

Развертывание и настройка GitlabCI

--

Для начала работ вам понадобится:

- *docker*
- *docker-compose*
- *gitlab*

Подробная установка и настройка всех необходимых инструментов описана в методических материалах к предыдущим урокам

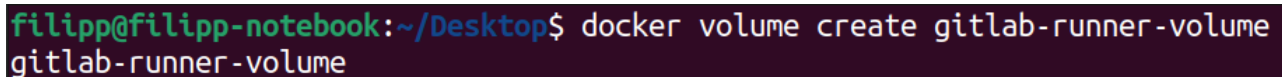
--

Часть 1. Установка gitlab-runner

Поскольку инструмент Gitlab у нас уже установлен, для запуска механизма CI/CD нам необходим проинсталлировать, зарегистрировать и настроить gitlab-runner

Сперва создадим *docker-volume* для gitlab-runner:

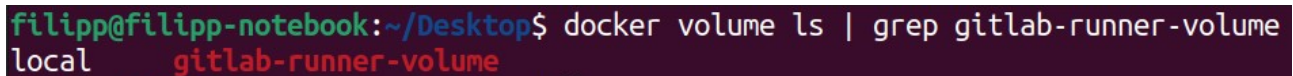
```
$ docker volume create gitlab-runner-volume
```



```
filipp@filipp-notebook:~/Desktop$ docker volume create gitlab-runner-volume
gitlab-runner-volume
```

Вызовем команду просмотра *docker-volumes*:

```
$ docker volume ls | grep gitlab-runner-volume
```



```
filipp@filipp-notebook:~/Desktop$ docker volume ls | grep gitlab-runner-volume
local      gitlab-runner-volume
```

После подготовки *docker-volume* запустим gitlab-runner в *docker-контейнере*:

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

```
filipp@filipp-notebook:~/Desktop$ docker run -d --name gitlab-runner --restart always \
> -v /var/run/docker.sock:/var/run/docker.sock \
> -v gitlab-runner-volume:/etc/gitlab-runner \
> gitlab/gitlab-runner:latest
30cd9ba26fc6d0da9096557eb14bc8a443fa65e64f2cbec9c186a9be4526aaa0
```

Вызовем команду отображения списка запущенных контейнеров:

```
$ docker ps
```

```
filipp@filipp-notebook:~/Desktop$ docker ps
```

