

# Летняя школа DevOps и CI/CD

GitLab runners & job execution

# GitLab architecture

- Рассмотрим, как устроен gitlab с точки зрения выполнения заданий
- Мы помним, что gitlab это комплексное приложение с множеством функций, сейчас интересует именно job execution
- Вне зависимости от типа установки (облачная или self-managed), у gitlab будут следующие элементы:
  - Сервер gitlab
  - Специальные хосты для выполнения заданий – раннеры (runners)

# GitLab server

- Главный компонент
- На нем хранится вся конфигурация
  - В том числе, конфигурация pipeline
- Он управляет запуском заданий – ставит их на выполнение
- На сервере хранятся результаты выполнения заданий
- Однако сам он задания не выполняет, а передает их специальному узлу – раннеру
  - Если точнее, раннер сам забирает готовые задачи по мере их появления

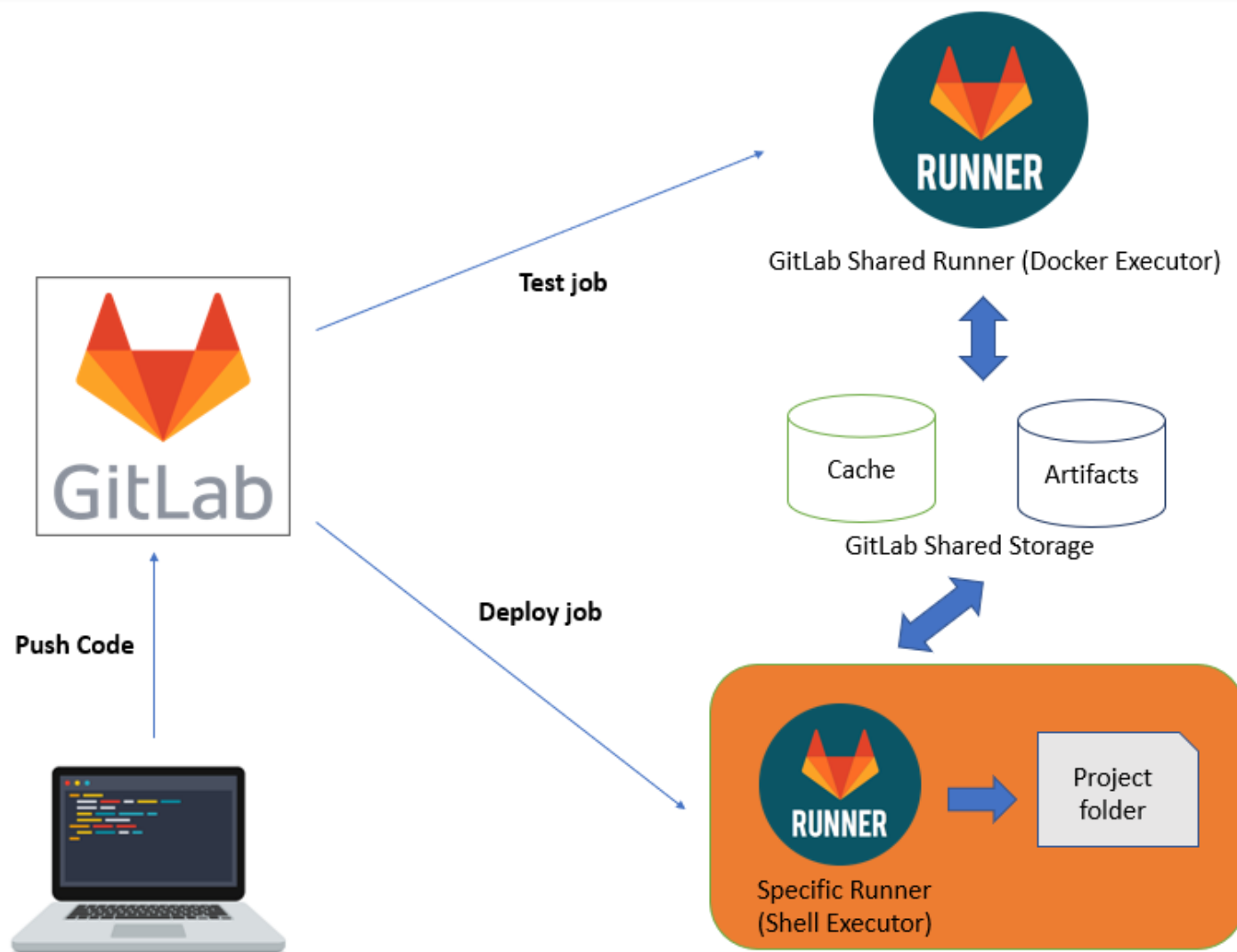
# GitLab runner

- Специальное приложение, которое работает совместно с GitLab CI для выполнения заданий
- Запускается и берет задание в работу в нужный момент
- Можно настроить необходимое количество раннеров в настройках GitLab
