



Лекция 12

Deploy, безопасность

Дмитрий Зайцев
Мартин Комитски



План на сегодня

- Автоматизация
 - Линтеры
 - Хуки
 - Контейнеры
- Безопасность
 - Глоссарий
 - Куки

Минутка бюрократии

- Внимание
- Отметки о посещении занятий
- Обратная связь о лекциях



Deploy,
безопасность

Вопросы с собеседа

- Как попасть в ваше приложение по сети
- Как код попадает в продакшн
- Что такое автоматизация
- Что происходит, после того как вы запустили код в мастер
- Кто должен отвечать за деплой

Вопросы с собеседа (специализированные)

- Сколько веток нужно для разработки
- Зачем разделять дев и прод окружения
- Что такое веб сервер
- Что такое DNS
- Что такое "сборка в облаке"
- Что такое ci/cd
- Что такое докер
- Что такое линтер

Автоматизация



Линтеры

Линтер – анализатор кода. Проверяет код на стилистические, синтаксические и специфичные для языка ошибки.

Зачем использовать?

- Повышение качества ПО
- Улучшение читаемости кода
- Сокращение времени на ревью

Когда использовать?

- Во время написания кода (IDE)
- Перед коммитом (гит хуки)
- При сборке приложения (после пуша)

GIT HOOKS

Хуки - команды/скрипты, которые будут выполнены до или после git команды (commit, push, pull, etc)

```
mv .git/hooks/pre-commit.sample .git/hooks/pre-commit
$ cat > .git/hooks/pre-commit << EOF
> echo 'OMG COOMIT IS DONE!'
> EOF
```

Дополнительно:

<https://githooks.com>

HUSKY

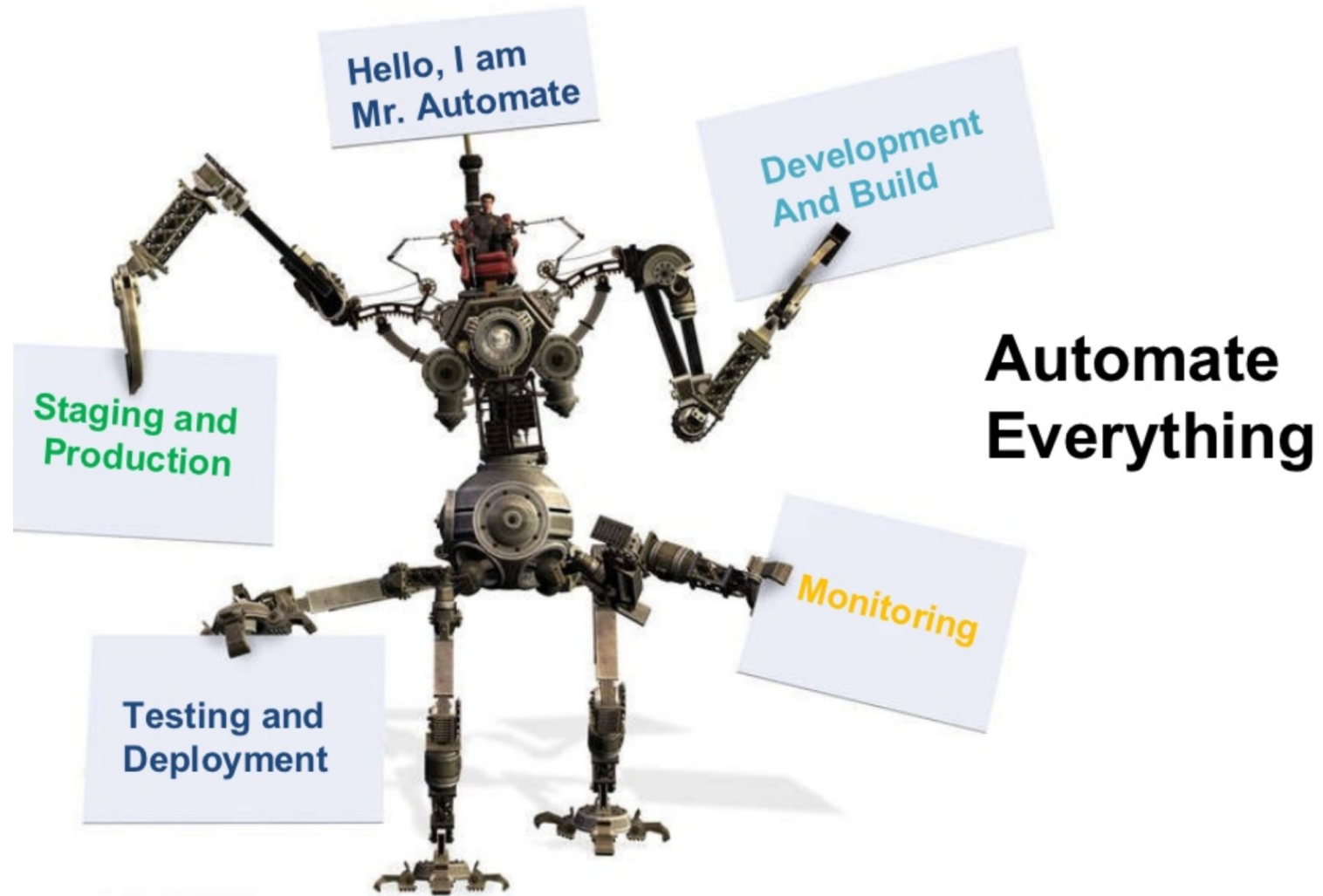
Позволяет описывать git хуки из package.json

```
$ npm i --saveDev husky lint-staged prettier
```