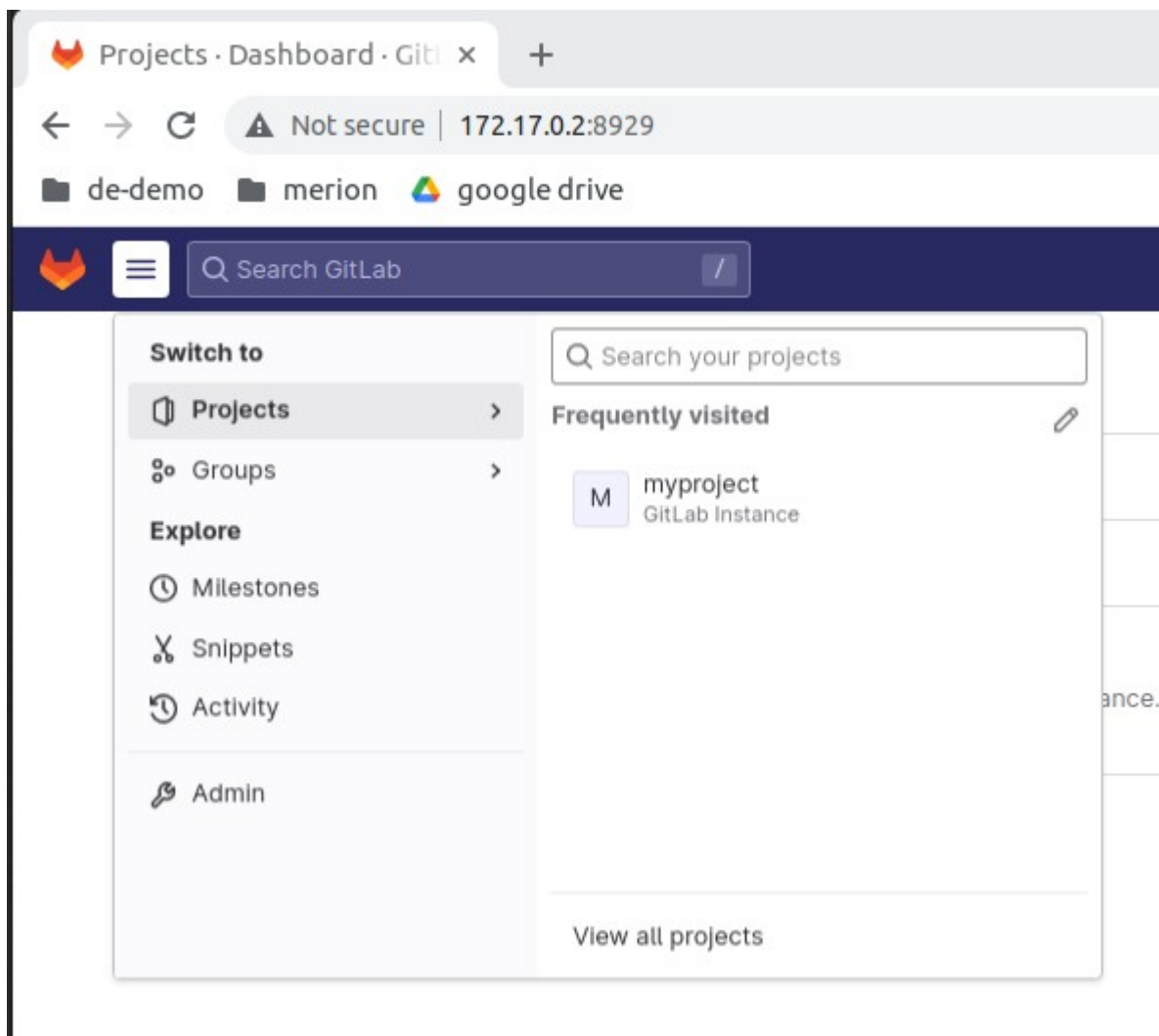
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
30cd9ba26fc6	gitlab/gitlab-runner:latest	"/usr/bin/dumb-init ..."	About a minute ago	Up About a minute