- Могут быть разные типы раннеров под разные задачи
- Это достаточно простое приложение, которое может быть установлено на разные ОС
  - Linux, Windows, Mac

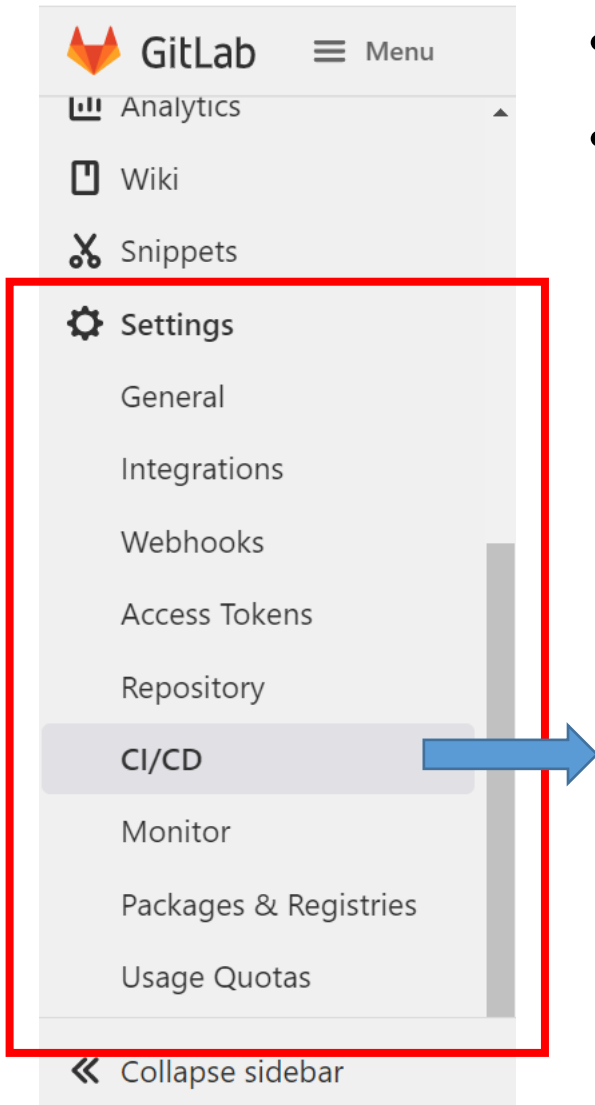
# GitLab runner

- Есть несколько видов раннеров с точки зрения доступности
- Shared runner
  - Доступен из любого проекта данного сервера gitlab
  - Доступен всем пользователям данного gitlab на конкурентной основе
  - Настраивается для экземпляра gitlab в целом
- Specific runner
  - Доступен для данного проекта только
- Group runner
  - Доступен для всех проектов конкретной группы

# GitLab runner



# GitLab runner



- Как узнать, какие раннеры доступны для проекта?
- Раздел Settings => CI/CD => Runners

## Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. [How do I configure runners?](#)

Register as many runners as you want. You can register runners as separate users, on separate servers, and on your local machine. Runners are either:

- **active** - Available to run jobs.
- **paused** - Not available to run jobs.

