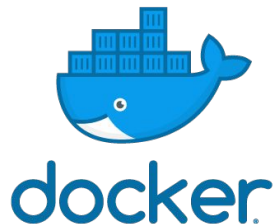


Обзорная экскурсия в



Дмитриев Дмитрий
SmartHead

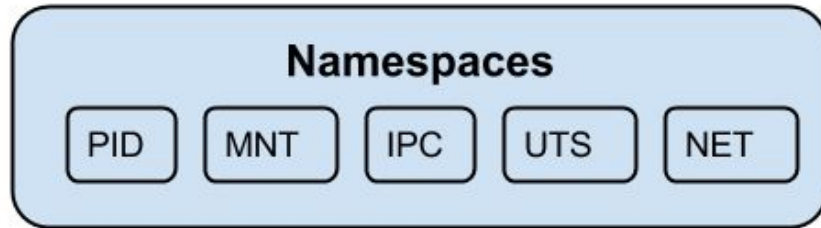
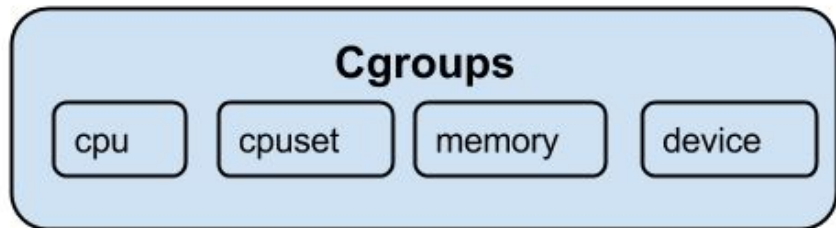
- Что такое docker?
- Что внутри контейнера?
- Отличия контейнеров от виртуальных машин
- Где применить?
- Создание образа и запуск контейнера
- Создание много-контейнерных приложений

Что такое docker

Docker – это инструмент для автоматизации развертывания приложений в виде переносимых, автономных контейнеров, выполняющихся в изолированном окружении.

Что такое docker

Docker – это инструмент для автоматизации развертывания приложений в виде переносимых, автономных контейнеров, выполняющихся в изолированном окружении.



Что такое docker

Docker – это инструмент для автоматизации развертывания приложений в виде переносимых, автономных контейнеров, выполняющихся в изолированном окружении.

Docker – это стандартизированный интерфейс для создания и управления контейнерами.

Что такое docker

Docker – это инструмент для автоматизации развертывания приложений в виде переносимых, автономных контейнеров, выполняющихся в изолированном окружении.

Docker – это стандартизированный интерфейс для создания и управления контейнерами.

Docker – это своеобразная система контроля версий для инфраструктуры.

Что внутри контейнера?

- Слой образа

Что внутри контейнера?

- Слой образа
- Образ

Образ

Layer 5 73dc7dba3d10

Layer 4 34c02351e4b1

Layer 3 501a36e590d1

Layer 2 e73defc464ab

Layer 1 dad26d192ae6

Что внутри контейнера?

