

# Базовое Администрирование Linux

Занятие 8



Дмитрий  
Молчанов

# LAMP: HTTP-сервер

---



1. LAMP
2. Обзор протокола HTTP
3. Медленные клиенты
4. nginx vs. apache
5. Ключевые моменты в конфигурации
6. Настройка apache
7. Настройка nginx
8. Сопрягаем nginx и apache

LAMP = Linux Apache Mysql Php.

В начале, когда эта аббревиатура только появлялась, она обозначала именно такой стек OS/Софта.

Linux – бесплатный, популярный

Apache – наиболее популярный и бесплатный http-сервер, достаточно простой в настройке, кроссплатформенный

Mysql – Кроссплатформенный, популярный, sql-сервер с кучей возможностей, которых более чем полостью хватало 99% web-проектов.

PHP – кроссплатформенный скрипто-язык с

Со временем apache стал терять ключевое место в этой связке. Чаще стали употребляться nginx+apache или просто nginx.

mysql все чаще стал заменять форками mysql – percona, Mariadb, nosql-решениями или postgresql.

Php, со временем, стал сдавать позиции python'у и ruby.

В общем сейчас классический LAMP все еще популярен, но раньше говоря web-сервер подразумевался, чаще всего, именно LAMP, а сейчас – http-сервер+СУБД+скриптовый язык

HTTP – HyperText Transfer Protocol.

Пожалуй самый популярный протокол в интернете, наряду с DNS и SMTP. Когда вы открываете web-страницу вы используете протокол http.

Для работы используется 80/tcp по умолчанию, но может использоваться любой другой порт. для https (HTTP Secure/SSL) используется 443/tcp.

HTTP – простой, гибкий, диалоговый,.

# HTTP - простой



- Текстовый (не бинарный)
- 7 базовых методов
  - 3 самых популярных – GET, POST, HEAD
- Запросы и ответы могут быть легко
  - сформированы «ручками»
  - Прочитаны глазами или отфильтрованы
- простой формат запроса:
  - Method URI ProtocolVersion
  - +headers

# HTTP - Диалоговый

---



HTTP – диалоговый протокол.

В пределах одного соединения идет обмен парами

Запрос – Ответ

# HTTP - гибкий



HTTP Может быть использован в качестве транспорта для какого-либо вашего протокола, т.к. используя запросы+заголовки и тело, в которое можно включить любую текстовую информацию – можно делать очень много. как пример REST API веб-приложений.



