СИСТЕМНОЕ АДМИНИСТРИРОВАНИЕ С НУЛЯ

ТЕОРИЯ И ПРАКТИКА ДЛЯ НАЧИНАЮЩИХ



Иван Гром

Системное администрирование с нуля

https://litres.ru/74799923
SelfPub; 2026

Аннотация

Эта книга поможет начать путь в системном администрировании с нуля. Вы узнаете, как устроены компьютеры и сети, освоите основы Linux и командной строки, разберётесь в Windows Server и Active Directory, DNS и DHCP, службах, правах доступа и хранении данных. Отдельные главы посвящены резервному копированию и восстановлению, мониторингу, безопасности, виртуализации и автоматизации.

Теория сопровождается примерами команд и конфигураций, пошаговыми лабораторными работами, задачами и разбором типичных неполадок. Практика рассчитана на изолированную виртуальную лабораторию, которую можно развернуть на домашнем компьютере.

Содержание

Для абсолютного новичка: начните отсюда 11
Правда, которую полезно знать сразу 12
Что нужно, чтобы начать 13
Мини-словарь самых базовых слов 14
Главная мысль профессии одной строкой 16
Как читать эту книгу, чтобы не бросить 17
Частые страхи — и правда о них 18
Ваши первые три шага прямо сейчас 19
Куда смотреть, если запутались 20
Как читать эту книгу 21
Кто такой системный администратор 22
Чем занимается сисадмин на практике 23
Сколько учиться и куда расти 23
Главный принцип, который всё объясняет 25
Linux или Windows 26
Как устроена книга 27
Глава 1. Роль системного администратора 28
Что вы на самом деле продаёте 28
Типы администраторов 29
Принципы работы профессионала 30
Инструменты админа 31
Жизнь админа: реактивная и проактивная работа 32

Практика к главе 1 33
Частые ошибки новичка 34
Глава 2. Как устроены компьютер и операционная система 35
Компьютер в четырёх словах 36
Что важно для сервера 36
RAID: защита данных на уровне дисков 38
Что делает операционная система 39
Процессы, службы, память 40
Файловые системы 41
Загрузка системы 42
Виртуализация: несколько серверов на одном 43
Практика к главе 2 44
Частые ошибки новичка 45
Глава 3. Сети: фундамент для администратора 46
Основные понятия на аналогии почты 47
IP-адресация и подсети 48
Модель OSI — без зауми 49
TCP и UDP 50
Статический и динамический адрес 51
Настройка сети в Linux 52
Инструменты диагностики — ежедневный хлеб админа 54
Алгоритм «сеть не работает» 55
Firewall 56
Коммутаторы, маршрутизаторы, VLAN 57

NAT и проброс портов 58
Практика к главе 3 59
Частые ошибки новичка 60
Глава 4. Linux: терминал и файловая система 61
Почему терминал 62
Структура команды 63
Файловая система: всё из корня «/» 64
Навигация и файлы 65
Редактор nano (и пара слов про vim) 66
Поиск и обработка текста 67
Пользователи и права (кратко, подробно — глава 5) 68
Диски и монтирование 69
Процессы 70
Архивы и передача файлов 71
Установка программ 72
История и горячие клавиши 73
Переменные окружения 74
Практика к главе 4 75
Частые ошибки новичка 76
Глава 5. Пользователи, группы и права доступа 77
Пользователи в Linux 78
Управление пользователями 79
Группы 80
Права на файлы 81
Изменение прав и владельца 82

sudo: контролируемые права root 83
Специальные биты (кратко) 84
Права в Windows (кратко, подробно — глава 13) 85
Практика к главе 5 86
Частые ошибки новичка 87
Глава 6. Службы, процессы и systemd 88
Что такое служба 89
systemctl — основной инструмент 90
Чтение логов через journalctl 92
Анатомия unit-файла 93
Таймеры systemd (альтернатива cron) 96
Цели (targets) и уровни загрузки 97
Процессы вне systemd 98
Практика к главе 6 99
Частые ошибки новичка 100
Глава 7. Своя лаборатория: виртуальные машины 101
Что такое виртуализация 102
Что установить 103
Создание VM с Ubuntu Server 104
Сеть виртуальных машин 105
Мини-инфраструктура для практики 106
Подключение по SSH 107
Альтернативы и дополнения 108
Homelab — ваш полигон и портфолио 109

Практика к главе 7 110
Частые ошибки новичка 111
Глава 8. DNS и DHCP: невидимый фундамент сети 112
DHCP: автоматическая раздача адресов 113
Как работает DHCP (процесс DORA) 113
Что раздаёт DHCP 113
Резервирование по MAC 114
Пример конфигурации (isc-dhcp-server) 114
DNS: телефонная книга интернета 116
Как работает разрешение имени 116
Типы записей 116
Прямая и обратная зоны 117
Внутренний DNS-сервер 117
Диагностика DNS 118
DNS и DHCP вместе 120
Практика к главе 8 121
Частые ошибки новичка 122
Глава 9. Веб-серверы и базы данных 123
Веб-серверы: nginx и Apache 124
Установка и запуск 124
Отдача статического сайта 124
Обратный прокси к приложению 125
HTTPS с Let's Encrypt 126
Логи и диагностика 127
Базы данных 128

Установка PostgreSQL 128
MySQL / MariaDB 129
Обязанности админа по базам 129
Репликация (вводно) 130
Связка «веб + база» 131
Практика к главе 9 132
Частые ошибки новичка 133
Глава 10. Файловые службы и общий доступ 134
Протоколы общего доступа 135
Samba: сетевые папки для Windows из Linux 136
NFS: общий доступ между Linux 137
Права на общих папках 138
SFTP: безопасная передача файлов 139
Централизованное хранилище и RAID 140
Резервное копирование файловых хранилищ 141
Практика к главе 10 142
Частые ошибки новичка 143
Глава 11. Почта, удалённый доступ и VPN 144
Почтовые службы: как работает email 145
Компоненты почтовой системы 145
Почему свой почтовый сервер — это сложно 146
Записи DNS для почты 146
Удалённый доступ: SSH (Linux) и RDP (Windows) 148
SSH — основной доступ к Linux-серверам 148

RDP — удалённый рабочий стол Windows 149
VPN: безопасное объединение сетей 150
Виды VPN 150
Популярные решения 150
Пример WireGuard (концептуально) 150
Практика к главе 11 152
Частые ошибки новичка 153
Глава 12. Диски, хранилища и управление данными 154
Устройство дисков в Linux 155
Разделы и файловые системы 156
Постоянное монтирование: /etc/fstab 156
LVM: гибкое управление томами 158
Мониторинг места — ежедневная обязанность 159
Ротация логов 160
SMART: здоровье дисков 161
Снимки и резервные копии 162
Практика к главе 12 163
Частые ошибки новичка 164
Глава 13. Windows Server: основы 165
Чем Windows Server отличается от обычной Windows 166
Роли и компоненты 167
Управление: GUI, Server Manager и PowerShell 168
PowerShell: терминал Windows 169
Управление пользователями (локальными) 171

Сеть в Windows Server 172
Службы и их логи 173
Обновления 174
Активация и лицензии 175
Практика к главе 13 176
Частые ошибки новичка 177
Глава 14. Active Directory: централизованное управление 178
Зачем нужен Active Directory 179
Ключевые понятия 180
Структура: лес → домен → OU → объекты 181
Установка контроллера домена 182
Конец ознакомительного фрагмента. 183

Иван Гром
Системное администрирование с нуля

Для абсолютного новичка: начните отсюда

Хотите стать системным администратором, но не знаете, с какой стороны подступиться? Слова «сервер», «домен», «бэкап», «firewall» звучат серьёзно? Эта короткая глава — для тех, кто начинает с полного нуля. Прочитайте спокойно — дальше будет легче.

Правда, которую полезно знать сразу

• **Сисадмином не рождаются.** Все начинали с «а как открыть эту чёрную штуку с командами». Разница только в часах практики.

• **Это посильнее, чем кажется.** Работа админа — это набор понятных действий: установить, настроить, проверить, починить. Каждое по отдельности несложное.

• **Главное качество админа — не гениальность, а аккуратность.** Делать по одному изменению, проверять, записывать, иметь путь назад. Этому можно научиться.

• **Практика в лаборатории решает всё.** Мы будем ломать и чинить виртуальные машины — их не жалко, они откатываются за секунды.

Что нужно, чтобы начать

1. **Компьютер** с 8 ГБ памяти (лучше 16) — чтобы запускать виртуальные машины.
2. **Интернет** — всё ПО бесплатное.
3. **Журнал администратора** (файл или тетрадь) — записывать, что настраивали и какими командами. Это ваша привычка профессионала с первого дня.
4. **Регулярность** — лучше понемногу каждый день.
5. **Спокойствие** — в этой профессии паника не чинит серверы, а аккуратность чинит.

Мини-словарь самых базовых слов

Не заучивайте — просто прочитайте, чтобы слова перестали пугать.

Сервер — Компьютер, который постоянно работает и что-то «отдаёт» (сайт, файлы, почту)

Терминал / консоль — Окно, где команды печатают текстом, а не кликают мышкой

Linux — Операционная система большинства серверов

Windows Server — Серверная версия Windows (домены, файловые службы)

Служба (сервис) — Программа, работающая в фоне постоянно (веб-сервер, база)

IP-адрес — «Адрес дома» компьютера в сети

Порт — «Номер квартиры»: за одним адресом много служб

DNS — «Телефонная книга»: превращает имя сайта в IP-адрес

DHCP — Служба, которая раздаёт устройствам адреса автоматически

Firewall — «Охранник», пропускающий только разрешённый трафик

Бэкап — Резервная копия данных на случай беды

Виртуальная машина (ВМ) — «Компьютер внутри компьютера» — наш безопасный полигон

Active Directory — Система централизованного управления компьютерами и пользователями в Windows-сети

SSH / RDP — Удалённое подключение к серверу (Linux / Windows)

Мониторинг — Наблюдение за состоянием систем, чтобы заметить проблему заранее

Встретите незнакомое слово дальше — загляните в **большой глоссарий** (Приложение Б).

Главная мысль профессии одной строкой

Плохой админ героически тушит пожары.

Хороший админ делает так, чтобы они не возникали.

Бэкапы, обновления, мониторинг, аккуратные изменения — всё это про предотвращение. Вся книга учит работать так, чтобы «не горело».

Как читать эту книгу, чтобы не бросить

1. **По порядку.** От устройства компьютера и сети — к серверам, службам и доменам. Фундамент сначала.