- Слой образа
- Образ
- Контейнер

Контейнер

Layer 6 51354a7e0bc1

Layer 5 73dc7dba3d10

Layer 4 34c02351e4b1

Layer 3 501a36e590d1

Layer 2 e73defc464ab

Layer 1 dad26d192ae6

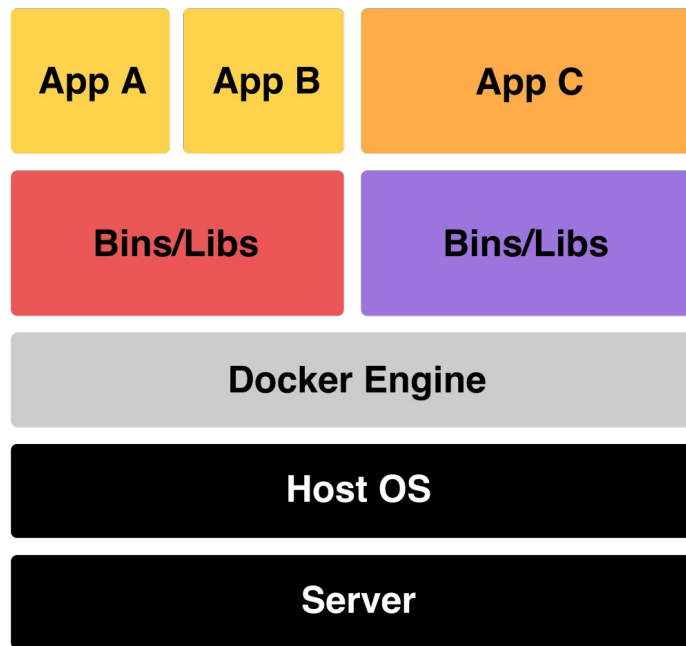
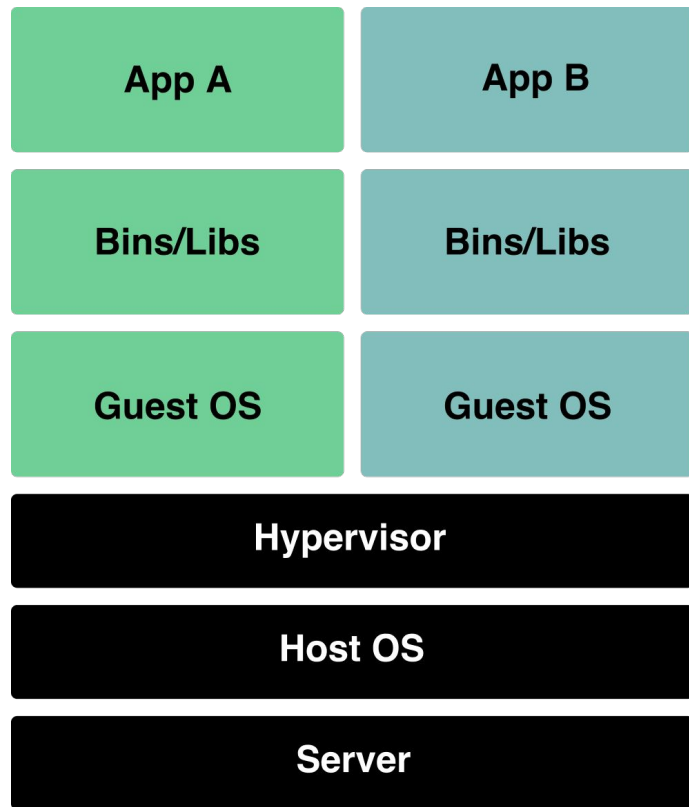
Что внутри контейнера?

- Слой образа
- Образ
- Контейнер
- Том

Что внутри контейнера?

- Слой образа
- Образ
- Контейнер
- Том
- DockerFile

Отличия контейнеров от виртуальных машин



Где применить?

- Тестирование

Где применить?

- Тестирование
- Доставка готового продукта

Где применить?

- Тестирование
- Доставка готового продукта
- Быстрое развертывание окружения (dev, stage, production)

Где применить?

- Тестирование
- Доставка готового продукта
- Быстрое развертывание окружения (dev, stage, production)
- Тиражируемость

