Run GitLab Runner in a container

We can run github runners locally. In this lab, we will register gitlab-runner on `gitlab.com`.

NOTE: Make sure to open new terminal and connect with your remote VM before running docker commands below:

```
ssh fenago@YOUR_VM_DNS_NAME
```

Username/Password: Will be provided by Instructor.

```
root@982e2c2fb78d:~# ssh root@gitlab-ansible-dev.courseware.io
root@gitlab-ansible-dev.courseware.io's password:
Welcome to Ubuntu 22.10 (GNU/Linux 5.19.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Feb 13 23:49:59 UTC 2023

  System load:  0.0703125          Users logged in:          1
  Usage of /:   13.0% of 154.96GB  IPv4 address for docker0: 172.17.0.1
  Memory usage: 15%                IPv4 address for eth0:    143.244.152.105
  Swap usage:   0%                 IPv4 address for eth0:    10.10.0.5
  Processes:    163                IPv4 address for eth1:    10.116.0.2

94 updates can be applied immediately.
68 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Mon Feb 13 23:48:16 2023 from 172.17.0.2
```

In this lab, you can use a configuration container to mount your custom data volume.

Create the Docker volume:

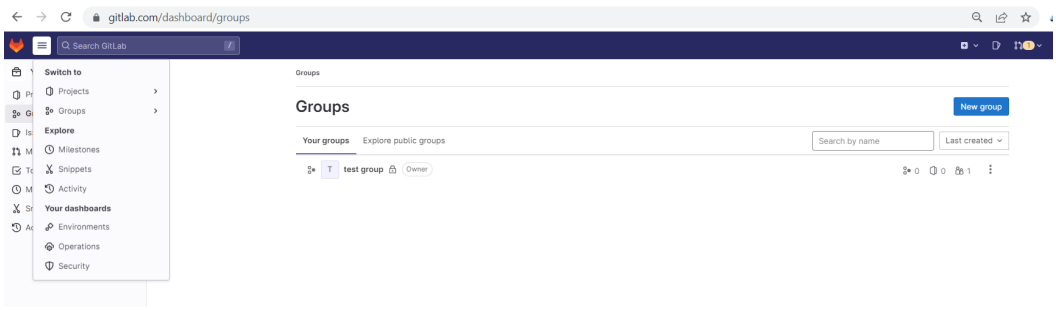
```
docker volume create gitlab-runner-config
```

Start the GitLab Runner container using the volume we just created:

```
docker run -d --name gitlab-runner --restart always \
    -v /var/run/docker.sock:/var/run/docker.sock \
    -v gitlab-runner-config:/etc/gitlab-runner \
    gitlab/gitlab-runner:latest
```