2. **Всё сразу в лаборатории.** В конце каждой главы — практика на виртуальных машинах. Это обязательно.

3. **Один сквозной проект.** Вы постепенно развернёте «мини-компанию»: сеть, сервер, домен, бэкапы, мониторинг. К концу это ваше портфолио.

4. **Документируйте.** Что настроили и как — в журнал. Это и учёба, и будущая рабочая привычка.

5. **Не бойтесь ломать ВМ.** Для того и снимки (snapshots): откатился — и как новенький.

Частые страхи — и правда о них

• **«Я сломаю рабочий компьютер».** Нет, учимся в виртуальных машинах. Основная система не пострадает.

• **«Нужно всё помнить наизусть».** Нет. Админы постоянно смотрят в документацию и шпаргалки. Понимать важнее, чем помнить.

• **«Терминал — это страшно».** Это просто текстовый способ отдавать команды. Привыкается за несколько дней.

• **«Windows или Linux — что учить?».** Оба: в реальных компаниях есть и то и другое. Книга охватывает обе системы, начиная с общих основ.

Ваши первые три шага прямо сейчас

1. Прочитайте **главы 1–2** — роль админа и устройство компьютера/систем.

2. Установите **VirtualBox** и поднимите виртуальную машину с **Ubuntu Server** (глава 7), сделайте снимок.

3. Освойте базовые команды терминала из главы 4 — фундамент всего.

Куда смотреть, если запутались

• **Шпаргалка команд** (Приложение А) — быстрый справочник (Linux, PowerShell, сеть).

• **Глоссарий** (Приложение Б) — расшифровка терминов.

• **Практикум с решениями** (в конце книги) — задачи с полным разбором по каждой теме. Не понял главу — порешай задачи по ней, и всё встанет на места.

Готовы? Вперёд, в главу 1. Помните: админом становятся не те, кто всё знает, а те, кто аккуратно делает и не бросает.

Как читать эту книгу

Эта книга для того, кто хочет стать системным администратором, но начинает с нуля. Вы научитесь устанавливать и обслуживать серверы, настраивать сети и службы, управлять пользователями, делать резервные копии и защищать систему. Простым языком, с примерами и практикой.

Предварительных знаний не требуется. Нужен компьютер (желательно 8–16 ГБ памяти, чтобы запускать виртуальные машины) и готовность практиковаться руками.

Кто такой системный администратор

Системный администратор (сисадмин) — это человек, который делает так, чтобы компьютеры, серверы, сети и службы в организации работали стабильно и безопасно. Если DevOps-инженер больше про автоматизацию доставки приложений, то сисадмин — это фундамент: он отвечает за «железо», операционные системы, сети, почту, файловые хранилища, учётные записи, резервные копии и безопасность инфраструктуры.

Простая аналогия: сисадмин — это завхоз и инженер здания одновременно. Он следит, чтобы было электричество (серверы работали), вода (сеть), отопление (службы), чтобы у каждого жильца был ключ от своей квартиры (учётные записи и права), чтобы здание не затопило (бэкапы) и чтобы в него не залезли чужие (безопасность).

Чем занимается сисадмин на практике

• Устанавливает и обслуживает **серверы** (Linux и Windows).

• Настраивает **сети**: IP-адресацию, маршрутизацию, firewall, VPN.

• Поднимает и поддерживает **службы**: веб, почту, базы данных, файловые серверы, DNS, DHCP.

• Управляет **пользователями и правами** (в том числе через Active Directory).

• Делает и проверяет **резервные копии**.

• **Мониторит** состояние систем и реагирует на сбои.

• Обеспечивает **безопасность**: обновления, доступы, защита от угроз.

• Автоматизирует рутину **скриптами**.

Сколько учиться и куда расти

Путь с нуля до junior-сисадмина (эникейщик → младший админ) — обычно 4–9 месяцев практики. Дальше можно расти в системного инженера, сетевого инженера, специалиста по виртуализации, в DevOps или в информационную безопасность. Сисадмин — отличная стартовая точка во всей IT-инфраструктуре. Всё ПО в книге бесплатное; практико-

ваться будем на виртуальных машинах.

Главный принцип, который всё объясняет

Запомните одну мысль: **администратор не чинит последствия, а предотвращает проблемы**. Хороший сисадмин незаметен: у него всё работает, бэкапы проверены, обновления установлены, мониторинг предупреждает заранее. Плохой — постоянно «тушит пожары», которые сам же и создал, забыв что-то настроить. Вся книга — про то, как делать правильно и заранее.

Linux или Windows

В реальных организациях есть и то, и другое. Поэтому книга охватывает обе системы:

• **Linux** — большинство серверов, веб, базы данных. Главный фокус.

• **Windows Server** — корпоративная среда, Active Directory, файловые службы, почта Exchange.

Начнём с основ, общих для обеих, затем разберём каждую.

Как устроена книга

Часть I (главы 1–3) — фундамент: роль админа, как устроены компьютер и сеть. Часть II (главы 4–7) — Linux: терминал, пользователи, файловые системы, службы, своя лаборатория. Часть III (главы 8–12) — сети и службы: DNS, DHCP, веб, базы, файловые серверы. Часть IV (главы 13–15) — Windows Server и Active Directory. Часть V (главы 16–22) — эксплуатация: бэкапы, мониторинг (обзор + подробные мануалы по Zabbix и Prometheus/Grafana), безопасность, автоматизация, виртуализация. Часть VI (главы 23–24) — карьера и план обучения. Приложения — шпаргалки, глоссарий, ресурсы.

В каждой главе: простое объяснение на аналогиях, примеры команд с разбором, что вы должны увидеть, практика (обязательна!) и частые ошибки новичка.

:::note Заведите «журнал администратора»: записывайте, что настраивали, какие команды выполняли, что сломалось и как починили. Это и обучение, и будущая документация, и привычка, которая отличает профессионала. Админ без документации — заложник собственной памяти. :::

Глава 1. Роль системного администратора

Что вы на самом деле продаёте

Сисадмин продаёт не «знание команд», а **стабильность и предсказуемость**. Бизнесу неважно, какую команду вы ввели, — важно, что почта работает, файлы доступны, сайт открывается, а если что-то сломается, вы быстро восстановите и это не повторится.

Типы администраторов

Эникейщик / helpdesk — Первая линия: помощь пользователям, принтеры, установка ПО

Системный администратор — Серверы, службы, сети, учётные записи

Сетевой администратор — Фокус на сетях: маршрутизаторы, коммутаторы, firewall, VPN

Администратор баз данных (DBA) — Базы данных: производительность, бэкапы, репликация

Инженер по виртуализации — VMware, Proxmox, Hyper-V, кластеры

DevOps / SRE — Автоматизация, CI/CD, облака (см. отдельную книгу)

Границы размыты, особенно в небольших компаниях, где один человек делает всё.

Принципы работы профессионала

1. **Сначала пойми, потом меняй.** Не трогай то, как работает, пока не понял, зачем оно так.

2. **Одно изменение за раз.** Меняешь десять вещей сразу — не поймёшь, что сломало.

3. **Всегда имей путь назад.** Перед изменением — бэкап конфига, снимок, запасной доступ.

4. **Документируй.** Что, когда, зачем изменил. Через полгода не вспомнишь.

5. **Автоматизируй рутину.** Повторяешь действие третий раз — напиши скрипт.

6. **Минимум привилегий.** Не работай под root/Administrator без нужды.

7. **Проверяй бэкапы восстановлением.** Непроверенный бэкап — это не бэкап.

:::warn Золотое правило изменений на удалённом сервере: всегда имейте запасной способ доступа и путь отката. Классическая катастрофа новичка — поменять сетевую настройку или firewall по SSH и потерять связь с сервером, до которого ехать через полгорода. Перед опасным изменением держите вторую открытую сессию и знайте, как откатиться. :::

Инструменты админа

• **Терминал** (Linux) и **PowerShell** (Windows) — основные рабочие инструменты.

• **SSH / RDP** — удалённый доступ к серверам.

• **Системы мониторинга** (Zabbix, Prometheus) — глаза админа.

• **Системы резервного копирования**.

• **Документация и тикет-системы**.

• **Виртуализация** (VirtualBox, Proxmox, VMware) — для тестов и продакшена.

Жизнь админа: реактивная и проактивная работа

• **Реактивная** — реакция на проблемы: «не работает почта», «забыл пароль». Неизбежна, но её должно быть мало.

• **Проактивная** — предотвращение: обновления, бэкапы, мониторинг, планирование ёмкости. Чем её больше, тем меньше пожаров.

Цель профессионала — сместить баланс в сторону проактивной работы. Новичок тушит пожары; мастер делает так, чтобы они не возникали.

Практика к главе 1

1. Своими словами (5–7 предложений) опишите, чем сисадмин полезен бизнесу и почему «незаметный админ» — это хороший админ.

2. Выпишите 7 принципов профессионала и приведите к каждому пример из бытовой жизни.

3. Найдите 2–3 вакансии «системный администратор junior» и выпишите требования. Это ваша карта на ближайшие месяцы.

4. Заведите журнал администратора (файл или блокнот), с которого начнёте записывать всю практику.

Частые ошибки новичка

• Менять несколько вещей сразу и не понимать, что сломало.

• Работать без бэкапа и пути отката.

• Не документировать — и забывать собственные настройки.

• Постоянно тушить пожары вместо их предотвращения.

• Работать всегда под администратором «чтобы не мешали права».

Глава 2. Как устроены компьютер и операционная система

Нельзя администрировать то, чего не понимаешь. Эта глава даёт фундамент: из чего состоит компьютер и сервер, что делает операционная система и где что хранится.

Компьютер в четырёх словах

• **Процессор (CPU)** — считает. Чем больше ядер и частота, тем больше одновременной работы.

• **Оперативная память (RAM)** — рабочий стол. Всё работающее лежит здесь. Выключили питание — очистилась.

• **Диск (HDD/SSD)** — шкаф для постоянного хранения. SSD быстрее HDD в разы.

• **Операционная система (ОС)** — управляющий, распределяющий ресурсы между программами.

