Упаковка приложения в контейнер

Вадим Калашников
Lead Software Engineer
в компании Wildberries

Проверка связи

Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включён звук
- обновите страницу вебинара (или закройте страницу и заново присоединитесь к вебинару)
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти

Поставьте в чат:

\+ если меня видно и слышно

— если нет

Вадим Калашников

О спикере:

- Разработчик на C++ более 15 лет
- Опыт разработки в областях: backend, embedded, kernel development, системное программирование, сети.
- С 2023 года Lead Software Engineer в компании Wildberries

Вспоминаем прошлое занятие

Вопрос: что означает «запах» «Стрельба дробью»?

Вспоминаем прошлое занятие

Вопрос: что означает «запах» «Стрельба дробью»?

Ответ: при выполнении любых модификаций приходится вносить множество мелких изменений в большое число классов

Вспоминаем прошлое занятие

Вопрос: как избавиться от одержимости элементарными типами?

Вспоминаем прошлое занятие

Вопрос: как избавиться от одержимости элементарными типами?

Ответ: по возможности укладывать небольшие задачи в классы

Вспоминаем прошлое занятие

Вопрос: что представляют собой цепочки сообщений?

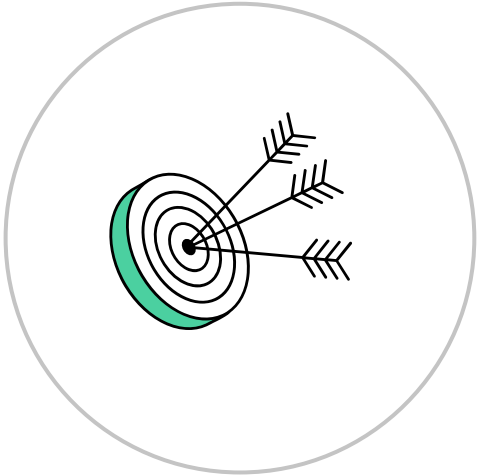
Вспоминаем прошлое занятие

Вопрос: что представляют собой цепочки сообщений?

Ответ: объект запрашивает данные у объекта, которых у него нет, и тот вынужден запрашивать их у другого объекта и т. д. вниз по иерархии

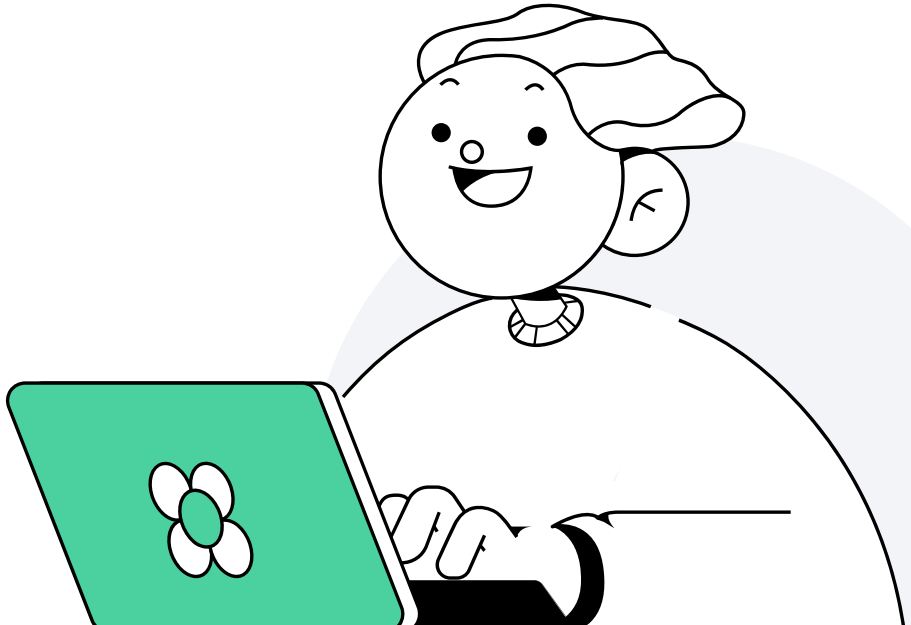
Цели занятия

- Узнаем, что такое Docker
- Сделаем свой образ и запустим его в качестве контейнера
- Разберём способы создания приложений Docker



