

Веб-сервер на Linux

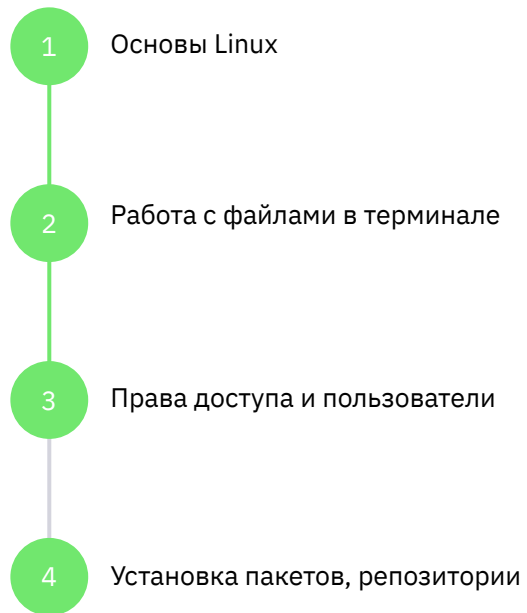




Содержание урока







План курса





Что будет на уроке сегодня

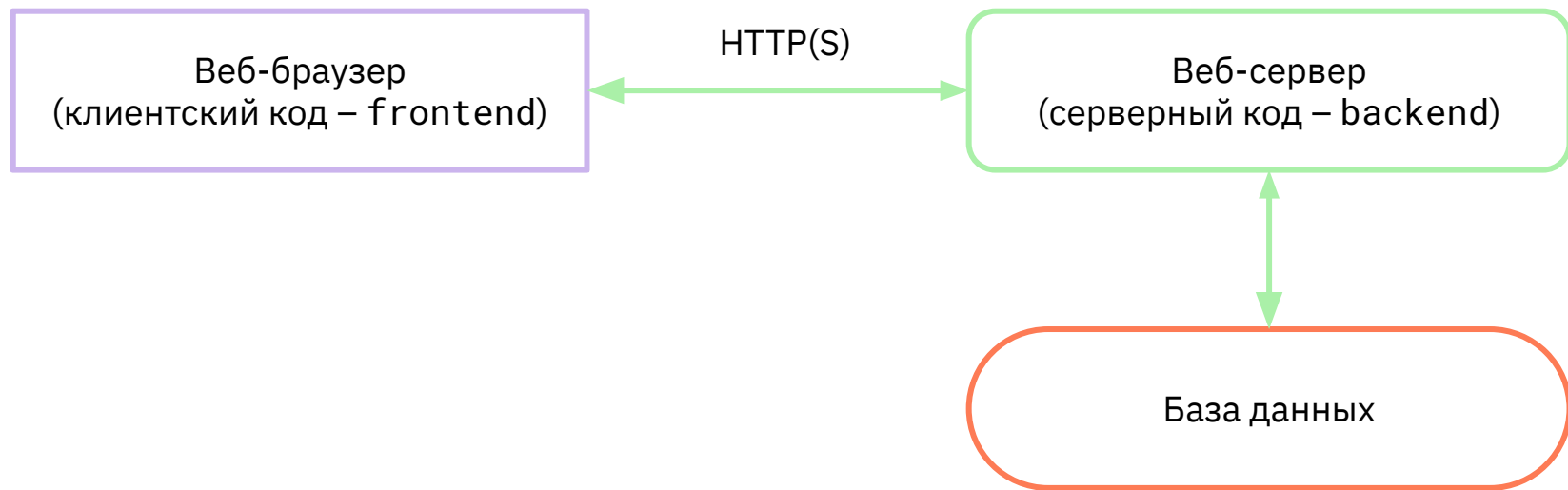
-  Узнаем, как устроены веб-приложения
-  Изучим принципы работы веб-сервера
-  Разберем схему работы обратного прокси
-  Познакомимся с СУБД MySQL



Компоненты веб-приложения



Компоненты веб-приложения





HTTP-запрос

```
GET /doc/test.html HTTP/1.1
```

```
Host: www.test101.com
```

```
Accept: image/gif, image/jpeg, */*
```

```
Accept-Language: en-us
```

```
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/4.0
```

```
Content-Length: 35
```

```
bookId=12345&author=Tan+Ah+Teck
```

