

## Lab 2 - Mise en place d'un specific runner

### Objectif :

- Savoir installer, enregistrer et utiliser des runners spécifiques.

1. Tout d'abord, on va installer le serveur Gitlab Runner sous la VM vagrant gitlab-runner.

Suivez les étapes suivantes pour installer GitLab Runner sous Ubuntu20.04:

- Tout d'abord, ajoutez le dépôt officiel de Gitlab :

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

```
vagrant@gitlab-runner:~$ curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh" | sudo bash
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           % Dload  % Upload   Total   Spent    Left   Speed
100 6885  100 6885    0     0  6180      0  0:00:01  0:00:01 --:--:-- 6180
Detected operating system as Ubuntu/focal.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Running apt-get update... done.
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/runner_gitlab-runner.list...done.
Importing packagecloud gpg key... done.
Running apt-get update... done.

The repository is setup! You can now install packages.
vagrant@gitlab-runner:~$
vagrant@gitlab-runner:~$
```

- Installez GitLab Runner :

*sudo apt install gitlab-runner*

```
vagrant@gitlab-runner:~$ sudo apt install gitlab-runner
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  docker-engine
The following NEW packages will be installed:
  gitlab-runner
0 upgraded, 1 newly installed, 0 to remove and 134 not upgraded.
Need to get 488 MB of archives.
After this operation, 533 MB of additional disk space will be used.
Get:1 https://packages.gitlab.com/runner/gitlab-runner/ubuntu focal/main amd64 gitlab-runner amd64 16.6.1 [488 MB]
5% [1 gitlab-runner 28.5 MB/488 MB 6%]
```

- Quelle est la version de Gitlab Runner ?

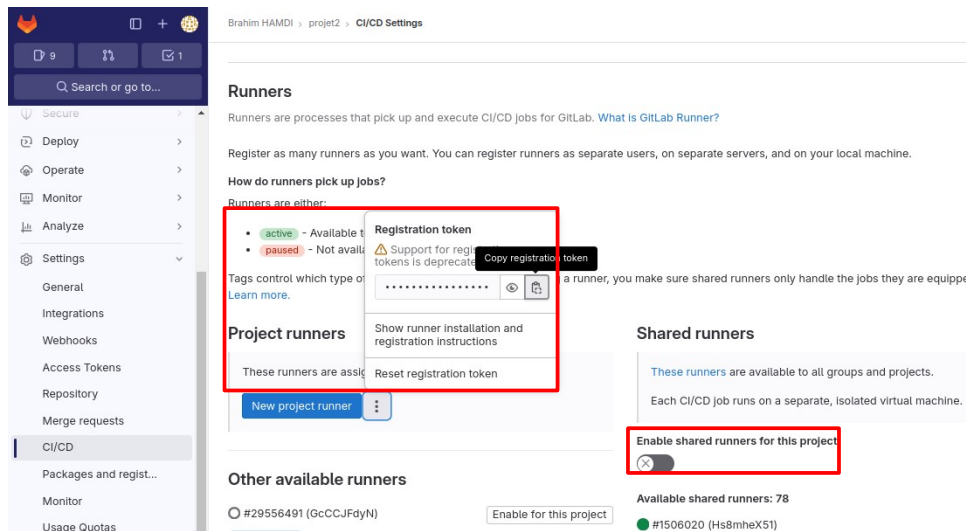
*sudo gitlab-runner -version*

```
vagrant@gitlab-runner:~$ sudo gitlab-runner -version
Version:      16.6.1
Git revision: f5da3c5a
Git branch:   16-6-stable
GO version:   go1.20.10
Built:        2023-11-24T21:11:36+0000
OS/Arch:      linux/amd64
vagrant@gitlab-runner:~$
vagrant@gitlab-runner:~$
```

2. Avant d'enregistrer le Runner dans GitLab, il faut disposer de l'URL du serveur GitLab et du token d'enregistrement. Vous pouvez les retrouver, soit au niveau d'un projet, soit au niveau d'un groupe.