HUSKY

```
"husky": {  
  "hooks": {  
    "pre-commit": "lint-staged"  
  }  
},  
"lint-staged": {  
  "src/**/*..{js,jsx,ts,tsx,json}": [  
    "prettier --write",  
    "git add"  
  ]  
},
```

CI/CD



Непрерывная интеграция

CI (Continuous Integration) - практика слияния выполненных разработчиками работ в основное хранилище/репозиторий (github/gitlab/bitbucket) – trunk/mainline. Непрерывно.

Непрерывная доставка

CD[E] (Continuous Delivery) - практика автоматизации всего процесса релиза ПО. Выполняется CI + подготовка приложения к выпуску на боевые сервера. Гарантируется высокое качество поставляемого ПО для возможности совершить релиз любое время.

Непрерывное развертывание

CD (Continuous Deployment) - выполняется CDE + автоматический деплой в продакшн с перезапуском сервером приложения при необходимости.

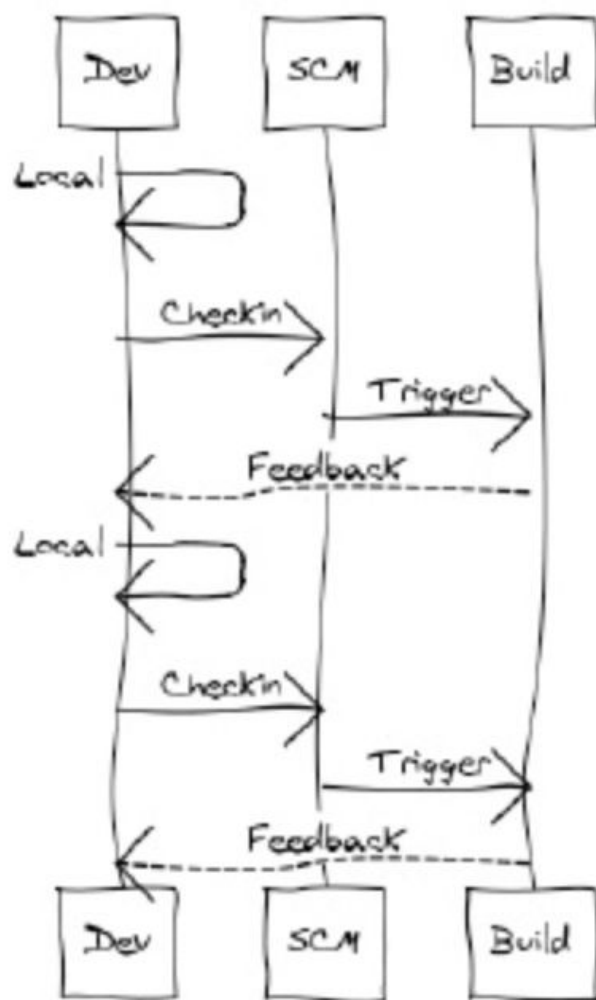
CDE & CD

CONTINUOUS DELIVERY



CONTINUOUS DEPLOYMENT

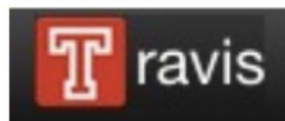




Сервисы



Jenkins



Docker

Docker – проект с открытым исходным кодом для автоматизации развертывания приложений в виде переносимых автономных контейнеров, способных выполняться в любой среде.

