

# Installation de Gitlab

## Installation

- Gitlab évoluant rapidement, il est difficile de maintenir un script d'installation/mise à jour. Il est plus simple pour l'installer de suivre le [tutoriel d'installation pas à pas pour Debian](#).
  - Préalablement, créer une base de données Mysql comme [indiqué sur le site de GitLab](#).
  - L'installation originelle de Gitlab sur Yilgarn a été réalisé en suivant les indications présentes dans le script `gitlab.sh` (voir ci-dessous)
- Pour basculer le site hébergent Gitlab en HTTPS, suivre [les indication concernant l'utilisation de HTTPS](#).

## Procédure de mise à jour

- Pour voir la version installer : `cat /home/git/gitlab/VERSION`
- Pour les **versions majeures** rechercher votre version dans [la liste des procédures de mise à jour](#).
- Pour les **mises à jour mineures** suivre les indication [pour patcher une version](#).

## Vérifier l'installation

- Se placer dans le dossier `/home/git/gitlab` : `cd /home/git/gitlab`
- Lancer les commandes :
  - `sudo -u git -H bundle exec rake gitlab:env:info RAILS_ENV=production`
  - `sudo -u git -H bundle exec rake gitlab:check RAILS_ENV=production`

## Services liées à Gitlab

- `service gitlab-workhorse restart`
- `service gitlab-unicorn restart`
- `service gitlab-sidekiq restart`
- `service redis-server restart` : doit fonctionner pendant l'opération de backup de Gitlab.

## Problèmes

- [Trouble-Shooting-Guide](#)

## Problème pour mettre en place Letsencrypt

- Ne pas modifier le fichier de conf Nginx du site Gitlab pour le support de SSL
  - ajouter cette ligne pour permettre la vérification par Certbot : `location ^~ /.well-known { root /home/git/gitlab/public; }`
- Lancer Certbot en root : `certbot certonly --webroot -w /home/git/gitlab/public -d git.clapas.org`
- Ajouter dans le fichier de conf Nginx du site Gitlab le support de SSL & HTTP2
- Et laisser la ligne pour la vérification par Certbot : `location ^~ /.well-known { root /home/git/gitlab/public; }`

## Problèmes de démarrage des services

- Si le service `gitlab-unicorn` ne veut pas démarrer à cause de ses dépendance, vérifier que le service `redis-server` est bien démarré.
- Si le service `redis-server` ne veut pas démarrer, vérifier que le dossier `/var/run/redis` existe bien avec les bons droits.

## Problème avec le backup

Vérifier que `mysqldump` est accessible. Vérifier la présence d'un lien vers la bonne version de Mysql dans `/usr/local/bin`.

## Problème avec Bundle

- Si le message suivant survient "Some gems seem to be missing from your vendor/cache directory.", essayer la commande suivante, pour réinstaller les paquets en local : `sudo -u git -H bundle`
- Pour remettre à plat Bundle :
  - `gem update --system`
  - `gem update bundler`

## Problème avec le cache de l'interface web

- Désactiver le cache dans le fichier :  
`/home/git/gitlab/config/environments/production.rb`
- Modifier le paramètre suivant : `config.cache_classes = false`