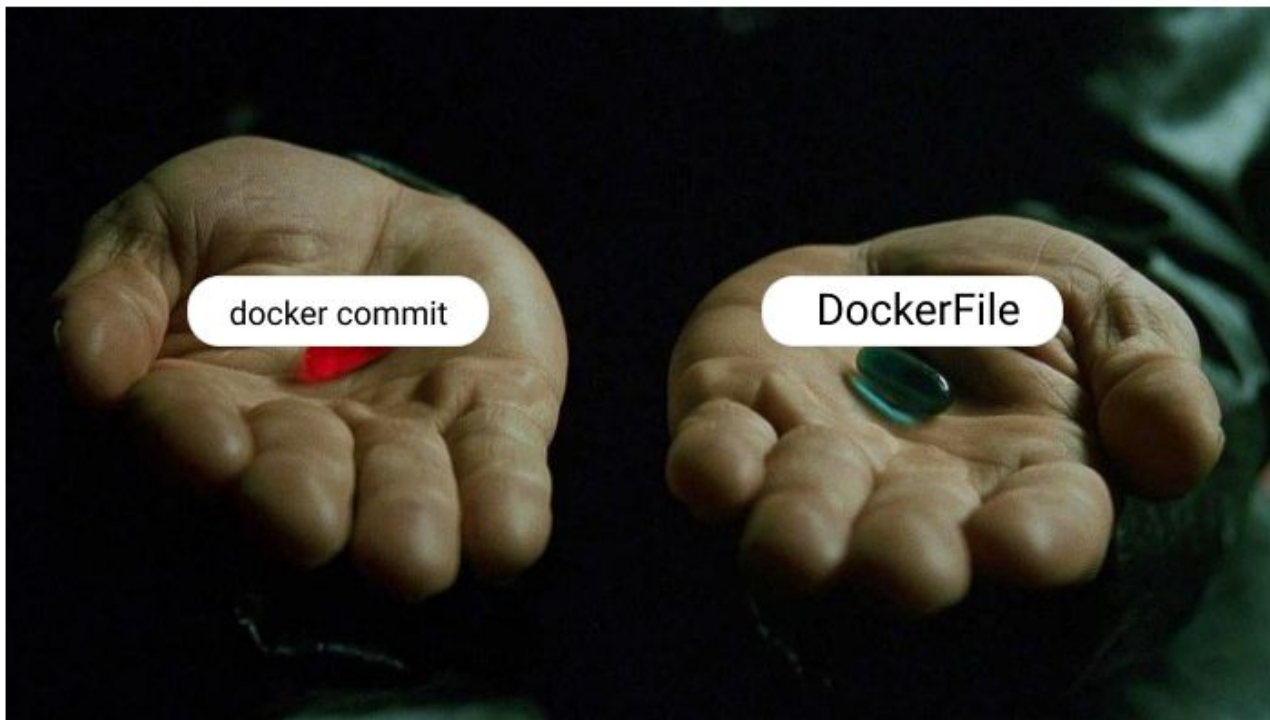
Где применить?

- Тестирование
- Доставка готового продукта
- Быстрое развертывание окружения (dev, stage, production)
- Тиражируемость
- Переиспользуемость

Где применить?

- Тестирование
- Доставка готового продукта
- Быстрое развертывание окружения (dev, stage, production)
- Тиражируемость
- Переиспользуемость
- Создание распределенных систем

Создание образа и запуск контейнера



Создание образа и запуск контейнера

```
FROM microsoft/dotnet:2.2-sdk
EXPOSE 80
EXPOSE 443
ENV ConnectionStrings__DefaultConnection="<connection_string>"
WORKDIR /app
COPY . .
RUN dotnet publish -c Release -o /out/
WORKDIR /out
ENTRYPOINT ["dotnet", "Sample.DockerApplication.Web.dll"]
```

Создание образа и запуск контейнера

- **711cb07ff8d1** example:latest /bin/sh -c #(nop) ENTRYPOINT ["dotnet" "Sample.DockerApplication.Web.dll"] (1.65 GB)
 - **63dfa24de8d2** /bin/sh -c #(nop) WORKDIR /out (1.65 GB)
 - **a12946152e32** /bin/sh -c dotnet publish -c Release -o /out/ (1.65 GB)
 - **1e11cfa7e59c** /bin/sh -c #(nop) COPY dir:4a5cce4e97979c991f21ce in . (1.63 GB)
 - **f9d0bdc6b643** /bin/sh -c #(nop) WORKDIR /app (1.61 GB)
 - **27dda73187fa** /bin/sh -c #(nop) ENV ConnectionStrings__DefaultConnection=<connection_string> (1.61 GB)
 - **63c9b173e3a2** /bin/sh -c #(nop) EXPOSE 443 (1.61 GB)
 - **450d2fd6ee9e** /bin/sh -c #(nop) EXPOSE 80 (1.61 GB)
 - **d5dbee97f68e** microsoft/dotnet:2.2-sdk /bin/sh -c dotnet help (1.61 GB)
 - scratch

Создание образа и запуск контейнера

```
FROM microsoft/dotnet:2.2-aspnetcore-runtime AS base
```

```
/* полезная нагрузка */
```

```
FROM microsoft/dotnet:2.2-sdk AS build
```

```
/* полезная нагрузка */
```

```
FROM build AS publish
```

```
/* полезная нагрузка */
```

```
FROM base AS final
```

```
WORKDIR /app
```