## Specific runners

These runners are specific to this project.

Set up a specific Runner for a project

1. [Install GitLab Runner and ensure it's running.](#)
2. Register the runner with this URL:

`https://git.miem.hse.ru/`

## Shared runners

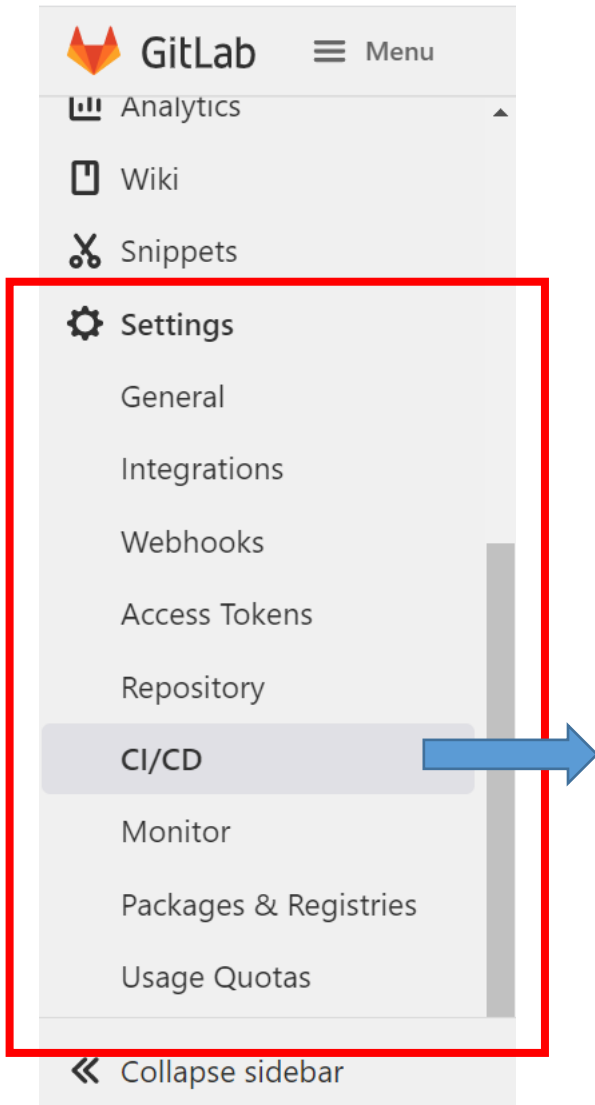
These runners are shared across this GitLab instance.

The same shared runner executes code from multiple projects, unless you configure autoscaling with [MaxBuilds](#) set to 1 (which it is on GitLab.com).

