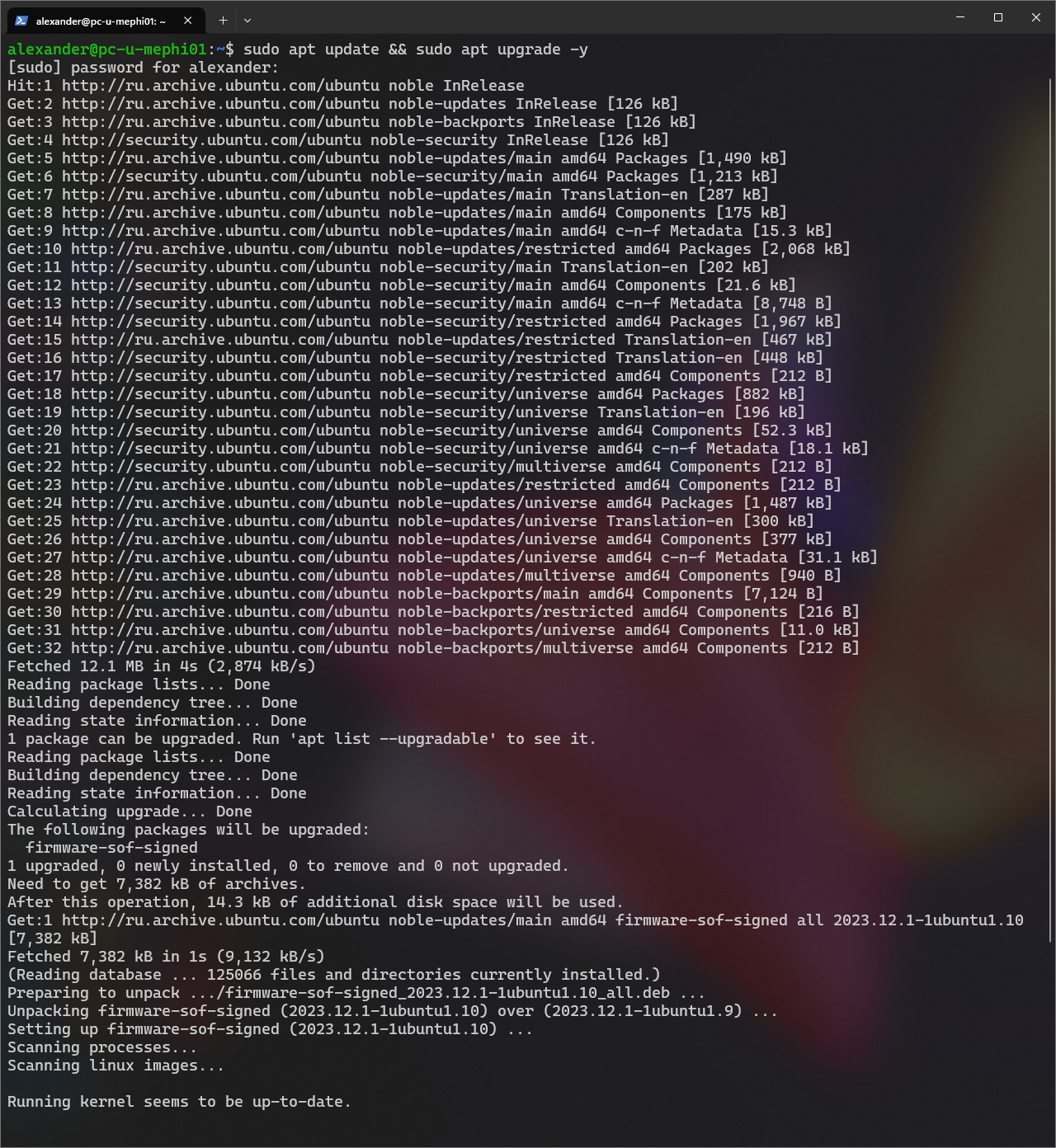
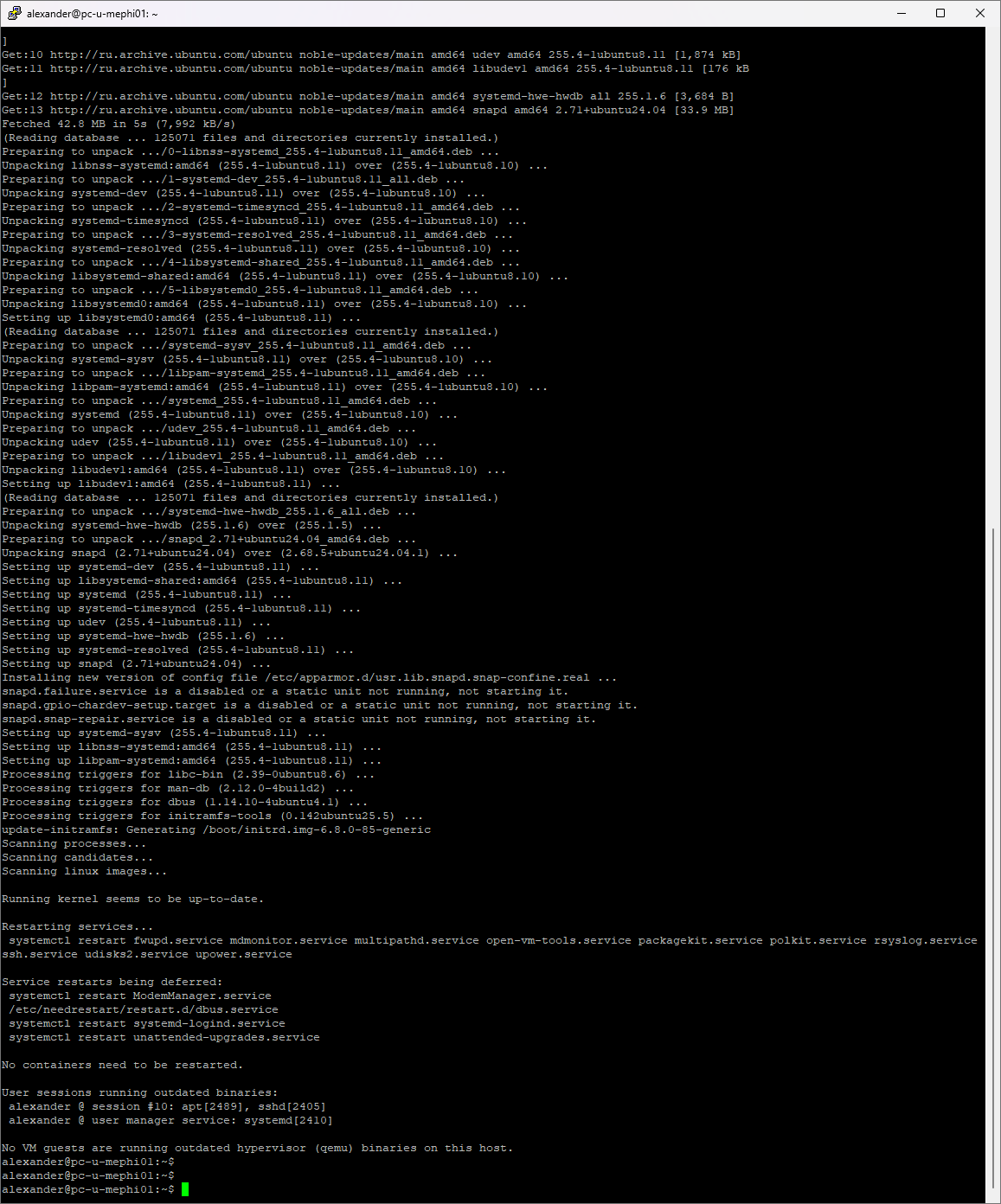
# Часть 1: Настройка