Контейнер с gitlab-runner успешно запущен

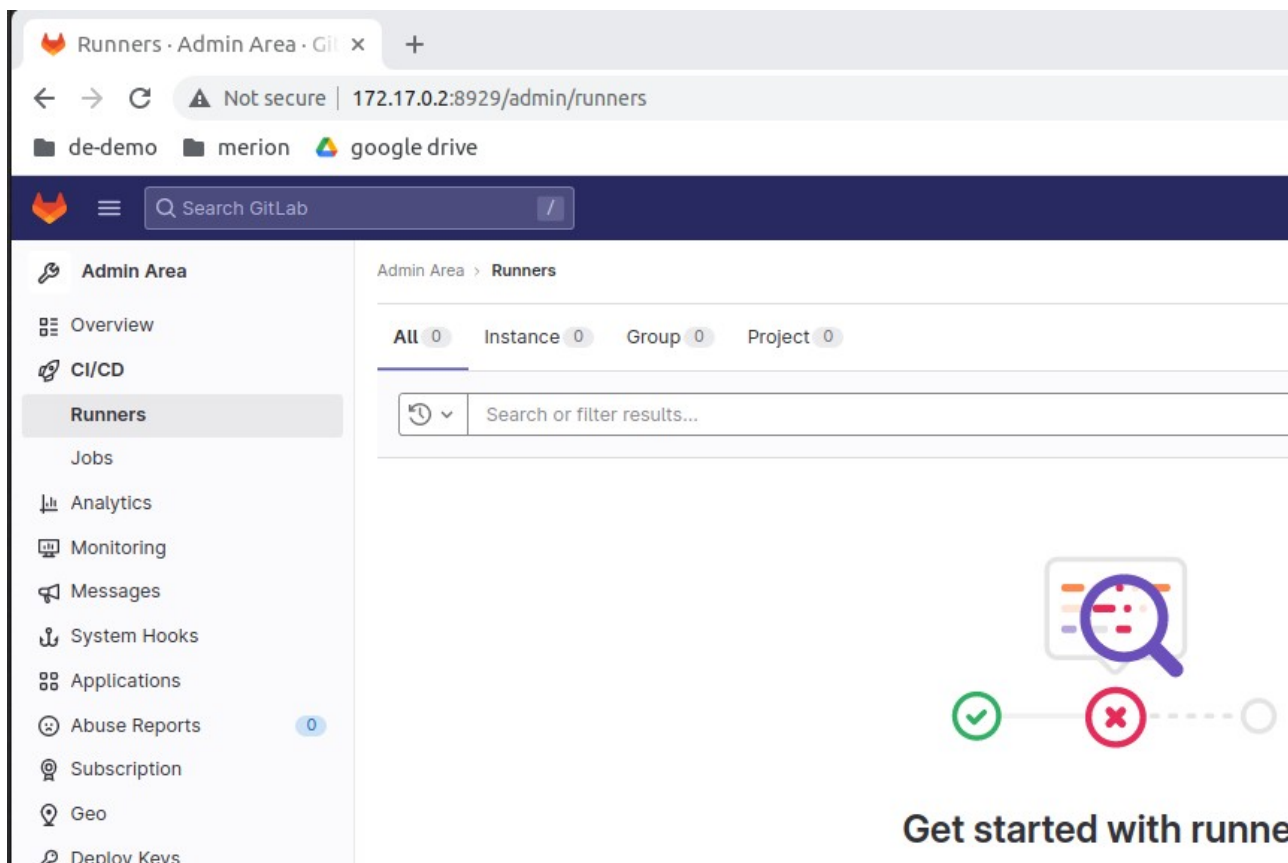
--

Часть 2. Регистрация gitlab-runner

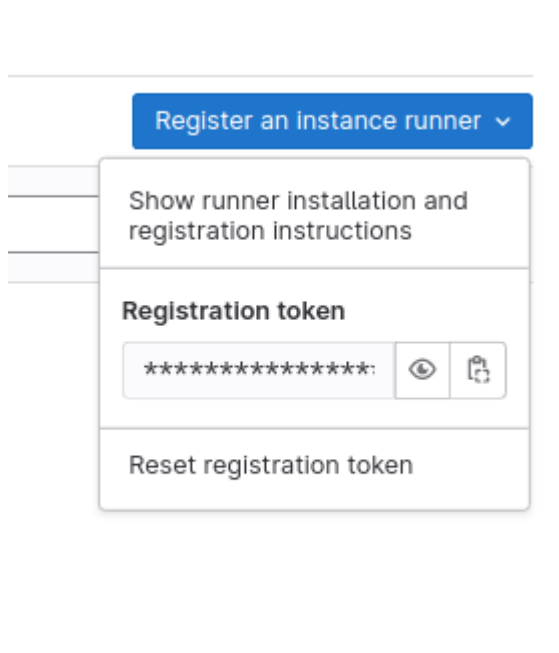
Перейдем в веб-интерфейс gitlab, открываем административную панель



Перемещаемся в раздел "CI/CD - Runners"



Нажимаем кнопку "Register an instance runner" и копируем Registration token (он нам понадобится в дальнейшем):



Возвращаемся в терминал, запускаем процедуру регистрации gitlab-runner (в команду ниже подставьте container-id запущенного gitlab-runner):

```
$ docker exec -ti 30cd9ba26fc6 bash
```

```
filipp@filipp-notebook:~/Desktop$ docker exec -ti 30cd9ba26fc6 bash
root@30cd9ba26fc6:/#
```

В открывшемся псевдотерминале контейнера выполните команду (подставьте url вашего экземпляра gitlab и скопированный ранее registration-token):

```
$ gitlab-runner register --url http://172.17.0.2:8929/ --
registration-token LcLQ8D69Jof9J6ayueoD
```

```
root@30cd9ba26fc6:/# gitlab-runner register --url http://172.17.0.2:8929/ --registration-token LcLQ8D69
Jof9J6ayueoD
Runtime platform                                arch=amd64 os=linux pid=58 revision=f86890c6 versio
n=15.8.1
Running in system-mode.
```

В интерактивном режиме введите запрашиваемые данные.

