

Homework_Lesson28_Report

Задание:

1. Изучите основы pipeline CI/CD и настройте интеграцию своего проекта с платформой управления исходным кодом (например, GitHub).
2. Определите этапы pipeline для своего проекта (например, сборка, тестирование, развертывание), и автоматизируйте каждый этап с помощью инструментов pipeline CI/CD, таких как Jenkins, Gitlab или Github Actions.
3. Создайте репозиторий на Github и добавьте исходный код вашего приложения.
4. Создайте Dockerfile для сборки образа приложения.
5. Установите VSC на свой компьютер.
6. Откройте ваш проект в VSC.
7. Установите расширения для языка программирования, которым вы пользуетесь (например, для Python - Python, для Java - Java Extension Pack и т.д.).
8. Установите расширения для интеграции с вашей выбранной CI/CD системой (например, Jenkins Pipeline Linter, GitLab Workflow Editor, CircleCI Orb for VS Code и т.д.).
9. Настройте подключение к вашей выбранной системе контроля версий (например, Git). Для этого можно использовать расширение GitLens или встроенный Git в VSC.
10. Настройте автоматическое форматирование кода и проверку ошибок в реальном времени в VSC, используя соответствующие расширения (например, Pylance для Python, ESLint для JavaScript и т.д.).
11. Определите удаленный сервер и разработайте решение для сборки вашего проекта на этом сервере с использованием механизмов параллельной сборки.
12. Проверьте свое решение, запустив pipeline CI/CD для вашего проекта, отслеживая метрики качества и оптимизируя процесс сборки с помощью параллельного исполнения.

Выполнение

У нас есть Супер мего приложение на pythone + flask которое открывает на порту 5050 html страничку с парой картинок app.py. Картинки у нас в папке со статическими файлами находится. Так же в папке tests у нас есть простейшие тесты, который проверяет статус ответа сервера 200 у нашего приложения, а также некоторый текст, который у нас присутствует на страничке. Тесты будут выполняться в пайплайне параллельно. В отличии от работы по Jankins2 диплом будет осуществляться в докер. На основании докер файла, находящегося в контейнере собирается Image. На основании этого image собирается контейнер и запускается на порту 5050 с именем myapp-container. До запуска контейнера присутствует этап Stop and Remove Old Container в ходе которого останавливаются и удаляются контейнеры с именем myapp-container что бы в дальнейшем контейнер

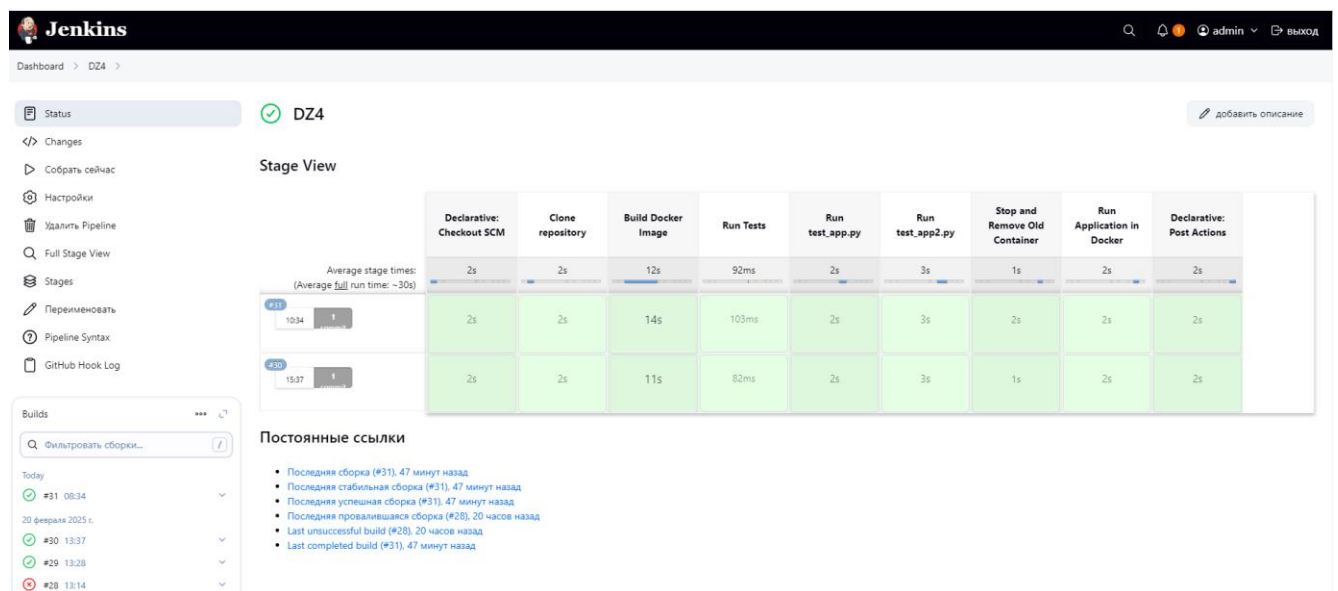
запустился корректно. По итогу пайплайна приходит сообщение на почту о его выполнении. Пайплайн так же запускается после комита с помощью webhook.