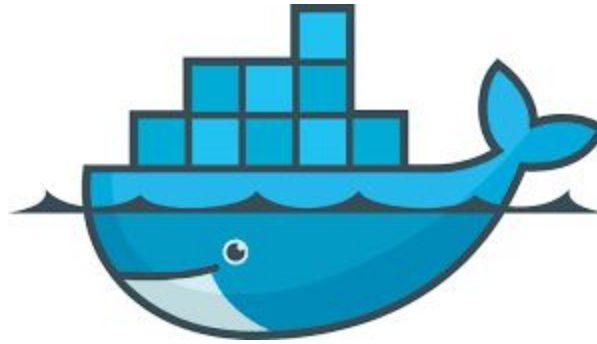
Цель





- Изоляция окружения
- Ограничение ресурсов
- Упрощение дистрибьюции

Дополнительно:

<https://docs.docker.com/get-started/>

Docker



-  Каждый компонент системы в отдельном контейнере
-  Контейнеры содержат в себе всю конфигурацию
-  Образы хранятся в registry
-  Образы версионятся

NGINX

NGINX – один из самых известных веб серверов. Способен выдерживать высокие нагрузки и реализовать архитектуру любой сложности. Огромный набор настроек – это плюс и минус nginx.

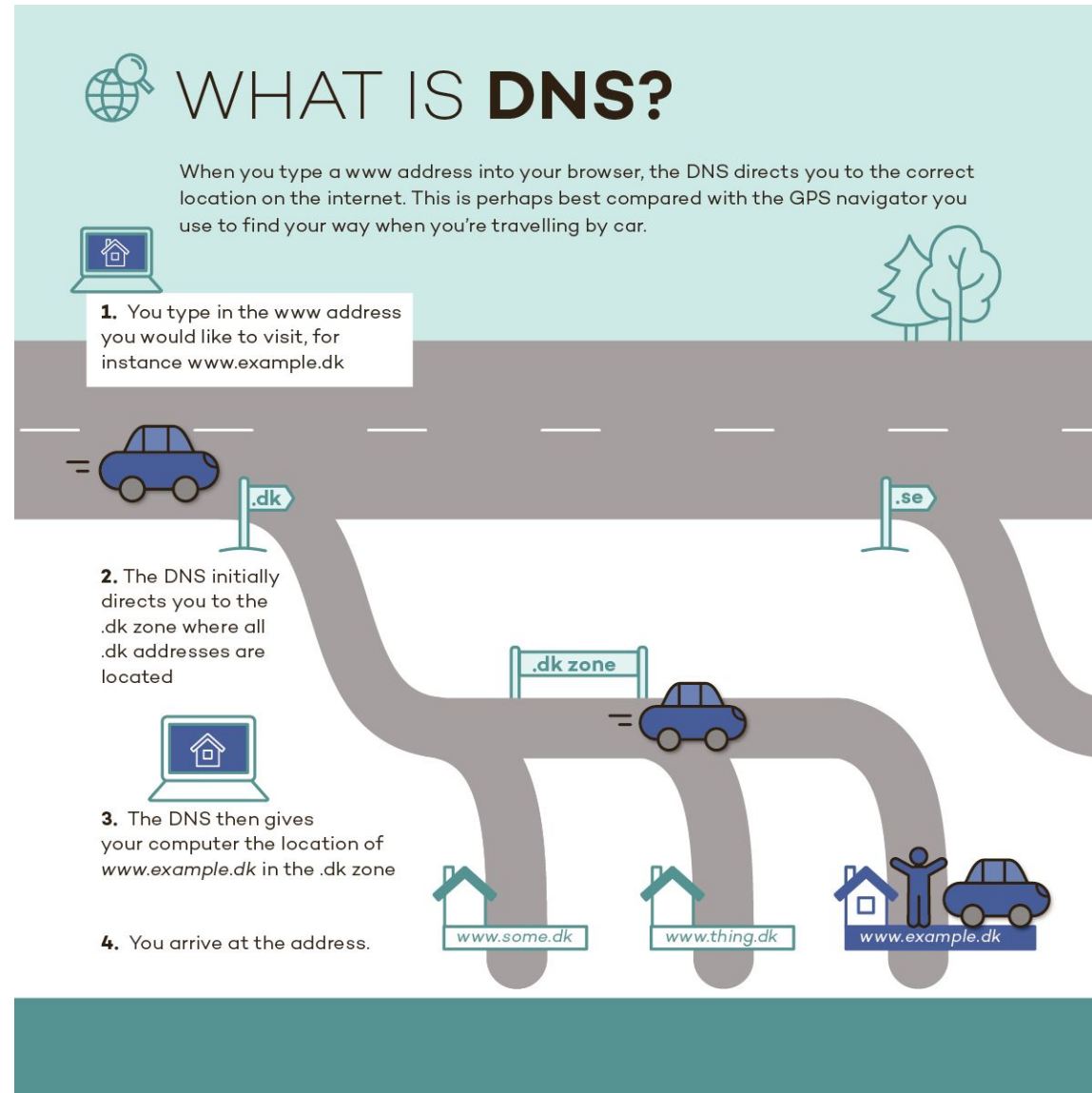
Основные кейсы для использования

- proxy
- reverse-proxy

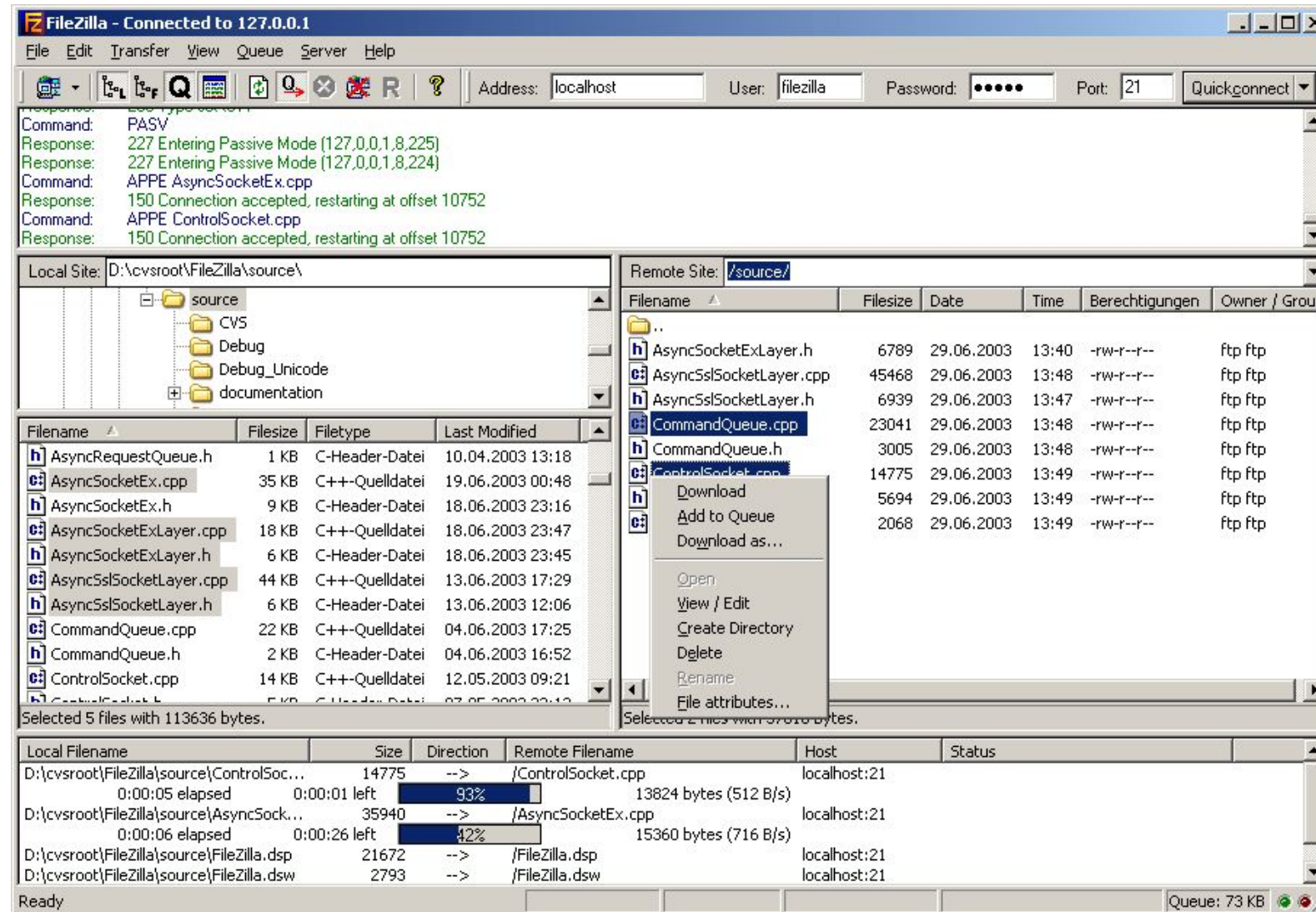
Caddy

Caddy – популярный веб сервер с поддержкой https и http2 из коробки.
Очень простой в настройке.

DNS



Long time ago...



NGINX+DOCKER+CRA

```
$ docker-compose build frontend
```

```
$ docker-compose up -d frontend
```

Dockerfile

```
# Stage 0, "build-stage" to build and compile the frontend
from node:11-alpine as build-stage
WORKDIR /app
COPY . ./
RUN npm i
RUN yarn install
RUN npm run build

# Stage 1, to have only the compiled app, ready for production
with Nginx