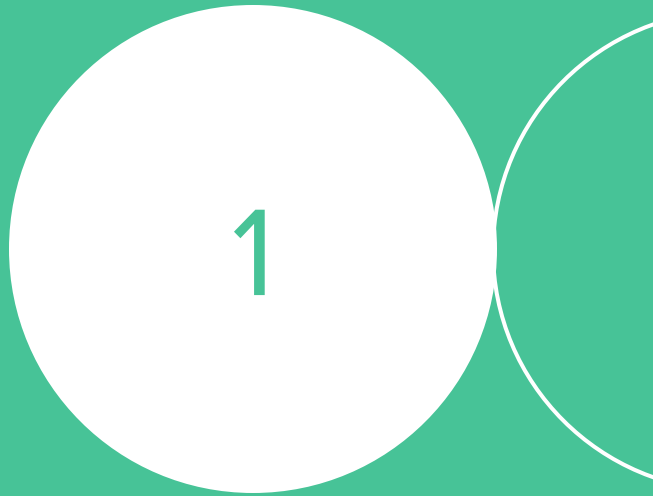
План занятия

1. Основы Docker
2. Создание приложений Docker
3. Итоги
4. Домашнее задание



*Нажми на нужный раздел для перехода

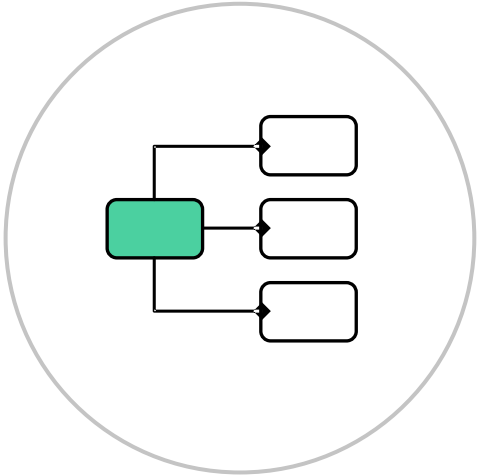
Основы Docker



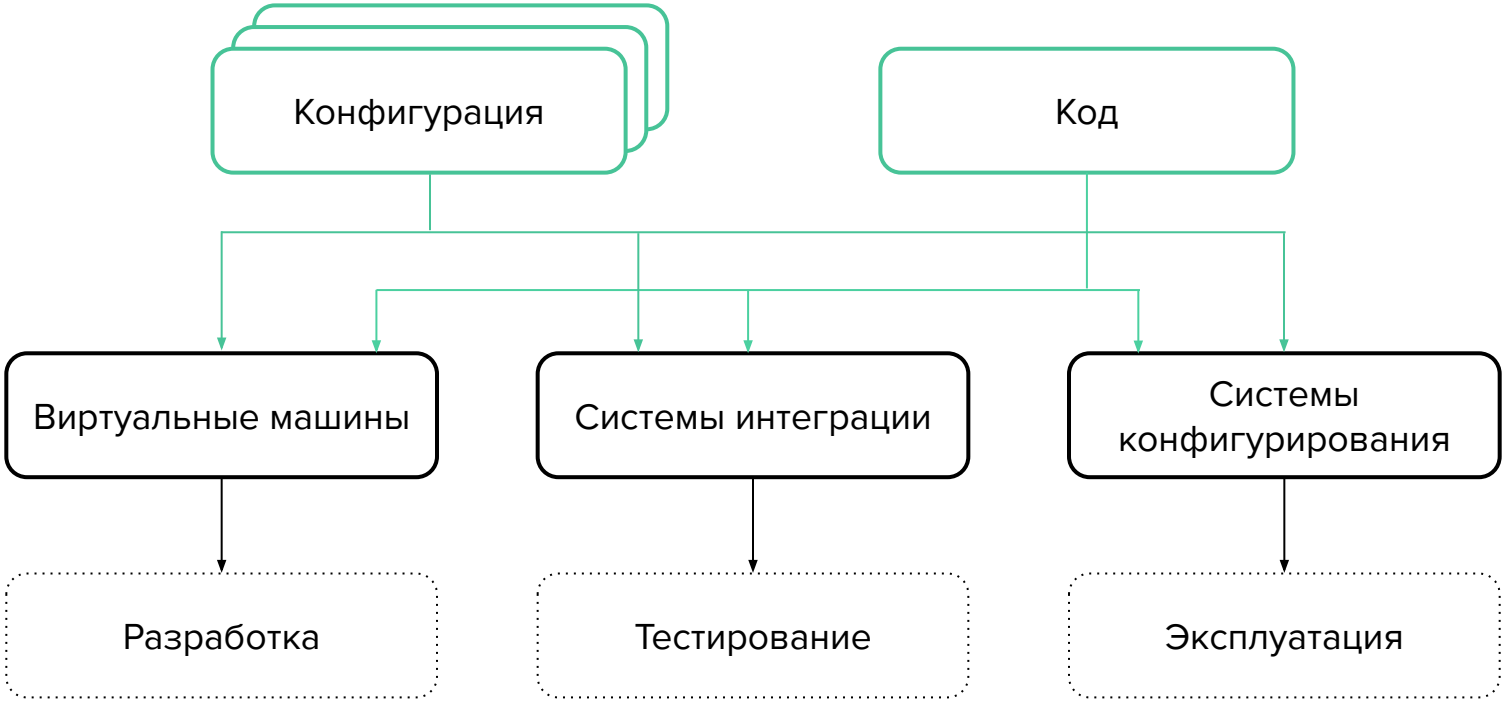
Docker — это платформа, которая позволяет создавать, поставлять и запускать любое приложение повсюду

Жизнь до Docker

До появления Docker в конвейере разработки обычно использовались комбинации различных технологий для управления движением программного обеспечения, такие как виртуальные машины, инструменты управления конфигурацией, системы управления пакетами и комплексные сети библиотечных зависимостей

