

# Pages

Support for domains and certificates is disabled. Ask your system's administrator to enable it.

## Enable gitlab pages

Register wildcard domain or mock it on your main DNS server to resolve to your Gitlab server IP:

```
*.doc.gitlab.org
```

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

```
gitlab_pages['enable'] = true
gitlab_pages['inplace_chroot'] = true
pages_external_url "http://doc.gitlab.org"
```

Apply changes:

```
gitlab-ctl reconfigure
gitlab-ctl restart
# or after minitor changes:
gitlab-ctl restart gitlab-pages
```

Log tails:

```
gitlab-ctl tail gitlab-pages
```

## Enable artifact preview

By default artifact created from build are downloadable (PDFs are shown im embedded browser!). There is a notice below each artifact:

```
The source could not be displayed because it is stored as a job artifact.
```

There is possibility to enable Gitlab pages feature to render view of artifact. Feature is implemented in weird way:

- when enabled, there is possibility to preview artifact by redirecting it special link on Gitlab pages server
- on request Gitlab pages server downloads it from Gitlab server and renders it

## Issues

When `inplace_chroot` is used (needed for Docker/LXC where full chroot is not possible because access right problems), 2 issues has to be resolved. More about side effect of using `inplace_chroot`: <https://gitlab.com/gitlab-org/gitlab-pages#caveats>

## No DNS resolution

Provide `resolv.conf` inside chrooted environment:

```
mkdir -p /var/opt/gitlab/gitlab-rails/shared/pages/etc
cp /etc/resolv.conf /var/opt/gitlab/gitlab-rails/shared/pages/etc/
```

NOTE: read side effects of this solution: <https://gitlab.com/gitlab-org/omnibus-gitlab/issues/4243>

Another solution is to force requests to IP address - check solution below  
[provide\\_url\\_to\\_gitlab\\_artifact\\_server](#)

## SSL cert is not trusted

Gitlab pages try to download artifact from main Gitlab server using main URL with https. When main gitlab instance is servered with not trusted certificate, gitlab pages cannot download it. Solution is to provide `cacerts.pem` inside chrooted environment

```
mkdir -p /var/opt/gitlab/gitlab-
rails/shared/pages/opt/gitlab/embedded/ssl/certs/
cp /opt/gitlab/embedded/ssl/certs/cacert.pem /var/opt/gitlab/gitlab-
rails/shared/pages/opt/gitlab/embedded/ssl/certs/
cat /opt/gitlab/embedded/ssl/certs/cacert.pem /usr/local/share/ca-
certificates/my_own_CA.crt > /var/opt/gitlab/gitlab-
rails/shared/pages/opt/gitlab/embedded/ssl/certs/cacert.pem
```

Another solution is to force requests to IP address - check solution below  
[provide\\_url\\_to\\_gitlab\\_artifact\\_server](#)

## Provide URL to gitlab artifact server

IT can only work if there is no redirect from http to https !

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

```
gitlab_pages['artifacts_server'] = true
gitlab_pages['artifacts_server_url'] = 'http://127.0.0.1/api/v4/'
```

# gitlabpages: more mime types

By default gitlab-pages shows only some files inside browser. Most of files (like .log) are server as application/octet-stream. According to Gitlab MR [MR](#) and this [issue](#) gitlab-pages daemon is now using embedded, compiled-in mime database [go-mimedb](#), but it doesn't work.

Digging inside gitlab-pages binary /opt/gitlab/embedded/bin/gitlab-pages shows strings related to default system mime database:

- /etc/mime.types
- /etc/apache/mime.types
- /etc/apache2/mime.types

From:

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

Permanent link:

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

Last update: **2020/06/03 11:17**