Что важно для сервера

Компонент: CPU (ядра) | На что влияет: Параллельная нагрузка | Совет админу: Больше ядер для многих служб/пользователей

Компонент: RAM | На что влияет: Сколько работает одновременно | Совет админу: Нехватка RAM = тормоза и падения, главный ресурс

Компонент: Диск SSD/NVMe | На что влияет: Скорость работы с данными | Совет админу: Базы и ОС — на SSD

Компонент: RAID | На что влияет: Надёжность дисков | Совет админу: Защита от выхода диска из строя (ниже)

Компонент: Сеть | На что влияет: Пропускная способность | Совет админу: Для файловых/веб-серверов критична

Компонент: Блок питания, охлаждение | На что влияет: Стабильность железа | Совет админу: Резервирование в серверах

RAID: защита данных на уровне дисков

RAID объединяет несколько дисков для надёжности и/или скорости:

Уровень: RAID 0 | Суть: Данные размазаны по дискам | Защита: Нет (один диск умер — всё пропало) | Применение: Только скорость, не для важного

Уровень: RAID 1 | Суть: Зеркало (копия на двух дисках) | Защита: Да (1 диск может умереть) | Применение: Системный диск

Уровень: RAID 5 | Суть: Данные + контроль на 3+ дисках | Защита: Да (1 диск) | Применение: Баланс места и надёжности

Уровень: RAID 6 | Суть: Как 5, но переживает 2 диска | Защита: Да (2 диска) | Применение: Большие массивы

Уровень: RAID 10 | Суть: Зеркало + разбивка | Защита: Да | Применение: Базы данных, высокая нагрузка

:::warn RAID — это НЕ замена резервным копиям. RAID защищает от физической смерти диска, но не от удаления файлов, вируса-шифровальщика, порчи данных или ошибки администратора. Эти случаи RAID честно продублирует на все диски. Нужны и RAID (доступность), и бэкапы (восстановление). Глава 16 — про бэкапы. :::

Что делает операционная система

ОС — посредник между программами и железом. Она:

- распределяет процессорное время между программами;
- управляет памятью;
- даёт доступ к файлам и устройствам;
- разделяет пользователей и их права;
- обеспечивает сеть.

Особый всемогущий пользователь: root в Linux, Administrator/SYSTEM в Windows. Работать под ним постоянно опасно.

Процессы, службы, память

• **Процесс** — работающая программа.

• **Служба (сервис, демон)** — фоновый процесс, работающий постоянно (веб-сервер, база).

• **Swap (подкачка)** — часть диска, используемая, когда кончается RAM. Спасает от падения, но очень медленная; постоянное использование swap — сигнал, что не хватает памяти.

Файловые системы

Файловая система определяет, как данные хранятся на диске:

ФС: ext4 | Где: Linux (по умолчанию) | Особенность: Надёжная, проверенная

ФС: XFS | Где: Linux (серверы, большие данные) | Особенность: Хороша для крупных файлов

ФС: ZFS / Btrfs | Где: Linux/BSD (продвинуто) | Особенность: Снимки, контроль целостности

ФС: NTFS | Где: Windows | Особенность: Стандарт Windows

ФС: FAT32 / exFAT | Где: Флешки, совместимость | Особенность: Простая, ограничения на размер файла

Загрузка системы

Понимание порядка загрузки помогает чинить «не грузится»:

Питание → BIOS/UEFI (проверка железа) → загрузчик (GRUB) →

ядро ОС → системные службы (systemd) → вход пользователя

Если сервер «не грузится», проблема локализуется по этой цепочке: железо? загрузчик? ядро? служба?

Виртуализация: несколько серверов на одном

Один физический сервер делят на много виртуальных машин (VM) с помощью гипервизора (VMware, Proxmox, Hyper-V, KVM). Плюсы: эффективное использование железа, изоляция, снимки, быстрое развёртывание. Почти вся современная инфраструктура виртуализирована. Подробнее — в главе 20.

Практика к главе 2

1. Посмотрите характеристики своего компьютера: CPU (ядра), RAM, тип диска. В Linux: lscpu, free -h, lsblk. В Windows: «Сведения о системе».

2. Своими словами объясните разницу между RAM и диском и почему нехватка RAM критична.

3. Составьте таблицу уровней RAID и для каждого запишите, от чего он защищает, а от чего — нет.

4. Объясните, почему RAID не заменяет бэкап. Приведите 3 случая, когда RAID не спасёт.

5. Нарисуйте цепочку загрузки и подпишите, что проверять на каждом этапе при сбое.

Частые ошибки новичка

• Считать RAID заменой резервным копиям.

• Игнорировать нехватку RAM (постоянный swap = тормоза).

• Ставить базы и ОС на медленные HDD вместо SSD.

• Не понимать порядок загрузки и теряться при сбое «не грузится».

Глава 3. Сети: фундамент для администратора

Сеть — это кровеносная система инфраструктуры. Сисадмин обязан понимать, как данные доходят от одного устройства к другому, уметь настраивать адресацию и диагностировать проблемы. Эта глава даёт практический минимум.

Основные понятия на аналогии почты

• **IP-адрес —** адрес дома (192.168.1.10).

• **Порт —** номер квартиры (веб — 80/443, SSH — 22, RDP — 3389).

• **MAC-адрес —** «серийный номер» сетевой карты, уникален в пределах локальной сети.

• **Шлюз (gateway) —** почтовое отделение, через которое пакеты уходят в другие сети.

• **DNS —** телефонная книга: имя → IP.

• **Маска подсети —** определяет, какие адреса «соседи» в одной сети.

IP-адресация и подсети

Адрес IPv4 — четыре числа 0–255 (192.168.1.10). Маска определяет границу сети:

IP: 192.168.1.10

Маска: 255.255.255.0 (или /24)

Сеть: 192.168.1.0 — соседи: 192.168.1.1 ... 192.168.1.254

Шлюз: 192.168.1.1 — обычно роутер

Запись /24 (CIDR) = маска 255.255.255.0 = 254 полезных адреса. Частные диапазоны (не маршрутизируются в интернет):

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16
- 127.0.0.1 — localhost.

CIDR: /24 | Маска: 255.255.255.0 | Адресов: 256 (254 полезных)

CIDR: /25 | Маска: 255.255.255.128 | Адресов: 128

CIDR: /16 | Маска: 255.255.0.0 | Адресов: 65 536

Модель OSI — без зауми

Данные упаковываются слоями. Админу достаточно понимать:

Уровень: L2 канальный | За что отвечает: Внутри одной сети по MAC | Примеры: Ethernet, Wi-Fi, коммутаторы | Где ломается: кабель, порт, VLAN

Уровень: L3 сетевой | За что отвечает: Между сетями по IP | Примеры: IP, маршрутизаторы | Где ломается: адресация, маршруты

Уровень: L4 транспортный | За что отвечает: Доставка приложению по порту | Примеры: TCP, UDP | Где ломается: порт закрыт, firewall

Уровень: L7 прикладной | За что отвечает: Смысл данных | Примеры: HTTP, DNS, SMTP | Где ломается: конфиг службы

Диагностика «снизу вверх» по этим уровням — метод решения большинства сетевых проблем.

TCP и UDP

• **TCP** — с установлением соединения и гарантией (веб, почта, SSH, базы).

• **UDP** — без гарантий, быстро (DNS, видео, VoIP, мониторинг).

Статический и динамический адрес

• **Статический** — задаётся вручную, не меняется. Для серверов обязателен (сервер должен быть по постоянному адресу).

• **Динамический (DHCP)** — выдаётся автоматически. Для рабочих станций и устройств.

Настройка сети в Linux

Современный Ubuntu использует netplan:
/etc/netplan/01-net.yaml

network:

version: 2

ethernets:

eth0:

addresses: [192.168.1.10/24]

routes:

- to: default

via: 192.168.1.1

nameservers:

addresses: [8.8.8.8, 1.1.1.1]
sudo netplan apply

ip a проверить адрес

ip route маршруты

Инструменты диагностики — ежедневный хлеб админа

ip a мои адреса

ip route таблица маршрутов

ping -c 4 8.8.8.8 доступность хоста

ping -c 4 ya.ru + проверка DNS

traceroute ya.ru путь пакета по узлам

dig ya.ru +short разрешение имени

nslookup ya.ru то же проще

ss -tulpn какие порты слушает машина

nc -zv host 443 открыт ли порт

curl -I http://site ответ веб-сервера

mtr ya.ru ping + traceroute в реальном времени

Алгоритм «сеть не работает»

1. **Есть ли адрес?** ip a — получили IP?
2. **Есть ли связь со шлюзом?** ping шлюз
3. **Есть ли выход наружу?** ping 8.8.8.8 (по IP, без DNS)
4. **Работает ли DNS?** ping ya.ru / dig ya.ru
5. **Открыт ли нужный порт?** nc -zv host port
6. **Отвечает ли служба?** curl, логи службы

:::note Классический приём: если ping 8.8.8.8 работает, а ping ya.ru — нет, значит сеть в порядке, а проблема в DNS. Это разделение «IP работает / имена не работают» мгновенно сужает поиск. Запомните его — сэкономит часы. :::

Firewall

Firewall пропускает только разрешённый трафик. В Linux — ufw (простой) или nftables/iptables:

sudo ufw default deny incoming запретить всё входящее

sudo ufw default allow outgoing разрешить исходящее

sudo ufw allow 22/tcp SSH

sudo ufw allow 80,443/tcp веб

sudo ufw enable

sudo ufw status numbered

Принцип: закрыто всё, открыто только нужное.

Коммутаторы, маршрутизаторы, VLAN

• **Коммутатор (switch)** — соединяет устройства в одной сети (L2).

• **Маршрутизатор (router)** — соединяет разные сети (L3), знает маршруты.

• **VLAN** — логическое разделение одной физической сети на изолированные (бухгалтерия, гости, серверы — отдельно). Основа сетевой безопасности.

NAT и проброс портов

NAT позволяет многим устройствам с частными адресами выходить в интернет через один публичный. Проброс портов (port forwarding) открывает доступ снаружи к конкретной внутренней службе.

Практика к главе 3

1. Определите на своей машине: IP, маску, шлюз, DNS. Объясните, кто есть кто.

2. Посчитайте вручную: сколько адресов в сети /24 и /25, какие «соседи» у 192.168.5.10/24.

3. Настройте статический IP на VM через netplan и проверьте ip a / ip route.

4. Пройдите алгоритм диагностики: отключите DNS (уберите nameservers) и убедитесь, что ping 8.8.8.8 работает, а ping ya.ru — нет. Верните DNS.

5. Настройте ufw: запретить всё входящее, кроме SSH и веб.

Частые ошибки новичка

• Давать серверу динамический адрес (DHCP) вместо статического.

• Путать проблему DNS с проблемой сети (используйте ping по IP).

