1 master -> origin/master Your configuration specifies to merge with the ref 'refs/heads/NEW\_SCRIPTS' from the remote, but no such ref was fetched.   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ ssh usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-b bind\_address] [-c cipher\_spec] [-D [bind\_address:]port] [-E log\_file] [-e escape\_char] [-F configfile] [-I pkcs11] [-i identity\_file] [-J [user@]host[:port]] [-L address] [-l login\_name] [-m mac\_spec] [-O ctl\_cmd] [-o option] [-p port] [-Q query\_option] [-R address] [-S ctl\_path] [-W host:port] [-w local\_tun[:remote\_tun]] [user@]hostname [command]   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ git remote -v origin https://gitlab.com/hamidou.mary/simple-bash-scripts (fetch) origin https://gitlab.com/hamidou.mary/simple-bash-scripts (push)   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ git remote remove origin

hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ git remote -v   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ git remote add origin git@gitlab.com:hamidou.mary/simple-bash-scripts.git   
 hamidou@DESKTOP-UIQRHPC:~/simple-bash-scripts$ git remote -v origin git@gitlab.com:hamidou.mary/simple-bash-scripts.git (fetch) origin git@gitlab.com:hamidou.mary/simple-bash-scripts.git (push)