Apache с HTTPS

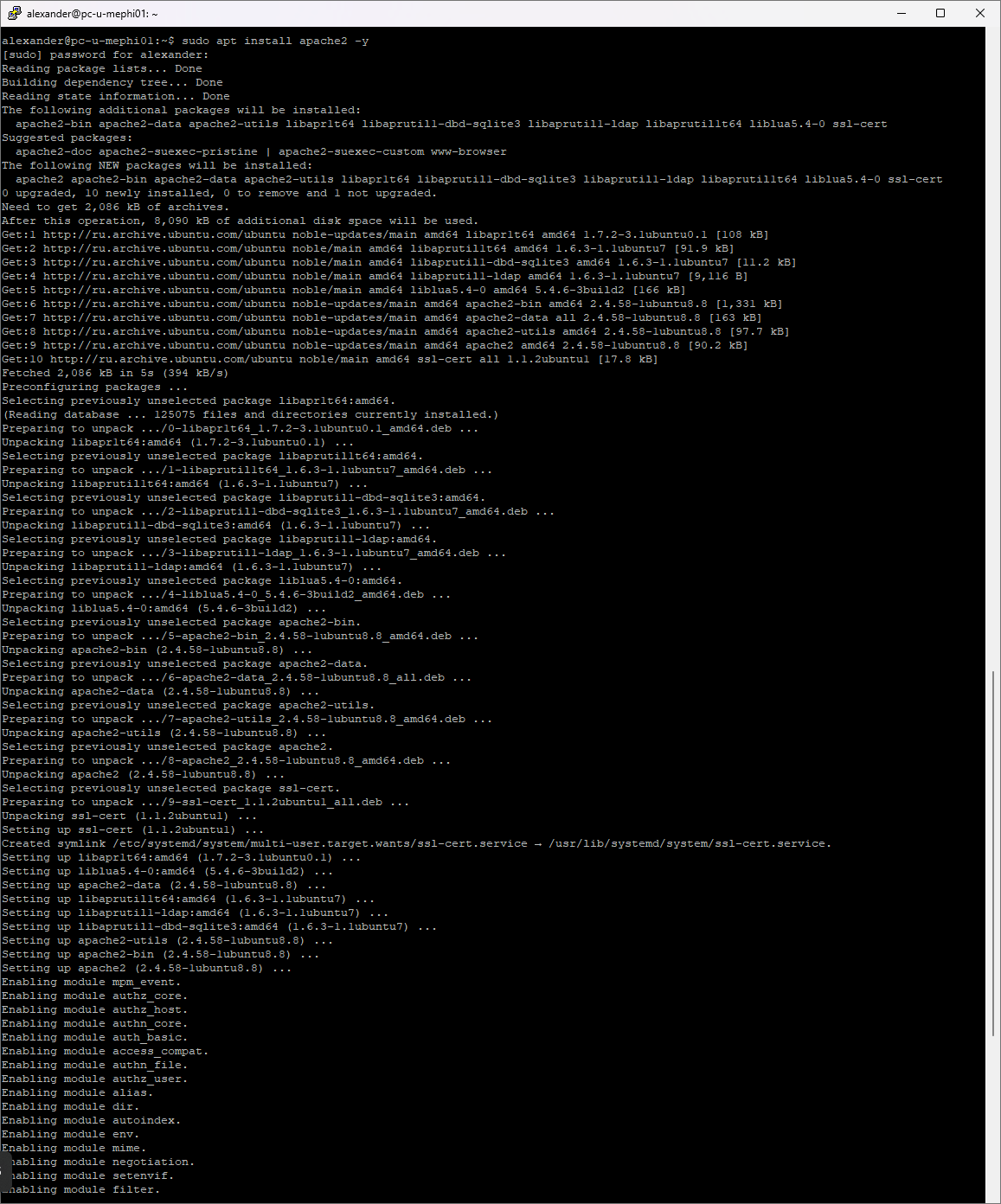
## Шаг 1: Установка Apache

### Обновление