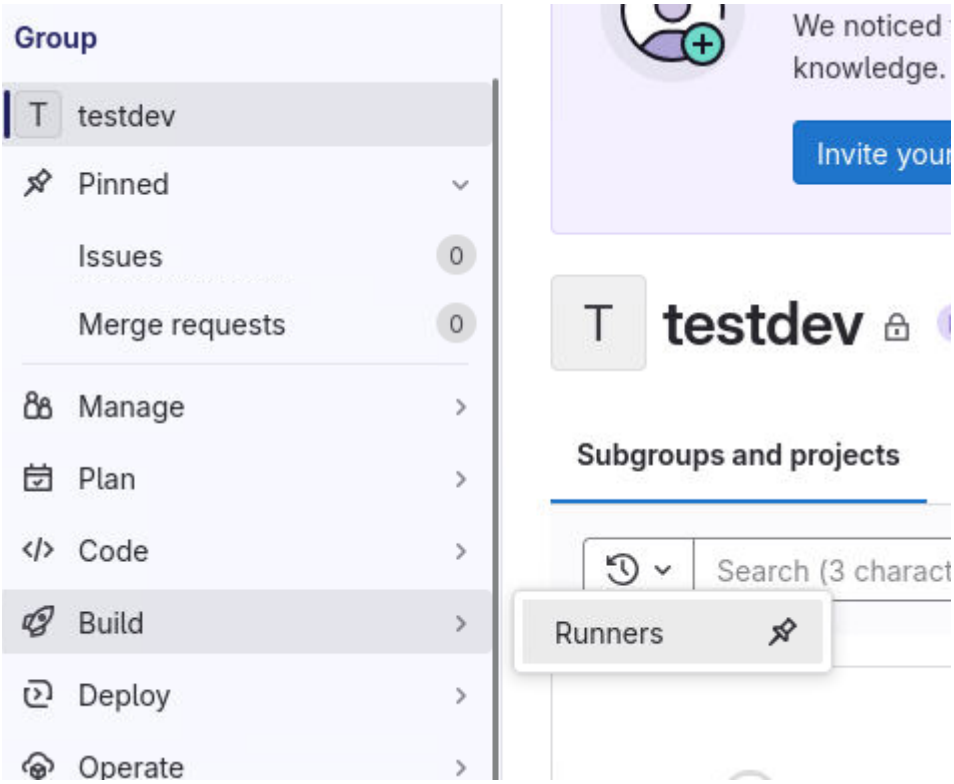
Group Runner Setup

1. On the top bar, select `Main menu` > `Groups` and find your group.

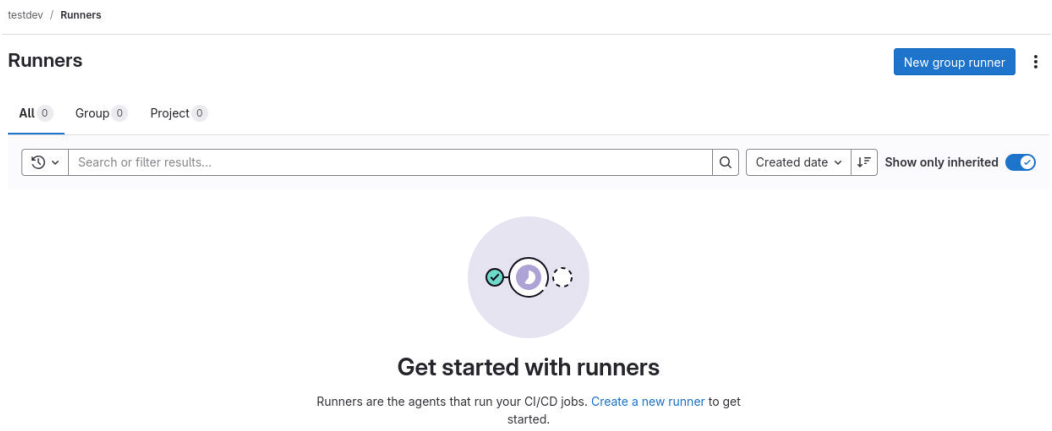


Note: If you don't have one, click `New Group` to create new one.

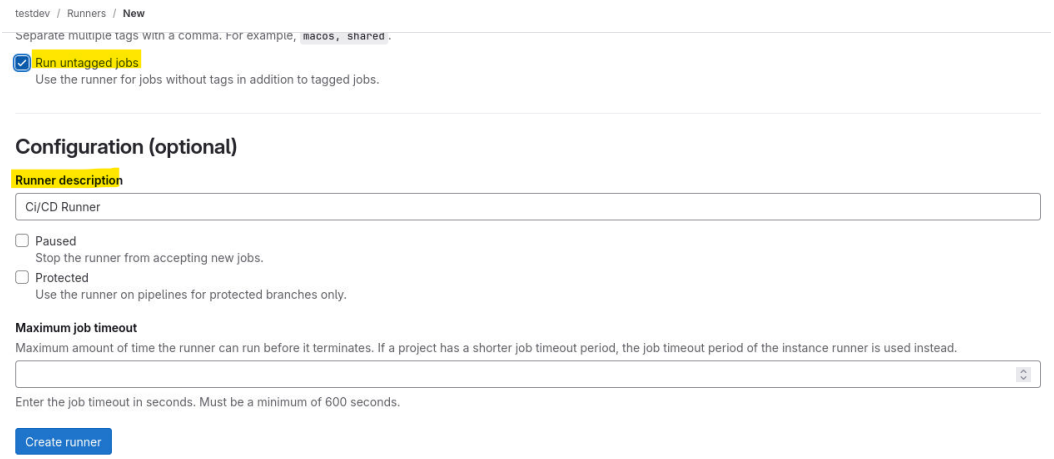
2. On the left sidebar, select `Build` > `Runners`.



