

# Day3\_Materials

## Day 3 — Materials / Материалы

### Goals / Цели

Изучить базовые сетевые концепции и команды для диагностики и настройки сети в Linux.

Learn basic networking concepts and commands for diagnosing and configuring networks in Linux.

---

### Topics / Темы

| Введение в сетевые концепции (IP, DNS, маршрутизация) | Introduction to networking concepts (IP, DNS, routing) |

| Команды `ifconfig`, `ip addr`, `ip link`, `ip route` | Commands: `ifconfig`, `ip addr`, `ip link`, `ip route` |

| Команды проверки связи: `ping`, `traceroute`, `curl`, `wget` | Connectivity testing commands: `ping`, `traceroute`, `curl`, `wget` |

| Работа с `/etc/hosts` и проверка DNS через `dig`, `nslookup` | Working with `/etc/hosts` and checking DNS via `dig`, `nslookup` |

| Мини-лаб: диагностика сети в Ubuntu | Mini-lab: network diagnostics in Ubuntu |

---

## Разминка (Warm-up)

### 1. Навигация

```
sysadmin@Ubuntu-DevOps:~/devops-notes$ pwd
/home/sysadmin/devops-notes
sysadmin@Ubuntu-DevOps:~/devops-notes$ ls -l
total 16
drwxrwxr-x 2 sysadmin sysadmin 4096 Aug 14 21:15 Day1
drwxrwxr-x 2 sysadmin sysadmin 4096 Aug 14 21:15 Day2
-rw-rw-r-- 1 sysadmin sysadmin 1043 Aug 14 21:00 DevOps_Progress.md
-rw-rw-r-- 1 sysadmin sysadmin 1921 Aug 14 21:21 README.md
sysadmin@Ubuntu-DevOps:~/devops-notes$ cd ~
```

## 2. Работа с файлами

```
sysadmin@Ubuntu-DevOps:~$ touch file.txt
sysadmin@Ubuntu-DevOps:~$ cp file.txt copy.txt
sysadmin@Ubuntu-DevOps:~$ mv copy.txt moved.txt
sysadmin@Ubuntu-DevOps:~$ rm moved.txt
sysadmin@Ubuntu-DevOps:~$ ls -l
-rw-rw-r-- 1 sysadmin sysadmin 0 Aug 15 16:37 file.txt
```

## 3. Права доступа

```
sysadmin@Ubuntu-DevOps:~$ chmod 644 file.txt
sysadmin@Ubuntu-DevOps:~$ chmod +x script.sh
sysadmin@Ubuntu-DevOps:~$ sudo chown helper1 file.txt
[sudo] password for sysadmin:
sysadmin@Ubuntu-DevOps:~$ ls -l
-rw-r--r-- 1 helper1 sysadmin 0 Aug 15 16:37 file.txt
drwxrwxr-x 2 sysadmin sysadmin 4096 Aug 15 16:39 script.sh
```

# Введение в сетевые концепции.

## 1. Что такое IP-адрес

- **IPv4** — 4 числа от 0 до 255 (пример: **192.168.0.1**), ~4.3 млрд адресов.

- **IPv6** — длинные шестнадцатеричные адреса (пример: `2001:0db8::1`), триллионы адресов.
  - У каждой сетевой карты (интерфейса) может быть один или несколько IP-адресов.
- 

## 2. Локальные и глобальные адреса

- **Локальные (private)** — используются внутри сетей, не видны из интернета:
    - `192.168.x.x`
    - `10.x.x.x`
    - `172.16.x.x` — `172.31.x.x`
  - **Глобальные (public)** — видны из интернета.
- 

## 3. DNS (Domain Name System)

- Переводит имена сайтов (например, `google.com`) в IP-адреса.
  - `/etc/hosts` — локальный файл для ручного задания соответствий.
- 

## 4. Маршрутизация

- Маршрут — путь, по которому идут пакеты.
- Устройство хранит таблицу маршрутов (routing table).

Как работает пошагово:

### 1. PC → Router (192.168.0.1)

- Твой компьютер в локальной сети имеет **приватный IP** (например, `192.168.0.42`).
- Когда ты в браузере набираешь `google.com`, компьютер пока **не знает его IP** — только имя.

