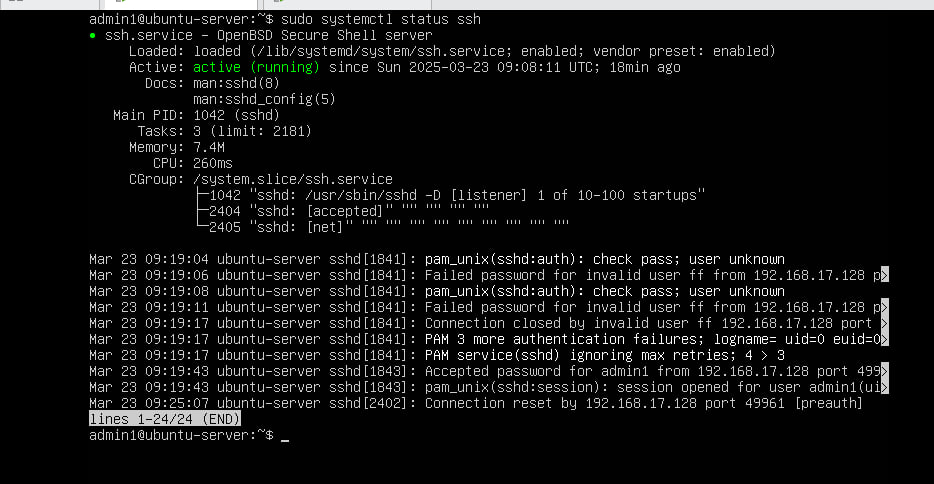
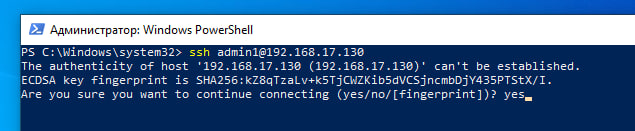
**Автоматизация настройки Linux-сервера**

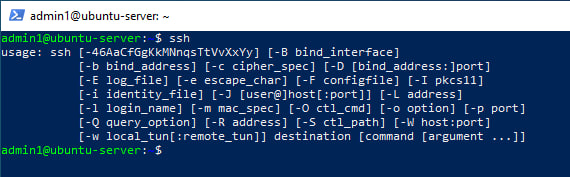
Я самостоятельно настроил с нуля полноценный Linux-сервер на Ubuntu 22.04 и автоматизировал процесс его конфигурации. Это не просто «чтобы работало», а полноценная продакшн-подготовка, как это делают в реальных компаниях. Я подходил к задаче как системный админ: чтобы всё было безопасно, чётко, логично и удобно для дальнейшей работы.

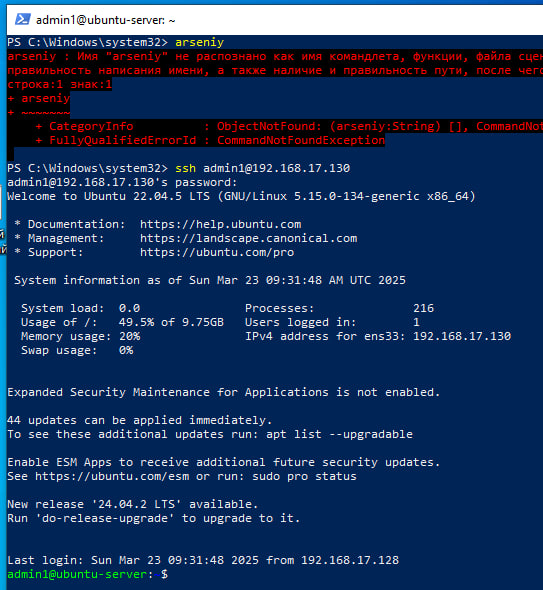
1. **Подключение Ubuntu по SSH с Windows**

Я начал настройку с подключения к Ubuntu-серверу по SSH с клиентской машины на Windows. Это подтвердило, что сеть и SSH-сервер работают корректно.



