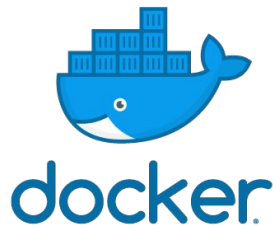
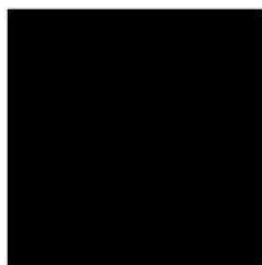


Обзорная экскурсия в



Дмитриев Дмитрий, SmartHead



[Redacted Name]

★ ★ ★ ★ ★ 04.08.2016

Не работает

Ответ разработчика

[Redacted Name]

от 04.08.2016

работает

Спойлеры

- Что такое docker?
- Контейнер != Образ
- DockerFile
- Multi-stage build и production
- Docker-Compose
- Взаимодействие контейнеров
- Вынесение данных из контейнера
- Tips&Tricks

Что такое docker?

Cgroups

cpu

cpuset

memory

device

Namespaces

PID

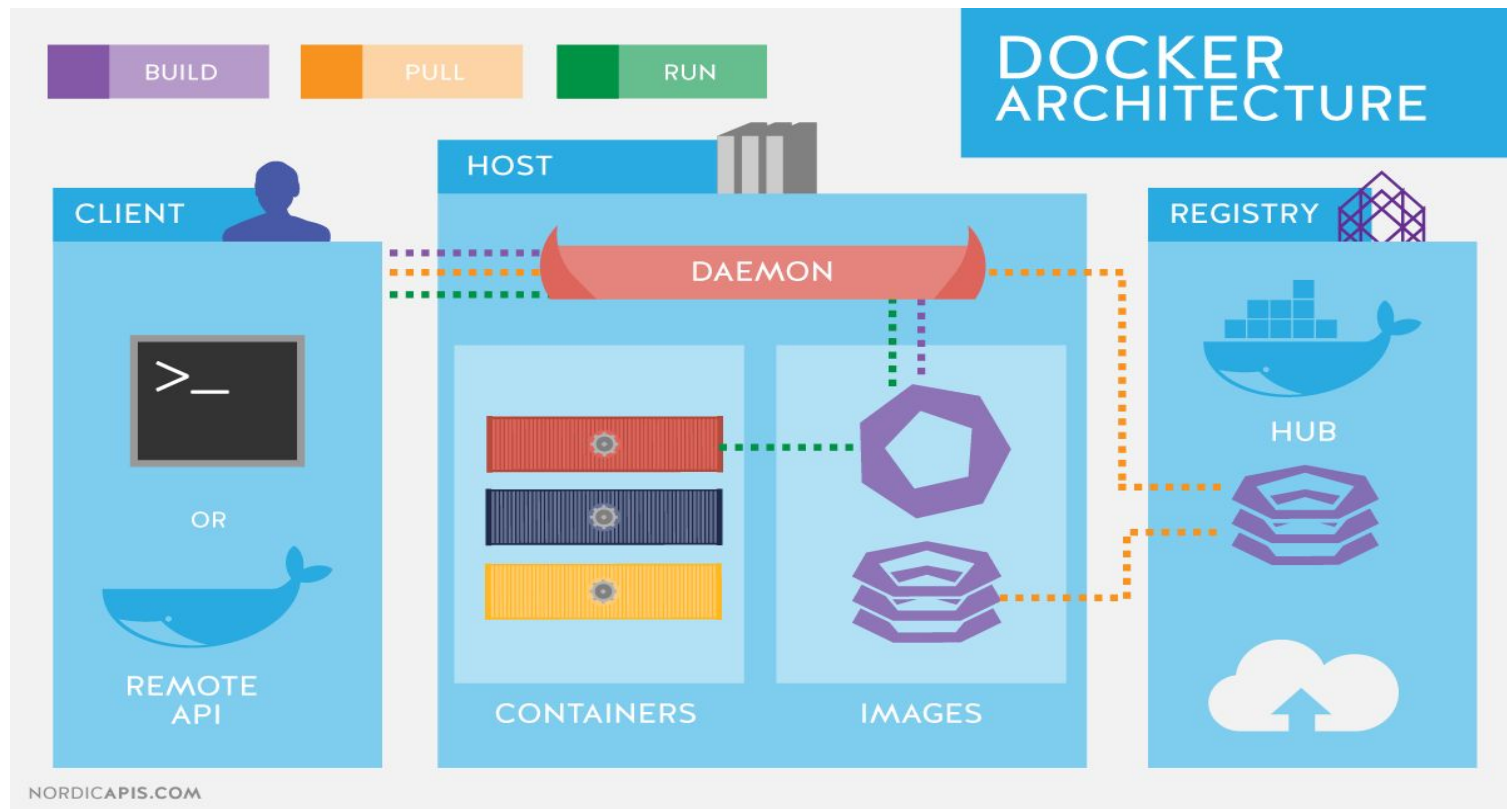
MNT

IPC

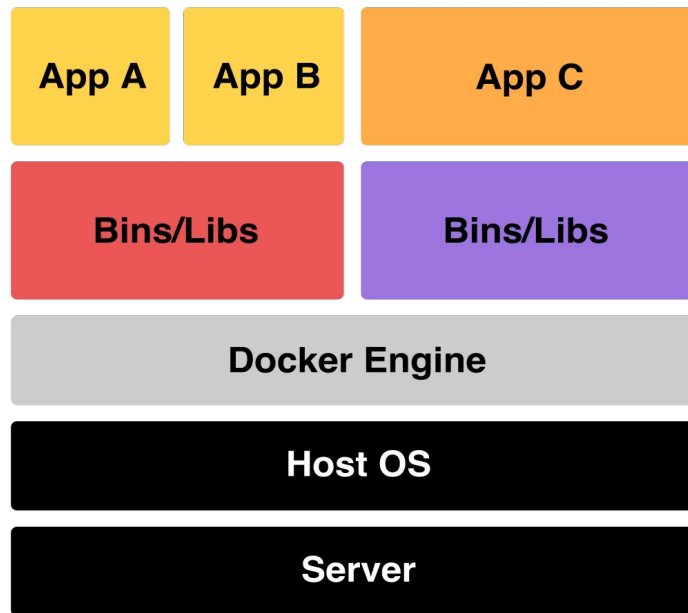
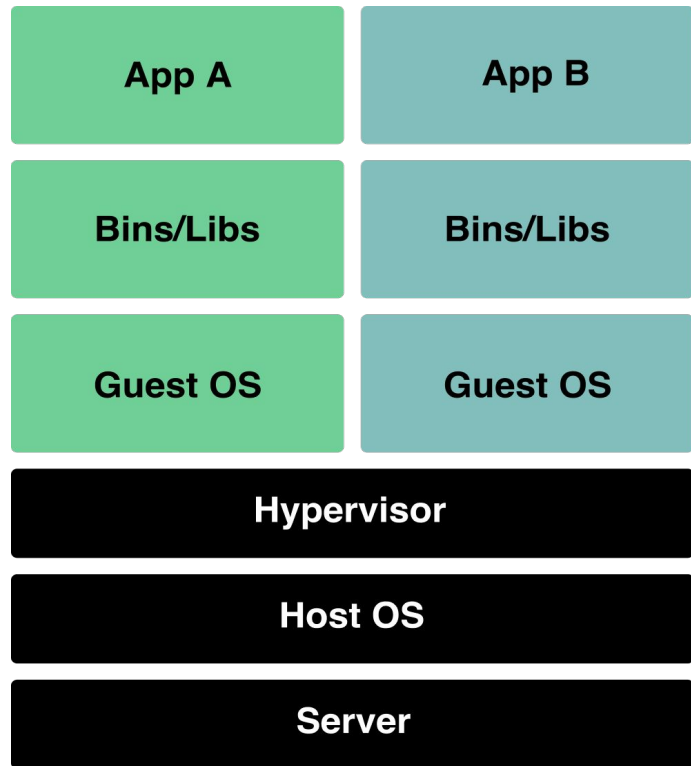
UTS