Значения "Enter the GitLab instance URL", "Enter the registration token", "Enter a description for the runner" оставьте по умолчанию (просто нажмите enter)

Значения "Enter tags for the runner (comma-separated)", "Enter optional maintenance note for the runner" оставьте пустыми (просто нажмите enter)

В поле "Enter an executor" укажите "docker"

В поле "Enter the default Docker image" укажите "alpine:3.17" (или любой другой образ, который вы хотите использовать в качестве образа по умолчанию)

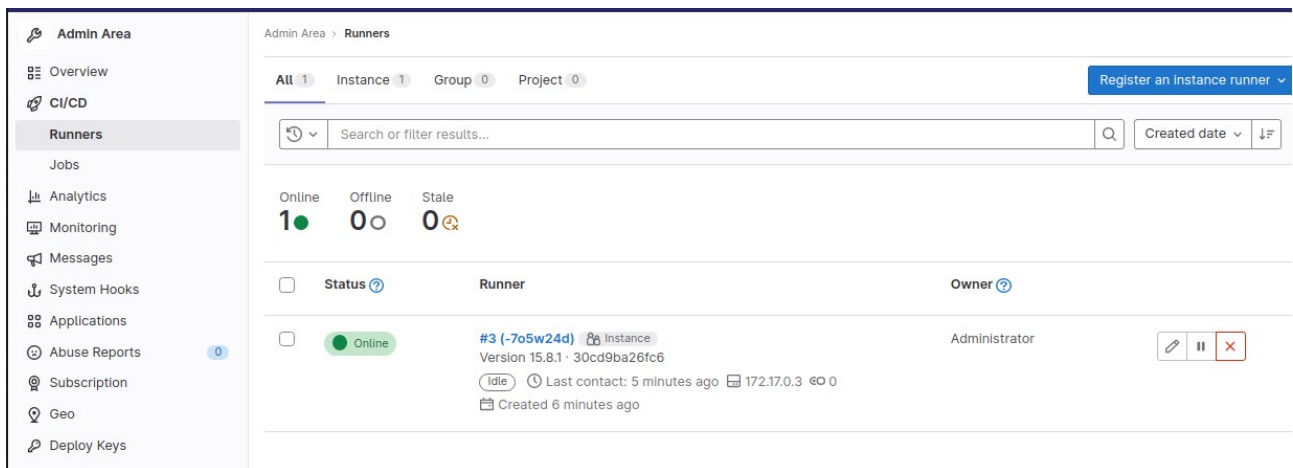
```
Enter the GitLab instance URL (for example, https://gitlab.com/):
[http://172.17.0.2:8929/]:
Enter the registration token:
[LcLQ8D69Jof9J6ayueoD]:
Enter a description for the runner:
[30cd9ba26fc6]:
Enter tags for the runner (comma-separated):

Enter optional maintenance note for the runner:

WARNING: Support for registration tokens and runner parameters in the 'register' command has been depre
cated in GitLab Runner 15.6 and will be replaced with support for authentication tokens. For more infor
mation, see https://gitlab.com/gitlab-org/gitlab/-/issues/380872
Registering runner... succeeded runner=LcLQ8D69
Enter an executor: parallels, shell, ssh, docker+machine, docker-ssh+machine, custom, docker, instance,
kubernetes, docker-ssh, virtualbox:
docker
Enter the default Docker image (for example, ruby:2.7):
alpine:3.17
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"
```

Переместитесь в веб-интерфейс gitlab, на страницу "CI/CD - Runners" в административной панели и обновите страницу:

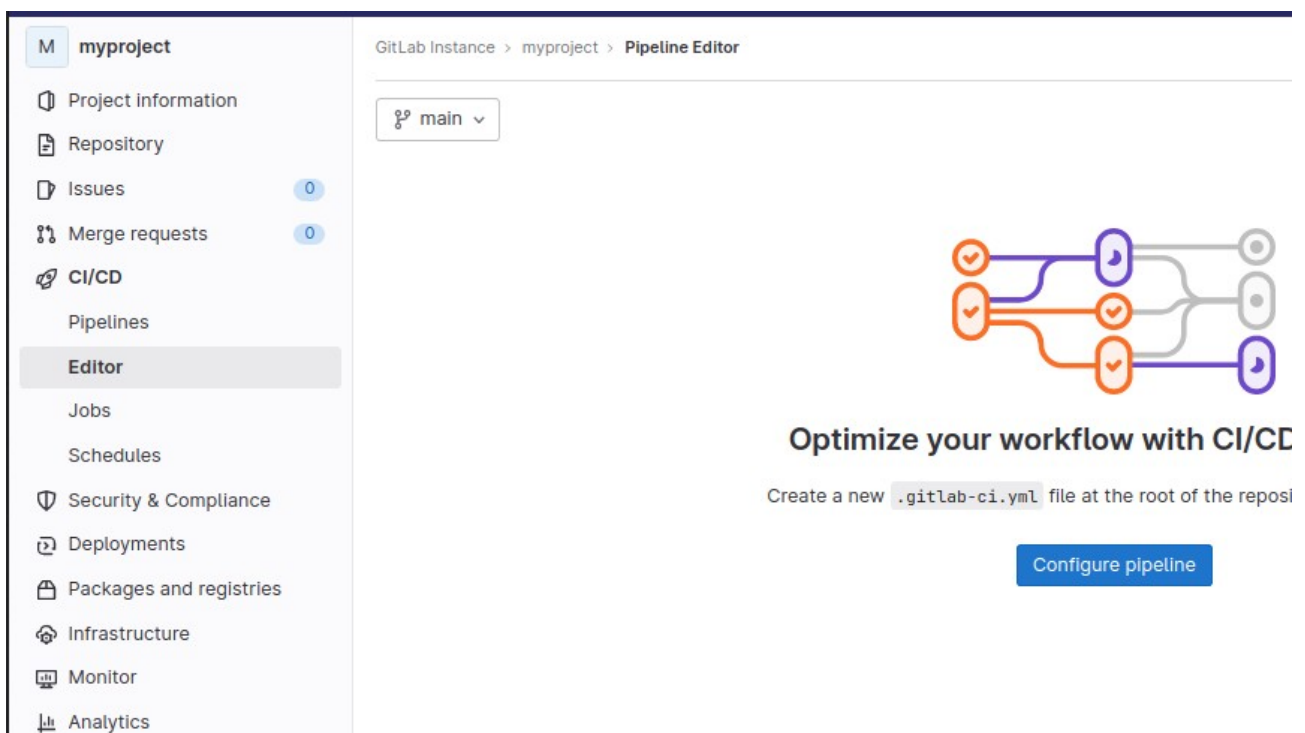


gitlab-runner успешно зарегистрирован и готов выполнять рабочие задания CI/CD

--

Часть 3. Проверка работоспособности установленного и зарегистрированного gitlab-runner

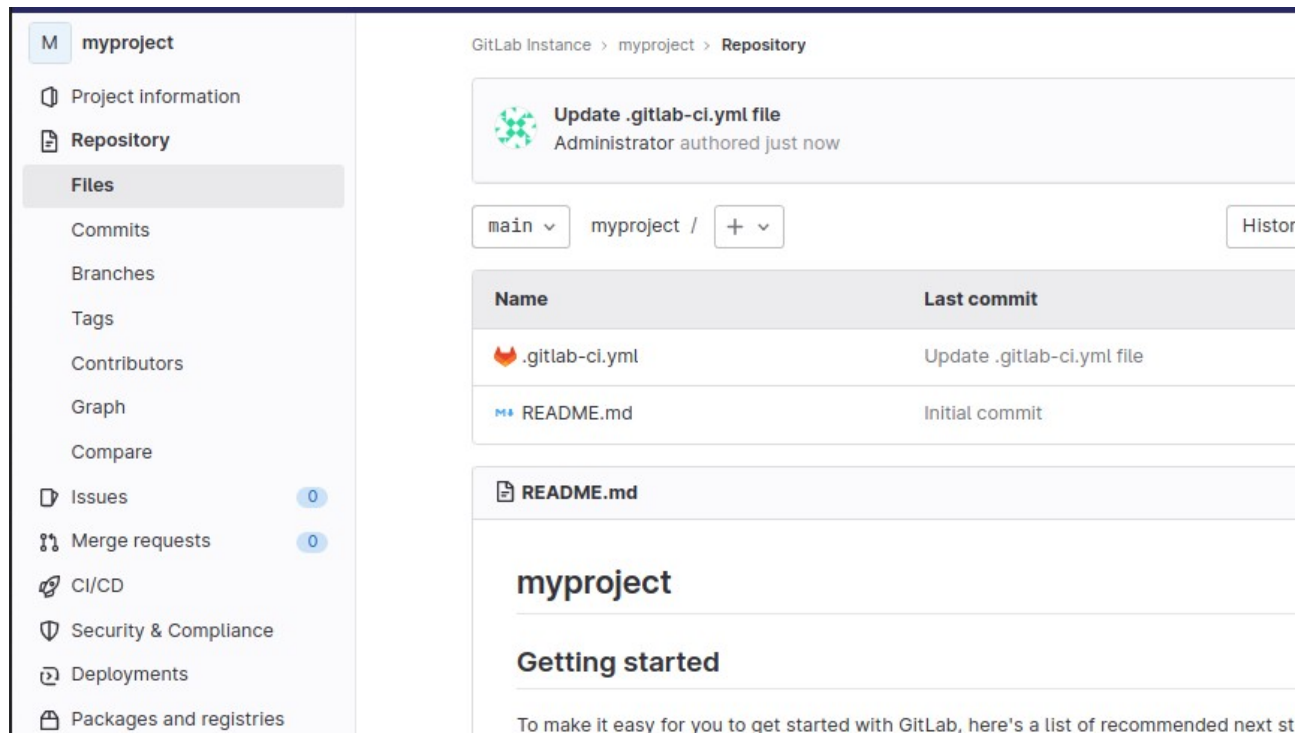
Перейдите в имеющийся проект в gitlab и находясь в нем, переместитесь на вкладку "CI/CD - Editor" и нажмите клавишу "Configure pipeline":