from nginx:1.15-alpine
COPY --from=build-stage /app/build /usr/share/nginx/html
COPY ./nginx.conf /etc/nginx/conf.d/default.conf
```

Nginx

```
server {  
    listen 80;  
    location / {  
        root /usr/share/nginx/html;  
        index index.html index.htm;  
        try_files $uri $uri/ /index.html =404;  
    }  
    include /etc/nginx/extra-conf.d/*.conf;  
}
```

docker-compose.yml

```
version: '3.6'
services:
  frontend:
    build:
      context: .
    volumes:
      - ./nginx.conf:/etc/nginx/conf.d/default.conf# to mount
        custom nginx.conf
    ports:
      - "80:80"
```

Дополнительно:

<https://docs.docker.com/compose/gettingstarted/>

Deploy?

Вопросы?





Перерыв! (10 минут)

Препо (с)

Безопасность

Вопросы с собеседа

- Что такое компьютерная безопасность?
- Какие последствия могут быть, если в приложении есть риск атак? Как снизить риск атак?

Вопросы с собеса (специализированные)

- Что такое cookie
- Что такое https
- Что такое CSRF
- Что такое XSS
- Что такое CSP
- Чем авторизация отличается от аутентификации Какие знаете способы аутентификации пользователя

Компьютерная безопасность

Процесс обеспечения:

- конфиденциальности данных
- целостности данных
- доступности данных

для пользователей или клиентов информационных систем

Конфиденциальность (confidentiality)

Система обеспечивает приватное хранение личных данных пользователя.

Атаки: раскрытие информации против воли пользователя. (Disclosure attacks)

Целостность (integrity)

Система обеспечивает надежное хранение личных данных.

Атаки: изменение или уничтожение данных. (Alteration attacks)

Доступность (availability)

Система обеспечивает доступ пользователя к данным.

Атаки: отказ от обслуживания. (Denial attacks)

Терминология

Ассет/актив/ценность (Asset) - объект, представляющий интерес для злоумышленника (личные данные пользователей, вычислительные ресурсы, репутация пользователя)

Терминология

Угроза (Threat) - действие, которое ведет к потере ценности актива (изменение прав собственности, уничтожение, повреждение или раскрытие актива)

Терминология

Уязвимость (Vulnerability) - слабое место в системе (передача пользовательских данных в GET параметре)

Терминология

Риск (Risk) - наличие в системе уязвимости, и угрозы. Возможность совершения атаки.

