

Gitlab

Download location: <https://packages.gitlab.com/gitlab/gitlab-ce>

Archive versions: <https://about.gitlab.com/downloads/archives/>

Configure Gitlab's web server:

<https://gitlab.com/gitlab-org/omnibus-gitlab/blob/master/doc/settings/nginx.md>

Get installed version info

```
gitlab-rake gitlab:env:info
```

See logs:

```
gitlab-ctl tail # all logs
```

```
tail -f /var/log/gitlab/gitlab-rails/production.log
```

```
gitlab-rake gitlab:app:check
```

```
gitlab-rake gitlab --tasks
```

```
gitlab-rails console
```

Restart services without downtime

```
sudo gitlab-ctl hup unicorn  
sudo gitlab-ctl hup puma  
sudo gitlab-ctl restart sidekiq  
sudo gitlab-ctl hup nginx
```

Start / stop / enable / disable

```
gitlab-ctl stop  
gitlab-ctl start  
  
systemctl disable gitlab-runsvdir.service  
systemctl enable gitlab-runsvdir.service
```

Configuration

| </etc/gitlab/gitlab.rb>

```
# will be started on :80 port  
external_url "http://gitlab.example.com"
```

```
external_port "8888"

# will be started on :443 port
# by default certificate file same as hostname is required:
#   /etc/gitlab/ssl/gitlab.example.com.key
#   /etc/gitlab/ssl/gitlab.example.com.crt
external_url "https://gitlab.example.com"
# redirect http to https
nginx['redirect_http_to_https'] = true

external_url "https://gitlab.example.com:2443"
nginx['ssl_certificate'] = "/etc/gitlab/ssl/gitlab.crt"
nginx['ssl_certificate_key'] = "/etc/gitlab/ssl/gitlab.key"

git_data_dir "/mnt/nas/git-data"
```

| [/etc/gitlab/gitlab.rb](#)

```
gitlab_rails['smtp_enable'] = true
gitlab_rails['smtp_address'] = "smtp.gmail.com"
gitlab_rails['smtp_user_name'] = "username"
gitlab_rails['smtp_password'] = "password"
gitlab_rails['smtp_domain'] = "gmail.com"
gitlab_rails['smtp_authentication'] = "login"
gitlab_rails['smtp_enable_starttls_auto'] = true

# If your SMTP server does not like the default 'From:
gitlab@localhost' you
# can change the 'From' with this setting.
gitlab_rails['gitlab_email_from'] = 'gitlab.mydomain.org'
gitlab_rails['gitlab_email_display_name'] = 'MyDomain.org Gitlab'
gitlab_rails['gitlab_email_reply_to'] = 'username@gmail.com'
```

```
sudo gitlab-ctl reconfigure
```

Uninstall

```
# Stop gitlab and remove its supervision process
sudo gitlab-ctl uninstall

# Debian/Ubuntu
sudo dpkg -r gitlab
```

Gitlab 7.4 empty repos issue

It is not possible to restore backup with empty repos.

Issue described here [Restore issue with empty projects](#)

- File to fix `/opt/gitlab/embedded/service/gitlab-rails/lib/backup`
- fix <https://github.com/gitlabhq/gitlabhq/commit/4e65d085b50844469a322d46773de04c16159ce1>

gitlab:check errors

```
Projects have satellites? ...
4/1 ... no
Try fixing it:
sudo -u git -H bundle exec rake gitlab:satellites:create
RAILS_ENV=production
If necessary, remove the tmp/repo_satellites directory ...
... and rerun the above command
For more information see:
doc/raketasks/maintenance.md
Please fix the error above and rerun the checks.
4/2 ... no
```

- Create missing satellites
 - `sudo gitlab-rake gitlab:satellites:create`
 - `sudo gitlab-ctl restart`
 - `sudo gitlab-rake cache:clear`

Ubuntu Precise Gitlab 7.4 update

* Create LXC container with the same version as your working gitlab instance

- Ubuntu LTS 12.04 Precise
- Gitlab 7.4 (`gitlab_7.4.3-omnibus.5.1.0.ci-1_amd64.deb`)
- copy `/etc/gitlab` form your working instance
- execute `gitlab-ctl reconfigure`
- [restore backup](#)

* Update Gitlab to latest version

- Follow official instructions to gitlab repos: [Install a GitLab CE Omnibus package](#)
- Upgrade gitlab version: [Update GitLab Community Edition](#)

* Make backup

* Create LXC container with latest Ubuntu Xenial release

- Follow official instructions to gitlab repos: [Install a GitLab CE Omnibus package](#)
 - edit `/etc/apt/sources.list.d/gitlab_gitlab-ce.list` and change distro to

trusty

- apt-get install gitlab-ce
- ~~gitlab-ctl reconfigure~~
- get /etc/gitlab from previous LXC container
- gitlab-ctl reconfigure to adapt directory structure to new configuration
- get gitlab backup from previous LXC container
- [restore backup](#)