Suivez les étapes suivantes pour inscrire un nouveau runner sur gitlab.com :

- Sur gitlab.com, créez un nouveau projet (ou utilisez le projet crée dans le TP1).
- Copiez le token depuis la section **Settings > CI/CD > Runners**.



Sur la même interface, désactivez les « Shared runners ».

3. Maintenant sur votre serveur gitlab runner, lancez la commande suivante pour inscrire un nouveau runner spécifique:

*gitlab-runner register*

Tapez les informations suivantes :

- *Gitlab instance URL* : <https://gitlab.com/>
- *Registration token* : Le token copié précédemment
- *Execuor* : docker
- *Image* : alpine
- Touche entrée pour les autres informations

```
vagrant@gitlab-runner:~$ gitlab-runner list
Runtime platform                                arch=amd64 os=linux pid=14227 revision=f5da3c5a version=16.6.1
Created missing unique system ID                system_id=s_bb3868b71379
Listing configured runners                      ConfigFile=/home/vagrant/.gitlab-runner/config.toml
vagrant@gitlab-runner:~$
vagrant@gitlab-runner:~$ gitlab-runner register
Runtime platform                                arch=amd64 os=linux pid=14259 revision=f5da3c5a version=16.6.1
WARNING: Running in user-mode.
WARNING: The user-mode requires you to manually start builds processing:
WARNING: $ gitlab-runner run
WARNING: Use sudo for system-mode:
WARNING: $ sudo gitlab-runner...

Enter the GitLab instance URL (for example, https://gitlab.com/):
https://gitlab.com/
Enter the registration token:
GR1348941KAfylnhazrtWw_8hsor
Enter a description for the runner:
[gitlab-runner]: test-runner
Enter tags for the runner (comma-separated):

Enter optional maintenance note for the runner:

WARNING: Support for registration tokens and runner parameters in the 'register' command has been deprecated in GitLab Runner 15.6 and will be replaced with support for authentication
tokens. For more information, see https://docs.gitlab.com/ee/ct/runners/new_creation_workflow
Registering runner... succeeded                  runner=GR1348941KAfylnha
Enter an executor: custom, parallels, shell, docker-autoscaler, instance, kubernetes, docker, docker-windows, ssh, virtualbox, docker-machine:
docker
Enter the default Docker image (for example, ruby:2.7):
alpine
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!

Configuration (with the authentication token) was saved in "/home/vagrant/.gitlab-runner/config.toml"
vagrant@gitlab-runner:~$ gitlab-runner list
Runtime platform                                arch=amd64 os=linux pid=14271 revision=f5da3c5a version=16.6.1
Listing configured runners                      ConfigFile=/home/vagrant/.gitlab-runner/config.toml
test-runner                                    Executor=docker Token=k_rTjcMA2AesdJruSXwy URL=https://gitlab.com/
vagrant@gitlab-runner:~$
```

Il est possible d'enregistrer un Runner plus rapidement et sans interactivité en lançant la commande :

```
gitlab-runner register --non-interactive --url "https://gitlab.com/" --registration-  
token "PROJECT_REGISTRATION_TOKEN" --executor "docker" --docker-image  
alpine:latest --run-untagged="true"
```

