

Homework_Lesson28_Report

Цель: Научиться работать с Jenkins Pipeline, Groovy, а также автоматизировать сборку, тестирование и развертывание приложения.

Основные шаги: Настроить Jenkins – установить и запустить Jenkins, если он еще не установлен.

Создать Jenkins Pipeline с использованием декларативного синтаксиса.

Определить агент (где будет выполняться сборка: любой сервер, Docker-контейнер и т. д.).

Добавить этапы (stages): Build – сборка приложения (например, с помощью Gradle или Maven). Test – запуск тестов. Deploy – развертывание приложения.

Определить шаги (steps) – команды, которые будут выполняться на каждом этапе (например, запуск команд в терминале sh, клонирование репозитория git).

Добавить условия выполнения (when) – запуск определенных этапов в зависимости от параметров (например, запуск только при изменениях в коде).

Использовать параметры (parameters) – например, выбор версии приложения или окружения.

Добавить обработку ошибок – предусмотреть проверку на ошибки, чтобы в случае проблем процесс не зависал. Запустить Pipeline в Jenkins и проверить его выполнение.

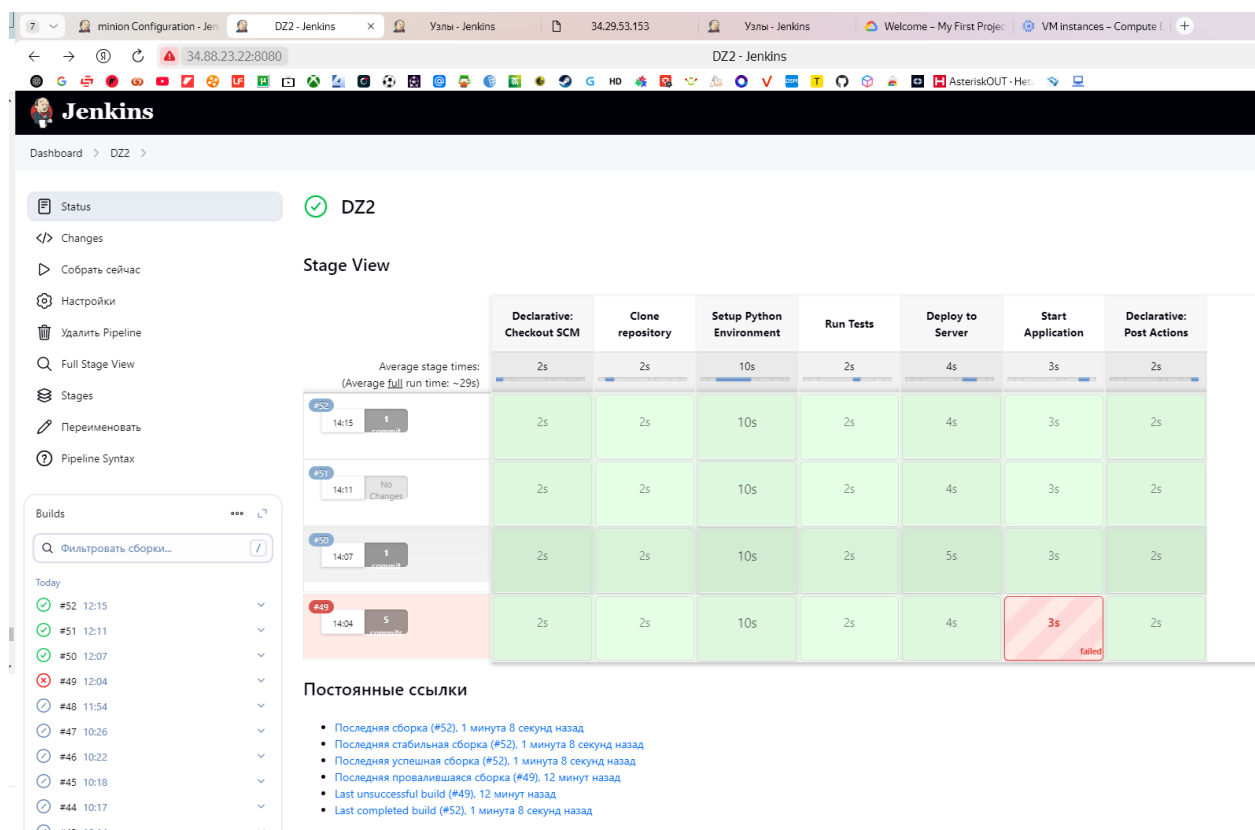
Задokumentировать Pipeline – описать структуру, шаги и параметры в файле README. Дополнительное задание (необязательно): Написать Groovy-скрипт, который: Клонировать репозиторий с веб-приложением. Собирает приложение (Gradle/Maven). Создает Docker-образ. Запускает контейнер с приложением.

Проверяет, что приложение работает. Автоматизация отчетов: Создать DSL (мини-язык) для генерации отчетов. Автоматизировать процесс сбора данных и формирование отчета. Добавить обработку ошибок (например, если данных нет).