NET

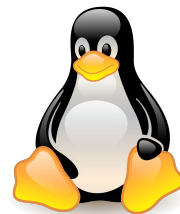
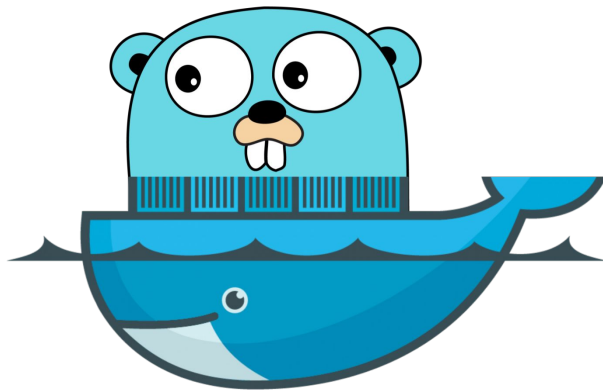
Что такое docker?



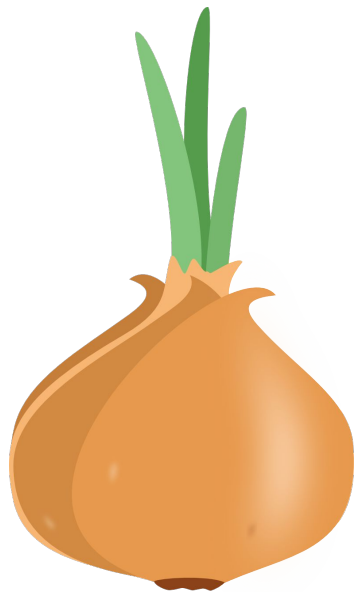
Что такое docker?



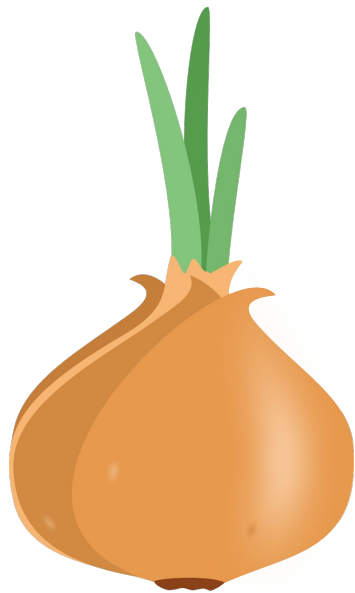
Что такое docker?



Контейнер != Образ



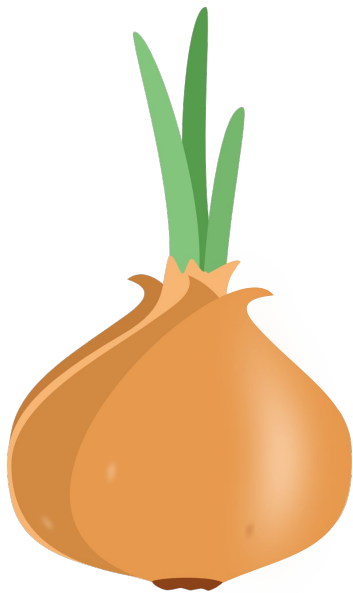
Контейнер != Образ



Образ

- Состоит из слоев

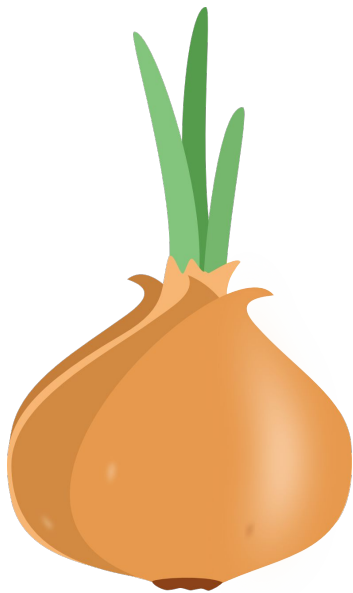
Контейнер != Образ



Образ

- Состоит из слоев
- Слои неизменяемы

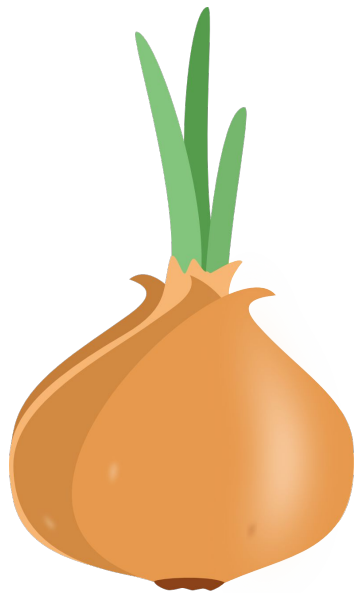
Контейнер != Образ



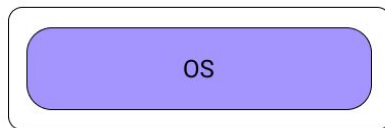
Образ

- Состоит из слоев
- Слои неизменяемы
- При изменении образа создается новый слой

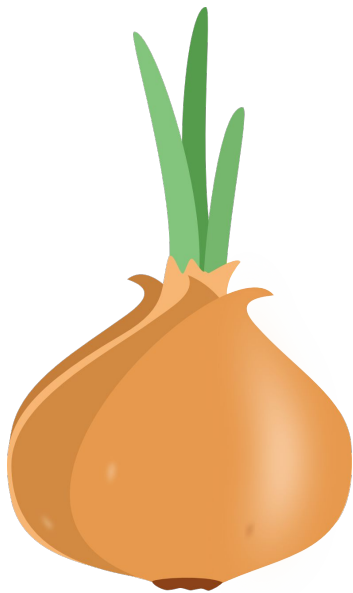
Контейнер != Образ



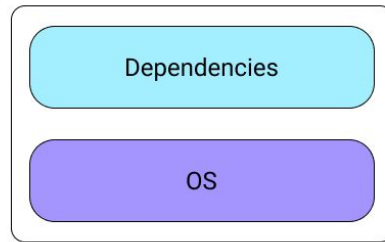
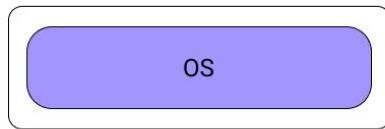
Образ