Терминология

Атака (Attack) - реализованный риск.

Не все риски ведут к атакам, но все атаки – результат реализации риска системы.

Терминология

Ослабление угроз (Mitigation) - процесс снижения рисков в системе за счет снижения количества уязвимостей или за счет обесценивания активов.

Атаки не могут быть ослаблены. Снижать можно только риски

Cookies

Cookies – механизм хранения информации на клиенте, обеспечивая таким образом возможность идентификацию пользователя и/или его действий. Куки являются заголовком HTTP протокола.

Назначение

- Управление сеансом (логин, просмотренная лента, корзина)
- Персонализация
- Мониторинг

Дополнительно:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies> - читать обязательно

Cookies

Нужные атрибуты:

- Domain
- Expires
- Max-Age
- Path

Нужные флаги:

- Secure
- HttpOnly
- SameSite

```
Set-Cookie: id=longid; Domain=otvet.mail.ru; Path=/rubrics; Expires=Wed,  
10 Dec 2024 07:28:00 GMT; Secure; HttpOnly; SameSite;
```

Cookies

<https://www.freecodecamp.org/news/web-security-hardening-http-cookies-be8d8d8016e1>

Аутентификация & авторизация

Аутентификация (Authentication) = логин + пароль (Кто ты?)

Процесс удостоверения, что некто действительно тот, за кого себя выдает.

Аутентификация & авторизация

Авторизация (Authorization) = доступы (permissions) (Что ты можешь?)

Набор правил, определяющих, кто какие имеет возможности

Дополнительно:

<https://stackoverflow.com/questions/6556522/authentication-versus-authorization>

HTTPS

http over tls

Дополнительно:

<https://hpbn.co/transport-layer-security-tls/>

Что такое?

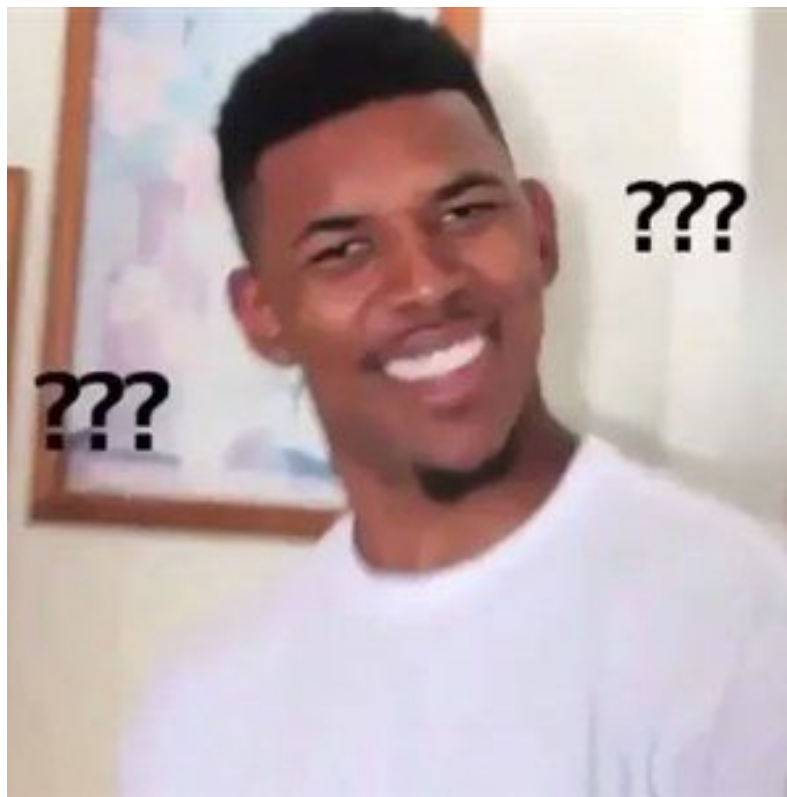
- XSS
- SQL Injection
- csrf
- ssrf

Что такое?

- same origin policy
- content security policy
- hsts HTTP Strict-Transport-Security

Безопасность?

Вопросы?