# Пример HTTP-запроса

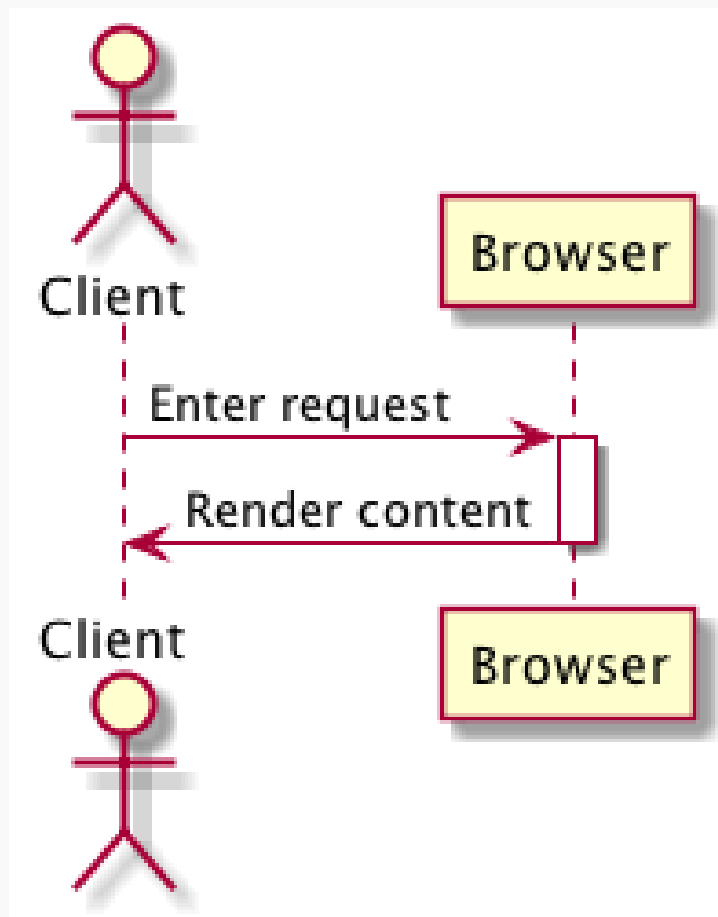


```
GET / HTTP/1.1
Host: mail.ru

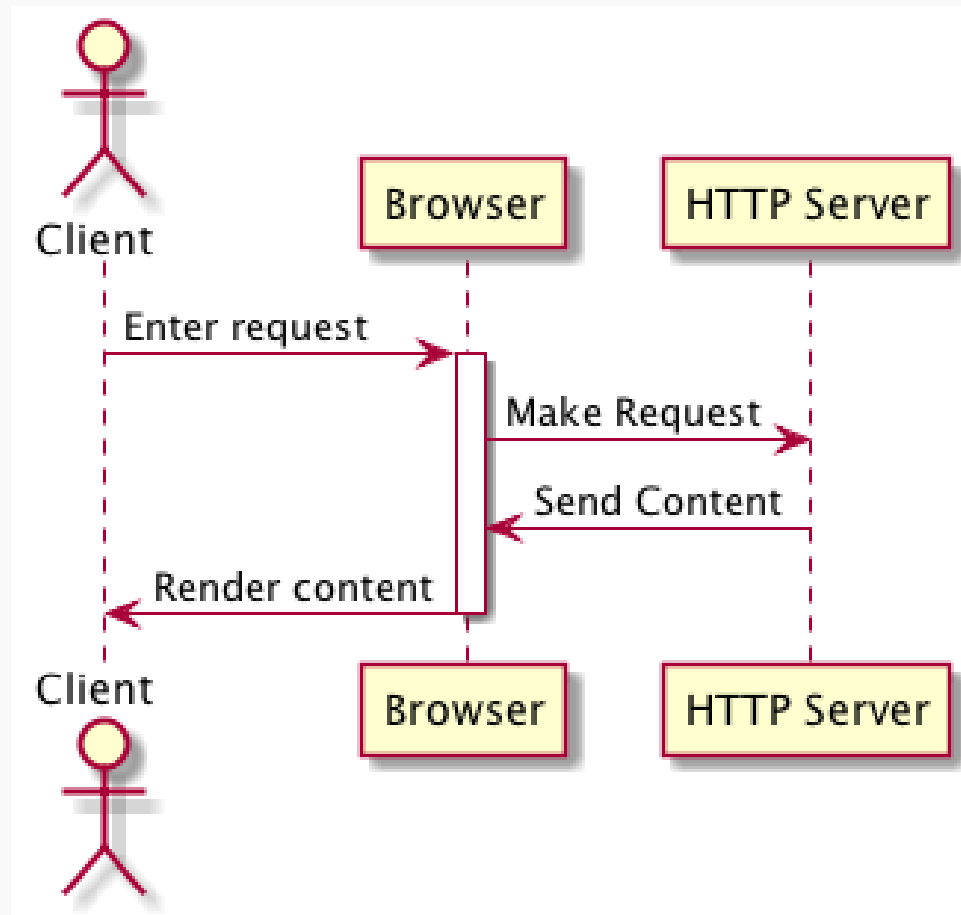
HTTP/1.1 302 Moved Temporarily
Server: nginx/1.6.2
Date: Sat, 14 Nov 2015 18:56:48 GMT
Content-Type: text/html
Content-Length: 37
Connection: keep-alive
Location: https://mail.ru
Set-Cookie: mrcu=04DA564783F033C81327AA79BC2E; expires=Tue, 11 Nov 2025
Cache-Control: no-cache,no-store,must-revalidate
Pragma: no-cache
Expires: Fri, 14 Nov 2014 18:56:48 GMT
Last-Modified: Sat, 14 Nov 2015 21:56:48 GMT
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block; report=https://cspreport.mail.ru/xxssp
X-Content-Type-Options: nosniff

<html><body>Redirect...</body></html>^]
```