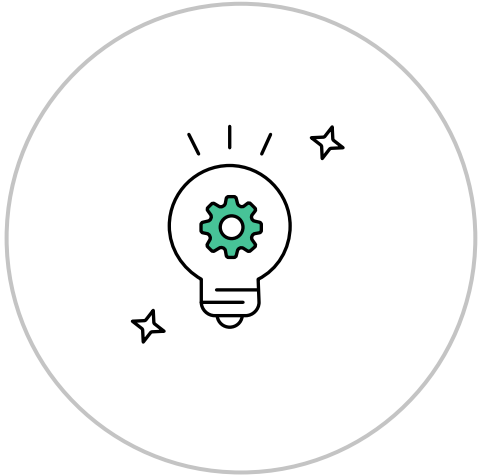


Жизнь до Docker

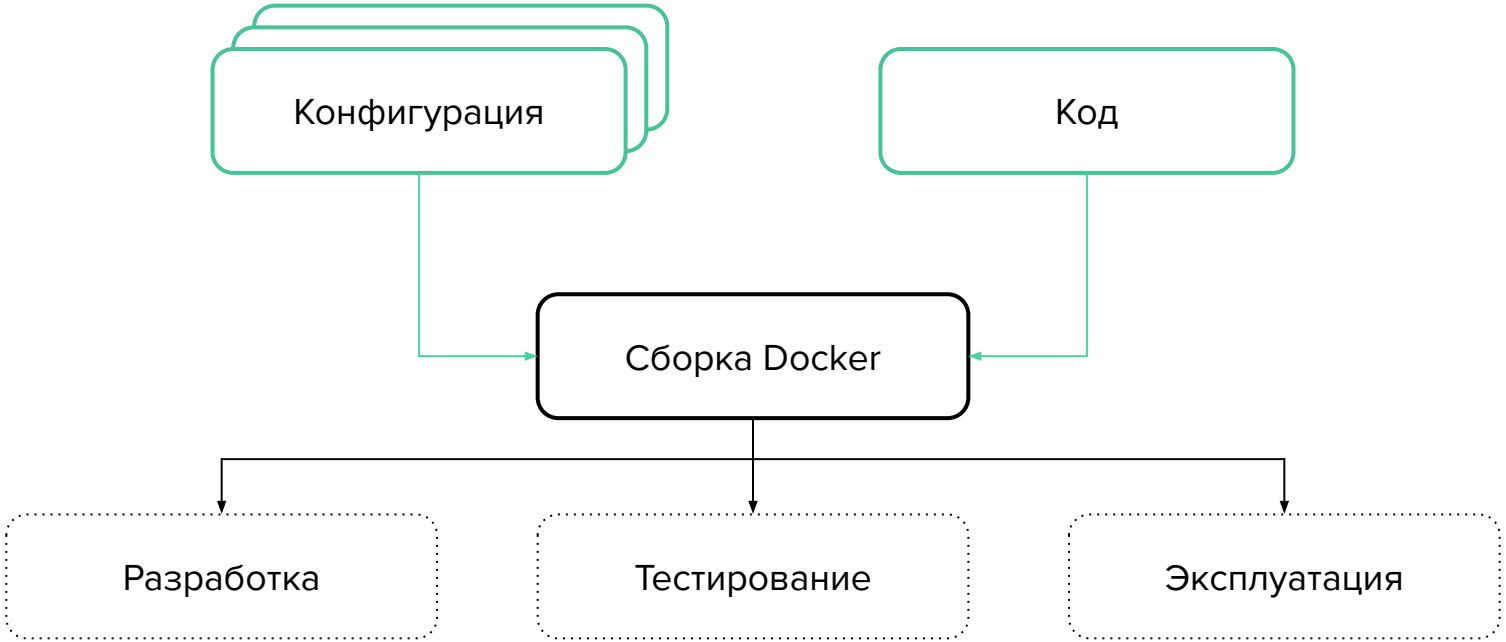


Жизнь с Docker

Docker позволил инженерам, вовлечённым в процесс разработки, эффективно говорить на одном языке, облегчая совместную работу. Всё проходит через общий конвейер к одному выходу, который можно использовать для любой цели — нет необходимости продолжать поддерживать запутанный массив конфигураций инструментов



Жизнь с Docker



Преимущества Docker

1. Замена виртуальных машин

Преимущества Docker

1. Замена виртуальных машин
2. Прототипирование программного обеспечения

Преимущества Docker

1. Замена виртуальных машин
2. Прототипирование программного обеспечения
3. Упаковка программного обеспечения

Преимущества Docker

1. Замена виртуальных машин
2. Прототипирование программного обеспечения
3. Упаковка программного обеспечения
4. Возможность для архитектуры микросервисов

Преимущества Docker

1. Замена виртуальных машин
2. Прототипирование программного обеспечения
3. Упаковка программного обеспечения
4. Возможность для архитектуры микросервисов
5. Моделирование сетей

Преимущества Docker

1. Замена виртуальных машин
2. Прототипирование программного обеспечения
3. Упаковка программного обеспечения
4. Возможность для архитектуры микросервисов
5. Моделирование сетей
6. Возможность производительности полного стека в автономном режиме

Преимущества Docker

1. Замена виртуальных машин
2. Прототипирование программного обеспечения
3. Упаковка программного обеспечения
4. Возможность для архитектуры микросервисов
5. Моделирование сетей
6. Возможность производительности полного стека в автономном режиме
7. Сокращение неизбежных расходов на отладку

Образ — это набор слоёв файловой системы и некоторые метаданные

Взятые вместе, они могут быть запущены как контейнеры Docker

Образ

Файлы образов занимают большую часть пространства.

