

Пример Dockerfile, основанного на образе Alpine, который содержит простое приложение на Python "Hello, World":

```
FROM alpine:latest

# Установка зависимостей
RUN apk update && \
    apk add --no-cache python3 py3-pip

# Копирование файлов приложения в контейнер
COPY app.py /

# Установка зависимостей Python
RUN pip3 install flask

# Определение переменной среды для запуска приложения
ENV FLASK_APP=/app.py

# Открытие порта
EXPOSE 5000

# Запуск приложения
CMD ["flask", "run", "--host=0.0.0.0"]
```

В этом примере Dockerfile мы используем образ Alpine Linux, устанавливаем Python 3 и Flask, копируем файл `app.py` (содержащий код для приложения "Hello, World") в контейнер, устанавливаем переменную среды `FLASK_APP`, чтобы указать Flask, какой файл следует запускать, открываем порт 5000 и используем команду `CMD` для запуска приложения Flask.

Создайте файл `app.py` в той же директории, где находится Dockerfile, и добавьте в него следующий код для простого приложения Flask "Hello, World":

```
```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello():
 return "Hello, World!"

if __name__ == '__main__':
 app.run()
```
```

После этого вы можете построить образ Docker, выполнив команду `docker build`:

```
```
docker build -t myapp .
```
```

После успешной сборки образа вы можете запустить контейнер, выполнив команду ``docker run``:

```

```
docker run -d -p 5000:5000 myapp
```

```

Теперь ваше простое приложение Flask "Hello, World" будет доступно на порту 5000 внутри контейнера, который привязан к порту 5000 на вашем хосте. Вы можете посмотреть его через команду ``curl localhost:5000``, чтобы увидеть "Hello, World!" сообщение.