• Включать firewall, забыв разрешить SSH, и терять доступ.

• Держать всё в одной плоской сети без VLAN.

• Не знать свой шлюз и маску — и не понимать, почему «соседи» недоступны.

Глава 4. Linux: терминал и файловая система

Большинство серверов — это Linux без графики, управляемый из терминала. Эта глава научит уверенно в нём работать. Освоив её, вы перестанете бояться «чёрного экрана».

Почему терминал

На сервере нет мышки и рабочего стола — только текстовые команды, часто по сети (SSH). Плюс терминала: любое действие можно записать в скрипт и повторить. Это основа автоматизации.

Структура команды

команда -флаги аргументы

ls -la /var/log

Файловая система: всё из корня «/»

/ — Корень
/etc — Настройки системы и служб
/var/log — Логи
/var/www — Сайты
/home/имя — Домашние папки пользователей
/root — Домашняя папка root
/tmp — Временные файлы
/usr/bin, /bin — Программы
/opt, /srv — Сторонние приложения и данные служб
/dev — Устройства (диски и т. д.)
/mnt, /media — Точки монтирования
/proc, /sys — Виртуальная информация о системе

Навигация и файлы

pwd где я

ls -la список, включая скрытые (начинаются с точки)

cd /etc/ssh / cd .. / cd ~

cat file показать файл

less file листать (q — выход)

head -n 20 / tail -n 20

tail -f /var/log/syslog следить за логом в реальном времени

cp / mv / rm / mkdir -p / touch

nano file простой редактор

Редактор nano (и пара слов про vim)

Для новичка nano проще всего: Ctrl+O сохранить, Ctrl+X выйти, Ctrl+W поиск. На многих серверах есть только vim: i — ввод текста, Esc — выход из ввода, :wq — сохранить и выйти, :q! — выйти без сохранения. Знать базовый выход из vim обязательно — иначе застрянете.

Поиск и обработка текста

find /etc -name "*.conf" поиск файлов по имени

find / -size +100M 2>/dev/null большие файлы

grep -r "error" /var/log/ поиск строки в файлах

grep -i "fail" auth.log без учёта регистра

cat access.log | grep 500 | wc -l сколько строк с 500

awk '{print $1}' file столбец

sed 's/old/new/g' file замена

Символ | (пайп) соединяет команды в цепочку — основа работы в терминале.

Пользователи и права (кратко, подробно — глава 5)

whoami кто я

sudo команда выполнить как root

chmod 640 file права числом

chown user:group file владелец

Права читаются rwxr-x---: владелец (rwx), группа (r-x), остальные (---). Числом: r=4, w=2, x=1.

Диски и монтирование

lsblk список дисков и разделов

df -h занятое место по файловым системам

du -sh /var/* размер папок

mount /dev/sdb1 /mnt подключить раздел

umount /mnt отключить

Постоянное монтирование настраивается в /etc/fstab.

:::warn Переполнение диска — одна из самых частых причин сбоев сервера. Регулярно проверяйте df -h. Когда диск забит на 100%, перестают работать службы, не пишутся логи, падают базы. Настройте мониторинг свободного места (глава 17) и чистите/ротируйте логи. :::

Процессы

ps aux все процессы

ps aux | grep nginx найти процесс

top / htop в реальном времени

kill PID завершить (мягко)

kill -9 PID принудительно

systemctl ... управление службами (глава 6)

Архивы и передача файлов

tar -czf arch.tar.gz /path создать архив gzip

tar -xzf arch.tar.gz распаковать

scp file user@host:/path копировать по SSH

rsync -av src/ user@host:/dst/ умная синхронизация

Установка программ

Debian/Ubuntu

sudo apt update && sudo apt install имя

RHEL/Rocky/Fedora

sudo dnf install имя

История и горячие клавиши

• **Tab —** автодополнение (жмите постоянно).

• **Стрелка вверх / Ctrl+R —** прошлые команды / поиск по истории.

• **Ctrl+C —** прервать. **Ctrl+L —** очистить экран.

• history — все команды.

Переменные окружения

echo $PATH где система ищет программы

export VAR=значение задать переменную

env все переменные

Практика к главе 4

1. Поставьте Ubuntu Server в VM (глава 7). Создайте папки и файлы, потренируйте навигацию.

2. Научитесь выходить из vim (:q!) — пригодится на чужих серверах.

3. Одной строкой с пайпами посчитайте число строк с «error» в любом логе.

4. Посмотрите df -h и du -sh /var/* — найдите, что занимает место.

5. Создайте архив папки через tar и распакуйте его в другое место.

6. Найдите через find все .conf файлы в /etc.

Частые ошибки новичка

- Бояться терминала.
- rm -rf без раздумий — в Linux нет корзины.
- Не следить за местом на диске (df -h).
- Застрять в vim, не зная, как выйти.
- Всё делать под root вместо sudo по необходимости.

Глава 5. Пользователи, группы и права доступа

Управление доступом — одна из главных обязанностей сисадмина. Кто может читать файл, запускать службу, входить на сервер — всё это настраивается через пользователей, группы и права. Ошибки здесь ведут либо к дырам в безопасности, либо к тому, что «ничего не работает».

Пользователи в Linux

Каждый пользователь имеет имя, числовой UID, домашнюю папку и оболочку. Информация хранится в /etc/passwd:

root:x:0:0:root:/root:/bin/bash

www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin

ivan:x:1000:1000:Ivan:/home/ivan:/bin/bash

Поля: имя, пароль (x = в другом файле), UID, GID, описание, домашняя папка, оболочка. **UID 0 = root** (полная власть). Служебные учётки (www-data) часто имеют оболочку /usr/sbin/nologin — под ними нельзя войти, и это правильно.

Пароли (в виде хешей) хранятся в /etc/shadow, доступном только root.

Управление пользователями

sudo adduser ivan создать пользователя (интерактивно)

sudo passwd ivan задать/сменить пароль

sudo usermod -aG sudo ivan добавить в группу sudo

sudo usermod -aG docker ivan добавить в группу docker

sudo deluser ivan удалить

sudo deluser --remove-home ivan удалить с домашней папкой

id ivan UID, GID, группы пользователя

su - ivan переключиться на пользователя

:::note Флаг -a в usermod -aG критичен: он ДОБАВЛЯЕТ в группу. Без -a (usermod -G) вы ЗАМЕНИТЕ все группы пользователя на указанную и выкинете его из остальных. Классическая ошибка новичка, из-за которой пользователь внезапно теряет доступы. Всегда -aG. :::

Группы

Группы объединяют пользователей с общими правами. Вместо раздачи прав каждому — дают группе.

sudo groupadd developers создать группу

sudo usermod -aG developers ivan добавить пользователя

groups ivan в каких группах пользователь

getent group developers кто в группе

Права на файлы

$ ls -l

-rw-r--r-- 1 ivan devs 1024 Oct 23 notes.txt

drwxr-x--- 2 ivan devs 4096 Oct 23 private

Разбор -rw-r--r--:

• 1-й символ: тип (- файл, d папка, l ссылка).

• Далее тройки: владелец (rw-), группа (r--), остальные (r--).

• r=чтение(4), w=запись(2), x=выполнение(1).

Числовая запись:

Права: rw-r--r-- | Число: 644 | Смысл: Файлы: владелец пишет, остальные читают

Права: rwxr-xr-x | Число: 755 | Смысл: Программы/папки: все выполняют/входят

Права: rw------- | Число: 600 | Смысл: Приватное: только владелец (ключи, секреты)

Права: rwxrwxrwx | Число: 777 | Смысл: Все всё (ОПАСНО, почти никогда не нужно)

Изменение прав и владельца

chmod 644 file права числом

chmod u+x script.sh добавить выполнение владельцу

chmod -R 755 /var/www рекурсивно для папки

chown ivan file сменить владельца

chown ivan:devs file владелец и группа

chown -R www-data:www-data /var/www рекурсивно

:::warn Никогда не ставьте chmod 777 «чтобы заработало». Это открывает файл на запись всем и является дырой в безопасности. Если что-то не работает из-за прав, разберитесь, какому пользователю нужен доступ, и дайте именно ему через владельца/группу. 777 — почти всегда признак того, что проблему «замяли», а не решили. :::

sudo: контролируемые права root

sudo позволяет обычному пользователю выполнять команды от root. Настройки — в /etc/sudoers (редактировать только через visudo!):

sudo visudo

примеры правил:

ivan ALL=(ALL) ALL ivan может всё через sudo

deploy ALL=(ALL) NOPASSWD: /bin/systemctl restart myapp без пароля, только эту команду

%developers ALL=(ALL) /usr/bin/apt группа developers может apt

Принцип минимальных привилегий: давайте через sudo только то, что реально нужно, а не «всё».

Специальные биты (кратко)

• **SUID** (s вместо x у владельца) — программа запускается с правами владельца файла. Нужен некоторым системным программам (passwd), но лишний SUID — риск безопасности.

• **Sticky bit** (t) — в папке вроде /tmp: удалять файлы может только их владелец, хотя писать могут все.

Права в Windows (кратко, подробно — глава 13)

В Windows права сложнее: ACL (списки контроля доступа) с детальными разрешениями (чтение, запись, изменение, полный доступ) для пользователей и групп. Управляются через свойства файла или icacls в командной строке. Концепция та же: кто и что может с объектом.

Практика к главе 5

1. Создайте пользователя, задайте пароль, добавьте его в группу sudo (через -aG!).

2. Создайте группу devs, добавьте двух пользователей, создайте папку с группой-владельцем devs и правами 770.

3. Поэкспериментируйте: войдите под пользователем не из группы и убедитесь, что доступа нет.

4. Через visudo дайте пользователю право на одну конкретную команду без пароля.

5. Разберите права 5 файлов в /etc и переведите их в числовую запись и обратно.

Частые ошибки новичка

• usermod -G без -a — выкидывает пользователя из других групп.

• chmod 777 вместо настройки правильного владельца.

• Редактировать /etc/sudoers напрямую, а не через visudo (ошибка может заблокировать sudo).

• Давать всем пользователям полный sudo вместо минимальных прав.

• Работать под root вместо обычного пользователя с sudo.

Глава 6. Службы, процессы и systemd

Сервер — это набор работающих служб: веб-сервер, база, почта, DNS. Управление службами — ежедневная работа админа. В современном Linux за это отвечает systemd.