Request Line

Request Headers

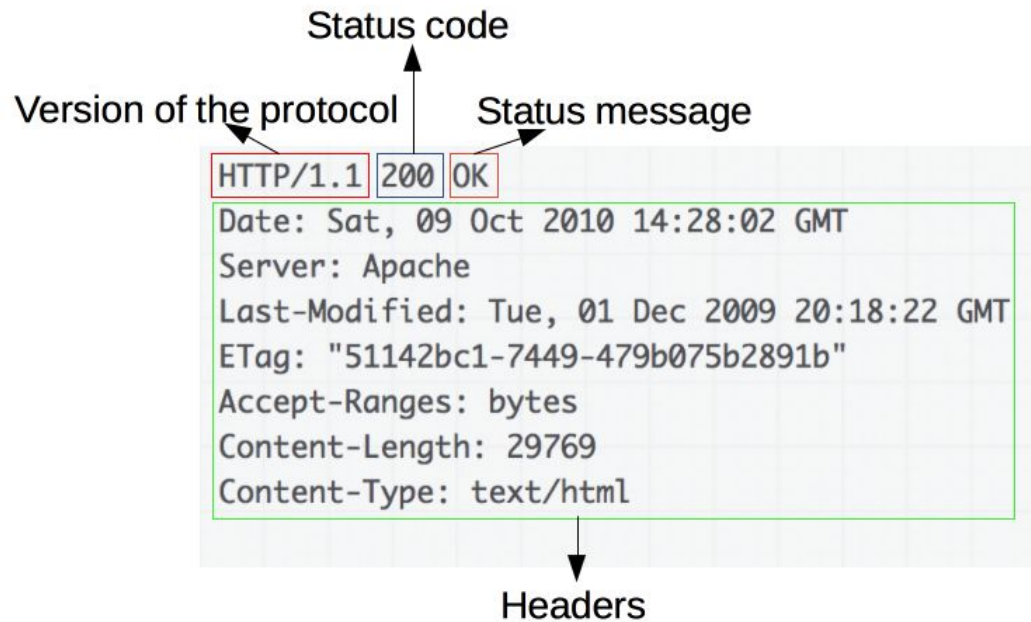
Request
Message
Header

A blank line separates header & body

Request Message Body



HTTP-ответ





Консольные утилиты для веб

- Получить URL в консоли:
 - `curl -L https://ya.ru/`
 - `wget https://yastatic.net/jquery/2.1.4/jquery.min.js`



Протоколы SSL/TLS и HTTPS

- SSL – первые версии защищенного протокола
- TLS – современный протокол
- HTTPS – работа протокола HTTP поверх защищенного канала SSL/TLS
- HTTP/2, HTTP/3 – по умолчанию используют защищенный вариант

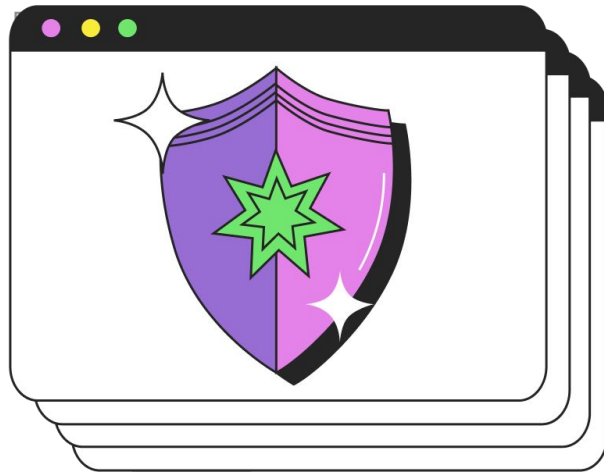
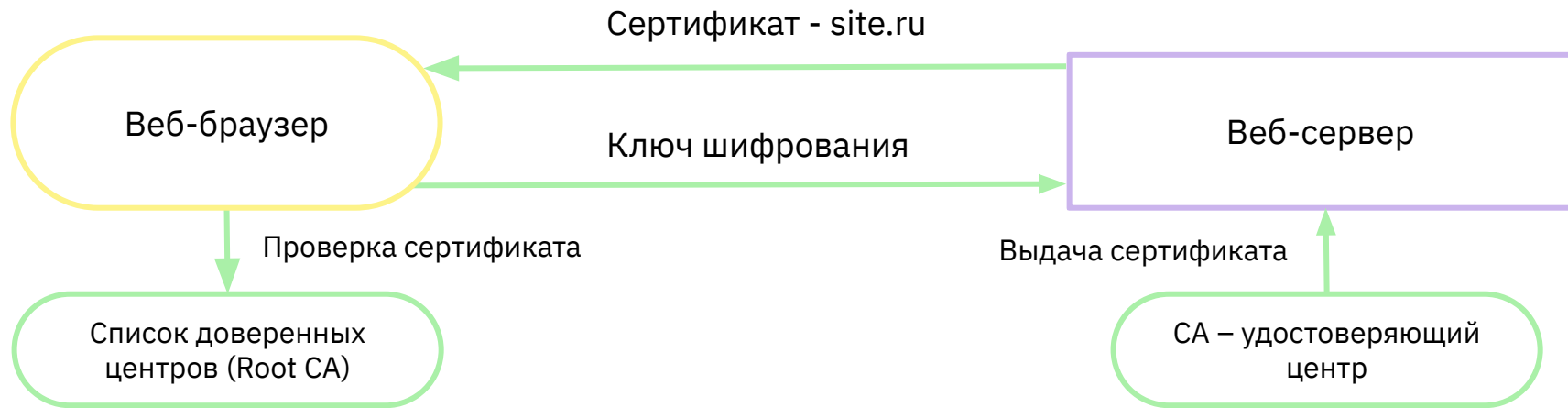