## Procédure d'installation suivie originellement sur Ylgarn :

## gitlab.sh

Procédure en cours de transformation en script :

```
#!/bin/bash
#
# Installation de GitLab
# Copyright : Jean-Pascal MILCENT, 2015
#
# Source :
https://gitlab.com/gitlab-org/gitlab-ce/blob/master/doc/install/installation.md
#
# Log :
# 2015-03-02 : création du script

echo "Enter the root password for GitLab :"
read GITLAB_ROOT_PASSWORD
while $GITLAB_ROOT_PASSWORD = null; do
    echo "Root password for GitLab is mandatory !"
    echo "Please enter the root password for GitLab :"
    read GITLAB_ROOT_PASSWORD
done

echo "Enter the domain name for GitLab :"
read GITLAB_DOMAIN_NAME
while $GITLAB_DOMAIN_NAME = null; do
    echo "Domain name for GitLab is mandatory !"
    echo "Please enter the domain name for GitLab :"
    read GITLAB_DOMAIN_NAME
done

#-----#
# Installation des dépendances
sudo apt-get install -y \
    build-essential \
    zlib1g-dev \
    libyaml-dev \
    libssl-dev \
    libgdbm-dev \
    libreadline-dev \
    libncurses5-dev \
    libffi-dev \
    curl \
    openssh-server \
    redis-server \
    checkinstall \
    libxml2-dev \
    libxslt-dev \
```

```
libcurl4-openssl-dev \
libicu-dev \
logrotate \
python-docutils \
pkg-config \
cmake \
libkrb5-dev

#-----
#-----#
# Install Git (v1.7.10 minimum)
sudo apt-get install -y git-core

# Make sure Git is version 1.7.10 or higher, for example 1.7.12 or 2.0.0
git --version

#-----
#-----#
# Install Ruby (v2.1 minimum) & Bundler

sudo apt-get install \
libruby2.1 \
ruby2.1 \
ruby2.1-dev \
rubygems-integration
# Create a link ruby to ruby2.1
sudo ln -s /usr/bin/ruby2.1 /usr/bin/ruby

# Install Bundler
sudo gem2.1 install bundler --no-ri --no-rdoc

#-----
#-----#
# Create a git user for GitLab:
sudo adduser --disabled-login --gecos 'GitLab' git

#-----
#-----#
# Database

echo 'To install Gitlab database with Mysql, see :
https://gitlab.com/gitlab-org/gitlab-ce/blob/master/doc/install/database\_mysql.md'

#-----
#-----#
# Redis

sudo apt-get install redis-server
```

```
# Configure redis to use sockets
sudo cp /etc/redis/redis.conf /etc/redis/redis.conf.orig

# Disable Redis listening on TCP by setting 'port' to 0
sed 's/^port .*/port 0/' /etc/redis/redis.conf.orig | sudo tee
/etc/redis/redis.conf

# Enable Redis socket for default Debian / Ubuntu path
echo 'unixsocket /var/run/redis/redis.sock' | sudo tee -a
/etc/redis/redis.conf
# Grant permission to the socket to all members of the redis group
echo 'unixsocketperm 770' | sudo tee -a /etc/redis/redis.conf

# Create the directory which contains the socket
sudo mkdir /var/run/redis
sudo chown redis:redis /var/run/redis
sudo chmod 755 /var/run/redis
# Persist the directory which contains the socket, if applicable
if [ -d /etc/tmpfiles.d ]; then
  echo 'd /var/run/redis 0755 redis redis 10d -' | sudo tee -a
/etc/tmpfiles.d/redis.conf
fi

# Activate the changes to redis.conf
sudo systemctl restart redis-server.service

# Add git to the redis group
sudo usermod -aG redis git

#-----#
# GitLab - Installation

# We'll install GitLab into home directory of the user "git"
cd /home/git

# Clone GitLab repository
sudo -u git -H git clone https://gitlab.com/gitlab-org/gitlab-ce.git -b 7-8-
stable gitlab

#-----#
# GitLab - Configuration

# Go to GitLab installation folder
cd /home/git/gitlab

# Copy the example GitLab config
sudo -u git -H cp config/gitlab.yml.example config/gitlab.yml

# Update GitLab config file, follow the directions at top of file
```

```
sudo -u git -H vim config/gitlab.yml

# Make sure GitLab can write to the log/ and tmp/ directories
sudo chown -R git log/
sudo chown -R git tmp/
sudo chmod -R u+rwx,go-w log/
sudo chmod -R u+rwx tmp/

# Create directory for satellites
sudo -u git -H mkdir /home/git/gitlab-satellites
sudo chmod u+rwx,g=rx,o-rwx /home/git/gitlab-satellites

# Make sure GitLab can write to the tmp/pids/ and tmp/sockets/ directories
sudo chmod -R u+rwx tmp/pids/
sudo chmod -R u+rwx tmp/sockets/

# Make sure GitLab can write to the public/uploads/ directory
sudo chmod -R u+rwx public/uploads

# Copy the example Unicorn config
sudo -u git -H cp config/unicorn.rb.example config/unicorn.rb

# Find number of cores
nproc

# Enable cluster mode if you expect to have a high load instance
# Ex. change amount of workers to 3 for 2GB RAM server
# Set the number of workers to at least the number of cores
sudo -u git -H vim config/unicorn.rb

# Copy the example Rack attack config
sudo -u git -H cp config/initializers/rack_attack.rb.example
config/initializers/rack_attack.rb

# Configure Git global settings for git user, useful when editing via web
# Edit user.email according to what is set in gitlab.yml
sudo -u git -H git config --global user.name "GitLab"
sudo -u git -H git config --global user.email "example@example.com"
sudo -u git -H git config --global core.autocrlf input

# Configure Redis connection settings
sudo -u git -H cp config/resque.yml.example config/resque.yml