Что такое служба

Служба (сервис, демон) — программа, работающая в фоне постоянно, обычно запускаемая при старте системы. Примеры: nginx, ssh, postgresql, cron. systemd их запускает, останавливает, перезапускает при падении и ведёт их логи.

systemctl — основной инструмент

systemctl status nginx состояние службы (жива? когда запущена?)

systemctl start nginx запустить

systemctl stop nginx остановить

systemctl restart nginx перезапустить

systemctl reload nginx перечитать конфиг без разрыва (если поддерживается)

systemctl enable nginx автозапуск при загрузке

systemctl disable nginx убрать из автозапуска

systemctl enable --now nginx включить автозапуск И запустить сразу

systemctl is-active nginx активна ли

systemctl list-units --type=service все службы

:::note Различайте start и enable. start запускает службу

сейчас, но после перезагрузки она не поднимется. enable добавляет в **автозапуск**, но не запускает сейчас. Классическая ошибка: настроили службу, start, всё работает — перезагрузили сервер, а служба не поднялась, потому что забыли enable. Привыкайте к enable --now. :::

Чтение логов через journalctl

systemd собирает логи всех служб в единый журнал:
journalctl -u nginx логи службы nginx

journalctl -u nginx -f следить в реальном времени

journalctl -u nginx --since "1 hour ago"

journalctl -u nginx -n 50 последние 50 строк

journalctl -p err только ошибки

journalctl -b логи с текущей загрузки

journalctl --disk-usage сколько места занимает журнал
«Служба не работает» → systemctl status (увидеть статус и последние строки) → journalctl -u служба (подробности). Это первый рефлекс админа.

Анатомия unit-файла

Службы описаны в unit-файлах. Системные — в /lib/systemd/system/, свои — в /etc/systemd/system/. Создадим службу для своего приложения:

```
# /etc/systemd/system/myapp.service

[Unit]

Description=Моё приложение

After=network.target

Requires=postgresql.service


[Service]

Type=simple

User=appuser

WorkingDirectory=/opt/myapp

ExecStart=/opt/myapp/start.sh
```

Restart=always

RestartSec=5

EnvironmentFile=/opt/myapp/.env

[Install]

WantedBy=multi-user.target

Разбор:

• [Unit] — описание и зависимости (After — запускать после сети; Requires — нужна база).

• [Service] — как запускать: от какого пользователя, какая команда (ExecStart), перезапуск при падении (Restart=always).

• [Install] — когда включать при автозагрузке.

Применение:

sudo systemctl daemon-reload перечитать unit-файлы после изменений

sudo systemctl enable --now myapp

systemctl status myapp

Restart=always — важная строка: если приложение упадёт,

systemd поднимет его снова. Проверьте: убейте процесс и посмотрите, что служба перезапустилась.

Таймеры systemd (альтернатива cron)

systemd умеет запускать задачи по расписанию через таймеры:

```
# myapp-backup.timer

[Timer]

OnCalendar=*-*-* 03:00:00

Persistent=true


[Install]

WantedBy=timers.target
```

Таймер связан с одноимённым .service. Преимущество перед cron — единое логирование через journalctl и зависимости.

Цели (targets) и уровни загрузки

systemctl get-default текущая цель загрузки

systemctl set-default multi-user.target без графики (серверы)

systemctl isolate rescue.target режим восстановления

multi-user.target — обычный серверный режим без графики; graphical.target — с рабочим столом.

Процессы вне systemd

ps aux | grep имя найти процесс

top / htop нагрузка в реальном времени

kill PID / kill -9 PID завершить / принудительно

nohup команда & запустить, не завершится при выходе

Но для постоянных служб всегда предпочитайте systemd, а не «запуск руками в фоне» — systemd перезапустит, залогирует и поднимет при загрузке.

Практика к главе 6

1. Управляйте службой nginx: status, stop, start, reload. Посмотрите разницу в systemctl status.

2. Проверьте разницу start/enable: запустите службу, перезагрузите VM, убедитесь, что без enable она не поднялась.

3. Напишите свою службу (простой скрипт-«приложение») с Restart=always. Убейте процесс и убедитесь, что systemd его поднял.

4. Почитайте логи своей службы через journalctl -u.

5. Создайте таймер systemd, запускающий бэкап-скрипт каждую ночь.

Частые ошибки новичка

• start без enable — служба не поднимается после перезагрузки.

• Забыть daemon-reload после изменения unit-файла.

• Запускать приложения «руками» в фоне вместо службы systemd.

• Не смотреть journalctl при проблемах со службой.

• Ставить Restart=always приложению, которое падает из-за ошибки конфига, — получить бесконечный цикл перезапусков (смотрите логи!).

Глава 7. Своя лаборатория: виртуальные машины

Администрированию нельзя научиться по книгам — только на реальных системах. Но экспериментировать на рабочих серверах нельзя. Решение — виртуальная лаборатория на своём компьютере: бесплатно, безопасно, с «откатом» через снимки.

Что такое виртуализация

Виртуальная машина (VM) — компьютер внутри вашего компьютера: своя ОС, диск, сеть, работает в окне. Гипервизор (программа виртуализации) делит ресурсы вашего ПК между VM. Для лаборатории это идеально: сломали VM — откатили снимок за секунды, а основная система цела.

Что установить

Компонент: VirtualBox (бесплатно) | Роль: Гипервизор для вашего ПК | Где взять: virtualbox.org

Компонент: Ubuntu Server, актуальный поддерживаемый LTS-выпуск | Роль: Основной учебный Linux-сервер | Где взять: ubuntu.com

Компонент: Rocky Linux / AlmaLinux | Роль: «RHEL-семейство» (dnf, другой опыт) | Где взять: rockylinux.org

Компонент: Windows Server (Evaluation) | Роль: Для практики с AD (главы 13–14) | Где взять: microsoft.com Evaluation Center

Для начала хватит VirtualBox + пары Ubuntu Server.

Создание VM с Ubuntu Server

1. Скачайте ISO-образ Ubuntu Server.
2. В VirtualBox: Создать → имя, тип Linux/Ubuntu 64-bit.
3. Память: 2048 МБ (для сервера без графики достаточно).
4. Диск: 20–25 ГБ динамический.
5. Подключите ISO, запустите, установите Ubuntu.
6. При установке **отметьте «Install OpenSSH server»** — чтобы подключаться по SSH.
7. После установки обновите систему: sudo apt update && sudo apt upgrade -y.
8. Сделайте **снимок (snapshot)** «чистая система».

:::note Снимки (snapshots) — ваша страховка. Перед любым рискованным экспериментом делайте снимок. Что-то сломали безнадёжно — откат за секунды, без переустановки. Это роскошь лаборатории, которой нет на реальных серверах, — пользуйтесь ей для смелых экспериментов. :::

Сеть виртуальных машин

VirtualBox даёт разные режимы сети:

Режим: NAT | Что даёт: VM имеет интернет, но её не видно извне | Когда: Простой доступ в интернет

Режим: Сетевой мост (Bridged) | Что даёт: VM — полноценный член вашей домашней сети | Когда: Чтобы видеть VM с основного ПК

Режим: Внутренняя сеть | Что даёт: VM видят только друг друга | Когда: Изолированная лаборатория

Режим: Host-only | Что даёт: VM видят друг друга и хост | Когда: Лаборатория + доступ с ПК

Для мини-инфраструктуры из нескольких серверов удобно: **Host-only** (вы подключаетесь с основного ПК) + при необходимости NAT (для интернета на VM).

Мини-инфраструктура для практики

Создайте несколько VM — это даст реалистичную среду:
srv-web — веб-сервер (nginx, приложение)

srv-db — база данных (PostgreSQL/MySQL)

srv-dns — DNS/DHCP (глава 8)

win-dc — Windows Server, контроллер домена (главы 13-14)

На слабой машине поднимайте по 1–2 VM за раз. Клонирование VirtualBox ускоряет создание похожих серверов.

Подключение по SSH

После установки узнайте адрес VM (ip a) и подключайтесь с основного компьютера:

ssh ivan@192.168.56.10

Настройте вход по ключу (удобнее и безопаснее):

ssh-keygen -t ed25519

ssh-copy-id ivan@192.168.56.10

ssh ivan@192.168.56.10 теперь без пароля

Альтернативы и дополнения

• **WSL2** (Windows) — быстрый Linux без VirtualBox, для изучения команд. Но без «настоящей» серверной сети.

• **Proxmox VE** — бесплатный гипервизор серверного класса; если есть отдельный старый компьютер, поставьте Proxmox и получите настоящий «домашний сервер» (homelab). Отличная практика и строчка в резюме.

• **Облачная VM** — арендуйте на пробный период реальный публичный сервер.

Homelab — ваш полигон и портфолио

Домашняя лаборатория (homelab) — лучший способ учиться и показать навыки работодателю. Старый ноутбук или мини-ПК с Proxmox, пара VM, свои службы — и вы практикуете всё из этой книги на «почти настоящей» инфраструктуре. Документируйте homelab в заметках — это готовое портфолио.

Практика к главе 7

1. Установите VirtualBox и поднимите VM с Ubuntu Server, отметив установку OpenSSH.

2. Сделайте снимок «чистой системы».

3. Настройте сеть так, чтобы подключаться к VM с основного ПК по SSH.

4. Настройте вход по SSH-ключу.

5. Склонируйте VM, чтобы получить вторую, и убедитесь, что они видят друг друга (ping).

6. Нарочно «сломайте» что-нибудь и откатите снимок — прочувствуйте силу снимков.

Частые ошибки новичка

• Не делать снимки и переустанавливать всё после первой поломки.

• Выделять VM слишком мало памяти (сервер тормозит).

• Запутаться в режимах сети и не понимать, почему VM не видят друг друга.

• Экспериментировать на рабочих системах вместо лаборатории.

• Не ставить OpenSSH при установке и мучиться без удалённого доступа.

Глава 8. DNS и DHCP: невидимый фундамент сети

Две службы, без которых сеть превращается в хаос. DHCP раздаёт устройствам адреса, DNS превращает имена в адреса. Админ обязан понимать и уметь их настраивать — и знать, что чинить, когда «интернет есть, а сайты не открываются».

DHCP: автоматическая раздача адресов

Представьте, что каждому новому сотруднику вы вручную выдаёте адрес, маску, шлюз и DNS. С сотней устройств это ад. DHCP делает это автоматически: устройство включается, просит настройки, получает их на время (lease).

Как работает DHCP (процесс DORA)

Устройство → DISCOVER "есть тут DHCP-сервер?"