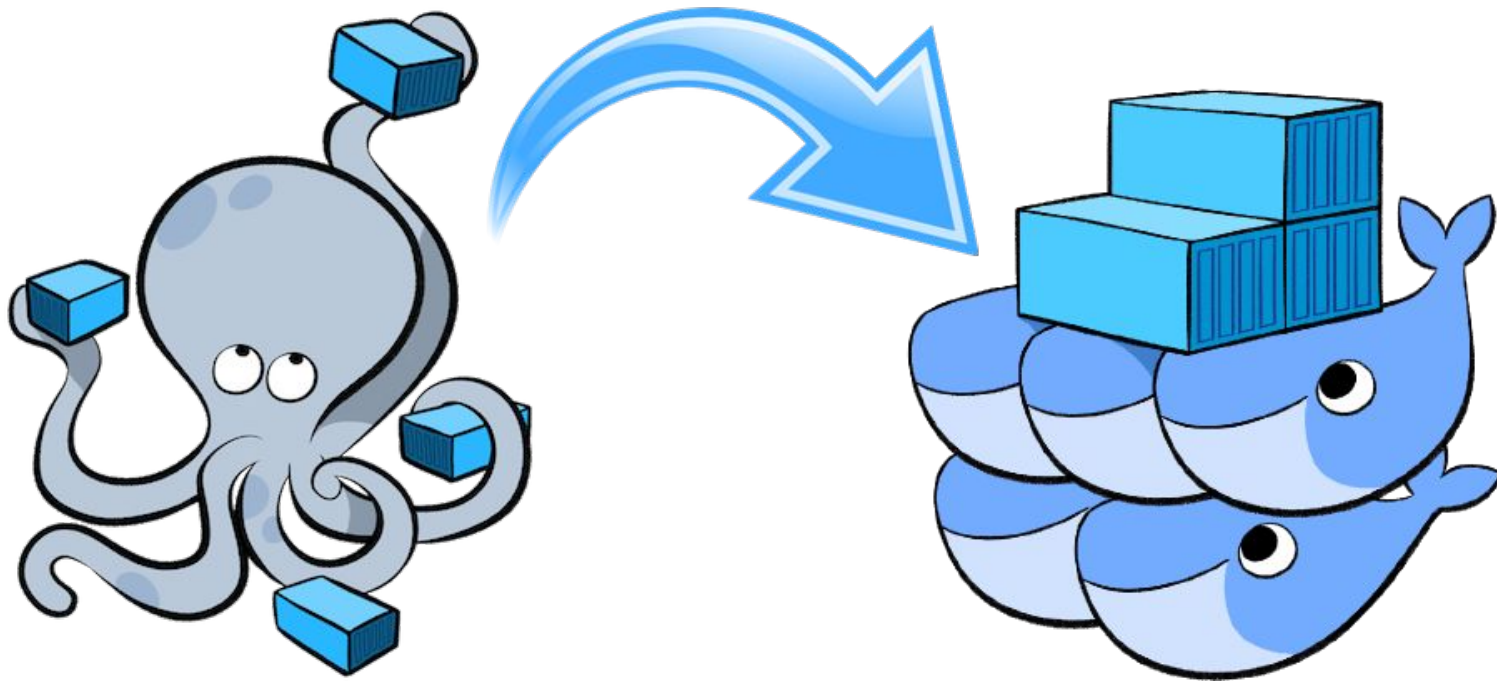
```
COPY --from=publish /app .
```

```
ENTRYPOINT ["dotnet", "Sample.DockerApplication.Web.dll"]
```


Создание образа и запуск контейнера

- **51354a7e0bc1** events.app:latest /bin/sh -c #(nop) ENTRYPOINT ["dotnet" "Sample.DockerApplication.Web.dll"] (251.89 MB)
 - **96b879fd0648** /bin/sh -c #(nop) COPY dir:9079eb30567caf70bec26 in . (251.89 MB)
 - **99de9245f2ab** /bin/sh -c #(nop) ENV ASPNETCORE_URLS=http://*:8082 (248.13 MB)
 - **73dc7dba3d10** /bin/sh -c #(nop) WORKDIR /app (248.13 MB)
 - **34c02351e4b1** /bin/sh -c #(nop) EXPOSE 443 (248.13 MB)
 - **501a36e590d1** /bin/sh -c #(nop) EXPOSE 8082 (248.13 MB)
 - **e73defc464ab** /bin/sh -c #(nop) WORKDIR /app (248.13 MB)
 - **dad26d192ae6** microsoft/dotnet:2.2-aspnetcore-runtime
 - scratch

