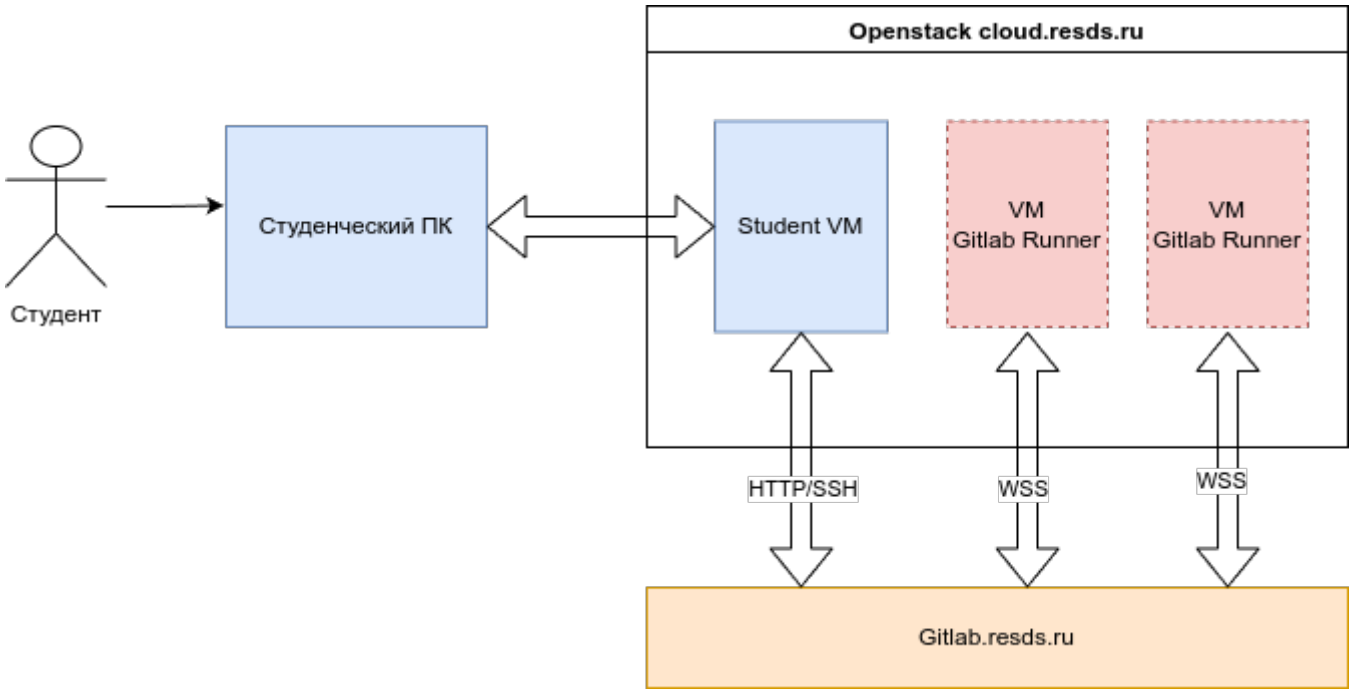
Автоматизация сборки и развертывания приложений. Gitlab CI.