Сервер → OFFER "вот тебе адрес 192.168.1.50"

Устройство → REQUEST "беру его"

Сервер → ACK "закреплено за тобой на N часов"

Что раздаёт DHCP

- IP-адрес и маску;
- шлюз по умолчанию;
- адреса DNS-серверов;
- иногда: NTP (время), домен, параметры загрузки по сети.

Резервирование по MAC

Важным устройствам (принтер, сервер) закрепляют постоянный адрес по MAC-адресу — удобство DHCP при предсказуемости статики.

:::note Серверам обычно дают **статический** адрес (или DHCP-резервирование), а не обычный динамический. Сервер должен быть всегда по одному адресу, иначе на него не настроишь ссылки, DNS-записи и доступы. Динамический DHCP — для рабочих станций, телефонов, гостей. :::

Пример конфигурации (isc-dhcp-server)

```
subnet 192.168.1.0 netmask 255.255.255.0 {

range 192.168.1.100 192.168.1.200; пул для выдачи

option routers 192.168.1.1; шлюз

option domain-name-servers 192.168.1.2; DNS

default-lease-time 3600;

}
```

```
host printer {

hardware ethernet 00:11:22:33:44:55;

fixed-address 192.168.1.50; резервирование

}
```

DNS: телефонная книга интернета

DNS превращает имена (server.company.local) в IP-адреса. Без DNS пришлось бы помнить цифры всех серверов.

Как работает разрешение имени

Клиент спрашивает резолвер: "какой IP у ya.ru?"

→ резолвер проверяет кэш

→ если нет: спрашивает корневые → зону .ru → сервер домена

→ получает IP, отдаёт клиенту и кэширует на TTL

Типы записей

Тип: A | Значение: Имя → IPv4 | Пример: server → 192.168.1.10

Тип: AAAA | Значение: Имя → IPv6 | Пример: server → ::1

Тип: CNAME | Значение: Псевдоним | Пример: www → server.company.local

Тип: MX | Значение: Почтовый сервер | Пример: → mail.company.local

Тип: PTR | Значение: IP → имя (обратная зона) | Пример: 192.168.1.10 → server

Тип: NS | Значение: DNS-серверы зоны | Пример:

Тип: TXT | Значение: Текст (SPF, DKIM, проверки) | Пример:

Тип: SRV | Значение: Расположение служб (важно для AD) | Пример:

Прямая и обратная зоны

• **Прямая зона**: имя → IP (основное использование).

• **Обратная зона**: IP → имя (PTR-записи). Нужна для почты, логов, диагностики.

Внутренний DNS-сервер

В организации поднимают свой DNS (BIND в Linux, DNS-роль в Windows Server), чтобы серверы и станции обращались друг к другу по именам, а не по адресам.

пример зоны BIND

$TTL 3600

@ IN SOA ns1.company.local. admin.company.local. (

2026102301 ; serial (увеличивать при каждом изменении!)

3600 1800 1209600 3600)

@ IN NS ns1.company.local.

ns1 IN A 192.168.1.2

web IN A 192.168.1.10

db IN A 192.168.1.20

www IN CNAME web

:::warn Serial-номер зоны (2026102301) нужно увеличивать при каждом изменении, иначе вторичные DNS-серверы не подхватят обновление. И помните про TTL: изменённая запись «живёт» в кэшах до истечения TTL. Это главная причина «я поменял DNS, а не работает» — подождите TTL или понизьте его заранее. :::

Диагностика DNS

dig ya.ru +short получить IP

dig @192.168.1.2 web.company.local спросить конкретный сервер

dig -x 192.168.1.10 обратный запрос (PTR)

nslookup web.company.local проще

systemd-resolve --status настройки DNS в системе

Классический приём: ping 8.8.8.8 работает, ping ya.ru — нет → проблема в DNS, а не в сети.

DNS и DHCP вместе

DHCP раздаёт станциям адрес DNS-сервера. DNS позволяет обращаться к серверам по именам. Часто в небольшой сети оба сервиса на одной машине (или на контроллере домена в Windows-среде, главы 13–14).

Практика к главе 8

1. Поднимите DHCP-сервер в лаборатории, настройте пул и резервирование по MAC. Проверьте, что клиент получает адрес.

2. Поднимите DNS-сервер (BIND или dnsmasq), создайте прямую зону с несколькими A-записями.

3. Проверьте разрешение имён через dig @ваш-сервер.

4. Создайте CNAME (псевдоним) и убедитесь, что он работает.

5. Смоделируйте проблему DNS: укажите неверный DNS и пройдите диагностику (ping по IP работает, по имени — нет).

Частые ошибки новичка

• Забыть увеличить serial зоны — изменения не распространяются.

• Не учитывать TTL при смене записей.

• Давать серверам динамические адреса DHCP.

• Не настроить обратную зону (ломается почта и диагностика).

• Указать станциям внешний DNS вместо внутреннего и потерять разрешение внутренних имён.

Глава 9. Веб-серверы и базы данных

Два самых частых типа служб, которые обслуживает админ: веб-серверы (отдают сайты и приложения) и базы данных (хранят данные). Эта глава — практический минимум по обоим.

Веб-серверы: nginx и Apache

Веб-сервер принимает HTTP-запросы и отдаёт страницы. Два главных:

• **nginx** — лёгкий, быстрый, отлично как обратный прокси и для статики. Современный стандарт.

• **Apache** — классика, гибкий, много модулей.

Разберём nginx как основной.

Установка и запуск

sudo apt install nginx

sudo systemctl enable --now nginx

curl localhost проверить

Конфиги: /etc/nginx/nginx.conf (главный), /etc/nginx/sites-available/ (сайты), /etc/nginx/sites-enabled/ (включённые).

Отдача статического сайта

server {

listen 80;

server_name site.company.local;

root /var/www/site;

index index.html;

location / {

try_files $uri $uri/ =404;

}

}

sudo ln -s /etc/nginx/sites-available/site /etc/nginx/sites-enabled/

sudo nginx -t ВСЕГДА проверять конфиг перед применением

sudo systemctl reload nginx

Обратный прокси к приложению

Приложение (на порту 8080) прячут за nginx:

server {

listen 80;

server_name app.company.local;

location / {

proxy_pass http://127.0.0.1:8080;

proxy_set_header Host $host;

proxy_set_header X-Real-IP $remote_addr;

}

}

:::note sudo nginx -t перед каждым reload — железное правило. Он ловит опечатки в конфиге до применения. reload применяет без разрыва соединений, restart — с кратким простоем. Опечатка + restart = упавший сайт. :::

HTTPS с Let's Encrypt

sudo apt install certbot python3-certbot-nginx

sudo certbot --nginx -d site.company.ru

Certbot получит бесплатный сертификат и настроит авто-

продление. Проверка: sudo certbot renew --dry-run.

Логи и диагностика

tail -f /var/log/nginx/access.log кто что запрашивает

tail -f /var/log/nginx/error.log ошибки (сюда при 502/504)

502/504 почти всегда = приложение за nginx не отвечает. Сначала проверяем бэкенд (curl 127.0.0.1:8080).

Базы данных

База данных хранит структурированные данные. Два мира:

• **Реляционные (SQL)**: PostgreSQL, MySQL/MariaDB. Таблицы, строгая схема, язык SQL.

• **NoSQL**: MongoDB (документы), Redis (ключ-значение, кэш). Гибкая структура.

Установка PostgreSQL

sudo apt install postgresql

sudo systemctl enable --now postgresql

sudo -u postgres psql войти в консоль БД

Базовые операции:

CREATE DATABASE appdb;

CREATE USER appuser WITH PASSWORD 'secret';

GRANT ALL PRIVILEGES ON DATABASE appdb TO appuser;

\l список баз

\du список пользователей

\q выход

MySQL / MariaDB

sudo apt install mariadb-server

sudo mysql_secure_installation базовая защита (пароль root, убрать тестовое)

sudo mysql
CREATE DATABASE appdb;

CREATE USER 'appuser'@'localhost' IDENTIFIED BY 'secret';

GRANT ALL ON appdb.* TO 'appuser'@'localhost';

FLUSH PRIVILEGES;

Обязанности админа по базам

• **Резервные копии** (критично! глава 16):
pg_dump appdb > appdb.sql PostgreSQL

mysqldump appdb > appdb.sql MySQL

• **Безопасность**: база слушает только localhost или внутреннюю сеть, сильные пароли, минимальные права пользователей.

• **Производительность**: индексы, мониторинг медленных запросов.

• **Обновления** и мониторинг места на диске.

:::warn База данных почти никогда не должна слушать весь интернет (0.0.0.0). Открытые наружу базы с дефолтными паролями — одна из главных причин утечек данных. База должна быть доступна только приложению: listen_addresses на localhost или внутреннюю сеть, firewall закрывает порт снаружи (5432 для PostgreSQL, 3306 для MySQL). :::

Репликация (вводно)

Для надёжности и нагрузки базы реплицируют: главный сервер (запись) + реплики (чтение). При сбое главного реплика может его заменить. Это тема для углубления, но знать о ней нужно.

Связка «веб + база»

Типичный стек: nginx (приём, HTTPS) → приложение (логика) → база данных (хранение). Админ обслуживает все три слоя: обновления, бэкапы, мониторинг, безопасность, диагностику.

Практика к главе 9

1. Поднимите nginx, отдайте статический сайт, затем настройте обратный прокси к простому приложению.

2. Получите HTTPS (если есть домен) или разберите самоподписанный сертификат локально.

3. Установите PostgreSQL, создайте базу, пользователя, выдайте права.

4. Сделайте дамп базы через pg_dump и восстановите его в новую базу.

5. Убедитесь, что база слушает только localhost, а firewall закрывает её порт снаружи.

6. Специально сломайте конфиг nginx и поймайте ошибку через nginx -t.

Частые ошибки новичка

- reload/restart nginx без nginx -t.
- Открывать порт базы данных в интернет.
- Дефолтные/слабые пароли баз.
- Не делать бэкапы базы (или не проверять восстановление).
- Путать 502/504 (проблема приложения) с проблемой nginx.

Глава 10. Файловые службы и общий доступ

Одна из классических задач сисадмина — организовать общий доступ к файлам: сетевые папки для отделов, обмен между Windows и Linux, централизованное хранение. Эта глава про файловые протоколы и службы.