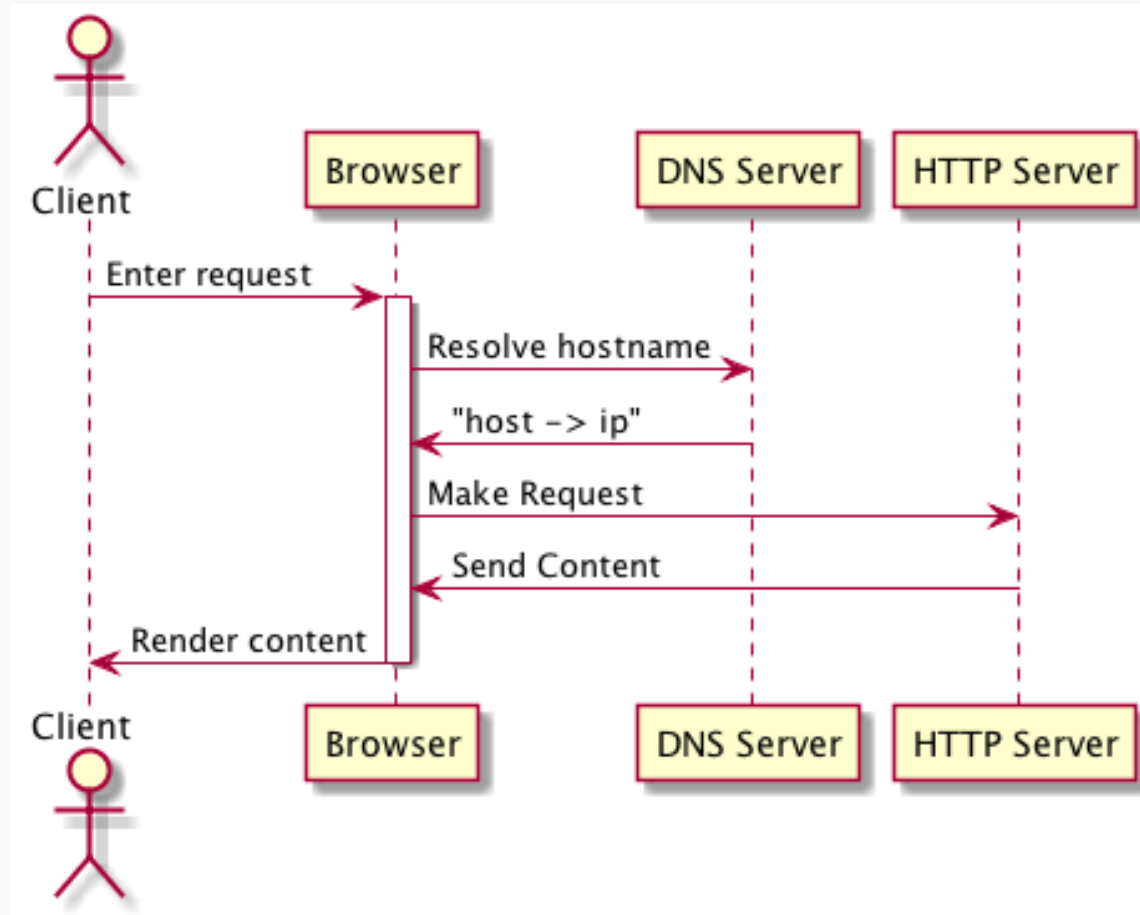
# Что такое web-запрос



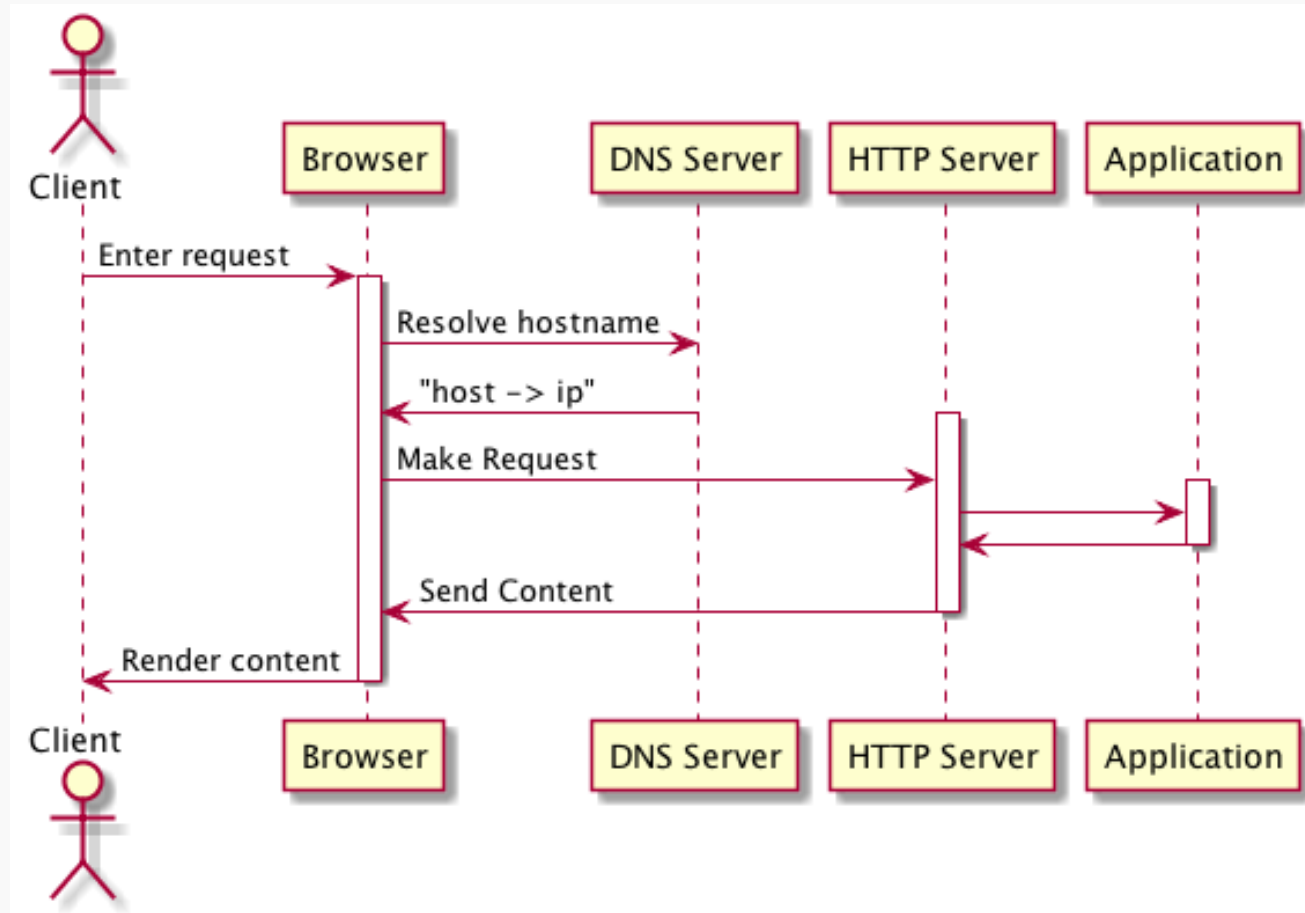
# Что такое web-запрос



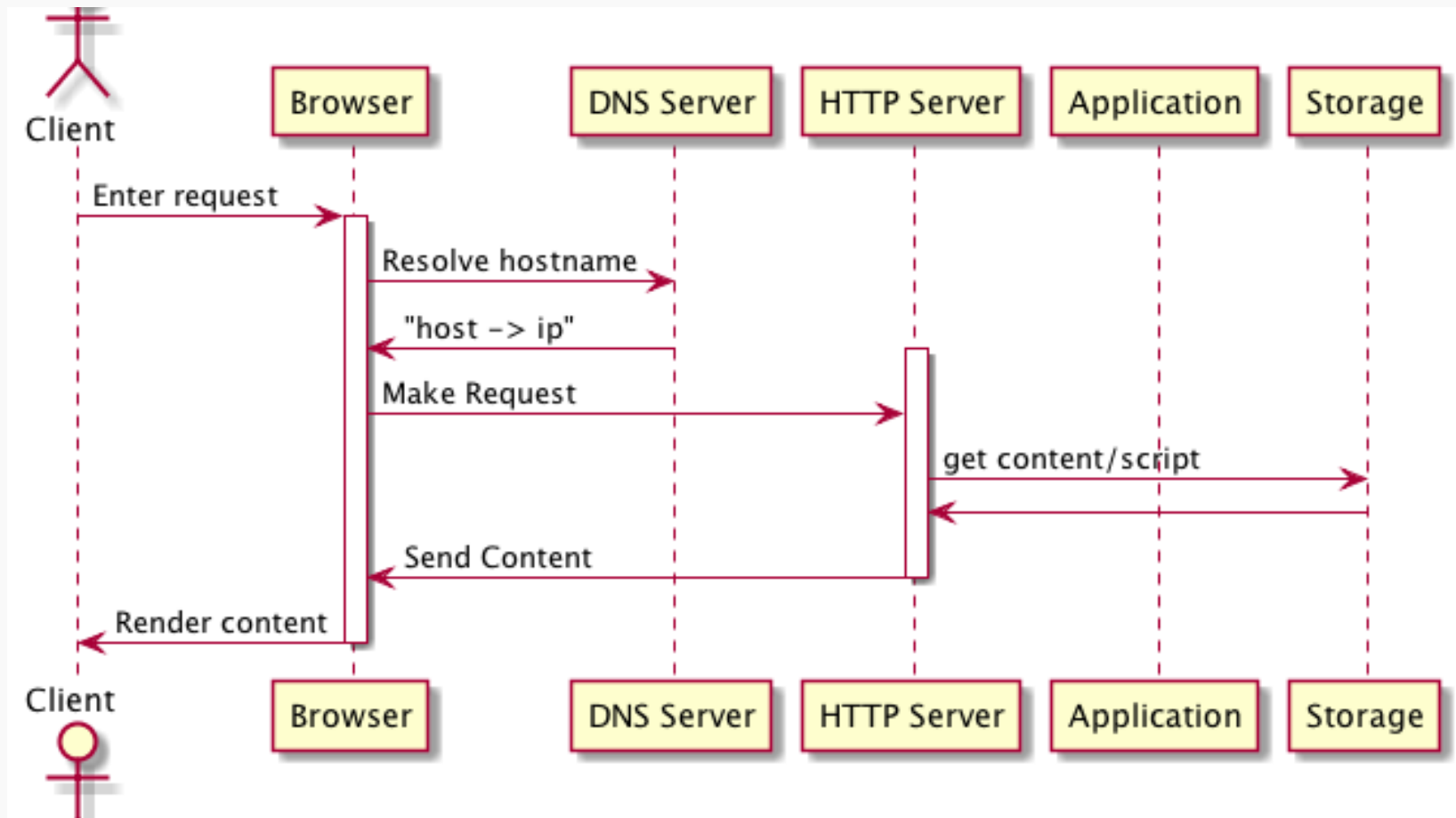
# Что такое web-запрос



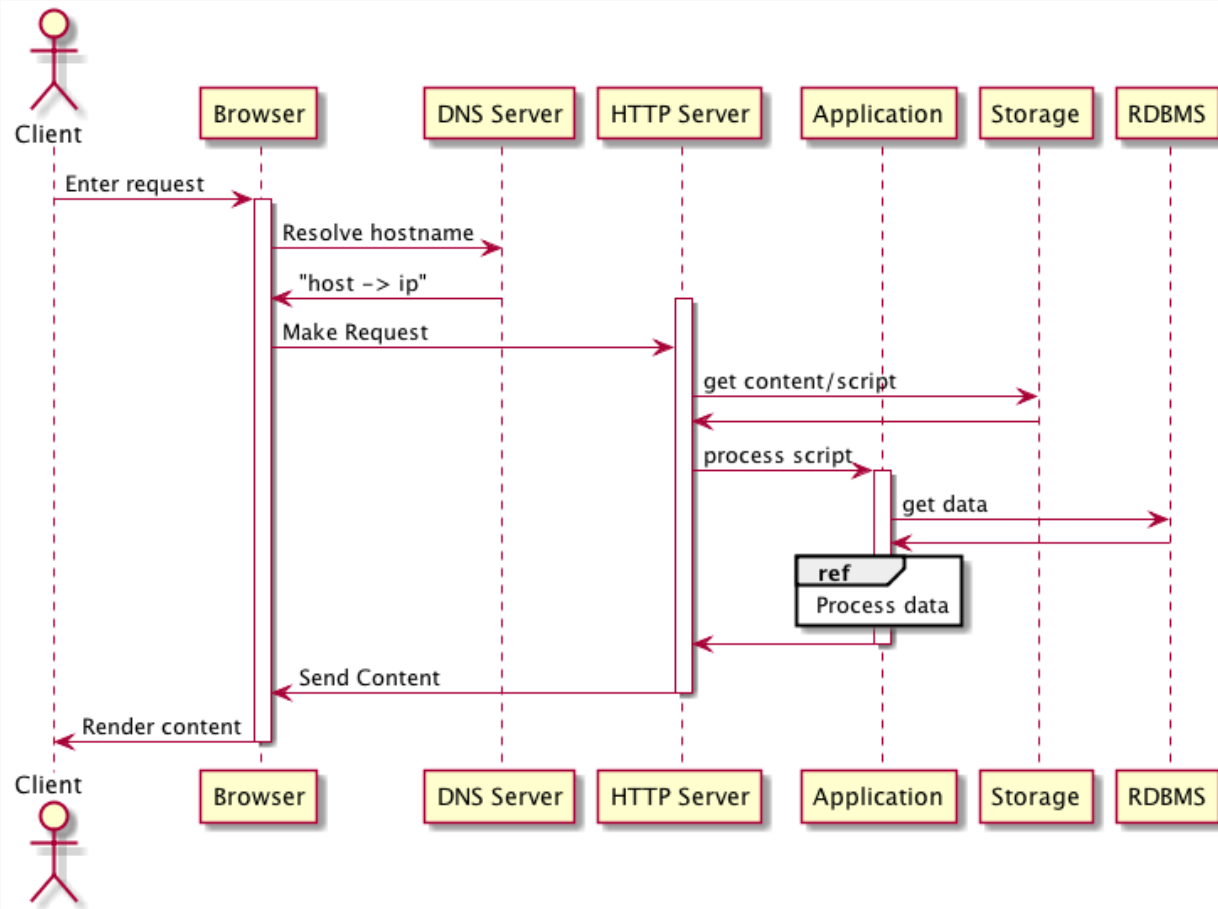
# Что такое web-запрос



# Что такое web-запрос



# Что такое web-запрос



# Медленные клиенты или nginx спешит на помощь

---



Медленные клиенты – клиенты которые выкачивают контент медленно, умышленно или вынужденно. Например клиент со старого модема или по медленному GPRS-соединению или атакующий, который забирает контент по 1 байту в секунду.



# nginx спешит на помощь



Ключевое отличие apache от nginx – метод обработки соединений.

nginx – event-based

apache – process-based, thread-based, event-based

Основная проблема process-based в том, что на одно соединение нужен один процесс, а php не работает\* ни с чем, кроме mpd-worker (process-based apache)

# Ключевые моменты конфигурации

---



- Контексты конфигурации
- Виртуальные сервера
- Наследование и переопределение.
- listen – что слушать
- document root – где лежит то, что отдавать
- 2 разных http сервера не могут слушать на одной паре ip:port

## HTTP Server

### Root Context:

- options
  - logs
  - docRoot
  - Bind/Listen:

addr1:80

addr2:81

### VirtualServer:

Names, docRoot

Directory

Location

### VirtualServer:

Names, docRoot

Directory

Location

### VirtualServer:

Names, docRoot

Directory

Location

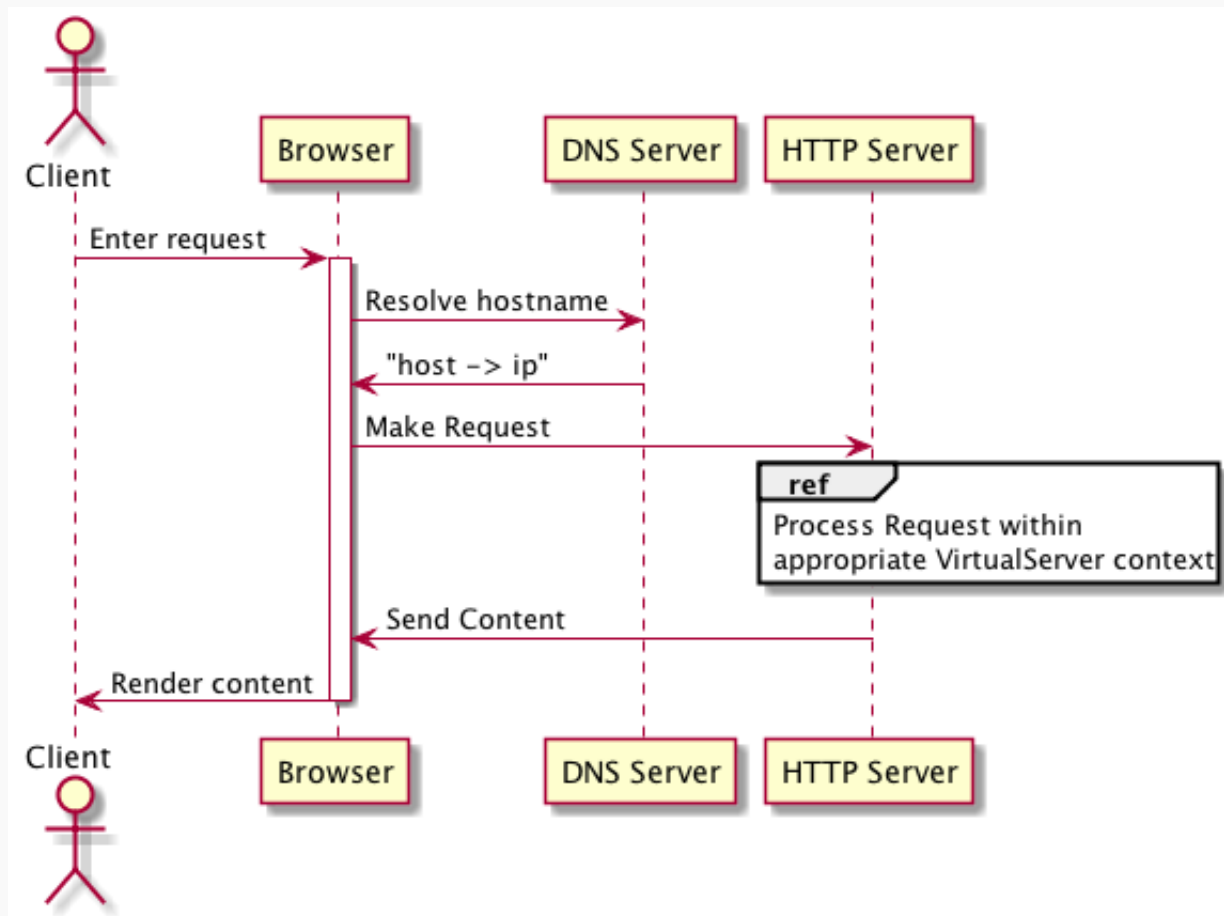
# Обработка HTTP-запроса

---



1. Принять соединение
2. Прочитать запрос
3. Проанализировать заголовок Host: запроса
4. Обработать запрос в соответствующем контексте VirtualServer
5. Если этому Host не соответствует ни один server\_name – обработать в default.

# Жизнь HTTP-запроса



# Настройка apache



- Ключевые параметры
  - Listen – [addr:]port которые слушать.
  - ServerName – основное имя сервера
  - ServerAlias – дополнительные имена сервера
  - NameVirtualHost – на каких Listen-портах обрабатывать VirtualHost'ы
  - VirtualHost – Контекст Виртуального сервера
  - include – включение в текущий контекст файла с директивами
  - LogFormat – описание формата лога
  - CustomLog – форматированный лог

# Ползености apache

---



- .htaccess – возможность поменять настройки в контексте текущего каталога. Не все, зависит от директивы (возможно переопределить или нет) и значения директивы AllowOverride контекста директории.

# Настройка nginx



- `worker_processes` – количество рабочих процессов
- `error_log` – лог ошибок
- `pid,user,include*`
- **events**
  - `worker_connections`
- **http** – контекст http-server'a
  - `access_log,log_format`
  - **server** – контекст виртуального сервера
    - `server_name`
    - `listen`
    - `root`



# Сопряжение nginx и apache



1. nginx слушает запросы на 80м порту и обрабатывает запросы к статическим файлам, запросы к динамическому контенту отдаются в apache.
2. nginx слушает запросы на 80м порту и перенаправляет ВСЕ запросы которые надо обработать к apache. Может использоваться для разгрузки apache от медленных клиентов.



**skype: dmolchanov**  
**email:**  
**dmolchanov@gmail.com**

**Спасибо за внимание!**