Enable shared runners for this project



# Shared runners



Раздел Settings => CI/CD => Runners

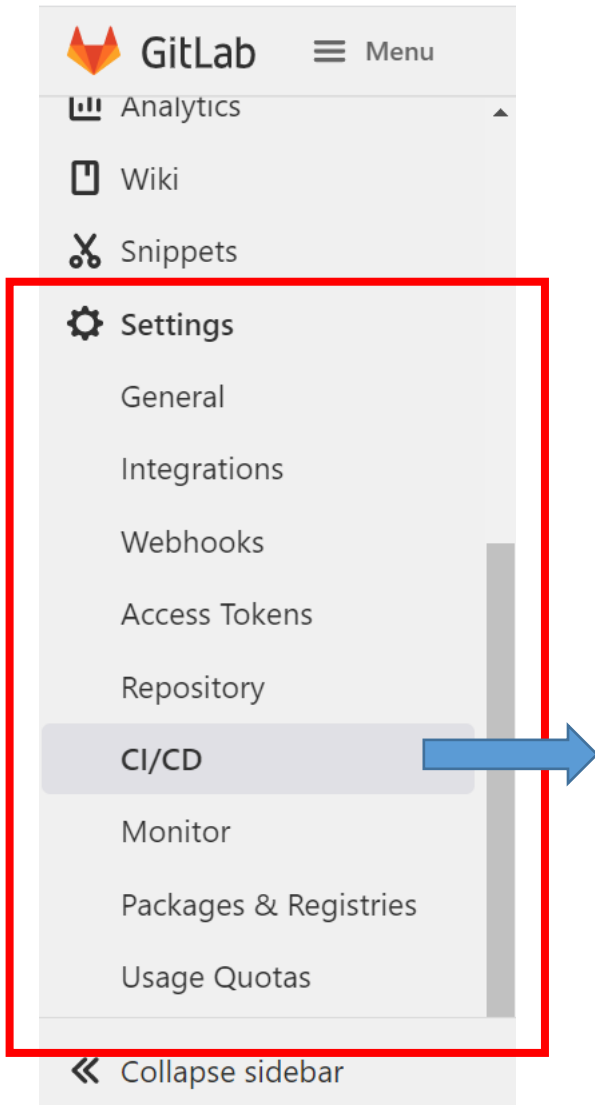
## Available shared runners: 4

- #73 (ZAM4rVjE)  
shared-runner-2-2  
docker
- #55 (9ZzsxiHx) 🔒  
shared cluster  
docker
- #60 (1qKTCpzJ) 🔒  
shared-cluster2  
video-lab-miem
- #72 (6ZBc681U)  
shared-runner-1-2  
docker

- Их настраивает администратор gitlab, но можно использовать
- Доступные раннеры видны в секции available shared runners



# Specific runners



Раздел Settings => CI/CD => Runners

## Available specific runners


<div><div>—</div><div>#94 (KPPnACsW)</div><div>🔒</div></div> <div>vbox-ubuntu-docker</div> <div><div>docker-linux</div><div>my</div></div>	<div><div>✎</div><div>⏸</div><div>Remove runner</div></div>
<div><div>●</div><div>#93 (q6qR52MA)</div><div>🔒</div></div> <div>my-windows-runner-2</div> <div><div>my</div><div>shell-runner</div></div>	<div><div>✎</div><div>⏸</div><div>Remove runner</div></div>
<div><div>●</div><div>#91 (C8a9KJPf)</div><div>🔒</div></div> <div>my-win-runner</div> <div><div>docker-run</div><div>my</div></div>	<div><div>✎</div><div>⏸</div><div>Remove runner</div></div>

- Можно настроить для конкретного проекта
- Достаточно установить приложение-агент на доступный хост
- Мы позже посмотрим, как установить и настроить раннер для своего проекта

# GitLab runner

- Runner это отдельные машины, их нужно настраивать и добавлять в настройках
- Смотрим, какие раннеры нам сейчас доступны
  - Надо зайти в раздел Settings -> CI/CD
  - Выбираем нужный (по тегу)
- В настройках задания надо указать тег нужного раннера

```
7  ∨ job:
8      stage: build
9      ∨ tags:
10         - docker
11  ∨ script:
12      # provide a shell script as argument for this keyword.
13      - echo "Hello World"
14
```



## Available shared runners: 4

● #55 (9ZzsxiHx) 🔒

shared cluster

docker

docker\_cache\_3

docker\_miem

● #12 (7dY1o7Pa) 🔒

shared runner 1

docker

docker\_cache\_1

docker\_itsoft

● #60 (1qKTCpzJ) 🔒

shared-cluster2

video-lab-miem

# Практика — установка своего runner

- Сейчас базовые раннеры — на основе ОС Linux
  - И чаще всего, это раннеры типа docker
- Но вообще, может быть установлен на разных ОС
  - Windows, Mac
- Установим и настроим свой раннер, и зарегистрируем его в проекте