0. Схема стенда



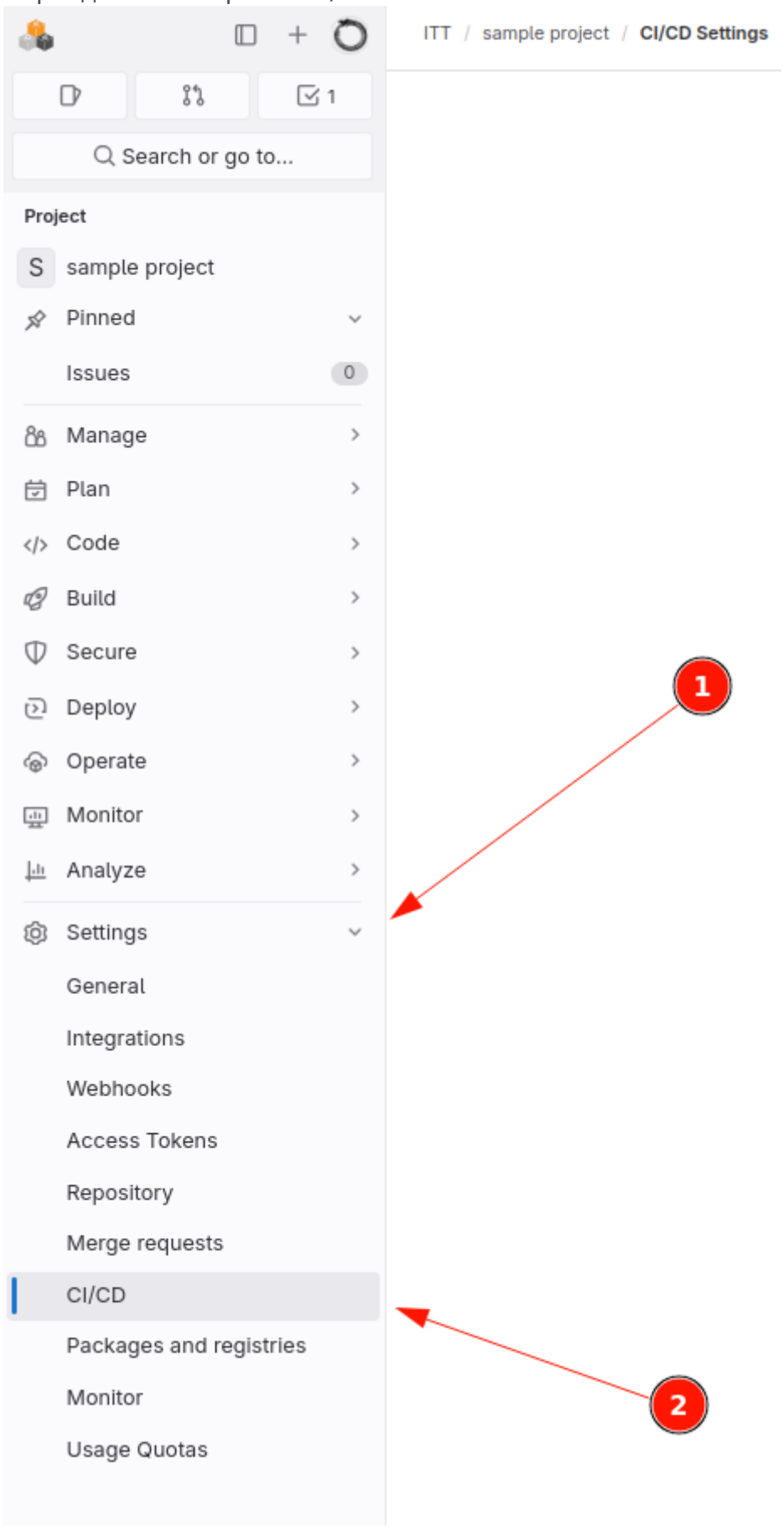
1. Подготовка репозитория

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

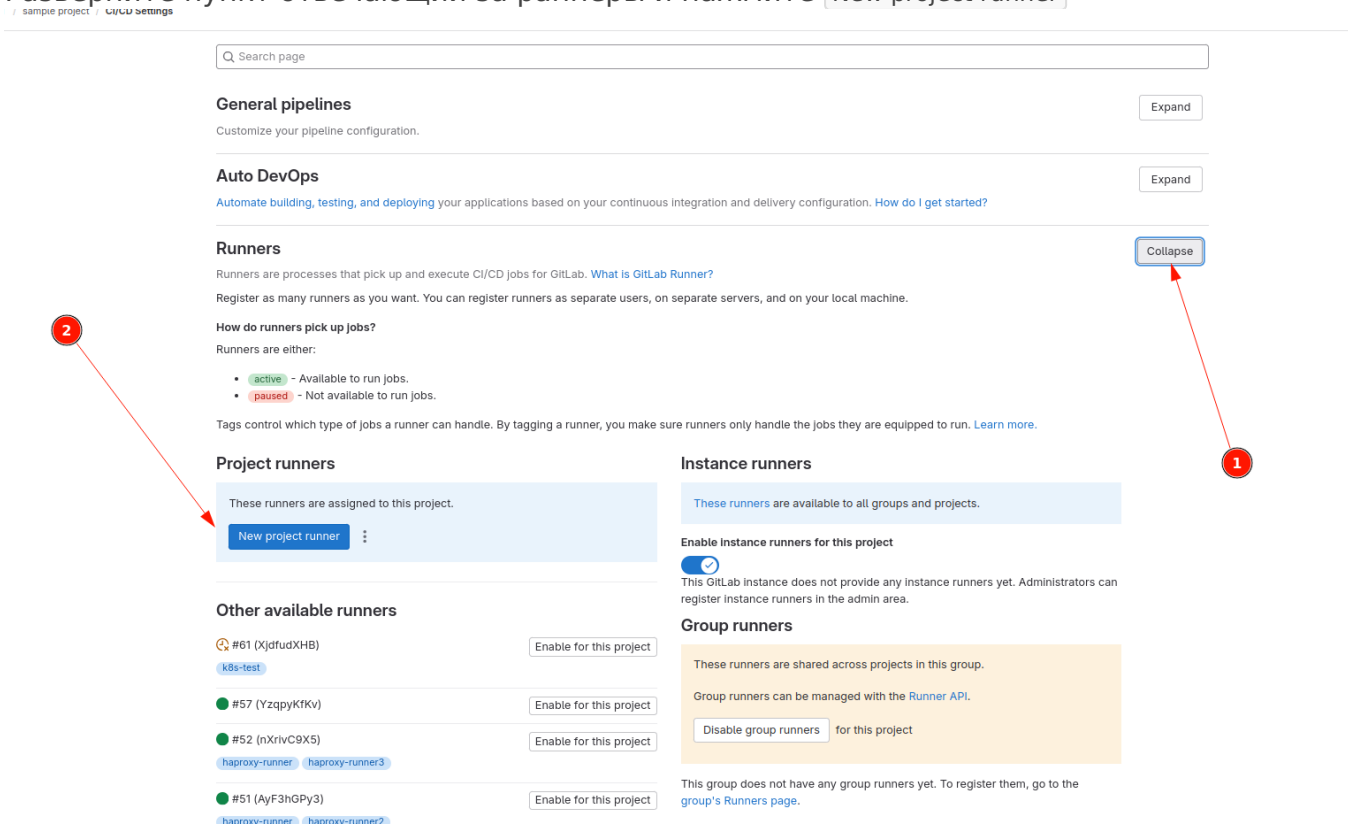
2. Подготовка раннера

1. Перейдите в настройки проекта

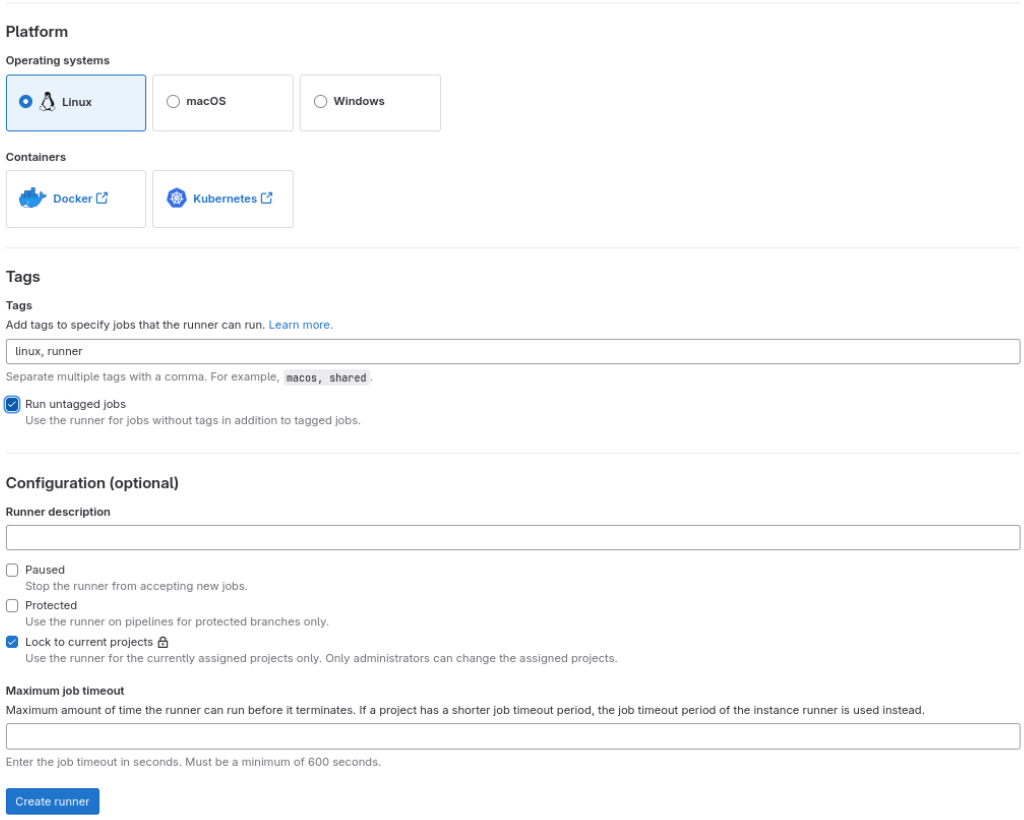
2. Перейдите в настройки CI/CD



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



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



5. Нажмите `Create Runner`
6. Перейдите на узел в котором должен будет работать раннер и установите его