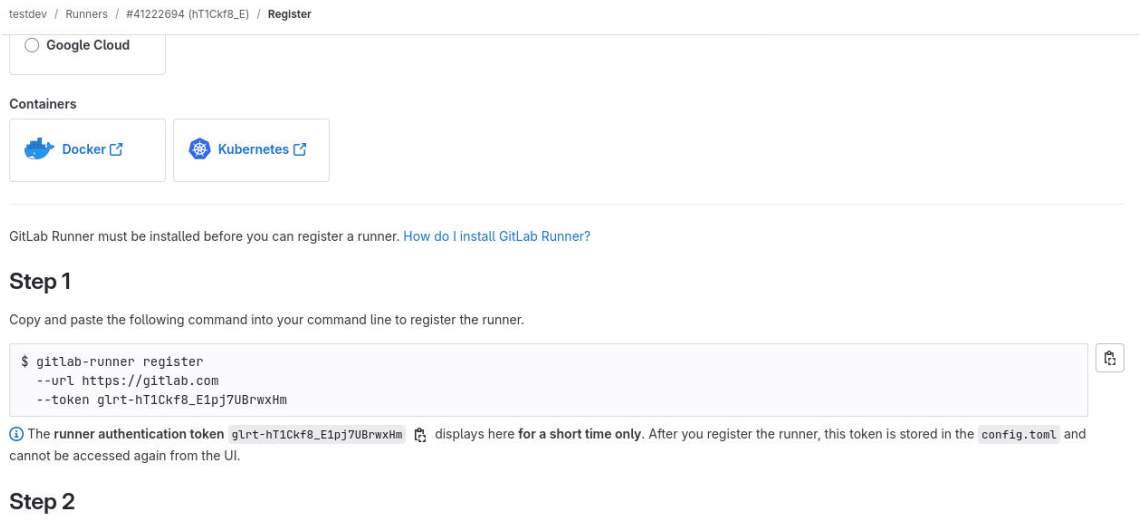
3. In the upper-right corner, select `New group runner`.



4. Check **Run untagged jobs**, add **Runner description** and create `New Runner`.



5. **Important:** Make sure to copy the token from the UI for next step (You will get different new token).



Register a Runner

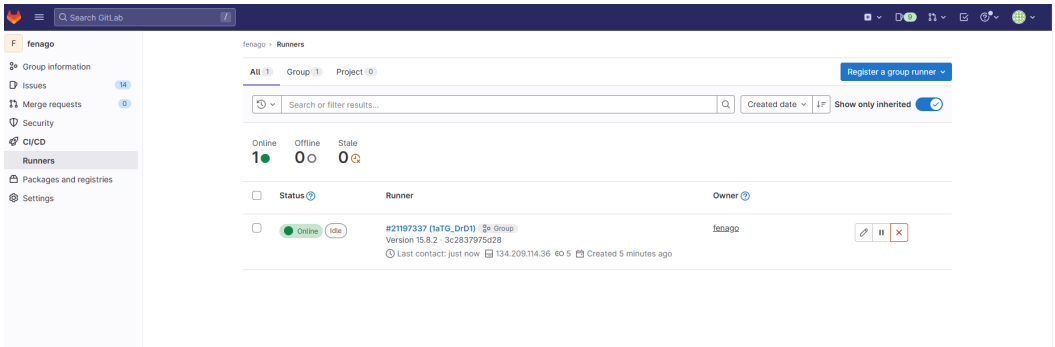
1. To register a runner using a Docker container:

```
docker run --rm -it -v gitlab-runner-config:/etc/gitlab-runner gitlab/gitlab-runner:latest register
```

2. Enter your GitLab instance URL (also known as the gitlab-ci coordinator URL): `https://gitlab.com/`
3. Enter the token you obtained to register the runner.
4. Enter a name for the runner
5. Provide the runner executor: enter **docker**.
6. If you entered docker as your executor, you are asked for the default image to be used for projects that do not define one in .gitlab-ci.yml: enter **ruby:2.7**