Протоколы общего доступа

Протокол: SMB/CIFS | Откуда: Windows (и Linux через Samba) | Для чего: Сетевые папки, принтеры; основной в Windows-среде

Протокол: NFS | Откуда: Unix/Linux | Для чего: Общий доступ между Linux-системами

Протокол: FTP/SFTP | Откуда: Универсально | Для чего: Передача файлов (SFTP — безопасный, через SSH)

Протокол: WebDAV | Откуда: Через HTTP | Для чего: Доступ к файлам по вебу

Samba: сетевые папки для Windows из Linux

Samba позволяет Linux-серверу отдавать папки, которые Windows видит как сетевые диски.

sudo apt install samba

Конфиг /etc/samba/smb.conf:

[shared]

path = /srv/shared

browseable = yes

read only = no

valid users = @smbgroup

sudo smbpasswd -a ivan задать Samba-пароль пользователю

sudo systemctl restart smbd

С Windows: \\192.168.1.10\shared. С Linux:

sudo mount -t cifs //192.168.1.10/shared /mnt -o username=ivan

NFS: общий доступ между Linux

```
# на сервере

sudo apt install nfs-kernel-server

echo "/srv/data 192.168.1.0/24(rw,sync,no_subtree_check)" | sudo tee -a /etc/exports

sudo exportfs -ra

sudo systemctl restart nfs-kernel-server


# на клиенте

sudo mount 192.168.1.10:/srv/data /mnt
```

Для постоянного монтирования — запись в /etc/fstab.

Права на общих папках

Главная сложность файловых служб — права. Нужно, чтобы отдел «Бухгалтерия» видел свою папку, а «Маркетинг» — свою, и никто не лез к чужому.

• В Linux/Samba — через пользователей, группы и права (глава 5).

• В Windows — через NTFS-права и общий доступ (ACL).

• Принцип: папка на группу, пользователь — в группу. Не раздавайте права каждому отдельно.

:::note При сетевом доступе действуют ДВА уровня прав одновременно: права самой файловой системы (NTFS/Linux) и права сетевой шары (Share/Samba). Итоговый доступ — наиболее строгое пересечение этих двух. Классическая путаница новичка: «в шаре разрешил, а не пускает» — потому что права файловой системы строже. Проверяйте оба уровня. :::

SFTP: безопасная передача файлов

SFTP работает поверх SSH — ничего дополнительно ставить не нужно, если есть SSH-сервер.

sftp ivan@192.168.1.10

> put localfile.txt загрузить

> get remotefile.txt скачать

> ls / cd / pwd навигация

Для ограниченного доступа настраивают chroot — пользователь видит только свою папку. FTP без шифрования (ftp, порт 21) использовать не стоит — пароли идут открытым текстом.

Централизованное хранилище и RAID

Для серьёзных объёмов используют:

• **NAS** (Network Attached Storage) — готовое сетевое хранилище.

• **Сервер с RAID** (глава 2) — защита от выхода диска из строя.

• **Квоты** — ограничение места на пользователя/группу, чтобы один не забил весь диск.

пример: место, занимаемое папками

du -sh /srv/shared/*

df -h свободное место

Резервное копирование файловых хранилищ

Файловые серверы — первые кандидаты на бэкап: там живые документы пользователей. Напоминание из главы 2: RAID ≠ бэкап. Шифровальщик зашифрует и зеркало RAID. Нужны отдельные резервные копии (глава 16).

Практика к главе 10

1. Поднимите Samba-шару на Linux-сервере и подключитесь к ней с Windows (или другого Linux).

2. Настройте доступ так, чтобы папку видела только определённая группа.

3. Поднимите NFS-шару между двумя Linux-VM, смонтируйте её.

4. Передайте файлы через SFTP.

5. Смоделируйте путаницу прав: разрешите доступ в шаре, но ограничьте на уровне файловой системы, и понаблюдайте результат.

6. Настройте постоянное монтирование через /etc/fstab.

Частые ошибки новичка

• Путать права шары и права файловой системы (итог — строжайшее из двух).

• Использовать незашифрованный FTP вместо SFTP.

• Раздавать права каждому пользователю вместо групп.

• Считать RAID заменой бэкапу файлового сервера.

• Не ставить квоты — один пользователь забивает весь диск.

Глава 11. Почта, удалённый доступ и VPN

Три важные темы: как устроена почта (одна из самых сложных служб), как безопасно подключаться к серверам издалека и как объединять сети через VPN.

Почтовые службы: как работает email

Почта кажется простой, но внутри — несколько протоколов и служб:

SMTP — Отправка и пересылка почты между серверами (порт 25, 587)

IMAP — Получение почты с сохранением на сервере (порт 993)

POP3 — Получение почты со скачиванием (порт 995, устаревает)

Путь письма:

Отправитель → SMTP своего сервера → (DNS: MX-запись получателя) →

SMTP сервера получателя → почтовый ящик → IMAP/POP3 → получатель читает

Компоненты почтовой системы

• **MTA** (Mail Transfer Agent) — пересылает почту: Postfix, Exim.

• **MDA** (Mail Delivery Agent) — раскладывает по ящикам: Dovecot.

• **Защита**: антиспам (SpamAssassin, Rspamd), антивирус.

Почему свой почтовый сервер — это сложно

:::warn Поднять почтовый сервер технически можно за час, но заставить мир принимать вашу почту (а не отправлять в спам) — отдельная большая задача. Нужны корректные записи DNS: MX (куда слать), SPF (кто вправе слать от домена), DKIM (цифровая подпись писем), DMARC (политика проверки), а также «чистый» IP-адрес с хорошей репутацией и обратная PTR-запись. Без этого письма уходят в спам. Поэтому многие организации используют готовые почтовые сервисы, а не поднимают свой. Для учёбы — поднимите в лаборатории, для продакшена — взвесьте, нужно ли оно. :::

Записи DNS для почты

MX company.ru → mail.company.ru куда доставлять почту

SPF TXT "v=spf1 mx -all" кто вправе слать от домена

DKIM TXT (публичный ключ подписи) проверка подлинности

DMARC TXT "v=DMARC1; p=quarantine" что делать с

подозрительным

PTR (обратная запись IP → имя) репутация

Удалённый доступ: SSH (Linux) и RDP (Windows)

SSH — основной доступ к Linux-серверам

ssh ivan@server

ssh -i ~/.ssh/key ivan@server -p 2222

scp file ivan@server:/path копировать файл
Безопасная настройка SSH-сервера (/etc/ssh/sshd_config):
PermitRootLogin no запретить вход root

PasswordAuthentication no только по ключам

Port 22 (иногда меняют, но это не защита сама по себе)

AllowUsers ivan deploy разрешённые пользователи
sudo systemctl restart ssh
Вход по ключу (безопаснее пароля):
ssh-keygen -t ed25519

ssh-copy-id ivan@server

RDP — удалённый рабочий стол Windows

Windows Server управляют через RDP (порт 3389). Правила безопасности:

• RDP **никогда** не открывают напрямую в интернет — только через VPN.

• Сильные пароли, ограничение пользователей, блокировка при подборе.

• Открытый в интернет RDP — цель №1 для вирусов-шифровальщиков.

:::note Открытый в интернет RDP (порт 3389) — одна из самых частых точек входа для атак шифровальщиков. Если нужен удалённый доступ к Windows-серверу извне — только через VPN или шлюз (RD Gateway), никогда напрямую. То же касается SSH: по возможности прячьте за VPN и используйте только ключи. :::

VPN: безопасное объединение сетей

VPN создаёт зашифрованный «туннель» через интернет: удалённый сотрудник или филиал работает так, будто находится внутри сети компании.

Виды VPN

• **Удалённый доступ** — сотрудник из дома подключается к сети офиса.
• **Site-to-site** — два офиса объединяются в одну сеть.

Популярные решения

WireGuard — Современный, простой, быстрый — лучший выбор для старта
OpenVPN — Проверенный, гибкий, повсеместный
IPsec — Корпоративный стандарт для site-to-site

Пример WireGuard (концептуально)

```
# сервер /etc/wireguard/wg0.conf

[Interface]
```

Address = 10.8.0.1/24

PrivateKey = <ключ сервера>

ListenPort = 51820

[Peer] клиент

PublicKey = <публичный ключ клиента>

AllowedIPs = 10.8.0.2/32
sudo wg-quick up wg0

sudo systemctl enable wg-quick@wg0
Клиент получает адрес из 10.8.0.0/24 и через туннель видит внутреннюю сеть.

Практика к главе 11

1. Нарисуйте путь письма от отправителя к получателю, подпишите протоколы и роль DNS (MX).

2. Объясните своими словами, зачем нужны SPF, DKIM, DMARC и почему без них почта уходит в спам.

3. Настройте безопасный SSH: вход по ключу, запрет root, ограничение пользователей.

4. Поднимите WireGuard между двумя VM и убедитесь, что они видят друг друга через туннель.

5. Объясните, почему RDP нельзя открывать напрямую в интернет.

Частые ошибки новичка

• Поднять почтовый сервер без SPF/DKIM/DMARC/PTR и удивляться спам-папке.

• Открыть RDP или SSH напрямую в интернет с паролем.

• Оставить вход root по SSH и аутентификацию по паролю.

• Считать смену порта SSH «защитой» (это лишь снижение шума, не защита).

• Путать протоколы отправки (SMTP) и получения (IMAP/POP3) почты.

Глава 12. Диски, хранилища и управление данными

Данные — главная ценность организации. Админ отвечает за то, где они лежат, как не кончится место, как пережить выход диска из строя и как расширяться. Эта глава — про управление хранилищем в Linux.

Устройство дисков в Linux

lsblk список дисков и разделов

sda первый диск

├──sda1 раздел 1 (например, /boot)

└──sda2 раздел 2

sdb второй диск

Диски — это /dev/sda, /dev/sdb; разделы — /dev/sda1 и т. д. NVMe-диски: /dev/nvme0n1.

Разделы и файловые системы

Чтобы использовать диск: создать раздел → создать файловую систему → примонтировать.

sudo fdisk /dev/sdb создать раздел (интерактивно)

sudo mkfs.ext4 /dev/sdb1 создать файловую систему ext4

sudo mkdir /mnt/data

sudo mount /dev/sdb1 /mnt/data примонтировать

df -h проверить

Постоянное монтирование: /etc/fstab

Чтобы диск монтировался при загрузке, добавляют запись в /etc/fstab:

использовать UUID (стабильнее, чем /dev/sdb1)

UUID=xxxx-xxxx /mnt/data ext4 defaults 0 2

UUID узнают командой blkid.