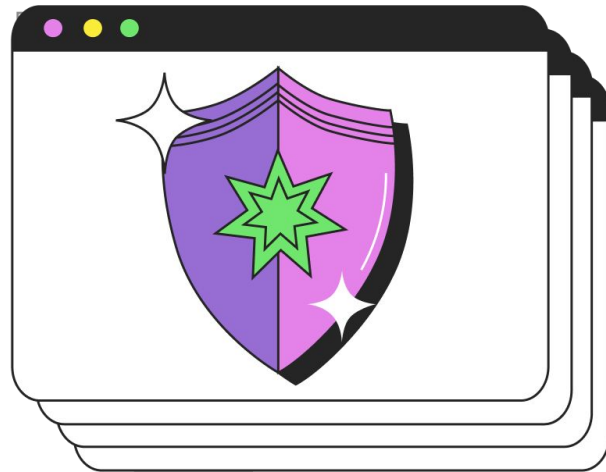


Схема работы HTTPS



Преимущества HTTPS

- Трафик шифруется при передаче
 - Нет перехвата данных
 - Нельзя изменить содержимое
- Подтверждение соответствия сервера домену
- Надежная доставка через цепочку хостов
- Поддержка новейших веб-технологий





Веб-сервер Nginx



Nginx

- Самый популярный сервер в мире
- Высокая производительность
- Масштабируемость по количеству подключений
- Открытый продукт (коммерческий Nginx Plus)
- Широкая функциональность
- Модульная архитектура
- Поддержка новых технологий
- Используется как сервер для статики, прокси, кэширующий сервер
- Не исполняет серверный код веб-приложения
- Разработчик первых версий: Игорь Сысоев

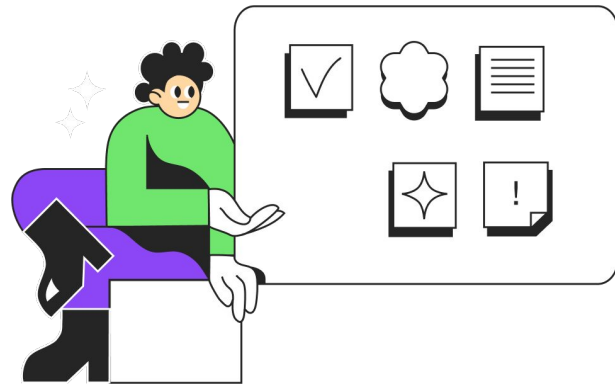


NGINX[®]
Part of F5



Конфигурация Nginx

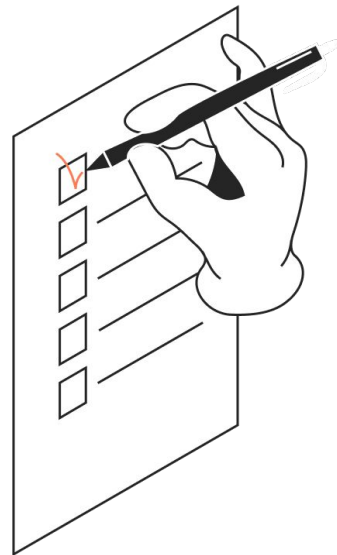
- Установка: `sudo apt install nginx`
- Тестирование конфигурации: `sudo nginx -t`
- Применить: `sudo systemctl reload nginx`
- Конфигурация: `/etc/nginx/*`
- Основной файл: `/etc/nginx/nginx.conf`
- Блоки: `server {}`
- Директивы: `server_name site.ru;`
- Переменные: `$uri`
- Документация: <http://nginx.org/ru/docs/>