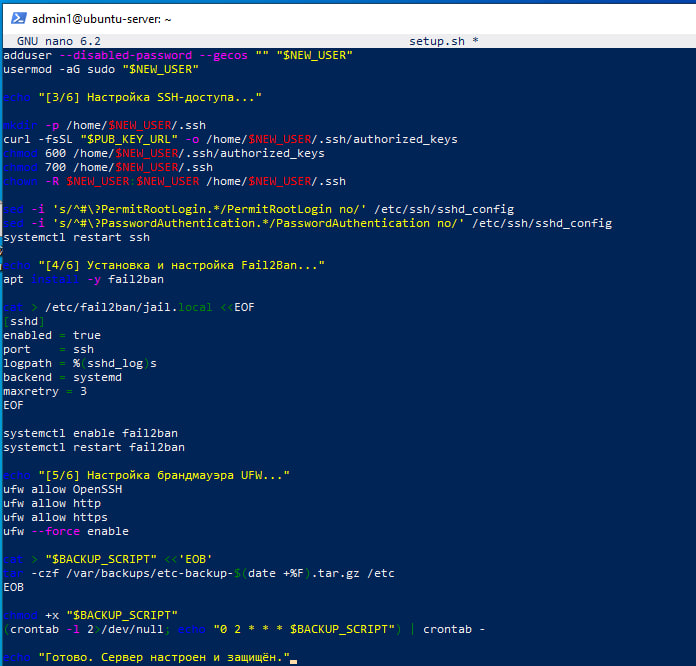


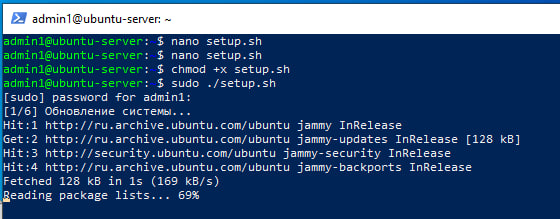




1. **Запуск скрипта автоматизации**

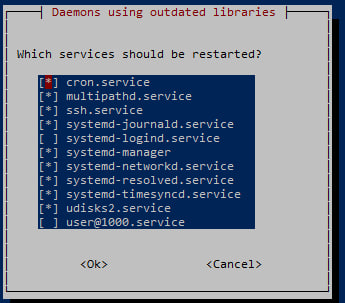
Я написал скрипт setup.sh, который автоматически выполняет все основные действия: обновление системы, создание пользователя, настройка SSH, fail2ban, UFW и cron. Далее я запустил его под sudo.





1. **Перезапуск служб после обновления**

После обновления система предложила перезапустить службы. Я выбрал рекомендованные и подтвердил перезапуск, чтобы изменения вступили в силу.



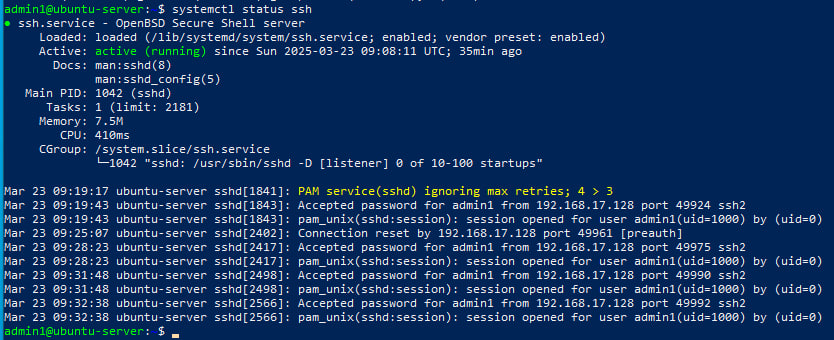
1. **Создание нового пользователя devadmin**

Скрипт создал нового пользователя devadmin, добавил его в группу sudo. Я проверил, что пользователь существует и имеет нужные права.



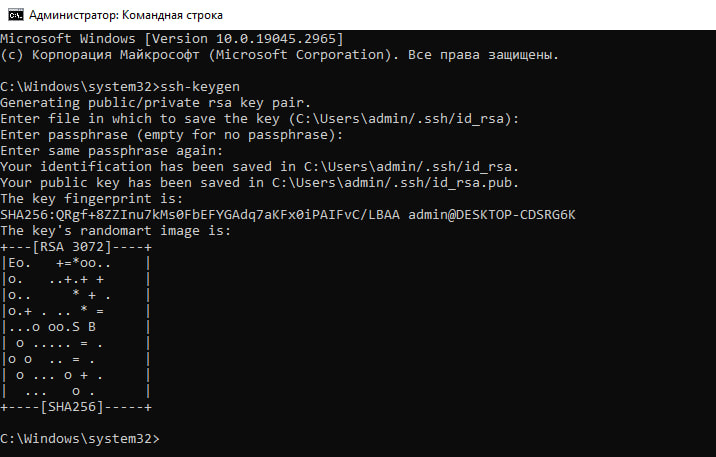
1. **Проверка статуса SSH**

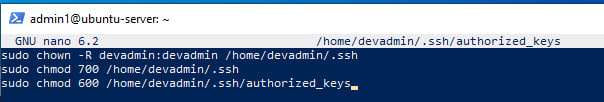
Я внёс изменения в конфигурацию SSH — отключил root-доступ и вход по паролю. Затем перезапустил SSH и убедился, что служба работает.



