

TP 4 - Mise en place d'un specific runner

Objectif :

- Savoir installer, enregistrer et utiliser des runners spécifiques.

Installer GitLab Runner

1. Installez GitLab Runner en suivant les étapes suivantes :

- 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

```
brahim@Training:~$ 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  5297      0  0:00:01  0:00:01 --:--:-- 5300
Detected operating system as Ubuntu/jammy.
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.
brahim@Training:~$
```

- Installez GitLab Runner :

sudo apt install gitlab-runner

```
brahim@Training:~$ sudo apt install gitlab-runner
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  bridge-utils dctrl-tools dkms libflashrom1 libftdi1-2 libgsoap-2.8.117 liblzfi net-tools ubuntu-fan
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Paquets suggérés :
  docker-engine
Les NOUVEAUX paquets suivants seront installés :
  gitlab-runner
0 mis à jour, 1 nouvellement installés, 0 à enlever et 8 non mis à jour.
Il est nécessaire de prendre 446 Mo dans les archives.
Après cette opération, 485 Mo d'espace disque supplémentaires seront utilisés.
Réception de :1 https://packages.gitlab.com/runner/gitlab-runner/ubuntu jammy/main amd64 gitlab-runner amd64 15.6.1 [446 MB]
11% [1 gitlab-runner 59,2 MB/446 MB 13%] 545 kB/s 11min 49s[
```

- Vérifiez que le service de gitlab-runner tourne et affichez la version

sudo gitlab-runner status

sudo gitlab-runner -version

```
brahim@Training:~$ sudo gitlab-runner status
Runtime platform                                arch=amd64 os=linux pid=33818 revision=133d7e76 version=15.6.1
gitlab-runner: Service is running
brahim@Training:~$
brahim@Training:~$ sudo gitlab-runner -version
Version:      15.6.1
Git revision: 133d7e76
Git branch:   15-6-stable
GO version:   go1.18.8
Built:        2022-11-24T15:24:48+0000
OS/Arch:      linux/amd64
brahim@Training:~$
```

Sinon, démarrez-le :

sudo gitlab-runner start

2. Maintenant on va accorder des permissions à gitlab-runner, on doit éditer le fichier */etc/visudo*.

- Ouvrir le fichier visudo et ajoutez la ligne suivante pour ajouter les privilèges du super-utilisateur à gitlab-runner :

gitlab-runner ALL=(ALL) NOPASSWD: ALL

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
gitlab-runner ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

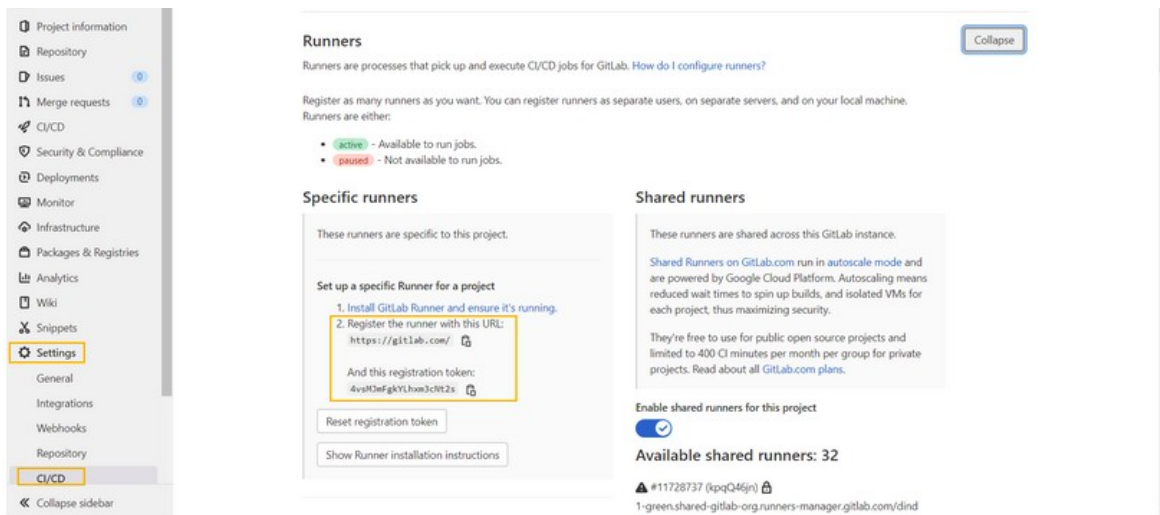
@includedir /etc/sudoers.d
gitlab-runner ALL=(ALL) NOPASSWD: ALL
```

Inscrire et tester un runner spécifique sur GitLab

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.

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

- Créez un nouveau projet sur GitLab.com et copier le token des specific runners.



- 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
- Exeutor : docker
- Image : alpine
- Touche entrée pour les autres informations