Пример конфигурации Nginx

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /var/www/html;  
  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name _;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```





Веб-сервер Apache



Веб-сервер Apache

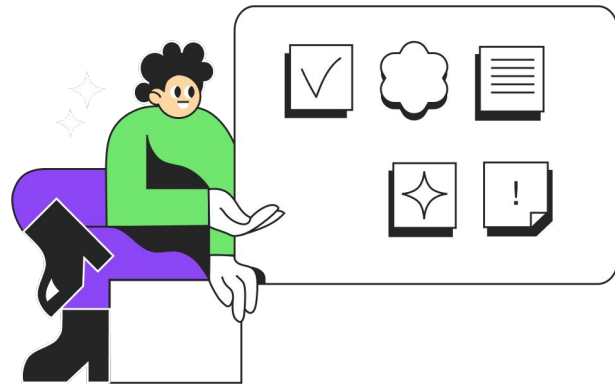
- Популярный веб-сервер
- Открытый продукт
- Максимальная функциональность
- Модульная архитектура
- Разные режимы работы (MPM – prefork, event...)
- Плохая масштабируемость по соединениям (prefork)
- Возможность исполнения серверного кода (модули)
- Исполняет роль backend-сервера
- Документация: <https://httpd.apache.org/docs/2.4/en/>





Конфигурация Apache

- Установка: `sudo apt install apache2`
- Тестирование конфигурации: `sudo apachectl -t`
- Применить: `sudo systemctl reload apache2`
- Конфигурация: `/etc/apache2/*`
- Основной файл: `/etc/apache2/apache2.conf`
- Блоки: `<VirtualHost></VirtualHost>`
- Директивы: `ServerName site.ru`
- Документация: <https://httpd.apache.org/docs/2.4/en/>





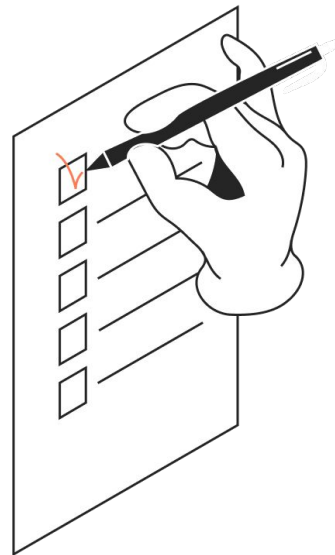
Пример конфигурации Apache

```
Listen 8081
<VirtualHost *:8081>
    ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    ErrorLog /var/log/httpd/error.log
    CustomLog /var/log/httpd/access.log combined

</VirtualHost>
```