# Установка своего runner на Windows

- Следуем официальной инструкции
  - <https://docs.gitlab.com/runner/install/windows.html>
- Скачиваем приложение с официального сайта
- Размещаем его в нужной директории (C:\GitLab-Runner)
  - Необходимо так же ограничить права на запись для директории и файла
- Следующие команды должны быть выполнены в командной строке с правами администратора

```
cd C:\GitLab-Runner
.\gitlab-runner.exe install
.\gitlab-runner.exe start
```

# Установка своего runner на Windows

- Проверяем, что сервис стартовал

```
PS D:\gitlab-runner> .\gitlab-runner.exe status
Runtime platform                                arch=amd64 os=windows pid=60
40 revision=bbcb5aba version=15.3.0
gitlab-runner: Service is running
```

- Если видим такое сообщение, то все ОК
- Раннер еще надо зарегистрировать в проекте, чуть позже посмотрим, как

# Установка своего runner на Linux

- Следуем официальной инструкции
  - <https://docs.gitlab.com/runner/install/linux-repository.html>
  - <https://docs.gitlab.com/runner/install/linux-manually.html>

```
sudo apt-get update
```

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

```
sudo apt-get install gitlab-runner
```

- Так же проверяем `gitlab-runner status`

# Регистрация своего runner


- Для начала зайдём в настройки проекта для получения необходимой информации
  - Раздел Settings => CI/CD => Runners
- Узнаем URL и Token

## Specific runners

These runners are specific to this project.

Set up a specific Runner for a project

1. [Install GitLab Runner and ensure it's running.](#)
2. Register the runner with this URL:

`https://git.miem.hse.ru/` 

And this registration token:

**YOUR TOKEN** 

Reset registration token

Show Runner installation instructions

# Регистрация своего runner

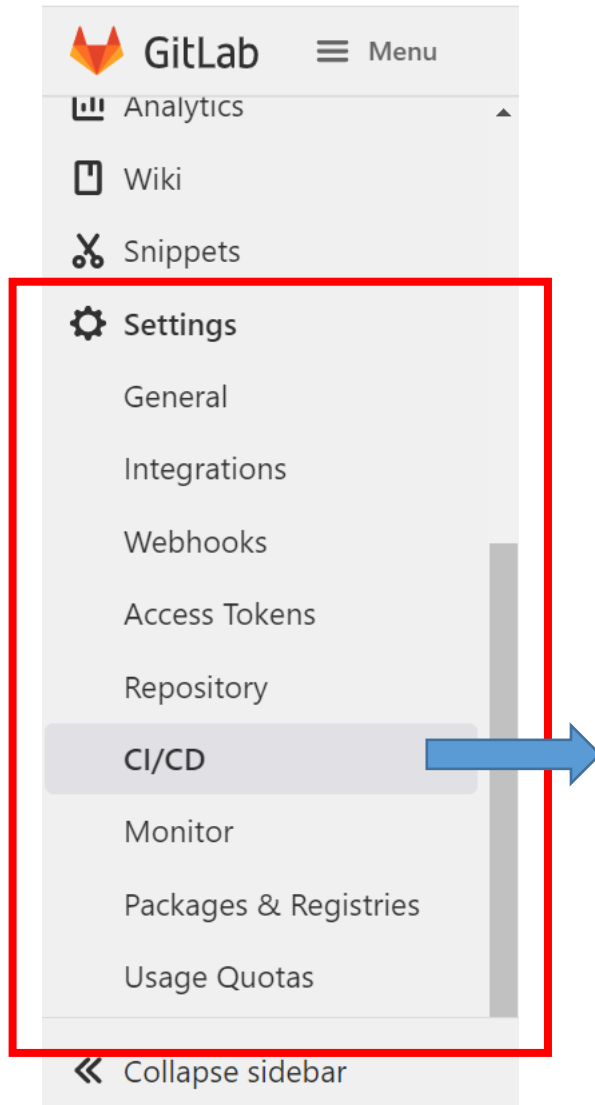
- В командной строке, выполним команду регистрации раннера

```
gitlab-runner register
```