```
brahim@Training:~$ gitlab-runner register
Runtime platform arch=amd64 os=linux pid=34184 revision=133d7e76 version=15.6.1
WARNING: The 'register' command has been deprecated in GitLab Runner 15.6 and will be replaced with a 'deploy' command. For more information,
see https://gitlab.com/gitlab-org/gitlab/-/issues/380872
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:
GR1348941s9yhFCwcv355ytLYW49A
Enter a description for the runner:
[Training]:
Enter tags for the runner (comma-separated):

Enter optional maintenance note for the runner:

Registering runner... succeeded runner=GR1348941s9yhFCwcv
Enter an executor: docker-ssh, ssh, virtualbox, docker+machine, docker-ssh+machine, instance, custom, docker, kubernetes, parallels, shell:
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/brahim/.gitlab-runner/config.toml"
brahim@Training:~$
```

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"
```

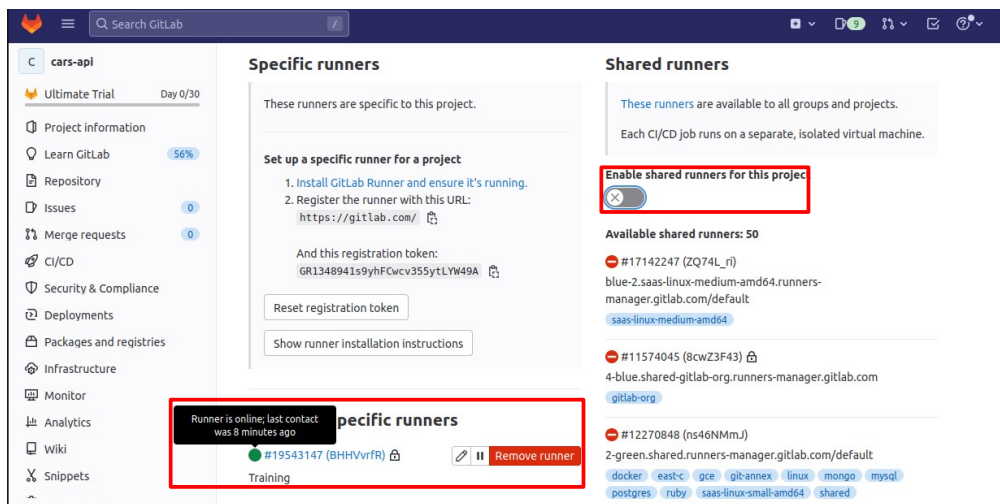
4. On peut vérifier si le runner a été bien enregistré et contacter le serveur GitLab.

- Taper la commande suivante pour afficher la liste des runners actifs :

```
gitlab-runner list
```

```
brahim@Training:~$ gitlab-runner list
Runtime platform arch=amd64 os=linux pid=34707 revision=133d7e76 version=15.6.1
Listing configured runners ConfigFile=/home/brahim/.gitlab-runner/config.toml
Training Executor=docker Token=BHHVvfrFRsYgGMchr5se- URL=https://gitlab.com/
brahim@Training:~$
```

- Sur gitlab.com, vérifiez via la même interface du token du projet si le runner a contacter le serveur. Vous devez aussi désactiver les *shared runners*.

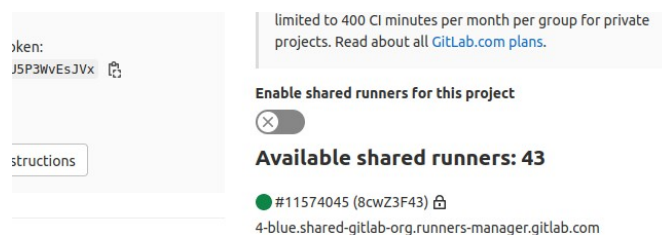


The screenshot shows the GitLab web interface for a project named 'cars-api'. On the left is a sidebar with navigation links. The main content area is divided into two panels. The 'Specific runners' panel shows a list of runners specific to this project, with one runner named 'Training' (ID: #19543147) listed. A red box highlights a status message: 'Runner is online; last contact was 8 minutes ago'. The 'Shared runners' panel shows a toggle for 'Enable shared runners for this project' which is currently disabled. Below this, a list of available shared runners is shown, including 'blue-2.saas-linux-medium-amd64.runners-manager.gitlab.com/default' and 'saas-linux-medium-amd64'.

Utiliser le runner spécifique sur GitLab

Dans cette partie nous allons faire les configurations nécessaires pour pouvoir exécuter les jobs de l'application cars-api par des runners spécifiques créés localement.

5. Pour pouvoir utiliser les specific runners, vous devez tout d'abord désactiver les shared runners.



- Modifiez le fichier de configuration `~/.gitlab-runner/config.toml` de gitlab-runner comme suit :

....

[runners.docker]

tls_verify = true

image = "alpine:latest"

privileged = true

disable_entrypoint_overwrite = false

oom_kill_disable = false

disable_cache = false

volumes = ["/certs/client", "/cache"]