# Change the Redis socket path if you are not using the default Debian /
Ubuntu configuration
sudo -u git -H vim config/resque.yml

#-----
#-----#
# Configure GitLab DB Settings
```

```
# MySQL only:  
sudo -u git cp config/database.yml.mysql config/database.yml  
  
# MySQL and remote PostgreSQL only:  
# Update username/password in config/database.yml.  
# You only need to adapt the production settings (first part).  
# If you followed the database guide then please do as follows:  
# Change 'secure password' with the value you have given to $password  
# You can keep the double quotes around the password  
sudo -u git -H vim config/database.yml  
  
# PostgreSQL and MySQL:  
# Make config/database.yml readable to git only  
sudo -u git -H chmod o-rwx config/database.yml  
  
#-----#  
# Install Gems  
cd /home/git/gitlab  
sudo -u git -H bundle -j4 install --deployment --without development test  
postgres aws  
  
#-----#  
# Install GitLab Shell  
  
# Run the installation task for gitlab-shell (replace `REDIS_URL` if  
needed):  
sudo -u git -H bundle exec rake gitlab:shell:install[v2.5.4]  
REDIS_URL=unix:/var/run/redis/redis.sock RAILS_ENV=production  
  
# By default, the gitlab-shell config is generated from your main GitLab  
config.  
# You can review (and modify) the gitlab-shell config as follows:  
sudo -u git -H vim /home/git/gitlab-shell/config.yml  
  
#-----#  
# Initialize Database and Activate Advanced Features  
  
# Type 'yes' to create the database tables.  
# When done you see 'Administrator account created:'  
sudo -u git -H bundle exec rake gitlab:setup RAILS_ENV=production  
GITLAB_ROOT_PASSWORD=$GITLAB_ROOT_PASSWORD  
  
#-----#  
# Install Systemd services  
  
cd /home/admin/scripts/services/systemd
```

```
wget -O gitlab-sidekiq.service https://gitlab.com/gitlab-org/gitlab-recipes/raw/master/init/systemd/gitlab-sidekiq.service
wget -O gitlab-unicorn.service https://gitlab.com/gitlab-org/gitlab-recipes/raw/master/init/systemd/gitlab-unicorn.service
# Note :
# - If you installed GitLab in other path than /home/git/gitlab change the service files accordingly.
# - Edit the files and change the names of other services needed by gitlab
# - Check the path of Bundle : /usr/bin/bundle

chmod 750 gitlab-*
sudo ln -s /home/admin/scripts/services/systemd/gitlab-* /etc/systemd/system

#Add redis-server systemd service
ln -s /lib/systemd/system/redis-server.service /etc/systemd/system/redis.service

#Reload systemd:
sudo systemctl daemon-reload

#Start the services:
sudo systemctl start gitlab-sidekiq.service gitlab-unicorn.service

#Enable them to start at boot:
sudo systemctl enable /home/admin/scripts/services/systemd/gitlab-*

#-----
# Setup Logrotate

sudo cp lib/support/logrotate/gitlab /etc/logrotate.d/gitlab

#-----
# Check Application Status

# Check if GitLab and its environment are configured correctly:
sudo -u git -H bundle exec rake gitlab:env:info RAILS_ENV=production

#-----
# Compile Assets
sudo -u git -H bundle exec rake assets:precompile RAILS_ENV=production

#-----
# Start Your GitLab Instance
sudo systemctl gitlab-* start
```

```
#-----
#-----#
# Nginx Site Configuration
sudo cp lib/support/nginx/gitlab /etc/nginx/sites-available/${GITLAB_DOMAIN_NAME}.conf
sudo ln -s /etc/nginx/sites-available/${GITLAB_DOMAIN_NAME}.conf
/etc/nginx/sites-enabled/${GITLAB_DOMAIN_NAME}.conf

# Change YOUR_SERVER_FQDN to the fully-qualified domain name of your host
# serving GitLab.
sudo vim /etc/nginx/sites-available/${GITLAB_DOMAIN_NAME}

# Test Nginx Configuration
sudo nginx -t

# Restart Nginx
sudo systemctl restart nginx.service

#-----
#-----#
# Double-check Application Status
sudo -u git -H bundle exec rake gitlab:check RAILS_ENV=production

#If all items are green, then congratulations on successfully installing
#GitLab!

#-----
#-----#
# Exim4 config for Gitlab

# On Debian 8, edit /etc/exim4/conf.d/main/01_exim4-config_listmacrosdefs
sudo vim /etc/exim4/conf.d/main/01_exim4-config_listmacrosdefs
# Add a new line : extract_addresses_remove_arguments=False
# Update exim4 config files :
update-exim4.conf.template -r
update-exim4.conf
# Check if new line exists in : /etc/exim4/exim4.conf.template and in
#/var/lib/exim4/config.autogenerated

# Or

# Edit /home/git/gitlab/config/application.rb
sudo -u Git - H vim /home/git/gitlab/config/application.rb
# Add a new line : config.action_mailer.sendmail_settings = { :arguments =>
"-i" }

#-----
#-----#
# Gitlab as personnal repository

# Create your personnal account on Gitlab
```

Last update:

2020/02/09      informatique:serveurs:installation-logiciels:gitlab <https://memos.clapas.org/informatique/serveurs/installation-logiciels/gitlab>  
15:06

```
# Then, sign in to Gitlab like root  
# Go to admin area and click on "Settings" menu  
# Disabled "Signup enabled"
```

From:

<https://memos.clapas.org/> - **Memos**



Permanent link:

<https://memos.clapas.org/informatique/serveurs/installation-logiciels/gitlab>

Last update: **2020/02/09 15:06**