

# Веб-сервер, прокси и туннелирование

---

1 апреля 2026 г.

# Что такое веб-сервер?

**HTTP Веб-сервер** — программное обеспечение, которое принимает HTTP-запросы от клиентов, обрабатывает и возвращает HTTP-ответы

## Принцип работы:

1. Клиент отправляет HTTP запрос
2. Веб-сервер обрабатывает запрос и возвращает HTTP ответ

## Примеры веб-серверов:

- Apache
- Nginx
- Python http.server

## Request

```
POST / HTTP/1.1
```

```
Host: developer.mozilla.org
```

```
User-Agent: curl/8.6.0
```

```
Accept: */*
```

```
Content-Type: application/json
```

```
Content-Length: 345
```

```
{  
  "data": "ABC123"  
}
```

Start line

Headers

Empty line

Body

## Response

```
HTTP/1.1 403 Forbidden
```

```
Server: Apache
```

```
Date: Fri, 21 Jun 2024 12:52:39 GMT
```

```
Content-Length: 678
```

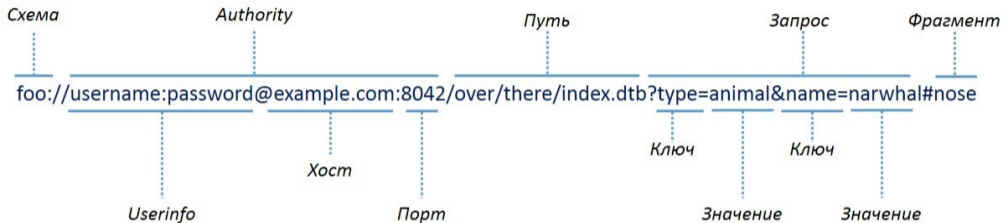
```
Content-Type: text/html
```

```
Cache-Control: no-store
```

```
<!DOCTYPE html>  
<html lang="en">  
(more data...)
```

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Messages>

# Uniform Resource Identifier (URI)



<https://habr.com/ru/articles/190154/>

## Что покажем:

- Запуск статического и динамического веб-сервера
- Обращение к веб-серверу из браузера, developers tools
- Обращение к веб-серверу с помощью curl

## Демо-репозиторий:

<https://github.com/prafdin/devops-course-demos/tree/webserver-proxy-tunneling>

**Прокси сервер** — сервер-посредник между клиентом (кто запрашивает ресурс) и сервером (кто предоставляет ресурс)

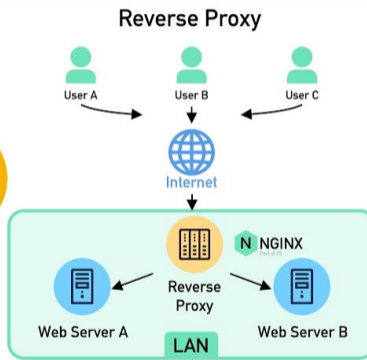
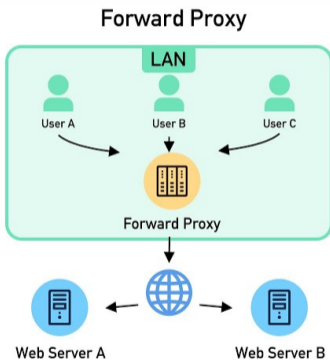
## **Forward proxy:**

- Клиент → Прокси → Интернет
- Скрывает клиента от сервера
- Пример: корпоративный прокси

## **Reverse proxy:**

- Интернет → Прокси → Сервер
- Скрывает сервер от клиента
- Пример: nginx перед приложением

# Forward Proxy vs. Reverse Proxy



### Что покажем:

- Маршрутизация по портам
- Маршрутизация по path
- Маршрутизация по заголовку Host

### Демо-репозиторий:

<https://github.com/prafdin/devops-course-demos/tree/webserver-proxy-tunneling>

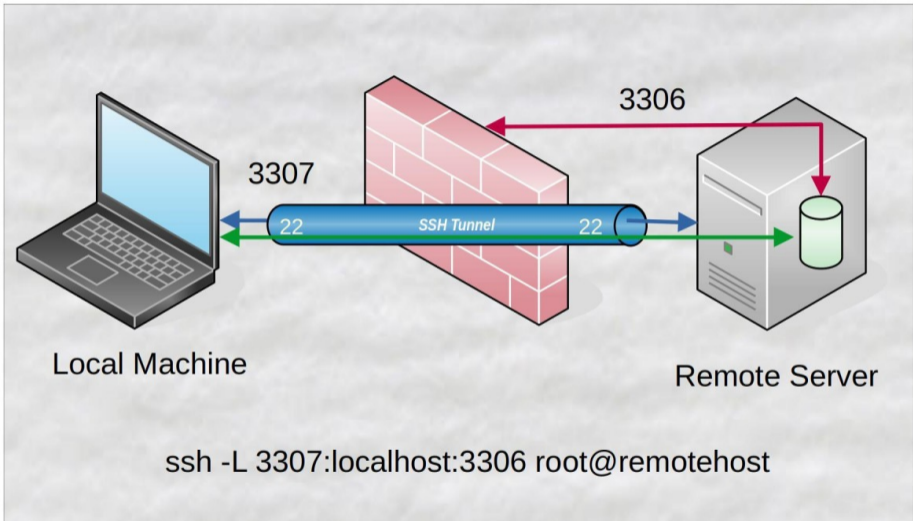
Туннелирование — технология инкапсуляции одного сетевого протокола в другой для создания канала передачи данных

