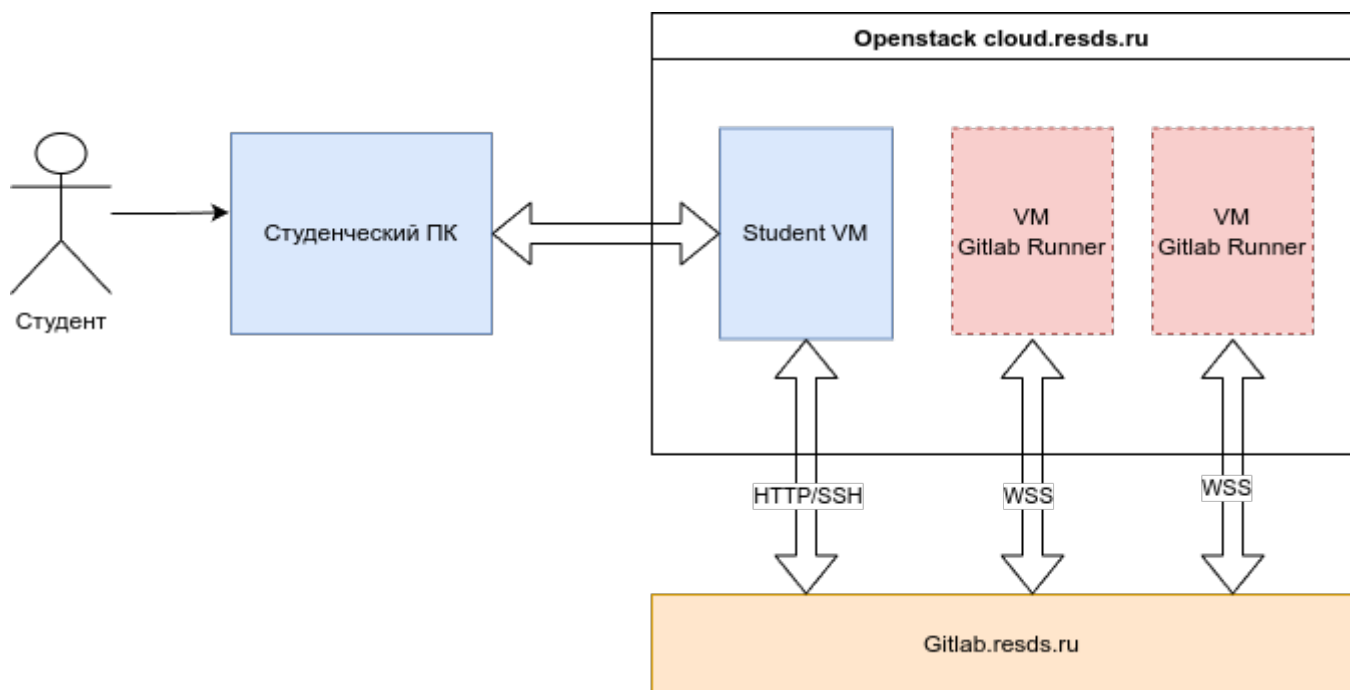


Автоматизация сборки и развертывания приложений. Gitlab CI.

