

---

# Gitlab Bulk Checkout

Feb 10, 2019



---

## Contents

---

<b>1</b>	<b>Features</b>	<b>3</b>
<b>2</b>	<b>Using Tox Build Script</b>	<b>5</b>
<b>3</b>	<b>Docker</b>	<b>7</b>
3.1	Build . . . . .	7
3.2	Usage . . . . .	7
<b>4</b>	<b>Gitlab Bulk Checkout Tool</b>	<b>9</b>
4.1	Getting Started . . . . .	9
<b>5</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>



Simple utility written in [Python](#) for the local repositories. In a Time of SaaS, Isac and microservices, it can happen that you need many small repositories at your local Machine for Development, so it didn't make fun to pull so many repositories.



# CHAPTER 1

---

## Features

---

- sort gitlab grops to fs folder
- append a prefix to the repos
- pull changes from origin





## CHAPTER 2

---

### Using Tox Build Script

---

```
source ~/development/lib/virtualenv/tox/bin/activate  
tox
```

The tox script create a installable dist under `./.tox/dist/*.tar.gz`



# CHAPTER 3

---

## Docker

---

For easy usage, the tool can be wrapped with docker, so you don't need a local python env for using.

### 3.1 Build

```
docker build -t nolte/gitlab-bulkcheckout .
```

ssh agent <https://kb.iu.edu/d/aeww>

### 3.2 Usage

```
docker run -it \
  --user=${UID}:${id -g $(whoami)} \
  -v $SSH_AUTH_SOCK:/ssh-agent \
  -e SSH_AUTH_SOCK=/ssh-agent \
  -e GROUPS_MAPPINGS=/app/gitlab_groupMapping.yml \
  -e GITLAB_TOKEN=$(pass /internet/gitlab.com/tokens/management) \
  -e PROJECTS_BASE=/tmp/bulkcheckout \
  -w /tmp/bulkcheckout \
  -v /tmp/bulkcheckout:/tmp/bulkcheckout \
  -v ${HOME}/.ssh:/home/builder/.ssh:ro \
  -v ${PWD}/gitlab_groupMapping.yml:/app/gitlab_groupMapping.yml:ro \
  nolte/gitlab-bulkcheckout -v checkoutbulk
```



---

## Gitlab Bulk Checkout Tool

---

### 4.1 Getting Started

#### 4.1.1 Usage

##### **gitlab\_bulkcheckout**

Console script for gitlab\_bulkcheckout.

```
gitlab_bulkcheckout [OPTIONS] COMMAND [ARGS]...
```

##### Options

**-v, --verbose**

Print debug information

**--debug-file** <debug\_file>

File to be used as a stream for DEBUG logging

##### **checkoutbulk**

Checkout all listed projects from configured groups.

```
gitlab_bulkcheckout checkoutbulk [OPTIONS]
```

##### Options

**-h, --host** <host>

Url from the Gitlab Server, like <https://gitlab.com>

**--private-token** <private\_token>  
Gitlab Private Access Token

**-m, --mapping-config-path** <mapping\_config\_path>  
Bulk Checkout configuration

**-b, --base** <base>  
The Basedirectory for the local checkout

### Environment variables

#### GITLAB\_HOST

Provide a default for `--host`

#### GITLAB\_TOKEN

Provide a default for `--private-token`

#### GROUPS\_MAPPINGS

Provide a default for `--mapping-config-path`

#### PROJECTS\_BASE

Provide a default for `--base`

### version

Display the current version.

```
gitlab_bulkcheckout version [OPTIONS]
```

### Config

Listing 1: example config

```
4564793:
  fs_prefix: "cookiecutter"
  path: templating
```

```
...
[gitlabGroupId]:
  # prefix for the local projects
  fs_prefix: "cookiecutter"
  # local folder for all projects from this group relative to $PROJECTS_BASE
  path: "templating"
...
```

## 4.1.2 Container Usage

For easy and quickly usage you can use the Dockerfile from [Dockerhub](#).

```
docker pull nolte/gitlab-bulkcheckout:latest
```

For wrapping the tool, and checkout to your local FS it is required that you define some additional *Container Run Parameters*.

Listing 2: execute checkout

```
docker run -it \
  --user=${UID}:${(id -g $(whoami))} \
  -v ${SSH_AUTH_SOCK}:/ssh-agent \
  -e SSH_AUTH_SOCK=/ssh-agent \
  -e GROUPS_MAPPINGS=/app/gitlab_groupMapping.yml \
  -e GITLAB_TOKEN=$(pass /internet/gitlab.com/tokens/management) \
  -e PROJECTS_BASE=/tmp/bulkcheckout \
  -w /tmp/bulkcheckout \
  -v /tmp/bulkcheckout:/tmp/bulkcheckout \
  -v ${HOME}/.ssh:/home/builder/.ssh:ro \
  -v ${PWD}/gitlab_groupMapping.yml:/app/gitlab_groupMapping.yml:ro \
  nolte/gitlab-bulkcheckout -v checkoutbulk
```

## Container Run Parameters

- user \${UID}:\${(id -g \$(whoami))}** The Container User will mapped to your user and group from the Host System, see [User](#).
- v \$SSH\_AUTH\_SOCK (optional)** Required for forward the User SSH Agent from the Host System to the container.
- e SSH\_AUTH\_SOCK (optional)** see -v \$SSH\_AUTH\_SOCK
- e GROUPS\_MAPPINGS** The Path to the checkout config (see: [gitlab\\_bulkcheckout](#), [Config Volume](#), [Config](#).)
- e GITLAB\_TOKEN** [Gitlab Access Token](#) for using the [Gitlab API](#) (see: [gitlab\\_bulkcheckout](#)) This example use the Commandline Tool [pass](#) for manage this secret.
- e PROJECTS\_BASE** The Local Checkout Location, this path should be mapped to your Host System, (see: [gitlab\\_bulkcheckout](#)).
- w /tmp/bulkcheckout** The Container [Workingdir](#).
- v /tmp/bulkcheckout:/tmp/bulkcheckout** Some Folder mounted to your Host System, used as checkout Base.
- v \${HOME}/.ssh:/home/builder/.ssh:ro** Mount your local SSH folder, for checkout the repositories over ssh.
- v \${PWD}/gitlab\_groupMapping.yml:/app/gitlab\_groupMapping.yml:ro** The Used config file (see: [Config](#))

Top-level package for project manager.





## CHAPTER 5

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



## g

`gitlab_bulkcheckout`, [11](#)



## Symbols

-debug-file <debug\_file>  
    gitlab\_bulkcheckout command line  
    option, 9

-private-token <private\_token>  
    gitlab\_bulkcheckout-checkoutbulk  
    command line option, 9

-b, -base <base>  
    gitlab\_bulkcheckout-checkoutbulk  
    command line option, 10

-h, -host <host>  
    gitlab\_bulkcheckout-checkoutbulk  
    command line option, 9

-m, -mapping-config-path  
    <mapping\_config\_path>  
    gitlab\_bulkcheckout-checkoutbulk  
    command line option, 10

-v, -verbose  
    gitlab\_bulkcheckout command line  
    option, 9

## G

gitlab\_bulkcheckout (*module*), 11

gitlab\_bulkcheckout command line  
    option  
    -debug-file <debug\_file>, 9  
    -v, -verbose, 9

gitlab\_bulkcheckout-checkoutbulk  
    command line option  
    -private-token <private\_token>, 9  
    -b, -base <base>, 10  
    -h, -host <host>, 9  
    -m, -mapping-config-path  
    <mapping\_config\_path>, 10