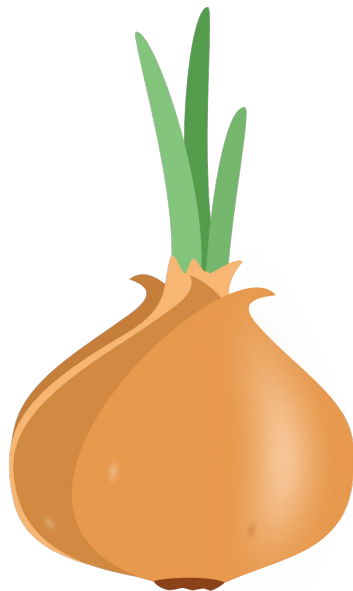
Контейнер != Образ



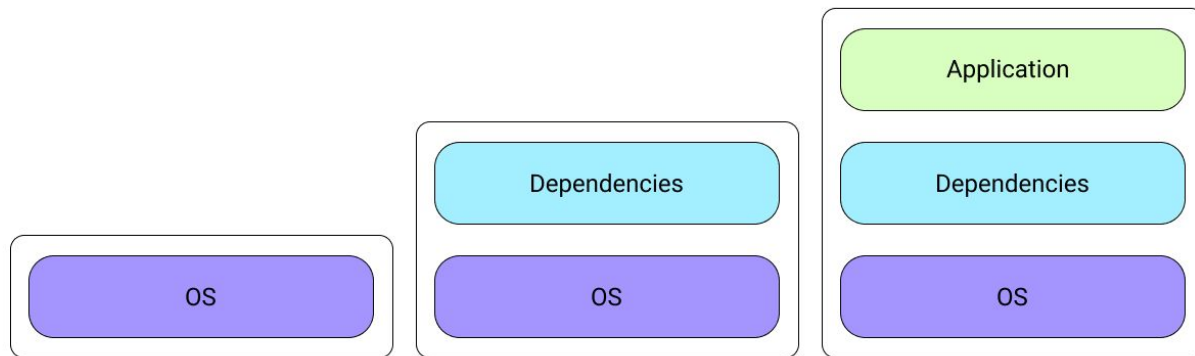
Образ



Контейнер != Образ



Образ



Контейнер != Образ

- Использует образ как шаблон



Контейнер

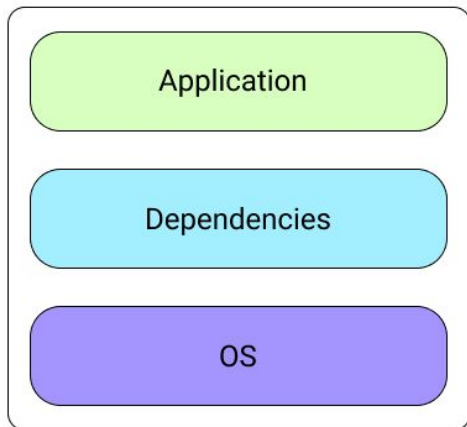
Контейнер != Образ

- Использует образ как шаблон
- Имеет слой доступный для записи



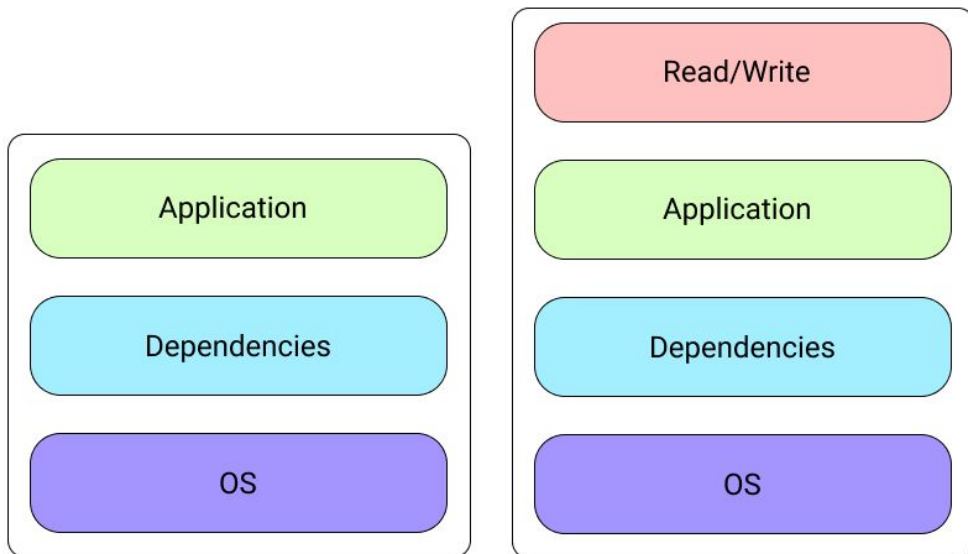
Контейнер

Контейнер != Образ



Контейнер

Контейнер != Образ



Контейнер

Контейнер != Образ

```
cbf2ff107de7 example:latest ENTRYPOINT ["dotnet" "App.dll"]  
(1.64 GB)  
47e49d6d139b ENV ASPNETCORE_URLS=http://*:8082 (1.64 GB)  
339e2781bf64 dotnet publish -c Release -o /app (1.64 GB)  
eaa324f93148 dotnet build -c Release -o /app (1.64 GB)  
5efae2b025ae COPY dir:7811293222 in . (1.64 GB)  
a5e29b6256f0 COPY file:2b33bc757 in . (1.62 GB)  
cf4b3ee54626 WORKDIR /src (1.62 GB)  
910d4027c6a3 EXPOSE 443 (1.62 GB)  
ea3eed67c634 EXPOSE 8082 (1.62 GB)  
f13ac9d68148 microsoft/dotnet:2.2-sdk dotnet help (1.62 GB)  
...
```

Контейнер != Образ

```
cbf2ff107de7 example:latest ENTRYPOINT ["dotnet" "App.dll"]  
(1.64 GB)  
47e49d6d139b ENV ASPNETCORE_URLS=http://*:8082 (1.64 GB)  
339e2781bf64 dotnet publish -c Release -o /app (1.64 GB)  
eaa324f93148 dotnet build -c Release -o /app (1.64 GB)  
5efae2b025ae COPY dir:7811293222 in . (1.64 GB)  
a5e29b6256f0 COPY file:2b33bc757 in . (1.62 GB)  
cf4b3ee54626 WORKDIR /src (1.62 GB)  
910d4027c6a3 EXPOSE 443 (1.62 GB)  
ea3eed67c634 EXPOSE 8082 (1.62 GB)  
f13ac9d68148 microsoft/dotnet:2.2-sdk dotnet help (1.62 GB)  
...
```

