

Развертывание приложений в Docker (Развертывание Python-скрипта в Docker)

Задание:

4) Развертывание приложений в Docker

- а) Общие требования:
 - Все действия выполняются на машине ControlVM. Выполнить развертывание Python-скрипта в Docker, настроить WordPress с использованием Docker Compose и развернуть базовый стек ELK для сбора и отображения логов.
- б) Развертывание Python-скрипта в Docker
 - 1. Напишите Python-скрипт 'py.py' в домашней директории пользователя **py.py**, который выполняет следующие задачи:
 - i. Проверяет наличие файла input.txt в рабочей директории root.
 - ii. Выводит сообщение с содержимым.
 - iii. Если файла input.txt нет, выводит сообщение об ошибке.
 - 2. Создайте Dockerfile для Python-скрипта 'py.py' ~~file-copy-python~~:
 - i. Используйте базовый образ python:3.8-alpine.
 - ii. Python-скрипт py.py должен выполняться внутри контейнера.
 - iii. Реализуйте копирование файла input.txt в контейнер (этот файл может содержать произвольный текст).
 - iv. Контейнер при запуске должен выводит содержимое файла input.txt, после чего завершать свою работу.
 - 3. Сборка и запуск контейнера:
 - i. Соберите Docker-образ с именем file-copy-python-~~ymt~~.
 - ii. Запустите контейнер и убедитесь, что содержимое файла выводится файл input.txt.

Вариант реализации:

ControlVM:

- В домашней директории пользователя **altlinux** из под пользователя **altlinux** создаём файл **py.py**:

```
vim ~/py.py
```

- Помещаем в него следующее содержимое:
 - Реализуем функционал согласно требованиям задания:

```
import os

def main():
    working_directory = os.path.expanduser("/root")
    file_path = os.path.join(working_directory, "input.txt")

    if os.path.exists(file_path):
        with open(file_path, "r", encoding="utf-8") as file:
            content = file.read()
            print(content)
    else:
        print("Ошибка: файл input.txt не найден в директории /root.")

if __name__ == "__main__":
    main()
```

- Проверяем функционал скрипта:
 - Вывод об ошибке в случае отсутствия файла:

```
[altlinux@controlvm bin]$ sudo python3 ~/py.py
Ошибка: файл input.txt не найден в директории /root.
[altlinux@controlvm bin]$
```

- Создаём файл и запускаем скрипт повторно:

```
[altlinux@controlvm bin]$ sudo su -
[root@controlvm ~]# echo "Hello au.team!" > input.txt
[root@controlvm ~]# exit
выход
[altlinux@controlvm bin]$ sudo python3 ~/py.py
Hello au.team!
[altlinux@controlvm bin]$
```

- В домашней директории пользователя **altlinux** из под пользователя **altlinux** создаём файл **Dockerfile**:

```
vim ~/Dockerfile
```

- Помещаем в него следующее содержимое:

- Реализуем функционал согласно требованиям задания:

```
FROM python:3.8-alpine
WORKDIR /root
COPY py.py .
COPY input.txt .
CMD ["python", "py.py"]
```

- Устанавливаем **docker**:

```
sudo apt-get install -y docker-engine
```

- Включаем и добавляем в автозагрузку службу **docker**:

```
sudo systemctl enable --now docker.service
```

- Запускаем сборку docker-образа из ранее описанного Dockerfile:
 - Имя собранного образа в данном случае будет **file-copy-python**
 - Запуск команды выполняется из домашнего каталога пользователя **altlinux**, в текущем каталоге находится сам **Dockerfile**, файл скрипта **py.py** и файл **input.txt**;

```
sudo docker build -t file-copy-python .
```

- Результат:

```
[altlinux@controlvm ~]$ sudo docker build -t file-copy-python .
[+] Building 86.5s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/python:3.8-alpine
=> [internal] load .dockerignore
=> [internal] load context: 2B
=> [1/4] FROM docker.io/library/python:3.8-alpine@sha256:3d93b1f77efce339aa77db726656872517b0d67837989aa7c4b35bd5ae7e81ba
=> resolve docker.io/library/python:3.8-alpine@sha256:3d93b1f77efce339aa77db726656872517b0d67837989aa7c4b35bd5ae7e81ba
=> sha256:3d93b1f77efce339aa77db726656872517b0d67837989aa7c4b35bd5ae7e81ba 10.29kB / 10.29kB
=> sha256:9f956f4ecf00137962b8da05d2ced2181cf22e23036a81daa61610e16ddd1078 1.73kB / 1.73kB
=> sha256:cd39247b02f92df84605637e0f1b7887434169aca105b8ee9cbcd2dbfc988ed5 4.95kB / 4.95kB
=> sha256:43c4264eed91be63b206e17d93e75256a6097070ce643c5e8f0379998b44f170 3.62MB / 3.62MB
=> sha256:bbd039b2f2dc0cbe5a1fb925d1006dc58892cd76b385af7243c1d668d7ce2491 455.12kB / 455.12kB
=> sha256:2fe9ed713579c200eb3576e25f4604247c6b5d36482a5bb477de3c8339274f0f 14.22MB / 14.22MB
=> sha256:86e037ebf8b67184256b8126401150c6d34e6d9977b5f320cb4a52e91e31fc41 247B / 247B
=> extracting sha256:43c4264eed91be63b206e17d93e75256a6097070ce643c5e8f0379998b44f170
=> extracting sha256:bbd039b2f2dc0cbe5a1fb925d1006dc58892cd76b385af7243c1d668d7ce2491
=> extracting sha256:2fe9ed713579c200eb3576e25f4604247c6b5d36482a5bb477de3c8339274f0f
=> extracting sha256:86e037ebf8b67184256b8126401150c6d34e6d9977b5f320cb4a52e91e31fc41
=> [internal] load build context
=> transferring context: 528B
=> [2/4] WORKDIR /root
=> [3/4] COPY py.py .
=> [4/4] COPY input.txt .
=> exporting to image
=> exporting layers
=> writing image sha256:4ba2cf9d476cc4c56472ff6f93a61385574d495b60c420765e07562ea2e39984
=> naming to docker.io/library/file-copy-python
[altlinux@controlvm ~]$
```

- Проверяем наличие собранного образа:

```
[altlinux@controlvm ~]$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
file-copy-python latest 4ba2cf9d476c About a minute ago 47.4MB
[altlinux@controlvm ~]$
```

- Запускаем docker-контейнер из собранного образа и проверяем что содержимое файла **input.txt** выводится на экран:

```
sudo docker run --rm file-copy-python
```

- Результат:

```
[altlinux@controlvm ~]$ sudo docker run --rm file-copy-python
Hello au.team!
[altlinux@controlvm ~]$
```

Последнее изменение: четверг, 16 января 2025, 08:36

Обратная связь

Подпишитесь

Вы используете гостевой доступ (Вход)

Сводка хранения данных

Тема оформления сайта разработана

conecti.me