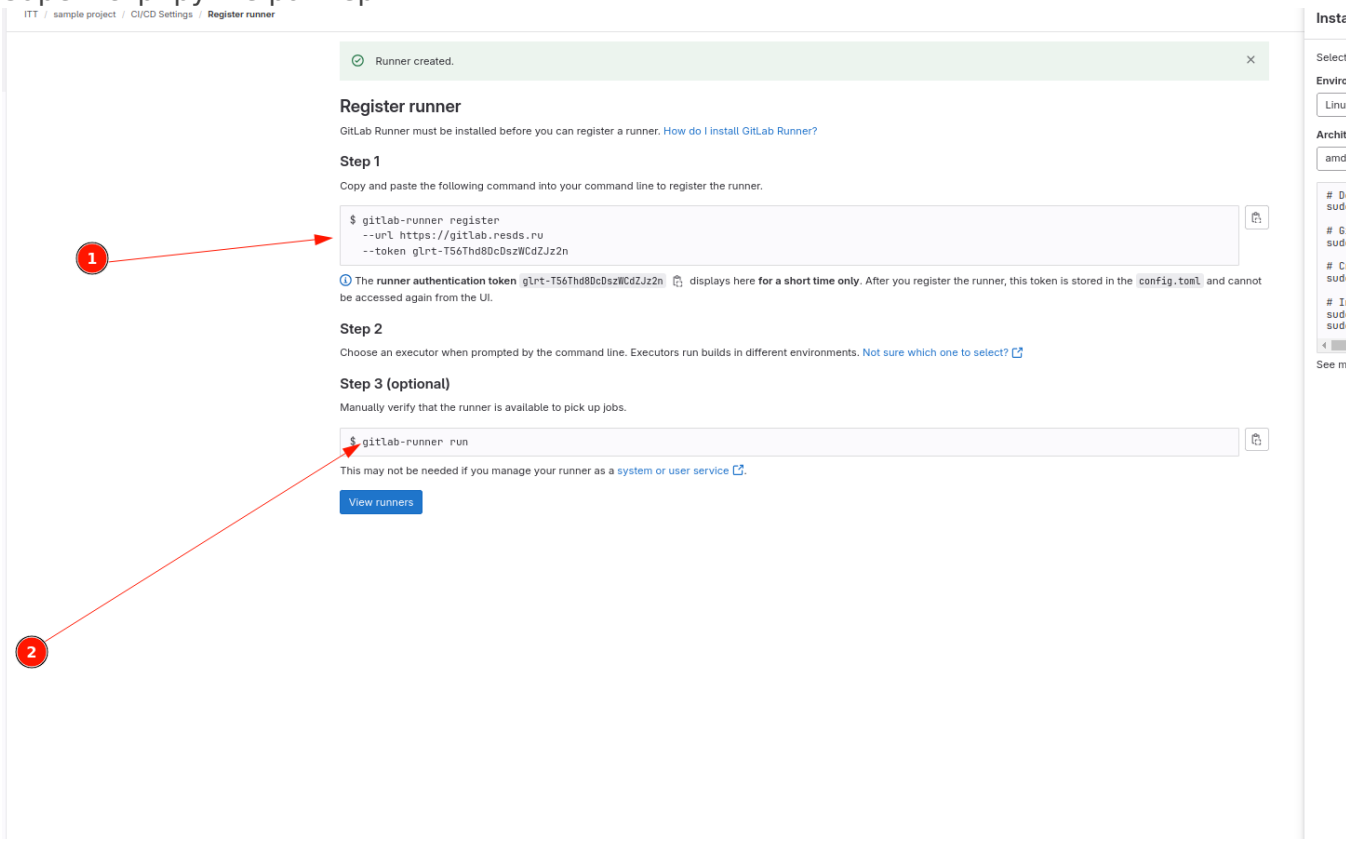
```
sudo curl -L --output /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64

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



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