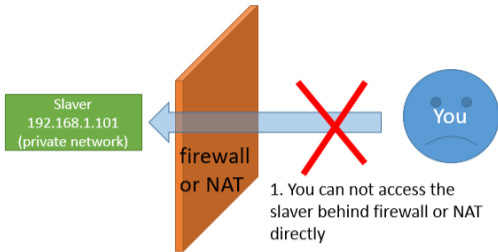
## Назначение:

- Обход сетевых ограничений (NAT, firewall)
- Обеспечение безопасности передачи данных

## Принцип работы:

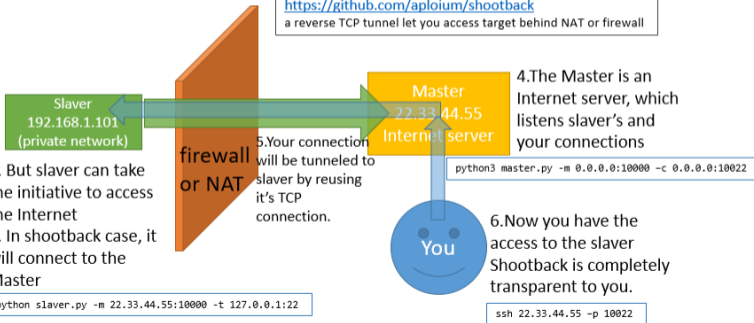
1. Исходный трафик упаковывается в другой протокол
2. Передается через промежуточный сервер
3. Распаковывается на стороне получателя





1. You can not access the slaver behind firewall or NAT directly

<https://github.com/aploium/shootback>  
a reverse TCP tunnel let you access target behind NAT or firewall



2. But slaver can take the initiative to access the Internet  
3. In shootback case, it will connect to the Master

```
python slaver.py -m 22.33.44.55:10000 -t 127.0.0.1:22
```

5. Your connection will be tunneled to slaver by reusing it's TCP connection.

4. The Master is an Internet server, which listens slaver's and your connections

6. Now you have the access to the slaver  
Shootback is completely transparent to you.

```
ssh 22.33.44.55 -p 10022
```

# Демонстрация: Туннелирование через SSH

## Что покажем:

- Запуск веб-сервера на удаленной машине
- Создание SSH туннеля: локальный порт → удаленный сервис
- Доступ к удаленному серверу через локальный порт
- Перенаправление запросов с удаленного сервера на локальный порт

## Демо-репозиторий:

<https://github.com/prafdin/devops-course-demos/tree/webserver-proxy-tunneling>

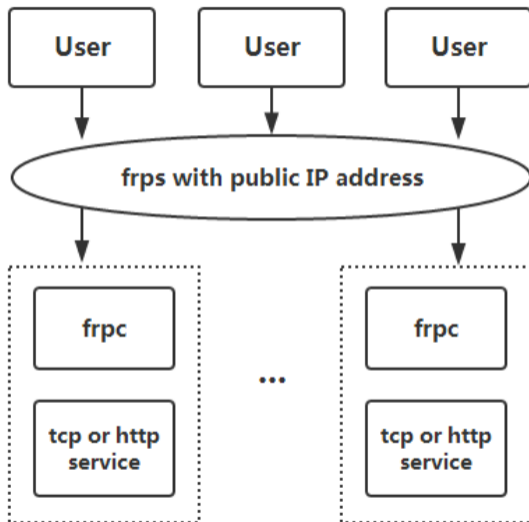
**Fast Reverse Proxy (FRP)** — обратный прокси сервер с поддержкой туннелирования до клиентов

### Возможности FRP:

- HTTP туннели
- TCP туннели
- UDP туннели
- Работа через NAT и firewall

### Архитектура:

Клиент → FRP Server ← FRP Client ← Сервер



### Что покажем:

- Установка и настройка FRP клиента
- Создание HTTP туннеля для веб-сервера
- Создание TCP туннеля для SSH доступа
- Доступ к внутренним сервисам через внешний домен

### Демо-репозиторий:

<https://github.com/prafdin/devops-course-demos/tree/webserver-proxy-tunneling>

- **HTTP** — основа веб-коммуникации (заголовки, статус коды)
- **Forward Proxy** — клиент → прокси → интернет
- **Reverse Proxy** — интернет → прокси → сервер
- **Туннелирование** — создание канала передачи данных через сеть

**Вопросы?**