Обзор пайплайна:

Этот пайплайн выполняет следующие шаги:

1. Клонировать репозиторий из GitHub.
2. Собирает Docker-образ с приложением.
3. Запускает тесты в контейнерах параллельно.
4. Останавливает и удаляет старый контейнер (если он есть).
5. Разворачивает новое приложение в Docker-контейнере.
6. Отправляет уведомления по электронной почте в случае успеха или ошибки.

Скрин выполненного пайплайна.



Проверяем webhooks

Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

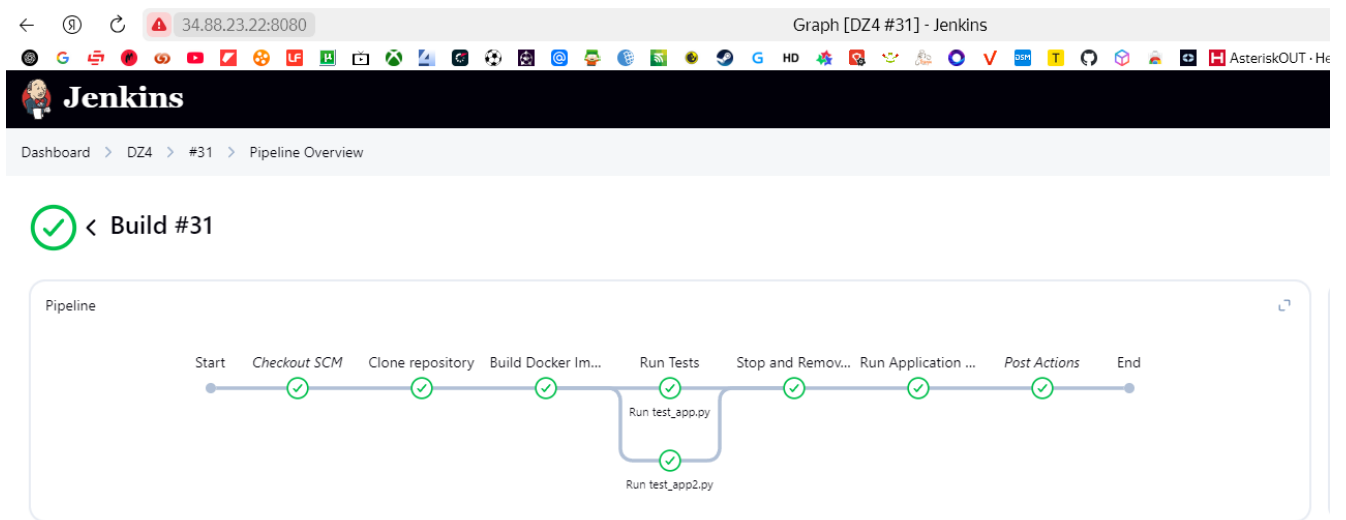
✓ <http://34.88.23.22:8080/github-web...> (push)

Last delivery was successful.

Edit

Delete

Проверим параллельное выполнение тестов.



Проверяем запущен ли контейнер с именем myapp-container.

```
alex5482671a1@minion:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
9cca899b4c3b   myapp-image  "python app.py"          54 minutes ago Up 54 minutes  0.0.0.0:5050->5050/tcp, [::]:5050->5050/tcp  myapp-container
alex5482671a1@minion:~$
```

Проверим доступно ли наше приложение после дилпоя в контейнер. Порт 5050.


Information

Levchenko Alexey Viktorovich

Group: DOS24-onl



Topic: webserver

IP Address: [192.168.1.210:8080](#)



Created by [Levchenko Alexey](#)

Уведомление на почту.

Jenkins Job SUCCESS: DZ4 #31  Входящие 



alex1436183@gmail.com

кому: мне ▾

Jenkins job DZ4 (31) успешно выполнен!

Проверить можно тут: <http://34.88.23.22:8080/job/DZ4/31/>

 Ответить

 Переслать