Выполнение

У нас есть Супер мего приложение на pythone + flask которое открывает на порту 8080 html страничку с парой картинок app.py. Картинки у нас в папке со статическими файлами находится. Так же в папке tests у нас есть простейший тест, который проверяет статус ответа сервера 200 у нашего приложения, а также присутствует слово в ответе Information. Так же вместе с нашим приложением лежит скрипт на баше start_app.sh. Он запускает приложение и записывает его pid в файл. Если при запуске файл с pid процесса присутствует он закрывает этот pid и запускает приложение заново и снова записывает pid. Это сделано для того что бы после диплоя оставить приложение запущенным и при повторном диплое прошлая версия приложения останавливалась.

Скрин выполненного пайплайна.

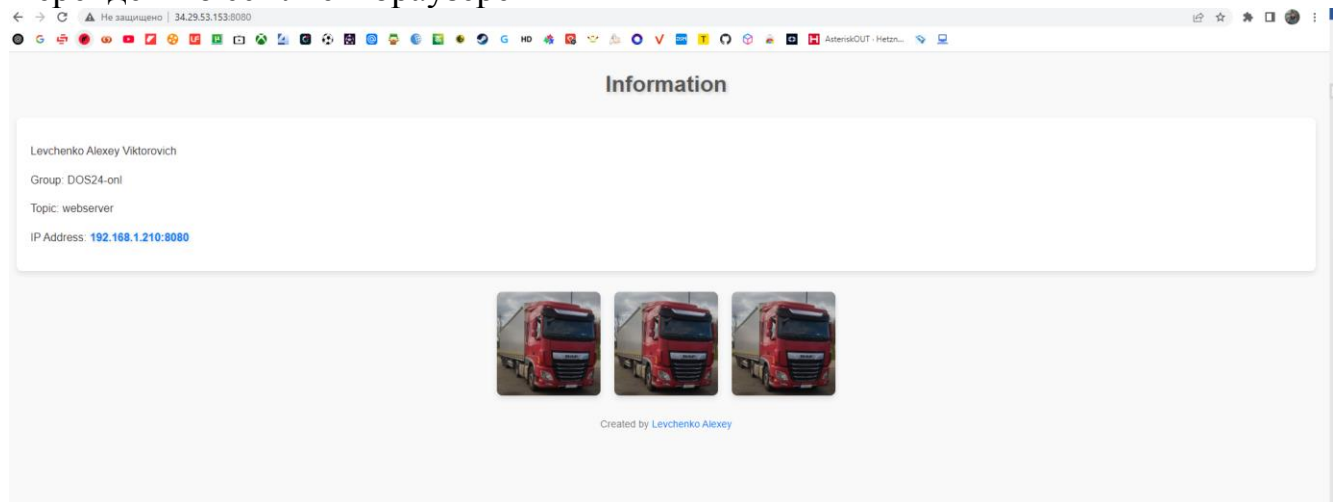


Наш пайплайн работает на агенте. При запуске клонирует репозиторий далее создает виртуальное окружение и устанавливает зависимости, запускает тесты, деплоит наше приложение на машину с миньоном в каталог /var/www/myapp далее запускает приложение с помощью скрипта на баше и в конце выводит сообщение о результате выполнения пайплайна.

Проверяем запустилось ли наше приложение

```
alex5482671al@minion:/var/www/myapp$ sudo netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN      4862/python3
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      357/systemd-resolve
tcp        0      0 127.0.0.1:6010          0.0.0.0:*               LISTEN      1054/sshd: alex5482
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN      357/systemd-resolve
tcp6       0      0 :::22                   :::*                    LISTEN      1/init
tcp6       0      0 :::1:6010               :::*                    LISTEN      1054/sshd: alex5482
udp        0      0 127.0.0.54:53           0.0.0.0:*               357/systemd-resolve
udp        0      0 127.0.0.53:53           0.0.0.0:*               357/systemd-resolve
udp        0      0 10.128.0.2:68           0.0.0.0:*               521/systemd-network
udp        0      0 127.0.0.1:323           0.0.0.0:*               665/chronyd
udp6       0      0 :::1:323                 :::*                    665/chronyd
alex5482671al@minion:/var/www/myapp$
```

Перейдем по ссылке в браузере



Файлы с приложением

```
alex5482671al@minion:/var/www/myapp$ ls -l
total 40
-rw-rw-r-- 1 jenkins jenkins 3875 Feb 18 12:15 Jenkinsfile
drwxrwxr-x 2 jenkins jenkins 4096 Feb 18 12:15 __pycache__
-rw-rw-r-- 1 jenkins jenkins 877 Feb 18 12:43 app.log
-rw-rw-r-- 1 jenkins jenkins 5 Feb 18 12:15 app.pid
-rw-rw-r-- 1 jenkins jenkins 3270 Feb 18 12:15 app.py
-rw-rw-r-- 1 jenkins jenkins 1259 Feb 18 10:23 start_app.py
-rw-rw-r-- 1 jenkins jenkins 979 Feb 18 12:15 start_app.sh
drwxrwxr-x 3 jenkins jenkins 4096 Feb 18 12:15 static
drwxrwxr-x 3 jenkins jenkins 4096 Feb 18 12:15 tests
drwxrwxr-x 6 jenkins jenkins 4096 Feb 18 12:15 venv
```