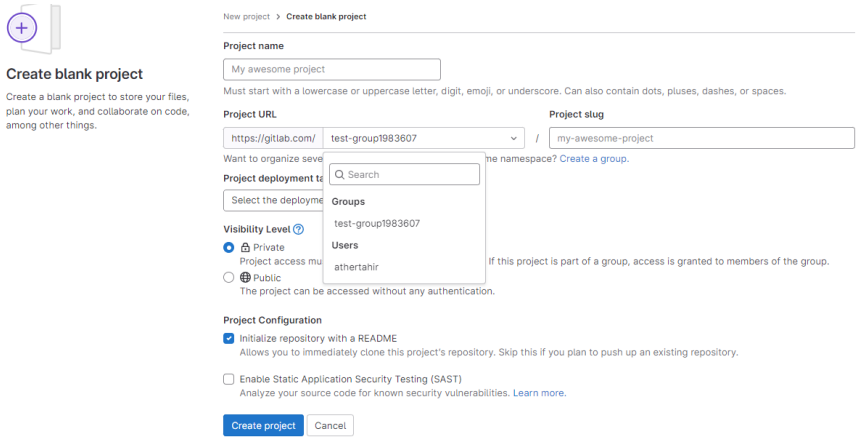
```
ubuntu@ubuntu-novnc:~$ docker run --rm -it -v gitlab-runner-config:/etc/gitlab-r
unner gitlab/gitlab-runner:latest register
Runtime platform                                    arch=amd64 os=linux pid=7 re
vision=66269445 version=17.3.1
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
https://gitlab.com/
Enter the registration token:
glrt-hT1Ckf8_E1pj7UBrwxHm
Verifying runner... is valid                        runner=hT1Ckf8_E
Enter a name for the runner. This is stored only in the local config.toml file:
[b8308774d7a7]:
Enter an executor: docker-windows, docker+machine, instance, shell, parallels, v
irtualbox, docker, custom, ssh, kubernetes, docker-autoscaler:
docker
Enter the default Docker image (for example, ruby:2.7):
ruby:2.7
Runner registered successfully. Feel free to start it, but if it's running alrea
dy the config should be automatically reloaded!

Configuration (with the authentication token) was saved in "/etc/gitlab-runner/c
```