Из-за изоляции, которую обеспечивает каждый контейнер, они должны иметь собственную копию любых необходимых инструментов, включая языковые среды или библиотеки

Метаданные содержат информацию о переменных среды, пробросе портов, томах и других деталях

Слой — это набор изменений в файлах

Различия между первой и второй версиями MyApplication хранятся в этом слое

Контейнер — это запущенный экземпляр образа.

У вас может быть несколько контейнеров, запущенных с одного образа

Ключевые команды Docker

Команда	Назначение
docker build	Собрать образ Docker
docker run	Запустить образ Docker в качестве контейнера
docker commit	Сохранить контейнер Docker в качестве образа
docker tag	Присвоить тег образу Docker

Создание приложений Docker

2

Способы создания нового образа Docker

Команда	Назначение
Команды Docker	Запуск контейнера с помощью docker run. Создание нового образа с помощью docker commit
Dockerfile	Сборка из известного базового образа и с помощью ограниченного набора простых команд, записанных внутри одного файла
Инструмент управления конфигурацией	То же самое, что и Dockerfile, но контроль над сборкой передаётся более сложному инструменту управления конфигурацией
Импорт набора файлов	Из пустого образа импортируется файл TAR с необходимыми файлами

Реализация Dockerfile

Dockerfile — это текстовый файл, содержащий серию команд

```
1  FROM node
2      LABEL maintainer ian.miell@gmail.com
3      RUN git clone -q https://github.com/docker-in-practice/todo.git
4      WORKDIR todo
5      RUN npm install > /dev/null
6      EXPOSE 8000
7      CMD ["npm","start"]
```

Запуск Dockerfile

`docker build [path to Dockerfile]`

- Команда приводит к созданию нового образа с выводом его идентификатора
- Docker загружает файлы и каталоги по пути, предоставленному команде docker build
- Каждый шаг сборки последовательно нумеруется, начиная с 1, и выводится командой
- Для экономии места каждый промежуточный контейнер удаляется, перед тем как продолжить
- Формируется окончательный идентификатор образа для этой сборки, готовый к присвоению тега

Присвоение тега

`docker tag [id] [tag]`

Присвоение тега может осуществляться для того, чтобы было удобно обращаться к данному образу

Запуск контейнера

`docker run [tag]`

- Подкоманда docker run запускает контейнер
- Подкоманда docker diff показывает, какие файлы были затронуты с момента создания экземпляра образа как контейнера

Итоги

3

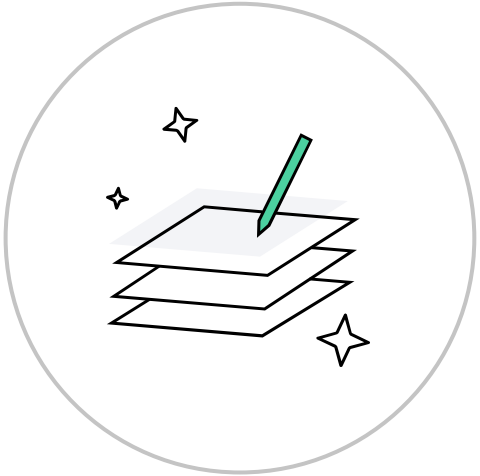
Итоги занятия

1. Узнали, что такое Docker
2. Сделали свой образ и запустили его в качестве контейнера
3. Разобрали способы созданий приложений Docker

Домашнее задание

Давайте посмотрим ваше **домашнее задание**.

1. Вопросы по домашней работе задавайте в чате группы
2. Задачи можно сдавать по частям
3. Зачёт по домашней работе ставят после того, как приняты все задачи



Дополнительные материалы

- Установка Docker Toolbox
- Установка Docker Desktop
- Работа с Docker через Visual Studio Code

Задавайте вопросы и пишите отзыв о лекции