Перед нами готовый шаблон для тестового пайплайна, состоящий из трех этапов (build, test и deploy). Шаблон носит иллюстративный характер и призван продемонстрировать лишь

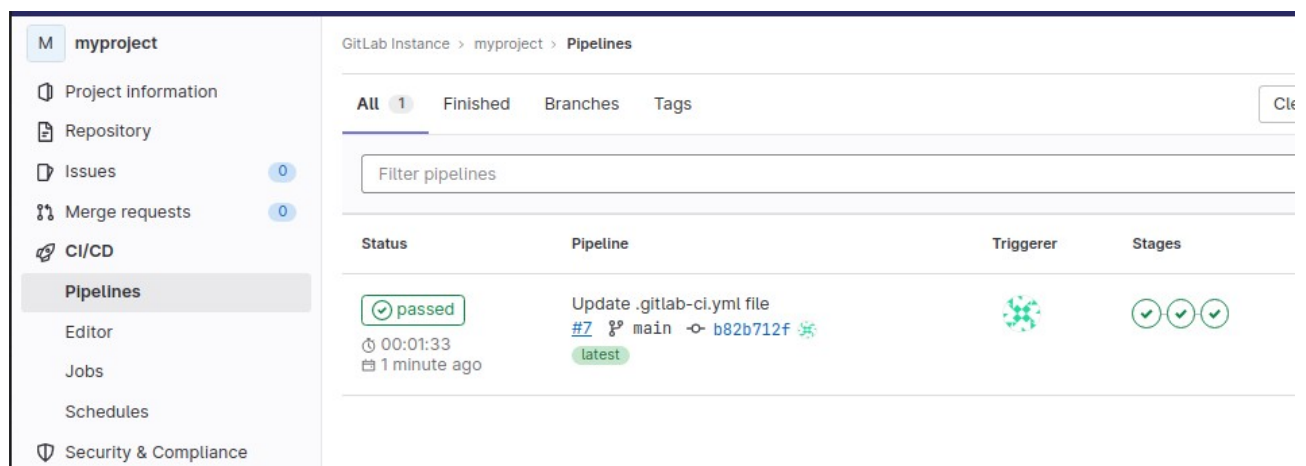
готовность gitlab-runner обрабатывать рабочие задания CI/CD. Сохраняем темплейт как есть нажатием клавиши "Commit changes"






После сохранения изменений в нашем репозитории появится файл, содержащий описание пайплайна (`gitlab-ci.yml`):



Name	Last commit
 .gitlab-ci.yml	Update .gitlab-ci.yml file
 README.md	Initial commit

При внесении изменений в структуру проекта (добавление файла `gitlab-ci.yml` им тоже является), CI/CD пайплайн запустится автоматически. Убедимся в этом, переместившись на вкладку "CI/CD - Pipelines":



Status	Pipeline	Triggerer	Stages
 passed 00:01:33 1 minute ago	Update .gitlab-ci.yml file #7 main b82b712f latest		  

Пайплайн успешно запущен и выполнен.

Мы установили и зарегистрировали gitlab-runner и проверили его работоспособность, запустив CI/CD-пайплайн в проекте gitlab