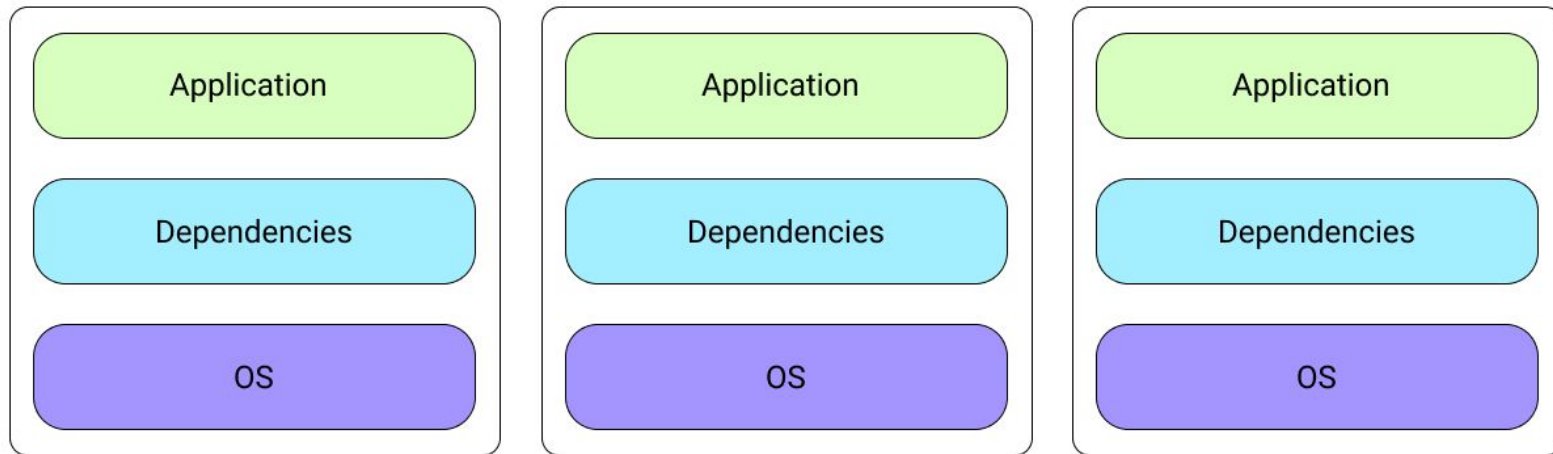
Контейнер != Образ

```
cbf2ff107de7 example:latest ENTRYPOINT ["dotnet" "App.dll"]  
(1.64 GB)  
47e49d6d139b ENV ASPNETCORE_URLS=http://*:8082 (1.64 GB)  
339e2781bf64 dotnet publish -c Release -o /app (1.64 GB)  
eaa324f93148 dotnet build -c Release -o /app (1.64 GB)  
5efae2b025ae COPY dir:7811293222 in . (1.64 GB)  
a5e29b6256f0 COPY file:2b33bc757 in . (1.62 GB)  
cf4b3ee54626 WORKDIR /src (1.62 GB)  
910d4027c6a3 EXPOSE 443 (1.62 GB)  
ea3eed67c634 EXPOSE 8082 (1.62 GB)  
f13ac9d68148 microsoft/dotnet:2.2-sdk dotnet help (1.62 GB)  
...
```

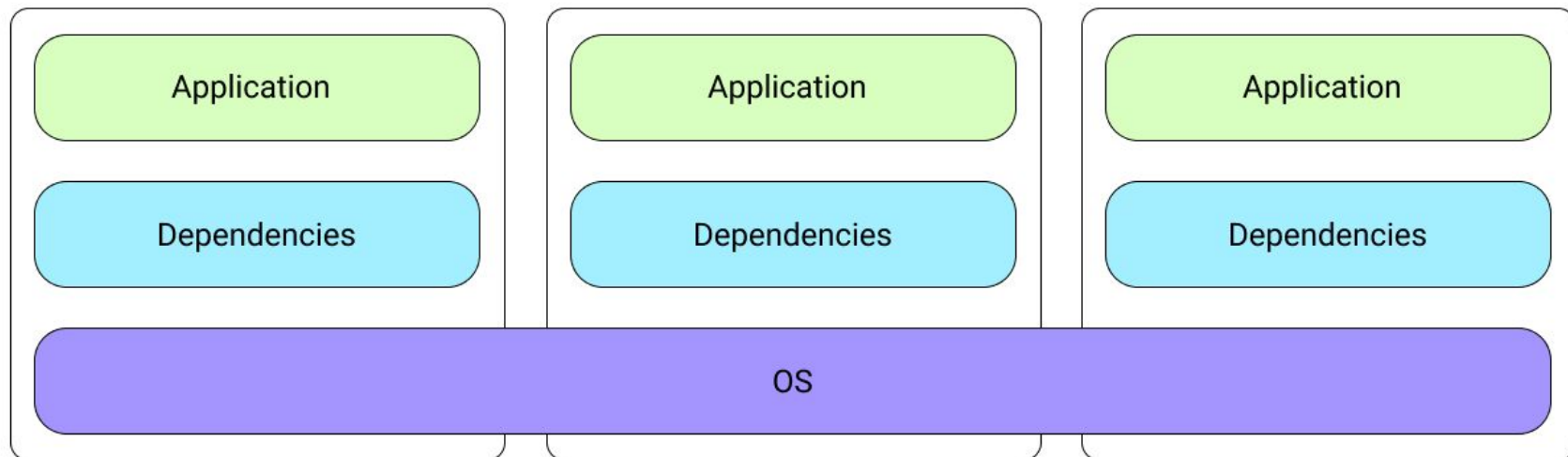
Контейнер != Образ

```
cbf2ff107de7 example:latest ENTRYPOINT ["dotnet" "App.dll"]  
(1.64 GB)  
47e49d6d139b ENV ASPNETCORE_URLS=http://*:8082 (1.64 GB)  
339e2781bf64 dotnet publish -c Release -o /app (1.64 GB)  
eaa324f93148 dotnet build -c Release -o /app (1.64 GB)  
5efae2b025ae COPY dir:7811293222 in . (1.64 GB)  
a5e29b6256f0 COPY file:2b33bc757 in . (1.62 GB)  
cf4b3ee54626 WORKDIR /src (1.62 GB)  
910d4027c6a3 EXPOSE 443 (1.62 GB)  
ea3eed67c634 EXPOSE 8082 (1.62 GB)  
f13ac9d68148 microsoft/dotnet:2.2-sdk dotnet help (1.62 GB)  
...
```