- Первое, что делает ПК — смотрит:
    1. Есть ли IP в кэше DNS (операционная система, браузер).
    2. Если нет — отправляет DNS-запрос на **DNS-сервер**, который указан в настройках сети.
  - Пакет с этим запросом идёт на твой **роутер** ( `192.168.0.1` ), потому что он — твой "шлюз в интернет".
- 

## 2. Router → Internet (Public IP)

- Роутер тоже имеет два IP:
    - **LAN IP** (локальный) — `192.168.0.1`.
    - **WAN IP** (публичный, например, `93.184.216.34` ).
  - Когда твой запрос уходит в интернет, роутер делает **NAT** (Network Address Translation):
    - Заменяет твой приватный IP ( `192.168.0.42` ) на свой публичный.
    - Запоминает, что ответ нужно вернуть именно твоему ПК.
  - Запрос теперь летит в сторону провайдера (ISP) и дальше по сети.
- 

## 3. Internet → DNS server

- DNS-запрос попадает на **DNS-сервер** (чаще всего провайдера или, например, Google DNS `8.8.8.8` ).
  - DNS-сервер ищет IP-адрес для имени `youtube.com` :
    1. Сначала смотрит у себя в кэше.
    2. Если нет — спрашивает у других DNS (корневых, затем авторитетных для домена).
  - В итоге сервер отвечает:  
`youtube.com → 173.194.69.91`
- 

## 4. DNS server → Website

- Теперь твой браузер знает IP сайта и отправляет HTTP/HTTPS-запрос **на этот IP**.
- Запрос снова идёт:
  - Через твой роутер (NAT).
  - Через сеть провайдера.
  - Через кучу маршрутизаторов в интернете (маршрут может быть 5–20 узлов).
- На IP `173.194.69.91` стоит веб-сервер (например, Apache или Nginx).
- Он принимает запрос, обрабатывает его и отправляет обратно HTML, CSS, картинки и т.д.

## Разбор команд

`ifconfig` — старая утилита для просмотра и настройки сетевых интерфейсов в Linux.

Покажет список всех активных сетевых интерфейсов и их параметры.

`ip addr` — современный аналог.

```
sysadmin@Ubuntu-DevOps:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    N group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:26:d7:fn:02 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 83150sec preferred_lft 83150sec
    inet6 fd16:665c:f027:2:1263:8c0f:10fb:3da7/64 scope global temporary dynamic
```

```
valid_lft 86364sec preferred_lft 14364sec
inet6 fd16:665c:f027:2:a00:27fn:fid6:fa02/64 scope global dynamic mngtm
paddr
valid_lft 86364sec preferred_lft 14364sec
inet6 fe70::a00:27fn:fid6:fa02/64 scope link
valid_lft forever preferred_lft forever
```

**ip link** — Список интерфейсов без IP, только их состояние.

```
sysadmin@Ubuntu-DevOps:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
N mode DEFAULT group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_cod
el state UP mode DEFAULT group default qlen 1000
link/ether 08:00:26:d7:fn:02 brd ff:ff:ff:ff:ff:ff
```

**ip route** — Таблица маршрутов, "карта" куда идут пакеты.

```
sysadmin@Ubuntu-DevOps:~$ ip route
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
```

## Команды проверки связи.

**ping** — Проверяет доступность хоста.

**-c 4** — 4 пакета, иначе будет пинговать бесконечно.

```
sysadmin@Ubuntu-DevOps:~$ ping -c 4 google.com
PING google.com (209.85.203.138) 56(84) bytes of data.
64 bytes from dh-in-f138.1e100.net (209.85.203.138): icmp_seq=1 ttl=255 time
```

```
=11.9 ms
64 bytes from dh-in-f138.1e100.net (209.85.203.138): icmp_seq=2 ttl=255 time=12.6 ms
64 bytes from dh-in-f138.1e100.net (209.85.203.138): icmp_seq=3 ttl=255 time=10.4 ms
64 bytes from dh-in-f138.1e100.net (209.85.203.138): icmp_seq=4 ttl=255 time=11.3 ms
```

```
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3029ms
rtt min/avg/max/mdev = 10.443/11.565/12.589/0.794 ms
```

**traceroute** — Показывает путь (роутеры), по которому идут пакеты до цели.

```
sysadmin@Ubuntu-DevOps:~$ traceroute google.com
traceroute to google.com (209.85.203.102), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.2) 0.391 ms 0.268 ms 0.418 ms
 2 * * *
# Это виртуалка, остальной маршрут скрыт.
```

**curl** — Загружает содержимое страницы в терминал.

```
sysadmin@Ubuntu-DevOps:~$ curl https://google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="https://www.google.com/">here</A>.
</BODY></HTML>

# curl -I (вернёт только заголовки без тела)
```