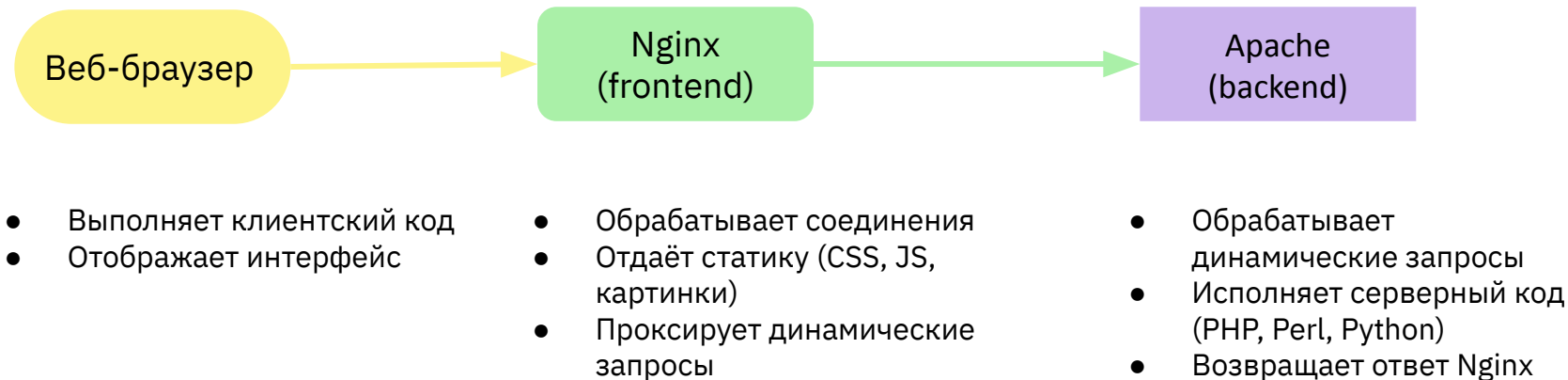




Динамический и статический контент. Reverse proxy



Схема Reverse proxy и типы контента

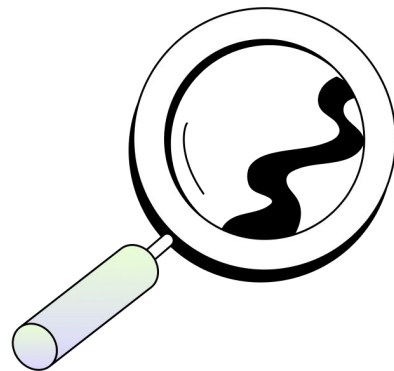




Пример конфигурации Nginx для Reverse proxy

```
# Динамические запросы
location / {
    proxy_pass http://localhost:8080;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Real-IP $remote_addr;
}

# Статические запросы
location ~* ^.+.(jpg|jpeg|gif|png|ico|css|zip|pdf|txt|tar|js)$ {
    root /var/www/html;
}
```





Обработка PHP в Apache

- Установка: `apt install libapache2-mod-php8.1 php8.1`
- Обработка PHP подключается автоматически, по расширению файла
- Файл для проверки:

```
<?php  
phpinfo();  
?>
```

- Проверяем: `curl http://localhost/info.php`





Обработка PHP в PHP-FPM

- Установка: `apt install php8.1-fpm`
- Замена Apache для работы с PHP
- Связь между Nginx и PHP-FPM через UNIX-сокеты
- Обработка происходит с помощью location в Nginx

```
location ~ \.php$ {  
    include snippets/fastcgi-php.conf;  
    root /var/www/html;  
    fastcgi_pass unix:/run/php/php8.1-fpm.sock;  
}
```



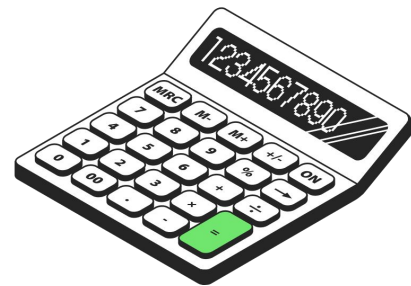


Система управления базами данных MySQL



СУБД MySQL

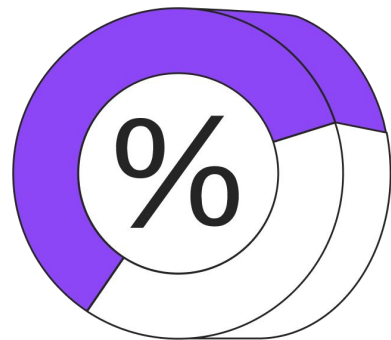
- Самая популярная СУБД для веб-приложений
- Клиент-серверная архитектура
- Несколько веток развития:
 - **Oracle MySQL**
 - Percona Server for MySQL
 - MariaDB
- Поддержка транзакций
- Репликация
- Поддерживает синтаксис SQL с расширениями
- Документация: <https://dev.mysql.com/doc/refman/8.0/en/>





Установка MySQL и первые шаги

- Установка: `apt install mysql-server-8.0`
- Заходим в консоль MySQL: `sudo mysql`
- Переходим в системную БД mysql: `use mysql;`
- Получаем список пользователей: `SELECT * FROM user\G`
- Создаём новую базу данных: `CREATE DATABASE gb;`
- Создаём таблицу: `CREATE TABLE test(i INT);`
- Создадим записи в таблице: `INSERT INTO test (i) VALUES (1),(2),(3),(4);`
- Сделаем выборку из таблицы: `SELECT * FROM test;`





Итоги занятия



На этом уроке мы

- 📌 Разобрали, из чего состоит веб-приложение
- 📌 Изучили основы протокола HTTP
- 📌 Узнали, что такое HTTPS
- 📌 Запустили веб-сервер Nginx
- 📌 Настроили схему обратного прокси с использованием Nginx
- 📌 Разобрали деление контента на динамический и статический
- 📌 Познакомились с сервисом PHP-FPM
- 📌 Установили СУБД MySQL и проверили базовые команды





Спасибо за внимание!