Issues

undefined method `tree' for #<Rugged::Tag

After update gitlab-ce:amd64 12.7.6-ce.0 to 12.8.1-ce.0. Problem browsing repo files when TAG is selected.

```
tail -f /var/log/gitlab/gitlab-rails/production.log
```

```
NoMethodError (undefined method `tree' for
#<Rugged::Tag::Annotation:0x00007f8981373c70>):
  lib/gitlab/git/rugged_impl/tree.rb:79:in `get_tree_entries_from_rugged'
  lib/gitlab/git/rugged_impl/tree.rb:37:in `tree_entries_from_rugged'
  lib/gitlab/git/rugged_impl/tree.rb:26:in
`tree_entries_with_flat_path_from_rugged'
  lib/gitlab/git/rugged_impl/use_rugged.rb:28:in `block in
execute_rugged_call'
  lib/gitlab/gitaly_client/storage_settings.rb:31:in `block in
allow_disk_access'
  lib/gitlab/temporarily_allow.rb:9:in `temporarily_allow'
  lib/gitlab/gitaly_client/storage_settings.rb:31:in `allow_disk_access'
  lib/gitlab/git/rugged_impl/use_rugged.rb:25:in `execute_rugged_call'
  lib/gitlab/git/rugged_impl/tree.rb:19:in `tree_entries'
  lib/gitlab/git/tree.rb:21:in `where'
  app/models/tree.rb:17:in `initialize'
  app/models/repository.rb:673:in `new'
  app/models/repository.rb:673:in `tree'
  app/graphql/resolvers/tree_resolver.rb:23:in `resolve'
  lib/gitlab/graphql/generic_tracing.rb:40:in `with_labkit_tracing'
  lib/gitlab/graphql/generic_tracing.rb:30:in `platform_trace'
  lib/gitlab/graphql/generic_tracing.rb:40:in `with_labkit_tracing'
  lib/gitlab/graphql/generic_tracing.rb:30:in `platform_trace'
  lib/gitlab/graphql/generic_tracing.rb:40:in `with_labkit_tracing'
  lib/gitlab/graphql/generic_tracing.rb:30:in `platform_trace'
  app/graphql/gitlab_schema.rb:39:in `multiplex'
  app/controllers/graphql_controller.rb:41:in `execute_multiplex'
  app/controllers/graphql_controller.rb:19:in `execute'
  app/controllers/application_controller.rb:478:in `set_current_admin'
  lib/gitlab/i18n.rb:55:in `with_locale'
  lib/gitlab/i18n.rb:61:in `with_user_locale'
  app/controllers/application_controller.rb:463:in `set_locale'
```

```

lib/gitlab/application_context.rb:46:in `block in use'
lib/gitlab/application_context.rb:46:in `use'
lib/gitlab/application_context.rb:19:in `with_context'
app/controllers/application_controller.rb:454:in `set_current_context'
lib/gitlab/error_tracking.rb:34:in `with_context'
app/controllers/application_controller.rb:554:in `sentry_context'
lib/gitlab/request_profiler/middleware.rb:17:in `call'
lib/gitlab/middleware/go.rb:20:in `call'
lib/gitlab/etag_caching/middleware.rb:13:in `call'
lib/gitlab/middleware/multipart.rb:117:in `call'
lib/gitlab/middleware/read_only/controller.rb:53:in `call'
lib/gitlab/middleware/read_only.rb:18:in `call'
lib/gitlab/middleware/basic_health_check.rb:25:in `call'
lib/gitlab/middleware/request_context.rb:23:in `call'
config/initializers/fix_local_cache_middleware.rb:9:in `call'
lib/gitlab/metrics/requests Rack middleware.rb:49:in `call'
lib/gitlab/middleware/release_env.rb:12:in `call'

```

/var/opt/gitlab perm

<https://gitlab.com/gitlab-org/omnibus-gitlab/issues/2374>

<https://github.com/gitlabhq/omnibus-gitlab/blob/master/docker/assets/update-permissions>

<https://gitlab.com/gitlab-org/omnibus-gitlab/blob/master/docker/assets/update-permissions>

FQDN issue

ERROR: External URL must include a FQDN

`external_url = 'gitlab.thefallenphoenix.net'`

```
external_url = 'gitlab.thefallenphoenix.net'
```

500 error webhooks

After pressing Test -> Push events, gitlab shows 500 ERROR page.

```

# gitlab-ctl tail

==> /var/log/gitlab/gitlab-rails/production.log <==

URI::InvalidURIError (URI::InvalidURIError):
  lib/gitlab/proxy_http_connection_adapter.rb:14:in `connection'
  app/services/web_hook_service.rb:73:in `make_request'
  app/services/web_hook_service.rb:26:in `execute'
  app/models/hooks/web_hook.rb:10:in `execute'

```

```
app/services/test_hooks/base_service.rb:22:in `block in execute'
```

New security settings were added. To allow remote network operations you should **enable** it **in** admin settings:
Outbound requests --> Allow requests to the **local** network from hooks and services

```
gitlab-rake check
```

[WARNING] Executing 4:Deadline Exceeded

During backup:

```
...
* project/repo ... [WARNING] Executing 4:Deadline Exceeded
Ignoring error on project/repo - Failed to backup repo
[DONE]
...
```

<https://gitlab.com/gitlab-org/gitlab-ce/issues/46721>

<https://gitlab.com/gitlab-org/gitlab-ce/issues/49690>

Fixed https://gitlab.com/gitlab-org/gitlab-ce/merge_requests/20810/diffs

The issue originally refers to the HasLocalBranches RPC which operates under the 'fast timeout' with a default of 10s. That is a very very generous timeout for running git for-each-ref -count=1 refs/heads. The most likely cause for the original issue is some sort of small outage, which is exactly what the timeouts are for.

Introduced already in version 12.4.0 [After upgrading to 12.4.0, timeouts while backing up large repos can back](#)

From:

<https://niziak.spox.org/wiki/> - **niziak.spox.org**

Permanent link:

<https://niziak.spox.org/wiki/git:gitlab>

Last update: **2021/12/10 10:45**