- При регистрации укажем этот URL, token, название раннера и теги, которые с ним будут связаны
- Тип runner укажем docker
  - Важно: у вас должен быть установлен docker на этой машине, т.е. docker работает в командной строке
  - С типами раннеров еще разберемся подробнее чуть позже



# Регистрация своего runner



Раздел Settings => CI/CD => Runners

## Available specific runners

<div><div>—</div><div>#94 (KPPnACsW)</div><div>🔒</div></div> <div>vbox-ubuntu-docker</div> <div><div>docker-linux</div><div>my</div></div>	<div><div>✎</div><div>⏸</div><div>Remove runner</div></div>
<div><div>●</div><div>#93 (q6qR52MA)</div><div>🔒</div></div> <div>my-windows-runner-2</div> <div><div>my</div><div>shell-runner</div></div>	<div><div>✎</div><div>⏸</div><div>Remove runner</div></div>
<div><div>●</div><div>#91 (C8a9KJPf)</div><div>🔒</div></div> <div>my-win-runner</div> <div><div>docker-run</div><div>my</div></div>	<div><div>✎</div><div>⏸</div><div>Remove runner</div></div>

- Если все прошло успешно, мы увидим новый раннер в настройках проекта

# GitLab runner

- Укажем любой из тегов нового раннера настройках задания, чтобы проверить, что он работает и выполняем задания

```
build-job:      # This job runs in the build stage, which runs first.
tags:
- shell-runner
stage: build
script:
- echo "Compiling the code..."
- ls
- echo "file is $PROPERTIES_FILE ... "
```

## Available specific runners

🔴 #94 (KPPnACsW) 🔒

✎ || Remove runner

vbox-ubuntu-docker

docker-linux my

🟢 #93 (q6qR52MA) 🔒

✎ || Remove runner

my-windows-runner-2

my shell-runner

🟢 #91 (C8a9KJPf) 🔒

✎ || Remove runner

my-win-runner




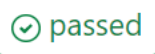



docker-run my

# Запуск задания

- Проверим запуск задания в разделе Job

devops > test2 > Jobs

All 31 Pending 0 Running 0 Finished 29

Status	Name	Job	Pipeline	Stage	Duration	Coverage
	lint-test-job	#153533  master  7c747622 docker-linux	#71572 by 	test	 00:00:16  4 minutes ago	
	unit-test-job	#153532  master  7c747622 docker-run	#71572 by 	test	 00:00:24  4 minutes ago	
	build-job	#153531  master  7c747622 shell-runner	#71572 by 	test	 00:00:05  3 minutes ago	

# Запуск задания

- В логах задания так же указан использованный раннер

```
1 Running with gitlab-runner 15.3.0 (bbcb5aba)
2   on my-windows-runner-2 q6qR52MA
3   ✓ Preparing the "shell" executor 00:00
4   Using Shell executor...
5   ✓ Preparing environment 00:01
6   Running on DESKTOP-HJ6IP71...
7   ✓ Getting source from Git repository 00:02
8   Fetching changes with git depth set to 50...
9   Reinitialized existing Git repository in D:/gitlab-runner/builds/q6qR52MA/0/devops/test
10  2/.git/
11  Checking out b3f60c97 as master...
12  git-lfs/2.13.1 (GitHub; windows amd64; go 1.14.12; git e896fc7a)
```

**Duration:** 4 seconds

**Timeout:** 1h (from project) ?

**Runner:** #93 (q6qR52MA) my-windows-runner-2

**Tags:** shell-runner

Commit [b3f60c97](#) 📄

Update .gitlab-ci.yml file

✓ Pipeline #71597 for master 📄