

Dockerize Flask приложение

№ урока: 8 Курс: Flask

Средства обучения: Python3 и любимая среда разработки, например, PyCharm.

Обзор, цель и назначение урока

Узнаем, что такое Docker и Docker контейнер, рассмотрим причины использования Docker, упакуем наше Flask приложение в Docker контейнер и развернем его на Heroku.

Изучив материал данного занятия, учащийся сможет:

- Создавать собственные Docker контейнеры.
- Использовать Docker и docker-compose в разработке.
- Развертывать Flask приложение на Heroku.
- Запускать Flask приложение в production mode с помощью Gunicorn.

Содержание урока

1. Docker, docker контейнеры
2. Упаковка Flask приложения в Docker контейнер
3. Docker-compose
4. Развертывание Flask приложения на Heroku и запуск с помощью Gunicorn

Резюме

- **Docker** - программное обеспечение, которое позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, а также предоставляет среду по управлению контейнерами. С помощью Docker вы можете быстро и последовательно доставлять приложения.
- **Docker Image** - образ в Docker является шаблоном только для чтения с набором инструкций для создания контейнера, они являются **иммутабельными**. Можно рассматривать его как snapshot.
- **Что такое контейнер?** — это работающий экземпляр образа. Каждый контейнер является изолированной и безопасной платформой, но может получить доступ к ресурсам, запущенным на других хостах или контейнерах, а также к хранилищам и базам данных.
- **Dockerfile** — это простой текстовый файл, в котором содержится список команд Docker-клиента. Это простой способ автоматизировать процесс создания образа.
- **ENTRYPOINT** - команда ENTRYPOINT определяет команду, которая должна выполняться при запуске контейнера.
- **CMD** - CMD используется для передачи аргументов по умолчанию для ENTRYPOINT или для выполнения специальной команды в контейнере.
- Если вам нужно при каждом запуске контейнера выполнять одну и ту же команду – используйте ENTRYPOINT, а в тех случаях, когда вы знаете, что при запуске контейнера вам понадобится передавать ему аргументы, которые могут перезаписывать аргументы, то используйте CMD.
- **Docker-compose** — это инструмент, который позволяет создавать и запускать несколько образов Docker одновременно. Вместо того, чтобы каждый раз проводить одни и те же команды, когда вы хотите запустить свое приложение, можно уместить их в одну – как только вы предоставите конкретную конфигурацию.

- Разница между Docker и docker-compose в том, что Docker применяется для управления отдельными контейнерами (сервисами), из которых состоит приложение, а docker-compose используется для одновременного управления несколькими контейнерами, входящими в состав приложения. Этот инструмент предлагает те же возможности, что и Docker, но позволяет работать с более сложными приложениями.
- **Heroku** - Heroku, одна из первых облачных платформ, которая появилась в июне 2007 года и изначально поддерживала только язык программирования Ruby, но на данный момент список поддерживаемых языков также включает в себя Java, Node.js, Scala, Clojure, Go, PHP и, конечно, Python.
- Развертывание веб-приложения в Heroku выполняется с помощью средства управления версиями git, поэтому приложение должно находиться в репозитории git. Heroku ищет файл под названием **Procfile** в корневом каталоге приложения для получения инструкций о том, как запустить приложение.
- **Gunicorn** - автономный веб-сервер с обширной функциональностью, предоставленной в удобном виде. Он изначально поддерживает различные фреймворки и адаптеры, что делает его чрезвычайно простой в использовании прямой заменой для многих серверов разработки.
Технически Gunicorn работает подобно Unicorn, популярному веб-серверу приложений Ruby. Они оба используют так называемую pre-fork модель (это значит, что главный процесс управляет инициированными рабочими процессами различного типа, создает сокеты и соединения, и т.п.).

Закрепление материала

- Что такое Docker?
- Какую задачу решает Docker?
- Что такое Docker Image?
- Что такое Docker Container?
- Что такое Dockerfile?
- Разница между Docker и docker-compose?
- CMD vs ENTRYPOINT
- Что такое Gunicorn?

Дополнительное задание

Задание

Вам нужно создать образ на основе docker/whalesay:latest. Контейнер должен иметь настроенный исполняемый файл, который читает сообщение из файла и передать его программе cowsay. Для этого нужно создать собственный Dockerfile и определить CMD и COPY.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Изучите материалы из рекомендуемых ресурсов.

Задание 3

Создайте собственный аккаунт на Docker Hub, скачайте образ Ubuntu и отправьте команду, чтобы отобразилось имя пользователя в системе (команда who).

Рекомендуемые ресурсы

Docker tutorial:

<https://docs.docker.com/get-started/overview/>

Best practices for writing Dockerfiles:

https://docs.docker.com/develop/develop-images/dockerfile_best-practices/

Docker Commands, Help & Tips:

<https://gist.github.com/bradtraversy/89fad226dc058a41b596d586022a9bd3>

Containerized Python Development:

<https://www.docker.com/blog/containerized-python-development-part-1/>

Heroku deploy:

<https://devcenter.heroku.com/articles/getting-started-with-python>