4. Vérifiez que le nouveau runner est ajouté au fichier `~/.gitlab-runner/config.toml`. C'est le fichier de configuration des runners spécifiques à l'utilisateur courant (vagrant).

```
vagrant@gitlab-runner:~$ cat .gitlab-runner/config.toml
concurrent = 1
check_interval = 0
shutdown_timeout = 0

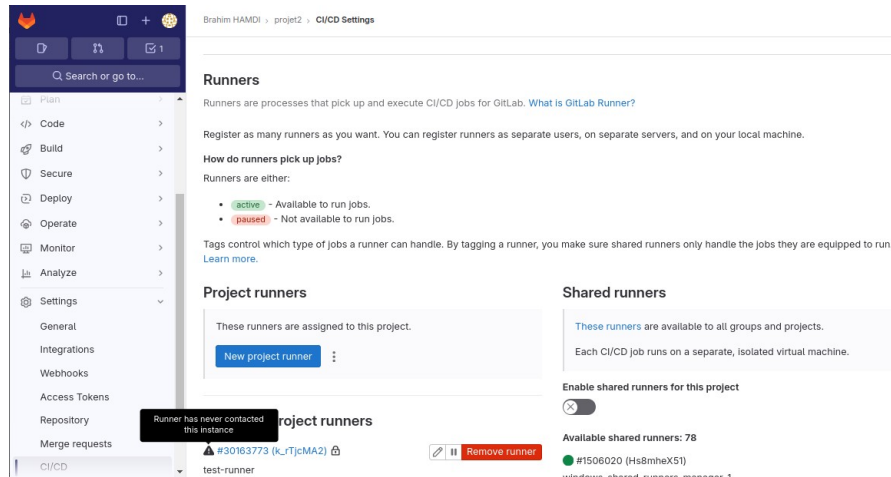
[session_server]
  session_timeout = 1800

[[runners]]
  name = "test-runner"
  url = "https://gitlab.com/"
  id = 30163773
  token = "k_rTjcMA2AesdJruSXwy"
  token_obtained_at = 2023-12-06T11:34:39Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "docker"
  [runners.cache]
    MaxUploadedArchiveSize = 0
  [runners.docker]
    tls_verify = false
    image = "alpine"
    privileged = false
    disable_entrypoint_overwrite = false
    oom_kill_disable = false
    disable_cache = false
    volumes = ["/cache"]
    shm_size = 0
    network_ntu = 0
vagrant@gitlab-runner:~$
vagrant@gitlab-runner:~$
```

Il est possible d'éditer ce fichier directement pour modifier/ajouter des runners de l'utilisateur courant. Notez bien qu'on n'a pas utilisé le mode root pour enregistrer le

nouveau runner (commande `sudo gitlab-runner register`). Le fichier de configuration des runners spécifiques au super-utilisateur root est `/etc/gitlab-runner/config.toml`.

5. Sur gitlab.com, vérifiez dans la même section si le nouveau runner a bien contacté le serveur gitlab.com.



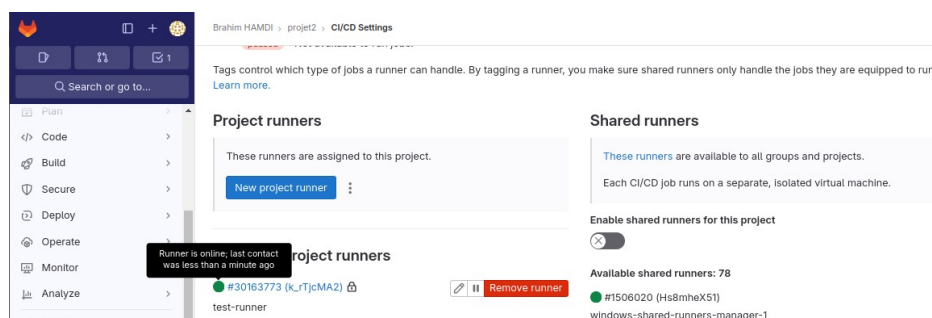
Le triangle indique que le runner n'a jamais contacté le serveur gitlab.com parce qu'il n'a pas démarré.

- Démarrer le nouveau runner avec la commande : `gitlab-runner run` (sans sudo)

```
vagrant@gitlab-runner:~$ gitlab-runner run
Runtime platform arch=amd64 os=linux pid=14307 revision=f5da3c5a version=16.6.1
Starting multi-runner from /home/vagrant/.gitlab-runner/config.toml... builds=0 max_builds=0
WARNING: Running in user-mode.
WARNING: Use sudo for system-mode:
WARNING: $ sudo gitlab-runner...

Configuration loaded builds=0 max_builds=1
listen_address not defined, metrics & debug endpoints disabled builds=0 max_builds=1
[session_server].listen_address not defined, session endpoints disabled builds=0 max_builds=1
Initializing executor providers builds=0 max_builds=1
```

- Vérifiez de nouveau que le runner a bien contacté gitlab.com



Le cercle indique que le runner a contacté le serveur gitlab.com