Домашнее задание №12

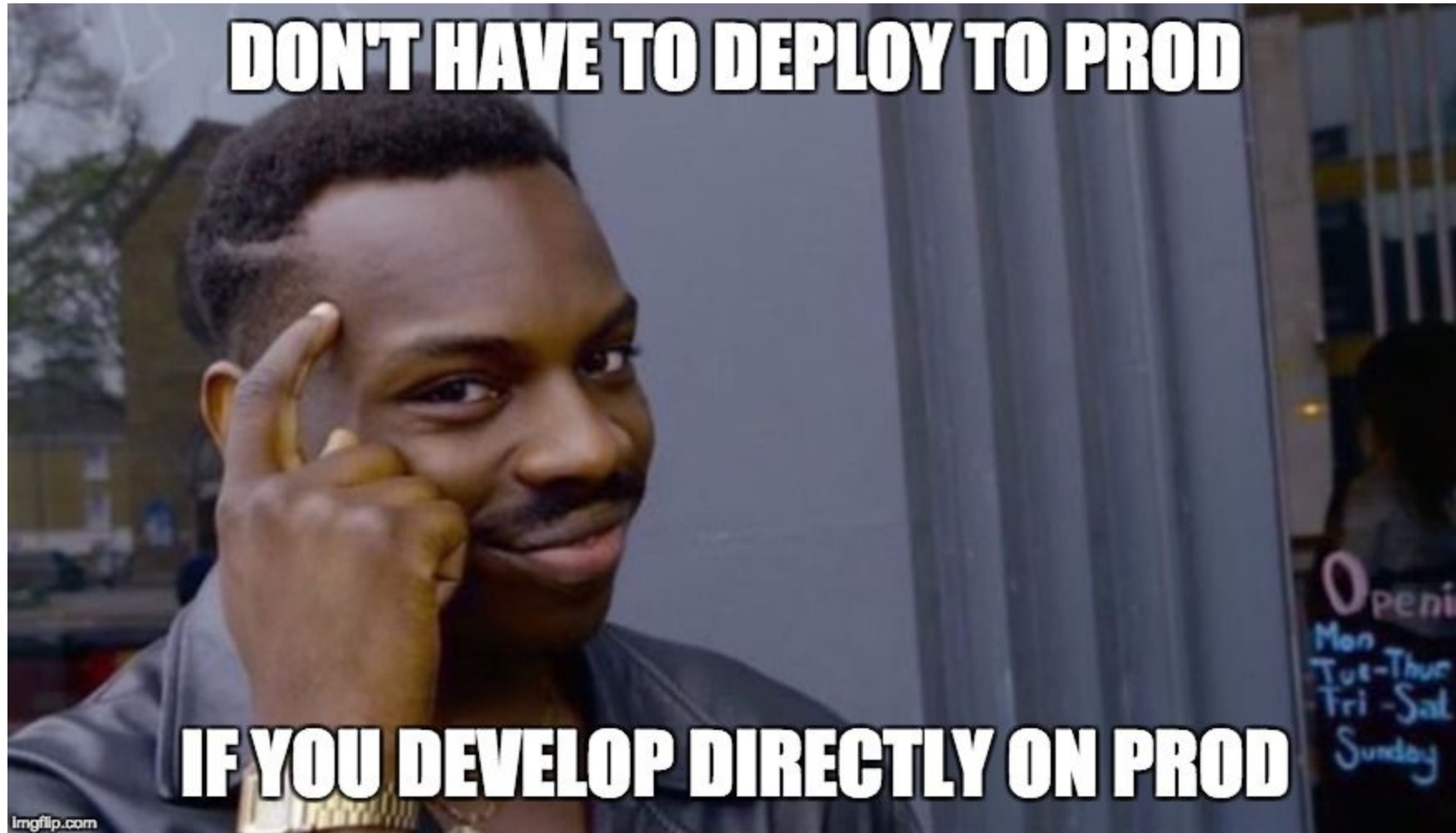
1. Linter

Расширенное описание задания, подсказки, а также презентации с лекций всегда есть в репозитории.

Срок сдачи

12 декабря

Мем дня



Спасибо за внимание!

Пока!

Присоединяйтесь к сообществу про образование в VK

- [VK Образование](#)