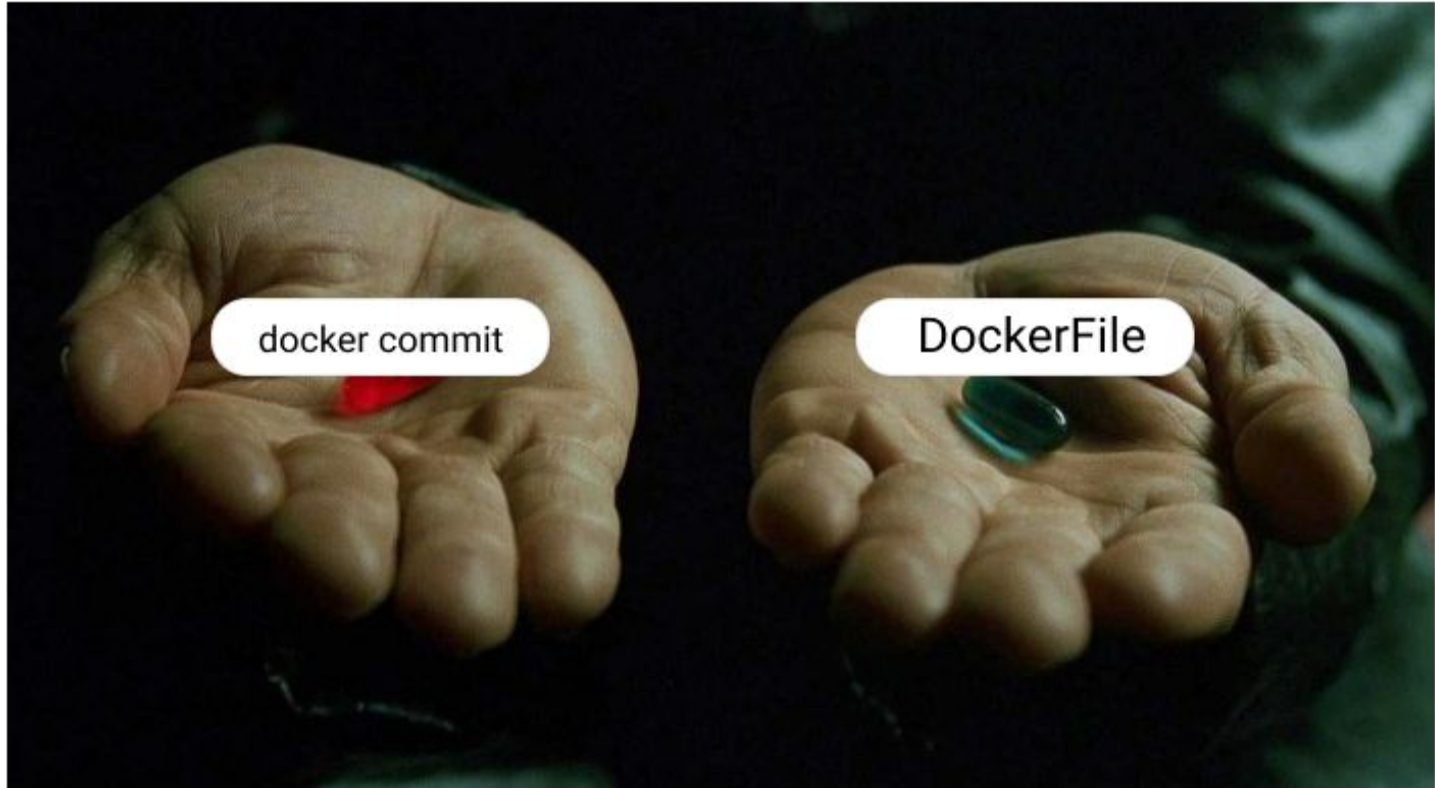
Контейнер != Образ



Контейнер != Образ



DockerFile



DockerFile

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

DockerFile

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

DockerFile

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

```
EXPOSE 80
```

```
EXPOSE 443
```

```
ENV DefaultConnection="<connection_string>"
```

```
WORKDIR /app
```

```
COPY . .
```

```
RUN dotnet publish -c Release -o /out/
```

```
WORKDIR /out
```

```
ENTRYPOINT ["dotnet", App.dll]
```

DockerFile

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

DockerFile

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

DockerFile

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

DockerFile

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


DockerFile

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

DockerFile

```
docker build \
  -t project_name:tag \
  -f path/to/DockerFile \
  ./path/to/context
```

DockerFile

```
docker build \
  -t project_name:tag \
  -f path/to/DockerFile \
  ./path/to/context
```

DockerFile

```
docker build \
  -t project_name:tag \
  -f path/to/DockerFile \
  ./path/to/context
```

DockerFile

```
docker build \
  -t project_name:tag \
  -f path/to/DockerFile \
  ./path/to/context
```

DockerFile

```
docker build \  
    -t project_name:tag \  
    -f path/to/DockerFile \  
    ./path/to/context
```

DockerFile



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

Multi-stage build in production

```
cbf2ff107de7 example:latest ENTRYPOINT ["dotnet" "App.dll"]
```

(1.64 GB)

```
47e49d6d139b ENV ASPNETCORE_URLS=http://*:8082 (1.64 GB)
```

```
339e2781bf64 dotnet publish -c Release -o /app (1.64 GB)
```

```
eea324f93148 dotnet build -c Release -o /app (1.64 GB)
```

```
5efae2b025ae COPY dir:7811293222 in . (1.64 GB)
```

```
a5e29b6256f0 COPY file:2b33bc757 in . (1.62 GB)
```

```
cf4b3ee54626 WORKDIR /src (1.62 GB)
```

```
910d4027c6a3 EXPOSE 443 (1.62 GB)
```

```
ea3eed67c634 EXPOSE 8082 (1.62 GB)
```

```
f13ac9d68148 microsoft/dotnet:2.2-sdk dotnet help (1.62 GB)
```

...

Multi-stage build in production

```
cbf2ff107de7 example:latest ENTRYPOINT ["dotnet" "App.dll"]  
(1.64 GB)  
47e49d6d139b ENV ASPNETCORE_URLS=http://*:8082 (1.64 GB)  
339e2781bf64 dotnet publish -c Release -o /app (1.64 GB)  
eaa324f93148 dotnet build -c Release -o /app (1.64 GB)  
5efae2b025ae COPY dir:7811293222 in . (1.64 GB)  
a5e29b6256f0 COPY file:2b33bc757 in . (1.62 GB)  
cf4b3ee54626 WORKDIR /src (1.62 GB)  
910d4027c6a3 EXPOSE 443 (1.62 GB)  
ea3eed67c634 EXPOSE 8082 (1.62 GB)  
f13ac9d68148 microsoft/dotnet:2.2-sdk dotnet help (1.62 GB)  
...
```

Multi-stage build и production



Multi-stage build и production



Multi-stage build in production

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

```
...
```

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

```
...
```

```
RUN dotnet build -c Release -o /app
```

```
FROM build AS publish
```

```
...
```

```
RUN dotnet publish -c Release -o /app
```

```
FROM base AS final
```

```
...
```

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

```
ENTRYPOINT ["dotnet", "App.dll"]
```

Multi-stage build in production

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

```
...
```

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

```
...
```

```
RUN dotnet build -c Release -o /app
```

```
FROM build AS publish
```

```
...
```

```
RUN dotnet publish -c Release -o /app
```

```
FROM base AS final
```

```
...
```

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

```
ENTRYPOINT ["dotnet", "App.dll"]
```

Multi-stage build in production

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

```
...
```

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

```
...
```

```
RUN dotnet build -c Release -o /app
```

```
FROM build AS publish
```

```
...
```

```
RUN dotnet publish -c Release -o /app
```

```
FROM base AS final
```

```
...
```

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

```
ENTRYPOINT ["dotnet", "App.dll"]
```

Multi-stage build in production

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

```
...
```

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

```
...
```

```
RUN dotnet build -c Release -o /app
```

```
FROM build AS publish
```

```
...
```

```
RUN dotnet publish -c Release -o /app
```

```
FROM base AS final
```

```
...
```

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

```
ENTRYPOINT ["dotnet", "App.dll"]
```

Multi-stage build in production

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

```
...
```

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

```
...
```

```
RUN dotnet build -c Release -o /app
```

```
FROM build AS publish
```

```
...
```

```
RUN dotnet publish -c Release -o /app
```

```
FROM base AS final
```

```
...
```

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

```
ENTRYPOINT ["dotnet", "App.dll"]
```


Multi-stage build in production

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

```
...
```

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

```
...
```

```
RUN dotnet build -c Release -o /app
```

```
FROM build AS publish
```

```
...
```

```
RUN dotnet publish -c Release -o /app
```

```
FROM base AS final
```

```
...
```

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

```
ENTRYPOINT ["dotnet", "App.dll"]
```