7. Open/Reload the runners webpage, you should see one runner available.



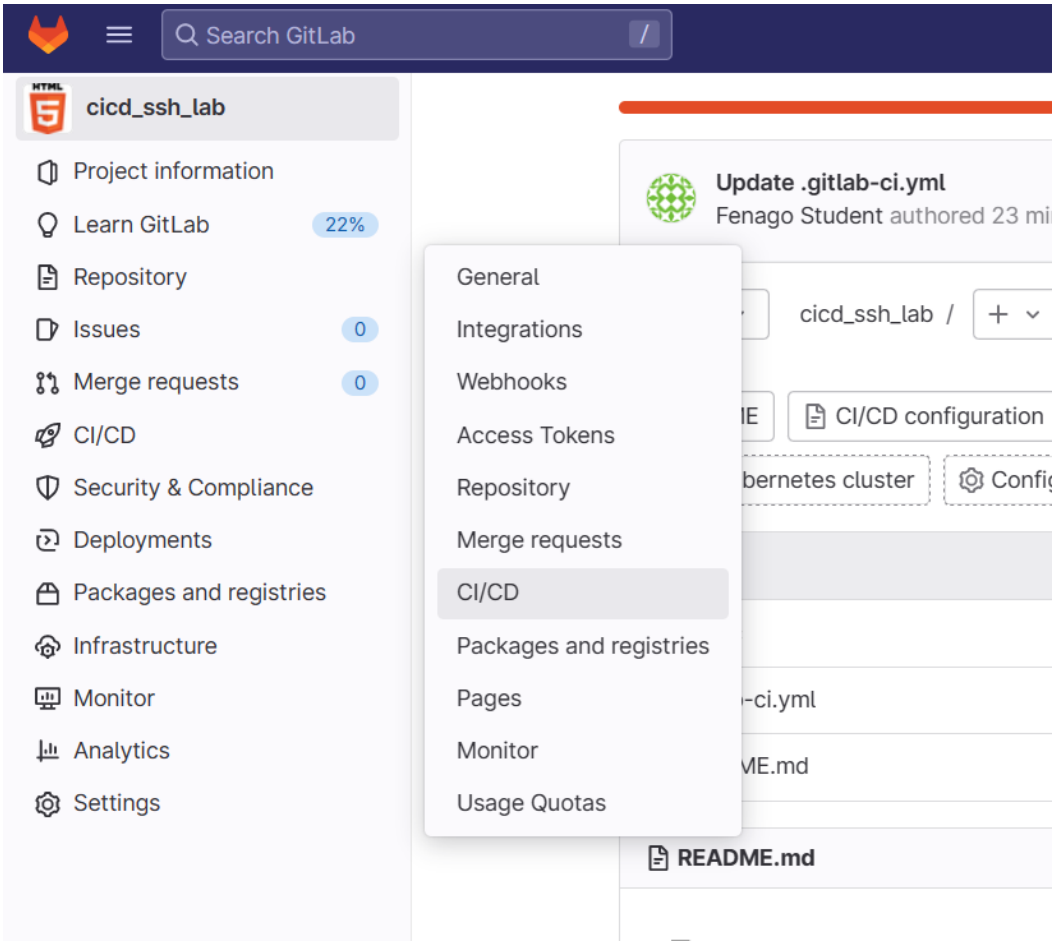
Using Gitlab Runner in CI/CD

1. **Important!** Make sure to create new project inside the group so that you can use the runner:

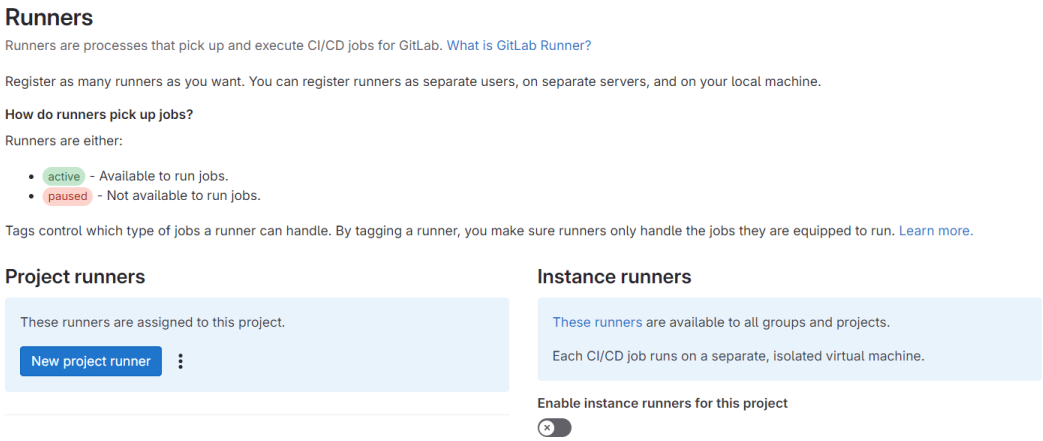


2. Disable instance runners for all the projects that you create. Otherwise, gitlab will fail your pipeline and ask for account verification.

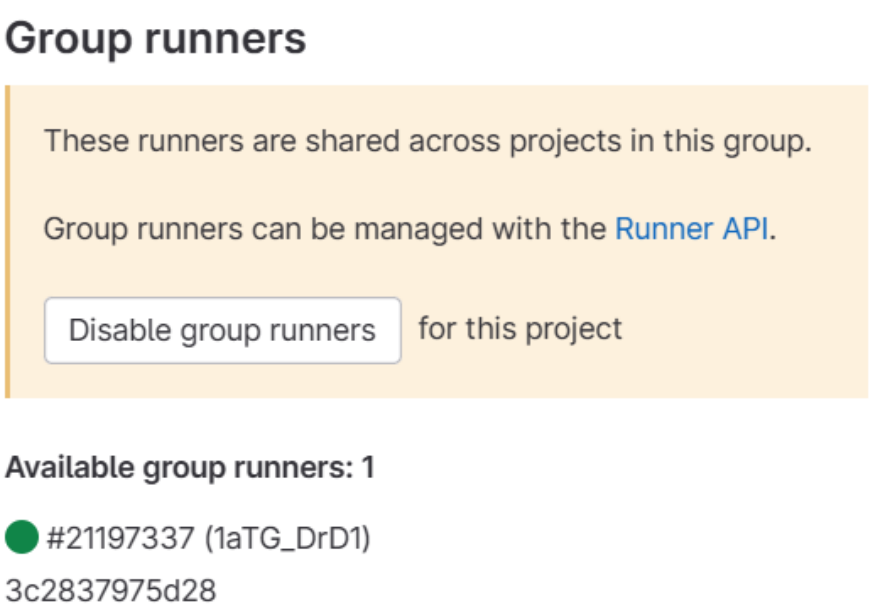
3. On the left sidebar, select `Settings` > `CI/CD`.



4. Expand `Runners` and disable instance runners for this project.



5. Scroll to `Group runners` and confirm one runner is available.



Important! You will need to disable shared runner for all gitlab projects.