

Задание день 1

Этап 1. Подготовка и настройка группы и репозитория

1. На сервере gitlab.com необходимо создать **приватную группу** FIO_devsec2022 (например ivanovii_devsec2022)
Подтверждающие выполнение: ссылка (URL) на группу (прикрепить в анкету)
2. **Добавить проверяющего** MIEM DevSecOps Service (@devsecops_service) в качестве Owner в данную группу – иначе мы не сможем проверить задание!
3. Необходимо создать **в данной группе приватный репозиторий** day1, инициализировать его readme (вы в нем Owner, проверяющий так же)
Подтверждающие выполнение: ссылка (URL) на репозиторий (прикрепить в анкету)
4. Создать ветку develop, сделать ее защищенной (в Settings -> Repository -> Protected branches) (ссылку на ветку прикрепить в анкету)
Подтверждающие выполнение: ссылка (URL) на ветку (прикрепить в анкету)

Вы можете сделать это непосредственно через веб интерфейс gitlab.com.

Если вы будете делать все локально через командную строку git, последовательность добавления ветки при работе в локальном репозитории: создание ветки и переключение на нее (git checkout -b), внесение изменений в readme, фиксирование изменений в локальном репозитории (git add, git commit), заливка новой ветки на сервер (git push --set-upstream origin develop).

5. Меняем права доступа на ветку develop. Делаем ветку develop защищенной (настройка прав: allowed to push – Maintainer, allowed to merge только Maintainer).
6. Меняем права доступа на main ветку (настройка прав: allowed to push – Maintainer, allowed to merge никто).

Результат работы: репозиторий с двумя ветками (master, develop), обе ветки защищенные.

Этап 2. Настройка gitlab runner и регистрация его в качестве group runner в вашей группе

1. По [инструкции](https://docs.gitlab.com/runner/install/docker.html#install-the-docker-image-and-start-the-container) устанавливаем ранер на свой компьютер

```
an.zavyalova@macbook-an ~ % docker volume create gitlab-runner-config
gitlab-runner-config
an.zavyalova@macbook-an ~ % 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
Unable to find image 'gitlab/gitlab-runner:latest' locally
latest: Pulling from gitlab/gitlab-runner
fb0b3276a519: Pull complete
90b6e83fab8a: Pull complete
dcd6fdef935d: Pull complete
Digest: sha256:43f8f8f9968d5ad04d46d960288cdc46e79f52de9e36823260f4f33c68b6a744
Status: Downloaded newer image for gitlab/gitlab-runner:latest
00b1cef545b6f22d9ee885a3c2d165e9f38799964a7c1ef28425dbadb0041453
an.zavyalova@macbook-an ~ % docker run --rm -it -v gitlab-runner-config:/etc/gitlab-runner gitlab/gitlab-runner:latest register
Runtime platform arch=amd64 os=linux pid=8 revision=0d4137b8 version=15.5.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
https://gitlab.com/
Enter the registration token:
Token
Enter a description for the runner:
[79804a0d1f62]: docker runner
Enter tags for the runner (comma-separated):
docker
Enter optional maintenance note for the runner:

Registering runner... succeeded runner=GR1348941nlu6n49b
Enter an executor: custom, docker-ssh, virtualbox, ssh, docker+machine, docker-ssh+machine, instance, kubernetes, docker, parallels, shell:
docker
Enter the default Docker image (for example, ruby:2.7):
alpine:3.16.2
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 "/etc/gitlab-runner/config.toml"
```

Рисунок 1 Запуск в docker с помощью docker volume

2. Открываем в gitlab.com группу, CI/CD -> Runners и используя кнопку “Register a group runner” узнаем групповой токен