Multi-stage build in production

```
2e43a04952a6 events.app ENTRYPOINT ["dotnet" "App.dll"]  
(252.22 MB)  
1bc5c75bfbcd COPY dir:df84299421 in . (252.22 MB)  
80cc7f8f1463 ENV ASPNETCORE_URLS=http://*:8082 (248.46 MB)  
d2e0add57b17 WORKDIR /app (248.46 MB)  
c925e067cfda EXPOSE 443 (248.46 MB)  
4fb77cd315bf EXPOSE 8082 (248.46 MB)  
3c39d1766866 WORKDIR /app (248.46 MB)  
45c933779d5c microsoft/dotnet:2.2-aspnetcore-runtime curl  
-SL --output aspnetcore.tar.gz
```

Multi-stage build in production

```
2e43a04952a6 events.app ENTRYPOINT ["dotnet" "App.dll"]
```

(252.22 MB)

```
1bc5c75bfbcd COPY dir:df84299421 in . (252.22 MB)
```

```
80cc7f8f1463 ENV ASPNETCORE_URLS=http://*:8082 (248.46 MB)
```

```
d2e0add57b17 WORKDIR /app (248.46 MB)
```

```
c925e067cfda EXPOSE 443 (248.46 MB)
```

```
4fb77cd315bf EXPOSE 8082 (248.46 MB)
```

```
3c39d1766866 WORKDIR /app (248.46 MB)
```

```
45c933779d5c microsoft/dotnet:2.2-aspnetcore-runtime curl
```

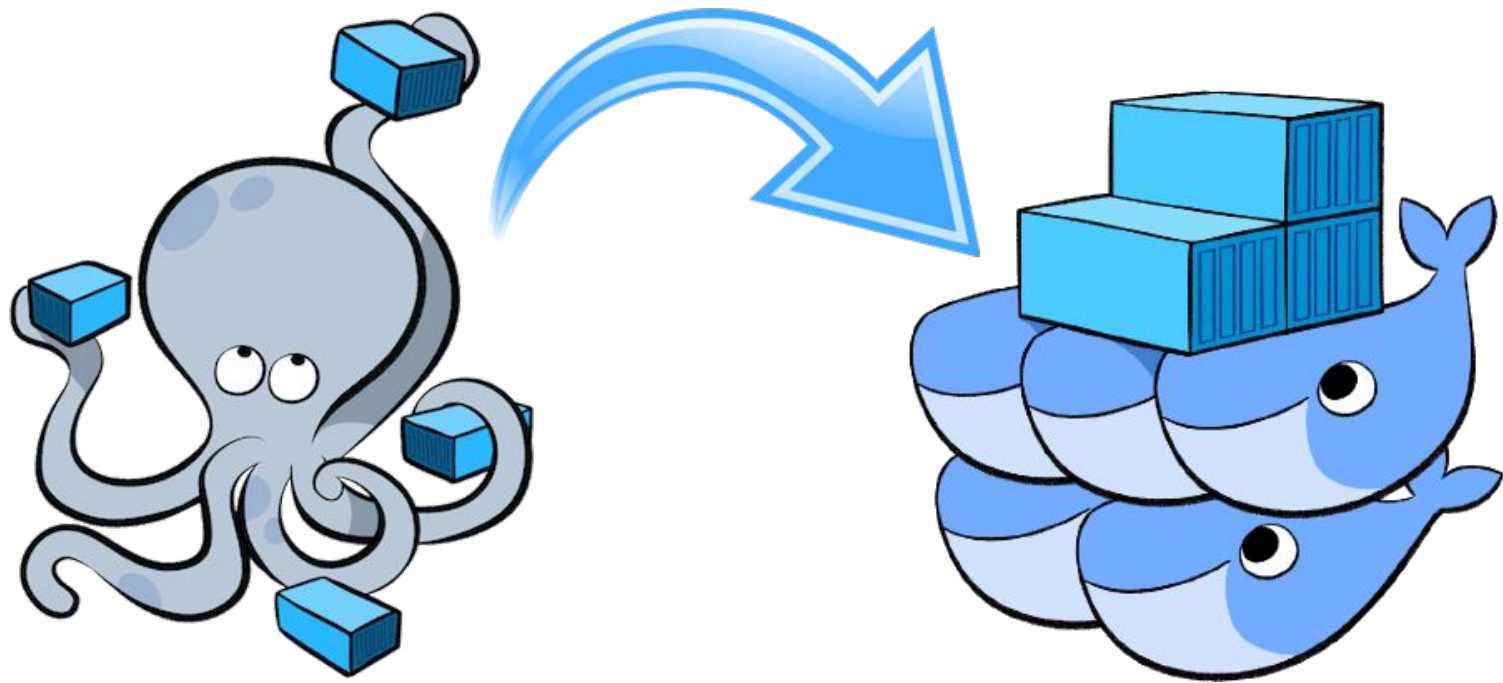
```
-SL --output aspnetcore.tar.gz (248.46 MB)
```

Multi-stage build в production



<https://docs.docker.com/develop/develop-images/multistage-build/>

Docker-Compose



Docker-Compose

- Запуск группы контейнеров

Docker-Compose

- Запуск группы контейнеров
- Настройка взаимодействия между контейнерами

Docker-Compose

- Запуск группы контейнеров
- Настройка взаимодействия между контейнерами
- Перезапуск контейнера в случае нештатных ситуаций

Docker-Compose

- Запуск группы контейнеров
- Настройка взаимодействия между контейнерами
- Перезапуск контейнера в случае нештатных ситуаций
- Масштабирование

Docker-Compose

```
version: '3'

services:
  proxy:
    image: <image>
    links:
      - backend
      - frontend
    ports:
      - "80:80"
```

Docker-Compose

```
version: '3'
```

```
services:
```

```
  proxy:
```

```
    image: <image>
```

```
    links:
```

```
      - backend
```

```
    ports:
```

```
      - "80:80"
```

```
    ...
```

Docker-Compose

```
version: '3'
```

```
services:
```

```
proxy:
```

```
  image: <image>
```

```
  links:
```

