



МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Цифровая кафедра

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 7

по дисциплине

«Непрерывная разработка и интеграция CI/CD»

Тема практической работы: Цикл настройки в CI/CD в GitLab

Выполнил студент группы 14

Стока И.П.

Руководитель практической работы

Волков М.Ю.

Практическая работа выполнена

«__»_____202__ г.

«Зачтено»

«__»_____202__ г.

Москва 2023г.

Необходимо произвести настройку цикла CI/CD в GitLab. Проект, находящийся в репозитории, написан на python, следовательно, будем использовать последний образ python.

Будут представлены следующие этапы сборки: build, test, package, deploy.

Соответственно, для начала проводим стадию сборки, далее тестирование, после чего стадию упаковки и в конце выгрузку.

Данная информация представлена на рисунке 1, в формате YAML, в файле .gitlab-ci.yml.

```
image: python:latest

stages:
  - build
  - test
  - package
  - deploy
  - notify

PythonBuild:
  stage: build
  script:
    - echo "Build python project.py"

PythonTest1:
  stage: test
  script:
    - echo "Lets try to do sime tests"

PythonTest2:
  stage: test
  script:
    - python project.py

PythonPackage:
  stage: package
  script:
    - echo "Package file"

PythonDeploy:
  stage: deploy
  script: echo "This project deploy from main brainch"
```

Рисунок 1 – Содержание файла .gitlab-ci.yml

Далее проверим работоспособность, перейдя во вкладку CI/CD Pipelines (Рисунок 2).



Рисунок 2 – Проверка работоспособности

Вопросы к практической работе

1. Совместимость - GitLab поддерживает интеграцию с такими платформами как: Docker, k8s, и многими другими, в том числе облачными. Метки и документация – имеет удобную систему меток, позволяющая отслеживать изменения, ошибки и запросы. Доступность – позволяет гибко настраивать доступ к репозиториями. IO система – быстрый и легкий импорт и экспорт данных. Аналитика и прогнозирование – присутствуют инструменты аналитики и планирования.

2. Различие в синтаксисе. GitLab более функционален, имеет возможности ручной работы в настройке контейнеров, мониторинга сред, также поддерживает возможность использовать собственный раннер и поддерживает множество различных отчетов.

3. Шаблоны написания pipeline для необходимого образа. Все шаблоны можно найти на следующей странице [GitLab CI/CD Examples | GitLab](#).

4.

```
image: python:latest
variables:
  PIP_CACHE_DIR: "${CI_PROJECT_DIR}/.cache/pip"
cache:
  paths:
    - .cache/pip
    - venv/

before_script:
  - python --version # For debugging
  - pip install virtualenv
  - virtualenv venv
  - source venv/bin/activate
test:
  script:
    - python setup.py test
    - pip install tox flake8 # you can also use tox
    - tox -e py36,flake8
run:
  script:
    - python setup.py bdist_wheel
    - pip install dist/*
artifacts:
  paths:
    - dist/*.whl
pages:
  script:
    - pip install sphinx sphinx-rtd-theme
    - cd doc
    - make html
    - mv build/html/ ../public/
artifacts:
```

```
    paths:
      - public
  rules:
    - if: $CI_COMMIT_BRANCH == $CI_DEFAULT_BRANCH
  deploy:
    stage: deploy
    script: echo "Define your deployment script!"
    environment: production
```

5. Описывает каким этапом является данный этап сборки из всех представленных, прописанных в stages

6. Описывает версию образа Docker, который он будет использовать в процессе CI/CD