0. Схема стенда



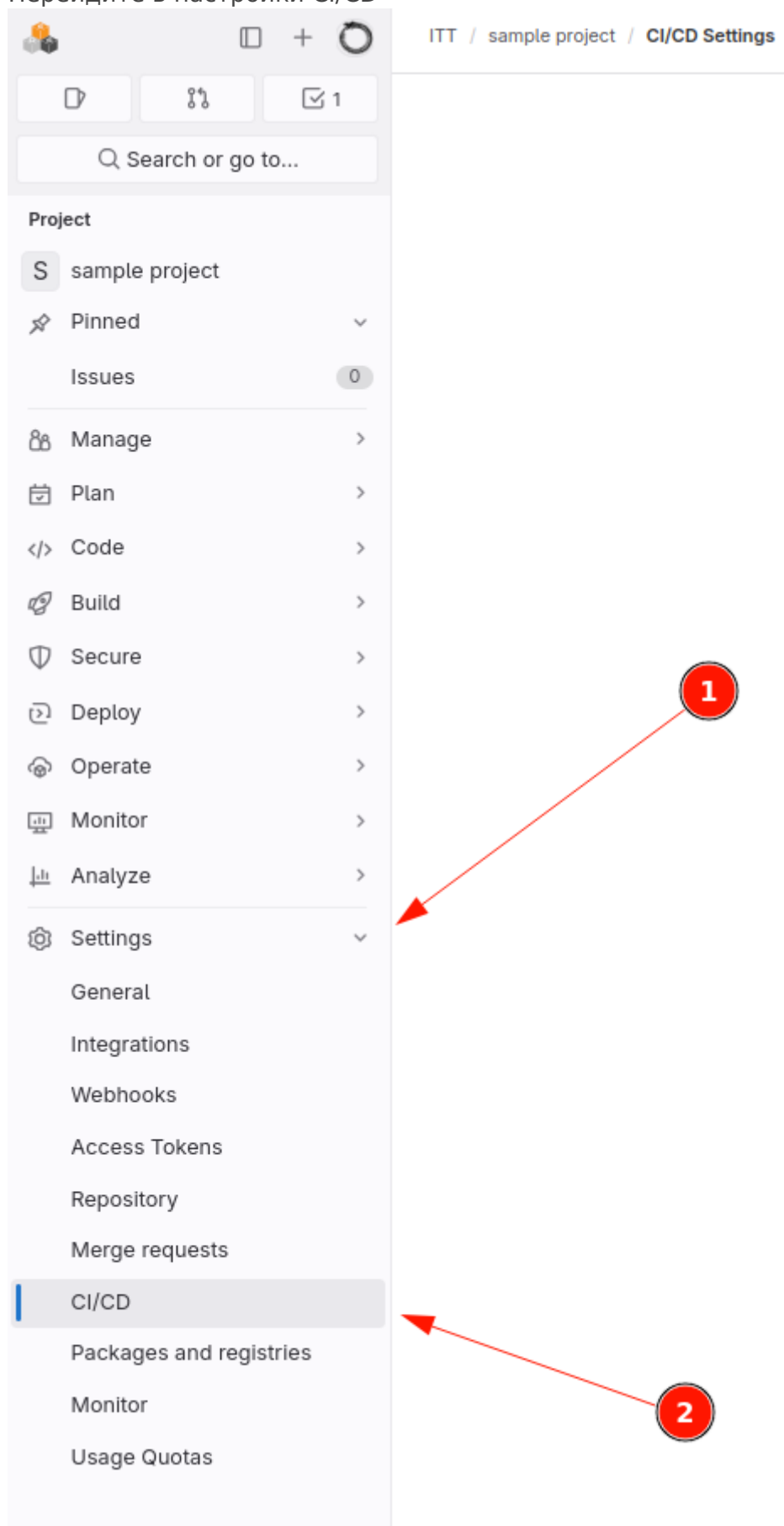
1. Подготовка репозитория

Сделайте форк репозитория: <https://gitlab.resds.ru/itt/sample-project>

2. Подготовка раннера

1. Перейдите в настройки проекта

2. Перейдите в настройки CI/CD



3. Разверните пункт отвечающий за раннеры и нажмите New project runner

sample project / ci/cd settings

Q Search page

General pipelines

Customize your pipeline configuration.

Expand

Auto DevOps

Automate building, testing, and deploying your applications based on your continuous integration and delivery configuration. [How do I get started?](#)

Expand

Runners

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

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

How do runners pick up jobs?

Runners are either:

active

 - Available to run jobs.

paused

 - Not available to run jobs.

Tags control which type of jobs a runner can handle. By tagging a runner, you make sure runners only handle the jobs they are equipped to run. [Learn more.](#)

Project runners

These runners are assigned to this project.