:::warn Ошибка в /etc/fstab может помешать серверу загрузиться (он зависнет на монтировании). Всегда проверяйте запись командой sudo mount -a ПЕРЕД перезагрузкой — она

применит fstab без ребута и покажет ошибки. Используйте UUID вместо /dev/sdX, потому что буквы дисков могут меняться. :::

LVM: гибкое управление томами

LVM (Logical Volume Manager) — слой абстракции, позволяющий менять размеры «на лету» и объединять диски. Вместо жёстких разделов — гибкие логические тома.

Физические диски (PV) → Группа томов (VG) → Логические тома (LV)

/dev/sdb, /dev/sdc → vg_data → lv_files, lv_db

sudo pvcreate /dev/sdb /dev/sdc пометить диски для LVM

sudo vgcreate vg_data /dev/sdb /dev/sdc создать группу

sudo lvcreate -L 50G -n lv_files vg_data создать том 50 ГБ

sudo mkfs.ext4 /dev/vg_data/lv_files

Главное преимущество — расширение без простоя:

sudo lvextend -L +20G /dev/vg_data/lv_files увеличить том

sudo resize2fs /dev/vg_data/lv_files растянуть ФС

Закончилось место — добавили диск в группу и расширили том, не останавливая службы. Это огромное удобство для админа.

Мониторинг места — ежедневная обязанность

df -h свободное место по ФС

df -i свободные inode (файлы кончаются и так!)

du -sh /var/* что занимает место

du -sh /* 2>/dev/null | sort -rh | head топ по размеру

ncdu / интерактивный анализатор (удобно)

:::note Диск может «кончиться» двумя способами: закончилось место (df -h показывает 100%) или закончились inode — внутренние записи о файлах (df -i). Второе случается при миллионах мелких файлов, и это сбивает с толку: место вроде есть, а файл не создаётся. Проверяйте оба. :::

Ротация логов

Логи растут и забивают диск. logrotate автоматически архивирует и удаляет старые:

```
# /etc/logrotate.d/myapp

/var/log/myapp/*.log {

daily

rotate 14 хранить 14 дней

compress сжимать старые

missingok

notifempty

}
```

Забытая ротиция логов — частая причина внезапно забитого диска.

SMART: здоровье дисков

Диски умирают, часто предупреждая заранее. SMART читает их самодиагностику:

sudo apt install smartmontools

sudo smartctl -a /dev/sda полная информация о здоровье диска

sudo smartctl -H /dev/sda краткий вердикт (PASSED/FAILED)

Настройте мониторинг SMART — предупреждение о «стареющем» диске позволяет заменить его до потери данных.

Снимки и резервные копии

• **Снимки LVM/ZFS** — мгновенная «фотография» состояния для консистентного бэкапа.

• **Резервные копии** — отдельная большая тема (глава 16).

Напоминание: RAID и снимки — не замена бэкапу (глава 2).

Практика к главе 12

1. Добавьте в VM второй диск, создайте раздел, ФС, примонтируйте, пропишите в fstab через UUID.
2. Проверьте fstab через mount -a до перезагрузки.
3. Настройте LVM: объедините два диска в группу, создайте том, затем расширьте его.
4. Исследуйте, что занимает место, через du и ncdu.
5. Настройте logrotate для какого-нибудь лога.
6. Посмотрите здоровье диска через smartctl.

Частые ошибки новичка

• Ошибка в /etc/fstab без проверки mount -a — сервер не грузится.

• Использовать /dev/sdX в fstab вместо UUID.

• Не мониторить место (df -h) и inode (df -i).

• Забывать про ротацию логов — диск забивается.

• Игнорировать SMART-предупреждения до смерти диска.

Глава 13. Windows Server: основы

В корпоративной среде рядом с Linux почти всегда есть Windows Server: на нём держат Active Directory, файловые службы, почту Exchange, терминальные серверы. Админ обязан уверенно работать в обеих системах. Эта глава — введение в Windows Server.

Чем Windows Server отличается от обычной Windows

Это серверная ОС: оптимизирована под службы, поддерживает роли (Active Directory, DNS, DHCP, файловый сервер), управляется централизованно и работает годами без перезагрузки. Варианты установки:

• **Desktop Experience** — с привычным графическим интерфейсом.

• **Server Core** — без графики, только командная строка и PowerShell (меньше ресурсов, меньше уязвимостей — предпочтительно для многих ролей).

Роли и компоненты

Windows Server строится из **ролей** — наборов функций:

Active Directory Domain Services (AD DS) — Контроллер домена, централизованное управление

DNS Server — Служба имён (критична для AD)

DHCP Server — Раздача адресов

File Server — Файловые службы, общие папки

Hyper-V — Виртуализация

Web Server (IIS) — Веб-сервер Microsoft

Remote Desktop Services — Терминальные серверы

Роли добавляют через «Server Manager» (графика) или PowerShell.

Управление: GUI, Server Manager и PowerShell

• **Server Manager** — центральная панель управления ролями и серверами.

• **PowerShell** — мощная командная оболочка, аналог терминала Linux. Позволяет автоматизировать всё.

• **Windows Admin Center** — современный веб-интерфейс управления.

• **RDP** — удалённый рабочий стол для доступа.

PowerShell: терминал Windows

PowerShell — это не старый cmd, а полноценный язык автоматизации. Команды называются Глагол-Существительное:

Get-Service список служб

Get-Service | Where-Object {$_.Status -eq "Running"}

Restart-Service -Name Spooler перезапустить службу

Get-Process процессы

Get-EventLog -LogName System -Newest 20 системные события

Get-WindowsFeature установленные роли

Install-WindowsFeature AD-Domain-Services установить роль

Get-NetIPConfiguration сетевые настройки

Test-Connection 8.8.8.8 аналог ping

:::note PowerShell работает с объектами, а не с текстом

(в отличие от bash). Get-Service | Where-Object {$_.Status -eq "Stopped"} фильтрует объекты-службы по свойству. Это мощнее текстовой обработки, но требует иного мышления. Для админа Windows PowerShell так же важен, как bash для Linux. :::

Управление пользователями (локальными)

Get-LocalUser список

New-LocalUser -Name "ivan" -Password (Read-Host -AsSecureString)

Add-LocalGroupMember -Group "Administrators" -Member "ivan"

В доменной среде пользователями управляют централизованно через Active Directory (глава 14), а не локально на каждой машине.

Сеть в Windows Server

Get-NetIPAddress адреса

New-NetIPAddress -InterfaceAlias "Ethernet" -IPAddress 192.168.1.10 -PrefixLength 24 -DefaultGateway 192.168.1.1

Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses 192.168.1.2

Серверу задают статический адрес — как и в Linux.

Службы и их логи

• Службы: оснастка services.msc или Get-Service/Set-Service в PowerShell.

• Логи: «Просмотр событий» (Event Viewer) — System, Application, Security. Аналог journalctl, но с классификацией по журналам. Get-EventLog / Get-WinEvent в PowerShell.

Обновления

Windows Update на серверах управляют контролируемо (часто через WSUS — централизованный сервер обновлений), чтобы патчи не ставились и не перезагружали сервер внезапно. Неустановленные обновления — частая причина взломов, но и неконтролируемая перезагрузка в рабочее время недопустима. Баланс — через планирование.

Активация и лицензии

Windows Server — платная ОС. Для обучения Microsoft даёт бесплатные **Evaluation**-версии (180 дней) — их и используйте в лаборатории.

Практика к главе 13

1. Поднимите Windows Server (Evaluation) в VM, задайте статический IP через PowerShell.

2. Освойте базовый PowerShell: Get-Service, Get-Process, фильтрация через Where-Object.

3. Установите роль (например, DNS или File Server) через Server Manager и через Install-WindowsFeature.

4. Создайте локального пользователя и добавьте в группу через PowerShell.

5. Изучите Event Viewer: найдите последние ошибки в журналах System и Application.

Частые ошибки новичка

• Игнорировать PowerShell и всё делать мышкой — не масштабируется.

• Давать серверу динамический адрес.

• Оставлять неконтролируемый Windows Update с автоперезагрузкой.

• Управлять пользователями локально вместо AD в доменной среде.

• Открывать RDP в интернет (глава 11).

Глава 14. Active Directory: централизованное управление

Active Directory (AD) — сердце корпоративной Windows-инфраструктуры. Это каталог, где хранятся все пользователи, компьютеры, группы и политики, и который позволяет управлять тысячами машин из одного места. Для сисадмина в корпоративной среде AD — обязательный навык.

Зачем нужен Active Directory

Представьте компанию на 500 человек без AD: на каждом из 500 компьютеров нужно вручную завести учётки, настроить права, политики. Сотрудник уволился — обойти 500 машин. Это невозможно. AD решает всё:

- **Единый вход (SSO)**: одна учётная запись работает на всех компьютерах домена.
- **Централизованное управление**: политики, права, настройки — из одного места.
- **Масштаб**: тысячи объектов управляются единообразно.
- **Безопасность**: централизованная аутентификация и контроль доступа.

Ключевые понятия

Домен (domain) — «Семья» пользователей и компьютеров под единым управлением

Контроллер домена (DC) — Сервер с базой AD, проверяет входы

Лес (forest) — Верхний уровень: один или несколько доменов

Дерево (tree) — Домены с общим именем (child.company.local)

OU (подразделение) — Папка для группировки объектов (по отделам)

Группа (group) — Набор пользователей с общими правами

GPO (групповая политика) — Правила и настройки, применяемые массово

LDAP — Протокол доступа к каталогу

Kerberos — Протокол аутентификации по «билетам»

Структура: лес → домен → OU → объекты

Лес: company.local

Домен: company.local

OU: Москва

OU: Бухгалтерия

Пользователи, компьютеры

OU: IT

OU: Питер

OU (Organizational Units) — это как папки, по которым раскладывают пользователей и компьютеры. На OU навешивают групповые политики — так настройки применяются к целым отделам.

Установка контроллера домена

```
# PowerShell на Windows Server

Install-WindowsFeature AD-Domain-Services -
IncludeManagementTools

Install-ADDSForest -DomainName "company.local"
```

После этого сервер становится контроллером домена. Он автоматически поднимает и DNS (AD без DNS не работает — это критическая связка).

:::note Active Directory полностью зависит от DNS. Клиенты находят контроллер домена через специальные DNS-записи (SRV). Если DNS настроен неверно — AD «ломается» необъяснимым образом: вход тормозит, политики не применяются, компьютеры «теряют» домен. Правило: клиенты и серверы домена должны использовать внутренний DNS (контроллер домена), а не внешний (8.8.8.8). Это причина 90% проблем с AD у новичков. :::

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, купив полную легальную версию на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.