1. **Создание SSH-ключа на клиентской ОС**

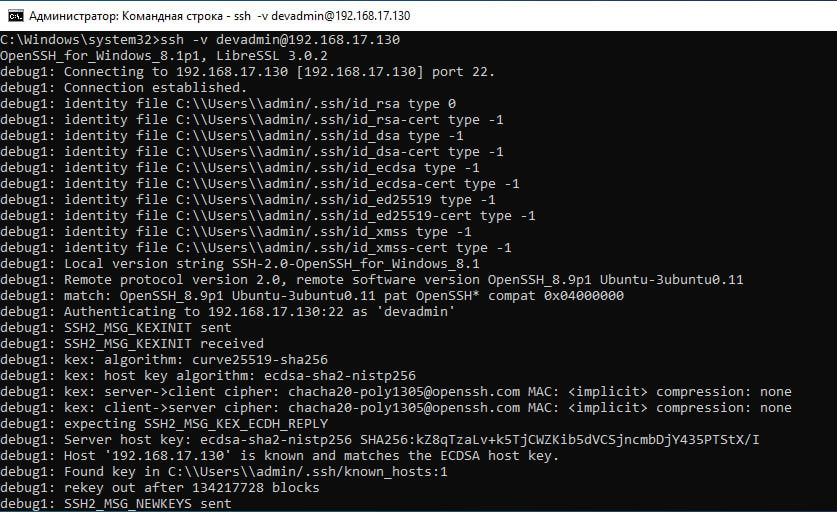
На клиентской Windows я создал SSH-ключ через ssh-keygen и добавил публичный ключ в Ubuntu, чтобы подключаться без пароля.





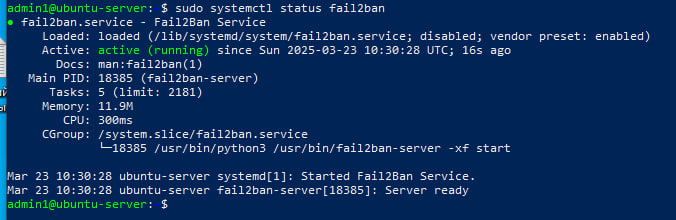
1. **Проверка входа по SSH-ключу**

После настройки ключа я успешно вошёл на сервер под пользователем devadmin по SSH без пароля — вход только по ключу. Это подтверждает, что всё работает безопасно.



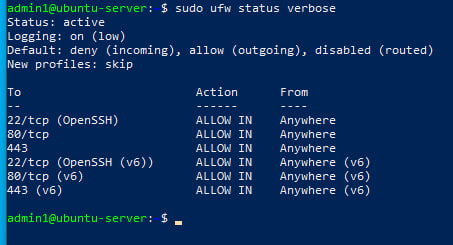
1. **Проверка Fail2Ban**

Я установил и настроил Fail2Ban для защиты от брутфорса. Jail sshd активен. Это повышает безопасность SSH-сервера.



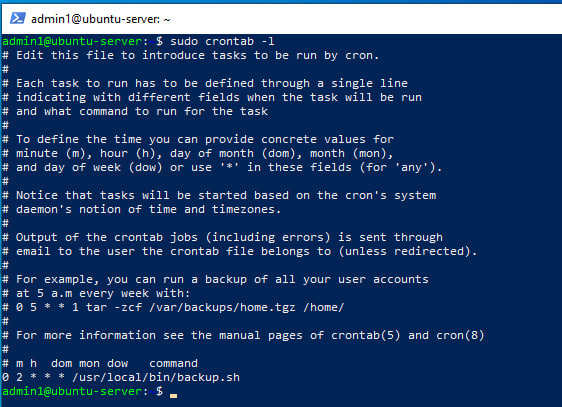
1. **Настройка UFW (брандмауэр)**

Я включил UFW, разрешил только нужные порты (22, 80, 443). Проверил правила — всё работает как надо, лишние подключения блокируются.



1. **Настройка cron-задачи на бэкап**

Создал скрипт backup.sh и добавил cron-задачу на ежедневный запуск. Проверил, что задача записана в sudo crontab -l



1. **Проверка результата бэкапа**

Запустил скрипт вручную и убедился, что в /var/backups появился архив с актуальной датой. Это подтверждает, что бэкап работает.