New project runner

Other available runners

#61 (XjfrudXHB)

k8s-test

Enable for this project

#57 (YzpyKfKv)

Enable for this project

#52 (nXrIVC9X5)

haproxy-runner

haproxy-runner3

Enable for this project

#51 (AyF3hGPY3)

haproxy-runner

haproxy-runner2

Enable for this project

Instance runners

These runners are available to all groups and projects.

Enable instance runners for this project

This GitLab instance does not provide any instance runners yet. Administrators can register instance runners in the admin area.

Group runners

These runners are shared across projects in this group.

Group runners can be managed with the [Runner API](#).

Disable group runners

for this project

This group does not have any group runners yet. To register them, go to the [group's Runners page](#).

4. Введите данные для раннера и добавьте тегов которые будут как-то обозначать общие признаки раннеров

Platform

Operating systems

Linux

macOS

Windows

Containers

Docker

Kubernetes

Tags

Tags

Add tags to specify jobs that the runner can run. [Learn more.](#)

linux, runner

Separate multiple tags with a comma. For example, `macos, shared`.

☒ Run untagged jobs

Use the runner for jobs without tags in addition to tagged jobs.

Configuration (optional)

Runner description

Paused

Stop the runner from accepting new jobs.

Protected

Use the runner on pipelines for protected branches only.

Lock to current projects

Use the runner for the currently assigned projects only. Only administrators can change the assigned projects.

Maximum job timeout

Maximum amount of time the runner can run before it terminates. If a project has a shorter job timeout period, the job timeout period of the instance runner is used instead.

Enter the job timeout in seconds. Must be a minimum of 600 seconds.

Create runner

5. Нажмите Create Runner

6. Перейдите на узел в котором должен будет работать раннер и установите его

```
sudo curl -L --output /usr/local/bin/gitlab-runner https://gitlab-runner-  
downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64
```

```
sudo chmod +x /usr/local/bin/gitlab-runner
```

```
sudo useradd --comment 'GitLab Runner' --create-home gitlab-runner --shell /bin/bash
```

```
sudo gitlab-runner install --user=gitlab-runner --working-directory=/home/gitlab-runner
```

7. Установите на узле раннера Docker

8. Зарегистрируйте раннер

Register runner

GitLab Runner must be installed before you can register a runner. [How do I install GitLab Runner?](#)

Step 1

Copy and paste the following command into your command line to register the runner.

```
$ gitlab-runner register  
--url https://gitlab.resds.ru  
--token glrt-T56Thd8DcDsZWCdZJz2n
```

The runner authentication token `glrt-T56Thd8DcDsZWCdZJz2n` displays here for a short time only. After you register the runner, this token is stored in the `config.toml` and cannot be accessed again from the UI.

Step 2

Choose an executor when prompted by the command line. Executors run builds in different environments. [Not sure which one to select?](#)

Step 3 (optional)

Manually verify that the runner is available to pick up jobs.

```
$ gitlab-runner run
```

This may not be needed if you manage your runner as a [system](#) or [user service](#).

[View runners](#)

Задание

1. Создайте новую ветку `feature/ci-cd`
2. В новой ветке создайте многоуровневый `.gitlab-ci.yml`
 1. Статическое тестирования кода(прогнать линтерами код)
 2. Сборка контейнеров и отправка в gitlab-registry
 3. Запуск собранного приложения с использованием ранее загруженного образа
3. Добавьте еще один раннер
4. Добавьте деплой на еще один сервер
 1. С помощью тегов сделайте деплой на 2 узла при запуске
 2. Повторяющийся код уберите с помощью [Anchors](#)
 3. Установите зависимости между запускаемыми шагами
5. Разделите запуск между узлами

1. Разделите 2 узла по ролям:

1. `dev`

2. `prod`

2. Запуск на `prod` осуществляется вручную после успешного на `dev`

6. Смерджите ветку в `main`

Версия #3

Тарабанов Илья Федорович создал 3 апреля 2024 15:25:02

Тарабанов Илья Федорович обновил 16 мая 2024 19:16:57