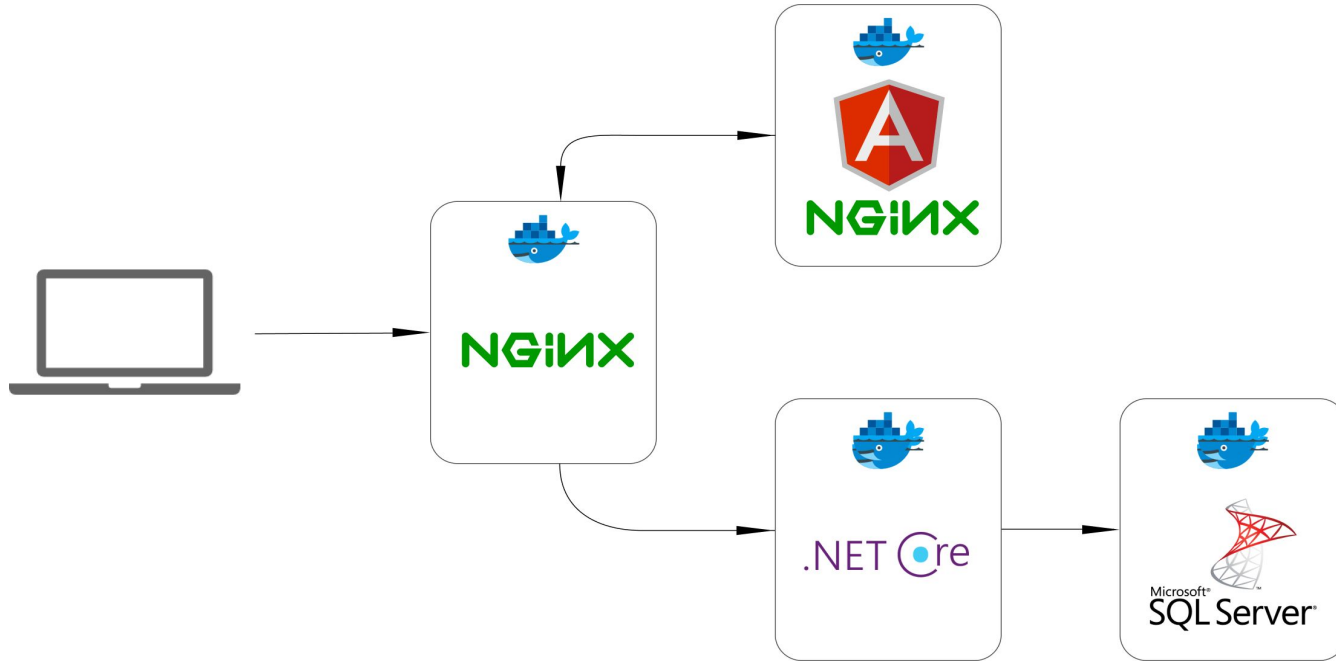
```
    - backend
```

```
  ports:
```

```
    - "80:80"
```

```
  ...
```

Docker-Compose



Docker-Compose



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

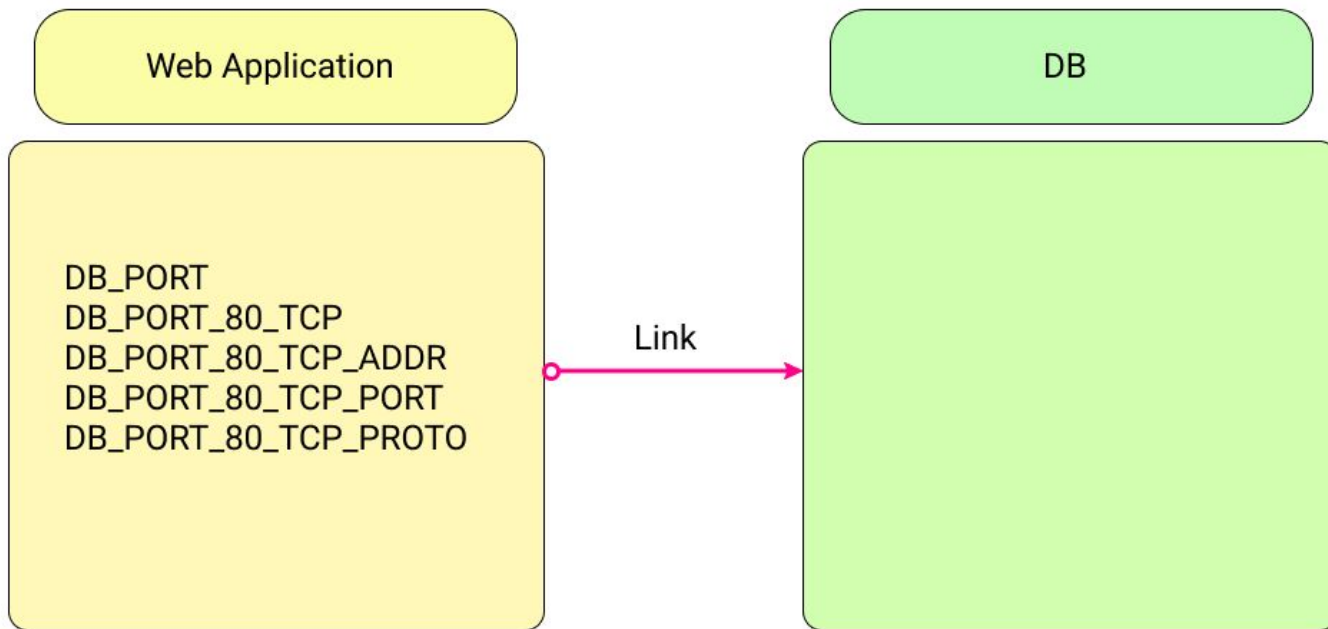
Docker-Compose



<https://docs.docker.com/compose/compose-file/>

Взаимодействие контейнеров

Links

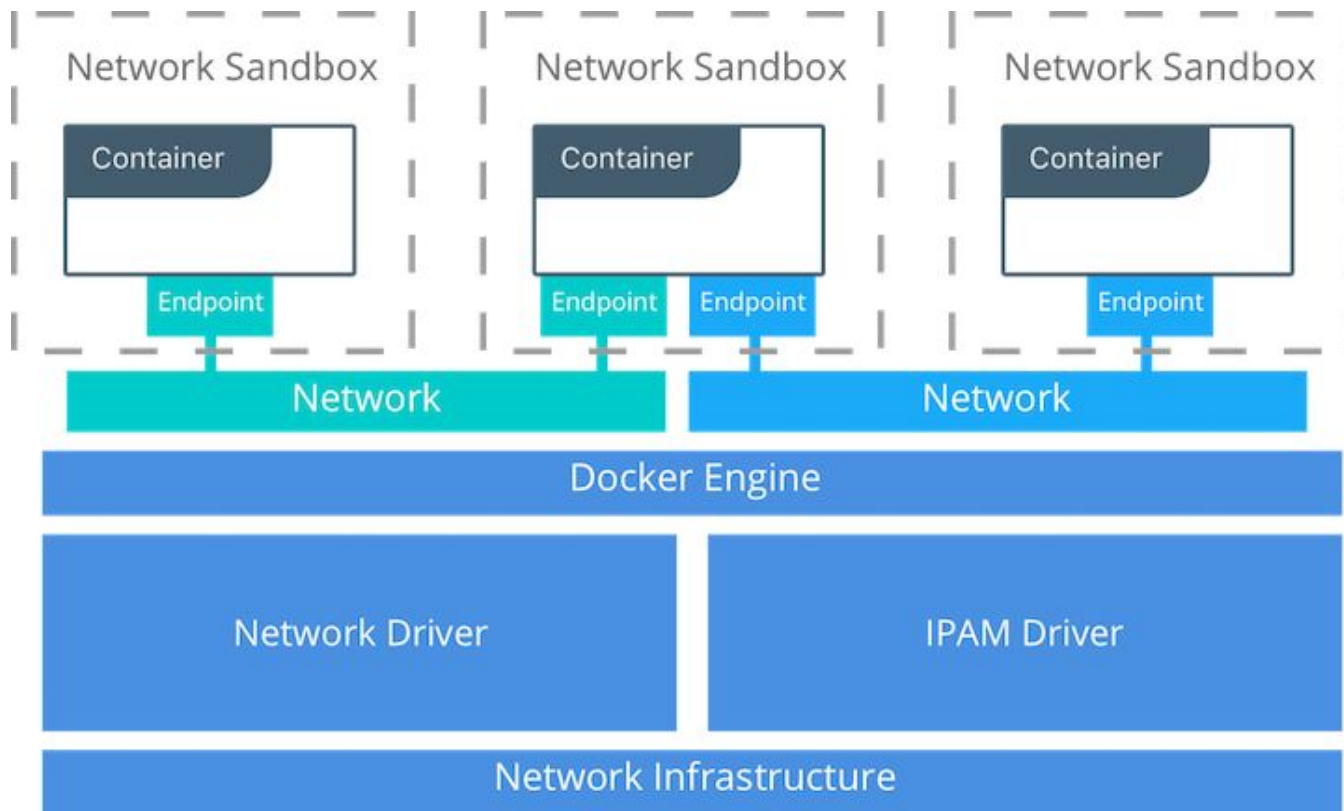


Взаимодействие контейнеров

Links

- Позволяет контейнерам взаимодействовать
- Хранит сетевую конфигурацию в переменных окружения
- Плохо масштабируется, или не масштабируется вовсе
- Deprecated