Создание много-контейнерных приложений



version: '3'

services:

frontend:

image: events.frontend

container_name: events.frontend

build:

context: ./frontend/

dockerfile: DockerFile

ports:

- "8081:8081"

depends_on:

- backend

proxy:

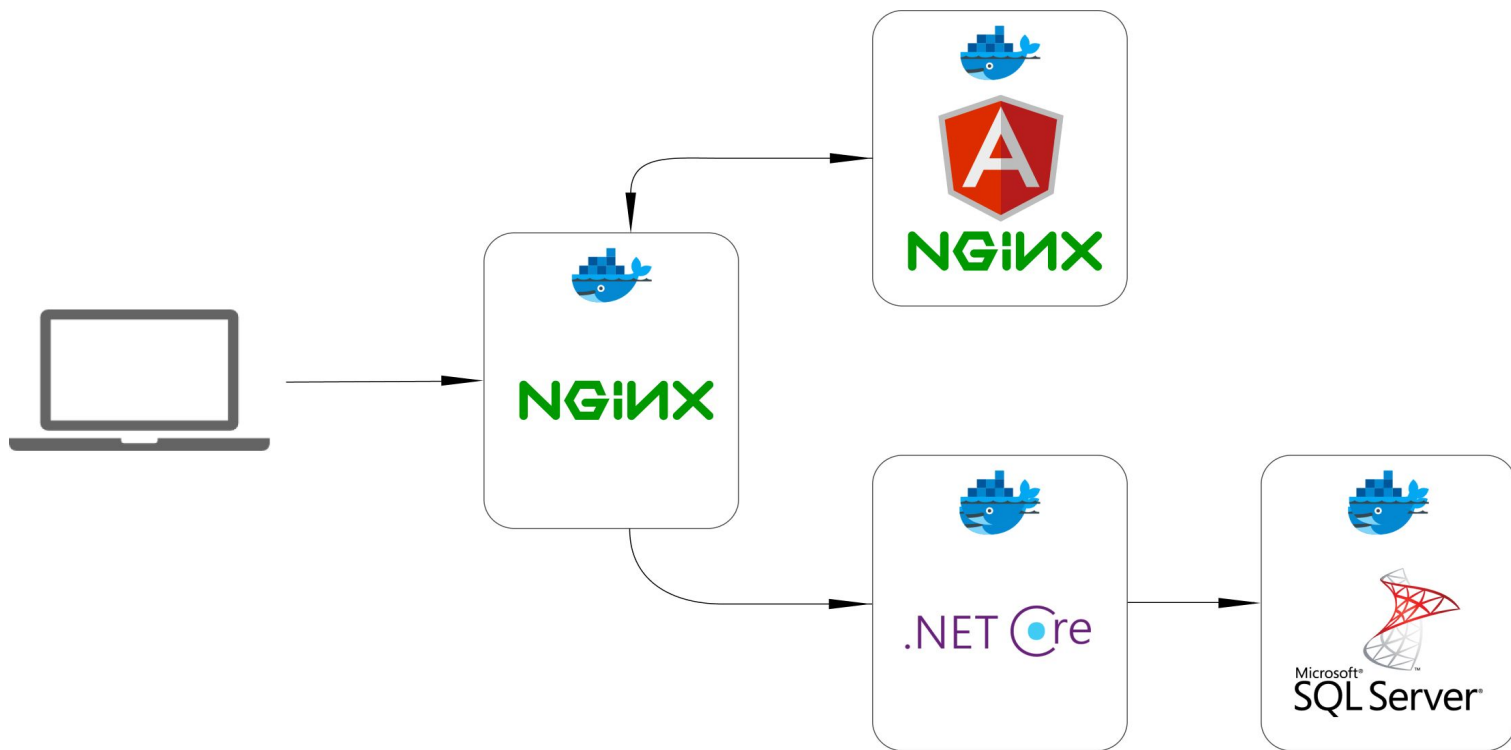
#полезная нагрузка

db:

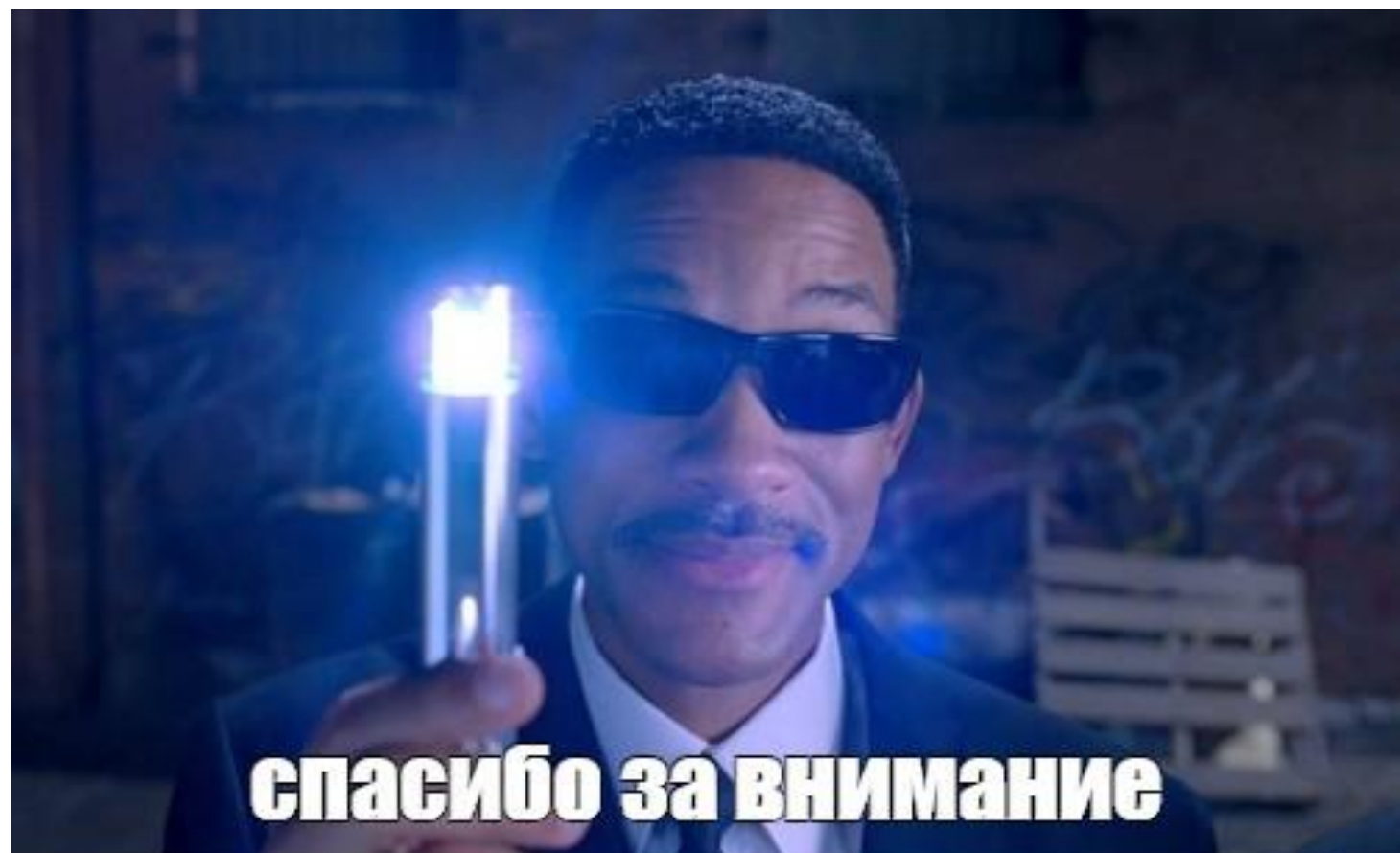
#полезная нагрузка

backend:

#полезная нагрузка



<https://github.com/qdimka/sample-docker-application>



спасибо за внимание

Ресурсы

<https://docs.docker.com/engine/reference/builder>

<https://github.com/veggemonk/awesome-docker>

<https://toster.ru/q/508290>

<https://habr.com/ru/company/flant/blog/336654>

<https://github.com/qdimka/sample-docker-application>