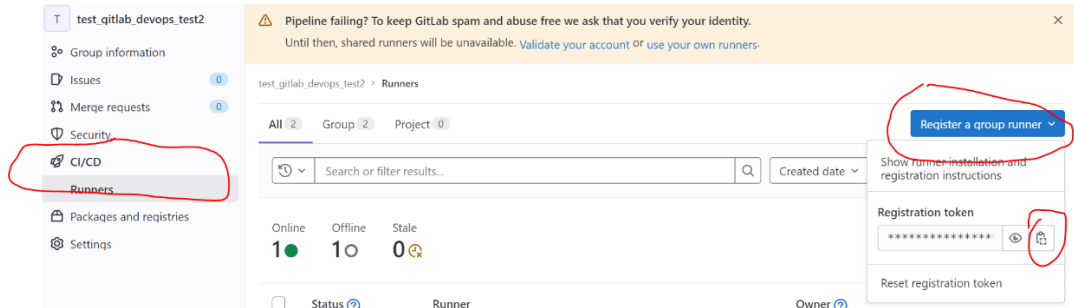


Рисунок 2 Узнать групповой токен для регистрации раннера

3. Регистрируем ранер, используя токен своей группы по инструкции

<https://docs.gitlab.com/runner/register/index.html#docker>

Следуйте инструкциям, регистрируем раннер типа docker.

Подтверждающие выполнение: group runner вашей группы с информацией о новом раннере виден в настройках CI группы.

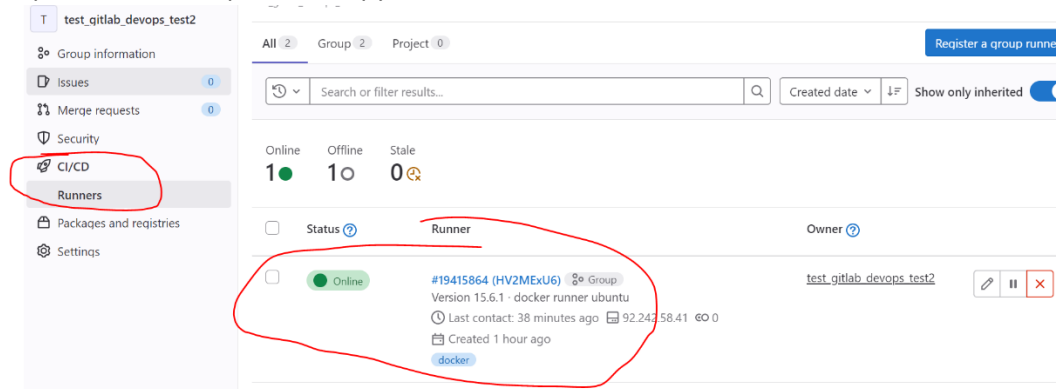


Рисунок 3 Информация о зарегистрированном раннере

4. В настройках ранера группы CI/CD => Runners => Редактировать поставьте галочку «Run unstaged jobs»

DevOps course > Runners > #18751176 (6JFSxpo2) > Edit

Online Group **Runner #18751176** created 8 hours ago

Details

Description

docker runner

Configuration

☐ Paused
Stop the runner from accepting new jobs.

☐ Protected
Use the runner on pipelines for protected branches only.

☒ Run untagged jobs
Use the runner for jobs without tags, in addition to tagged jobs.

Maximum job timeout

Enter the number of seconds. This timeout takes precedence over lower timeouts set for the project.

Tags

Рисунок 4 Run unstaged jobs

5. В настройках группы Settings => CI/CD => Runners необходимо выключить "Enable shared runners for this group"

T test_gitlab_devops_test2

Group information
Issues 0
Merge requests 0
Security
CI/CD
Packages and registries
Settings
General
Integrations
Webhooks
Projects
Repository
CI/CD

Variables store information, like passwords and secret keys, that you can use in job scripts. [Learn more.](#)

Variables can be:

- Protected:** Only exposed to protected branches or protected tags.
- Masked:** Hidden in job logs. Must match masking requirements. [Learn more.](#)

Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. [What is GitLab Runner?](#)

Enable shared runners for this group

☒

Enable shared runners for all projects and subgroups in this group.

Allow projects and subgroups to override the group setting

☒

Allows projects or subgroups in this group to override the global setting.

Рисунок 5 Выключить shared runners

Ответ на задание

Результат выполнения задания загрузите в форму:

<https://forms.yandex.ru/cloud/6384f666f47e730620b8b8ee/>