Если вы видите что-то необычное, просто сообщите мне.

Пример создания Pipeline на основе Gitlab-ci.

Рассмотрим пример создания pipeline на основе gitlab-ci.yaml

Давайте посмотрим какие пункты мы можем внести gitlab.

В gitlab-ci.yaml будет внесены пукнты 3-5 прошлой статьи. То есть, gitlab-ci будет отвечать за создание, тестирование и выпуск образа и развертывание его в рабочем окружении.

1. Добавим в проект файл gitlab-ci.yml.
2. Заполним его, для этого воспользуемся готовым шаблоном:

```
image: docker:latest

variables:
  DOCKER_DRIVER: overlay
  CONTAINER_TEST_IMAGE: $CI_REGISTRY/$CI_PROJECT_PATH:$CI_COMMIT_REF_NAME
  CONTAINER_RELEASE_IMAGE: $CI_REGISTRY/$CI_PROJECT_PATH:latest

services:
  - docker:dind

stages:
  - build
  - test
  - release
```

```
  - deploy

build:
  stage: build
  script:
    - docker login -u gitlab-ci-token -p $CI_JOB_TOKEN $CI_REGISTRY
    - docker build --pull -t $CONTAINER_TEST_IMAGE . --build-arg=secret_key=secret
    - docker push $CONTAINER_TEST_IMAGE
  only:
    - tags

 test:
   image: $CI_REGISTRY/$CI_PROJECT_PATH:$CI_COMMIT_REF_NAME
   stage: test
   services:
     - postgres:11-alpine
   variables:
     SECRET_KEY: runner
     POSTGRES_DB: runner
     POSTGRES_USER: runner
     POSTGRES_PASSWORD: runner
     POSTGRES_HOST: postgres
     POSTGRES_PORT: 5432
     TEST: 1
   script:
     - python3 /code/backend/manage.py qa
   only:
     - tags

release:
  stage: release
  script:
    - docker login -u gitlab-ci-token -p $CI_JOB_TOKEN $CI_REGISTRY
    - docker pull $CONTAINER_TEST_IMAGE
    - docker tag $CONTAINER_TEST_IMAGE $CONTAINER_RELEASE_IMAGE
    - docker push $CONTAINER_RELEASE_IMAGE
  only:
    - tags

deploy:
```

```
  image: kroniak/ssh-client
  stage: deploy
  script:
    - eval $(ssh-agent -s)
    - ssh-add <(echo "$SSH_PRIVATE_KEY")
    - ssh -o StrictHostKeyChecking=no -p $TESTPORT $TESTUSER@$STAGINGHOST make --
directory=$TESTPATH deployfront TAG=$CI_COMMIT_REF_NAME
  only:
    - tags
#  when: manual
```

Для начала разберем переменные используемые в данном файле конфигурации, но которые необхдимо задать в окружении gitlab(Settings->CI\CD->Variables->"Add variables" уберите галочку "Protected variable" для ознакомления она вам не понадобится):

- SSH_PRIVATE_KEY = приватная часть ключа для доступа к удаленной машине
- TESTPORT = порт на котором слушает ssh сервер
- TESTUSER = пользователь у которого есть публичная часть доступа к серверу.
- TESTHOST = адрес хоста,(ip или dns имя)
- TESTPATH = путь до папки проекта.

Все остальные переменные перечислены в секции `variables` (используются в местах где они начинаются с символа `$`) геренируются в процессе запуска скрипта. Берутся они из gitlab окружения, можно найти в документации к gitlab.

Теперь рассмотрим стадии, в приведенном примере в разделе `stages` их 4 стадии:

- build - сбор образа
- test - тестирование
- release - выпуск
- deploy - развертывание

Рассмотрим их более подробно:

Для начала общие части

- image - образ внутри которого будет выполняться скрипт
- stage - стади к которой относится текущая секция
- services - дополнительные сервисы необходимые для проекта
- script - непосредственно выполнение логики.

Revision #2

Created 2021-09-22 11:51:31 UTC by gasick

Updated 2021-09-27 16:14:40 UTC by gasick