Взаимодействие контейнеров



Взаимодействие контейнеров

Networks

- Позволяет контейнерам взаимодействовать
- Имеет разные драйвера (bridge, overlay, host, macvlan, custom)
- Хорошо масштабируются
- Позволяет использовать имя контейнера в качестве host name

Взаимодействие контейнеров

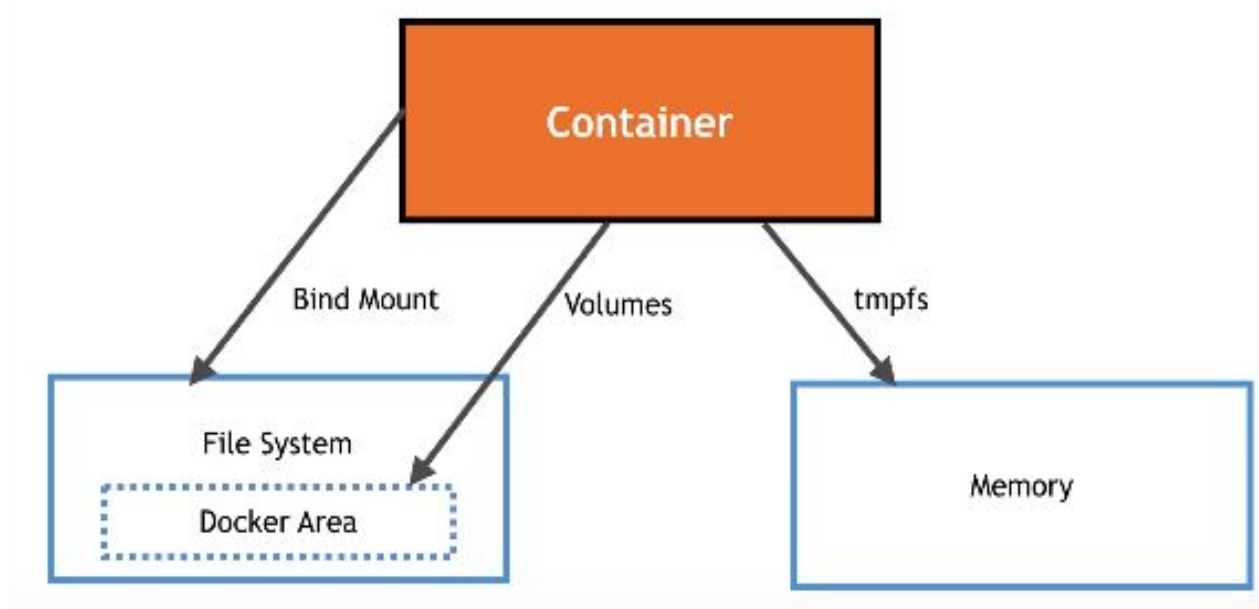


<https://docs.docker.com/network/>

Вынесение данных из контейнера

- Конфигурация приложений (`nginx.conf`)
- Данные приложения (базы данных)

Вынесение данных из контейнера



Вынесение данных из контейнера

Volumes

- Не зависят от хоста
- Могут использоваться несколькими контейнерами
- Управляются с помощью Docker CLI
- Могут быть вынесены в облако или на удаленный сервер

Вынесение данных из контейнера

Bind mount

- Директория хоста монтируется внутри контейнера
- Могут использоваться несколькими контейнерами
- Не управляются с помощью Docker

Вынесение данных из контейнера

Tmpfs

- Область памяти за пределами контейнера
- Не может использоваться несколькими контейнерами
- Доступно только в linux
- Существует, пока запущен контейнер

Вынесение данных из контейнера



<https://docs.docker.com/storage>

Tips&Tricks

- Храните данные за пределами контейнера
- Используйте multi-stage build
- Вы можете использовать .env файлы для задания множества переменных окружения внутри контейнера
- Если вы не хотите, чтобы папка или файл попали в образ, используйте файл **.dockerignore**

Tips&Tricks

- Создавайте образы для сборки ваших проектов
Данный шаг сократит время на развертывание проекта

```
FROM microsoft/dotnet
```

```
ENV PATH="${PATH}:/root/.dotnet/tools"
```

```
RUN dotnet tool install --global coverlet.console \  
    & dotnet tool install --global dotnet-sonarscanner
```

Tips&Tricks

FROM php:7.2-fpm

WORKDIR /var/www

RUN apt-get update

RUN apt-get update && apt-get install -y \
 <composer-deps> \

RUN apt-get clean && rm -rf /var/lib/apt/lists/*

RUN docker-php-ext-install pdo_mysql mbstring zip exif pcntl

RUN docker-php-ext-configure gd --with-gd
--with-freetype-dir=/usr/include/ --with-jpeg-dir=/usr/include/
--with-png-dir=/usr/include/

RUN docker-php-ext-install gd

RUN curl -sS https://getcomposer.org/installer | php --
--install-dir=/usr/local/bin --filename=composer

Tips&Tricks

- Вы можете использовать ранее созданную сеть в docker-compose.yaml:

```
networks:  
  existing_network:  
    external: true
```

Tips&Tricks

- Используйте префикс **x-** и **якоря** в уaml, чтобы не дублировать конфигурацию.
Соблюдайте DRY.

```
version: '3.5'
```

```
x-base-webapi: &base-webapi
```

```
  restart: always
```

```
services:
```

```
  development-webapi:
```

```
    <<: *base-webapi
```

```
  production-webapi:
```

```
    <<: *base-webapi
```

Tips&Tricks

- `# docker run --rm <image>` - удалит контейнер после завершения работы
- `# docker system prune` - удалит остановленные контейнеры и неиспользуемые сети.
- `# docker ps -a` - отобразит список всех контейнеров
- `# docker exec -it <container_id> bin/bash` - позволяет подключиться к контейнеру

Tips&Tricks

И самый главный лайфхак...

Tips&Tricks

Поможет выйти оттуда, куда вы вошли ...

Tips&Tricks

Он решит многие проблемы, которые у вас будут ...

Tips&Tricks

[illegible]

Tips&Tricks



<https://docs.docker.com>

Где применить?

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

Легких контейнеров и быстрых SSD Вам!
Спасибо за внимание!

Ресурсы

<https://docs.docker.com>

<https://github.com/veggie-monk/awesome-docker>

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

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

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