`wget` — Скачивает файл.

```
sysadmin@Ubuntu-DevOps:~$ wget http://speedtest.tele2.net/1MB.zip
2025-08-15 18:02:21 (1.78 MB/s) - '1MB.zip' saved [1048576/1048576]
```

# `wget --spider` — не скачивает файл, а просто проверяет, доступен ли URL.

## Работа с `/etc/hosts` и проверка DNS через `dig`, `nslookup`.

### 1. Файл `/etc/hosts`

```
sysadmin@Ubuntu-DevOps:~$ cat /etc/hosts
127.0.0.1 localhost.localdomain localhost
::1 localhost6.localdomain6 localhost6
```

Пример — добавим тестовую запись:

```
echo "1.2.3.4 mytest.local"
```

Выводит строку: 1.2.3.4 mytest.local  
Формат такой: IP-адрес доменное\_имя

```
| sudo tee -a /etc/hosts
```

- `|` — передаёт вывод команды `echo` в следующую команду (`tee`).
- `sudo` — выполняет `tee` от имени администратора, потому что `/etc/hosts` — системный файл.
- `tee -a` — **append** (добавить в конец файла, не перезаписывая).



- `/etc/hosts` — локальный файл, который система использует перед DNS-запросами.

```
sysadmin@Ubuntu-DevOps:~$ echo "1.2.3.4 mytest.local" | sudo tee -a /etc/hosts
[sudo] password for sysadmin:
1.2.3.4 mytest.local
sysadmin@Ubuntu-DevOps:~$ ping mytest.local
PING mytest.local (1.2.3.4) 56(84) bytes of data.
```

Это полезно для :

- Для тестирования сайтов до того, как их DNS-записи настроены.
- Для подмены домена на другой IP (например, локальный сервер).
- Для блокировки доменов (можно отправлять их на `127.0.0.1`).

## 2. `dig` — детальный DNS-запрос

`dig` — это утилита для DNS-запросов. (**Domain Information Groper**)

`dig` отправляет запрос к DNS-серверу и показывает подробный ответ:

- какой IP связан с доменом (A/AAAA записи),
- кто DNS-сервер для домена,
- время жизни записи (TTL),
- полный путь ответа.

```
sysadmin@Ubuntu-DevOps:~$ dig google.com

; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; →>HEADER<← opcode: QUERY, status: NOERROR, id: 19869
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1
```

```
# Нет информации о DNS-серверах домена (AUTHORITY)
```

```
:: OPT PSEUDOSECTION:
```

```
; EDNS: version: 0, flags:; udp: 65494
```

```
:: QUESTION SECTION:
```

```
;google.com.      IN    A
```

```
:: ANSWER SECTION:
```

```
google.com.      1560  IN    A 209.85.203.139
```

```
google.com.      1560  IN    A 209.85.203.138
```

```
google.com.      1560  IN    A 209.85.203.113
```

```
google.com.      1560  IN    A 209.85.203.102
```

```
google.com.      1560  IN    A 209.85.203.100
```

```
google.com.      1560  IN    A 209.85.203.101
```

```
:: Query time: 1 msec
```

```
:: SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
```

```
:: WHEN: Fri Aug 15 18:57:58 UTC 2025
```

```
:: MSG SIZE rcvd: 135
```

```
sysadmin@Ubuntu-DevOps:~$ dig +short google.com
```

```
# только IP (A-записи)
```

```
sysadmin@Ubuntu-DevOps:~$ dig google.com MX
```

```
# MX-записи (почтовые сервера)
```

```
sysadmin@Ubuntu-DevOps:~$ dig google.com NS
```

```
# NS-записи (имя-сервера домена)
```

### 3. **nslookup** — простой DNS-запрос (Name Server Lookup )

- Узнаёт IP по домену (**A** или **AAAA** запись).
- Узнаёт домен по IP (обратный запрос).
- Можно спрашивать конкретный DNS-сервер.

```
sysadmin@Ubuntu-DevOps:~$ nslookup google.com
Server:      127.0.0.53
Address: 127.0.0.53#53
```

Non-authoritative answer:

```
Name: google.com
Address: 209.85.203.139
Name: google.com
Address: 209.85.203.100
Name: google.com
Address: 2a00:1450:400b:c02::8a
Name: google.com
Address: 2a00:1450:400b:c02::8b
```

# nslookup проще и короче, но даёт меньше деталей.

## Practice / Практика

Выполнить команды для просмотра конфигурации сети, проверки соединений и диагностики проблем.

Execute commands to view network configuration, test connections, and diagnose issues.