пакетов системы



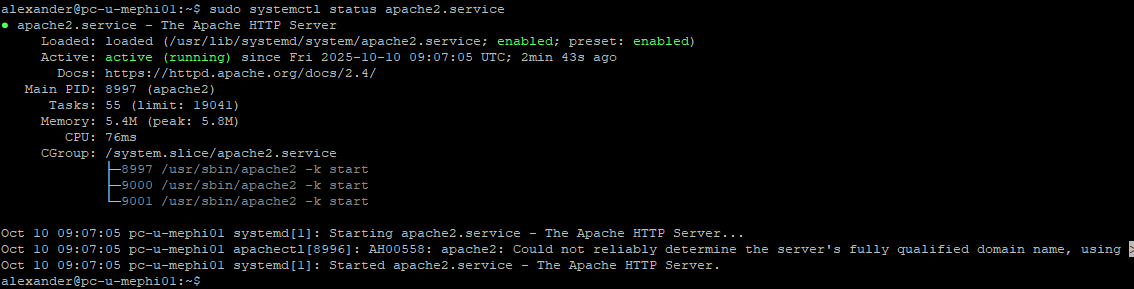


### Установка Apache



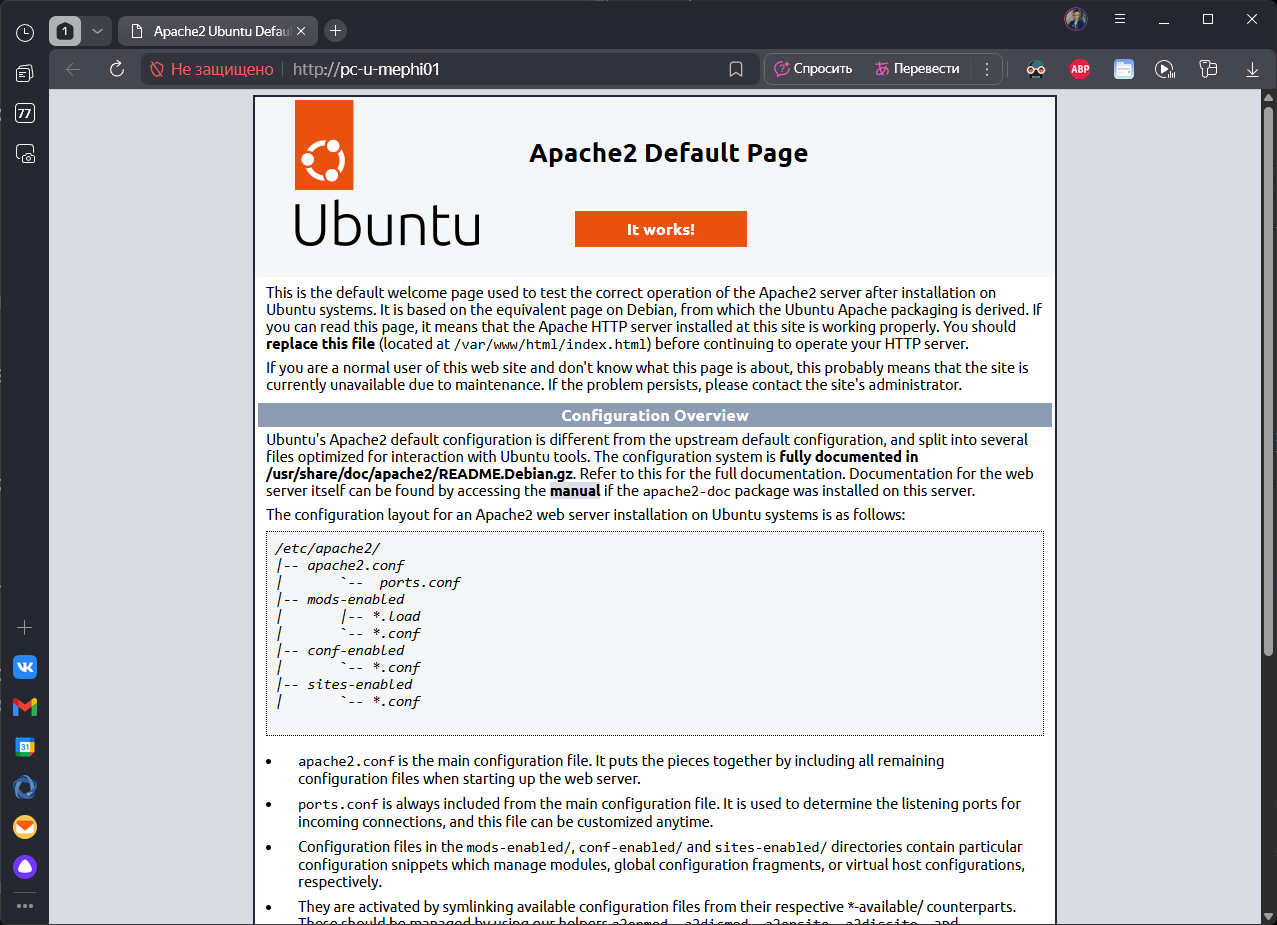
## Шаг 2: Проверка работы Apache

### Проверка статуса службы Apache

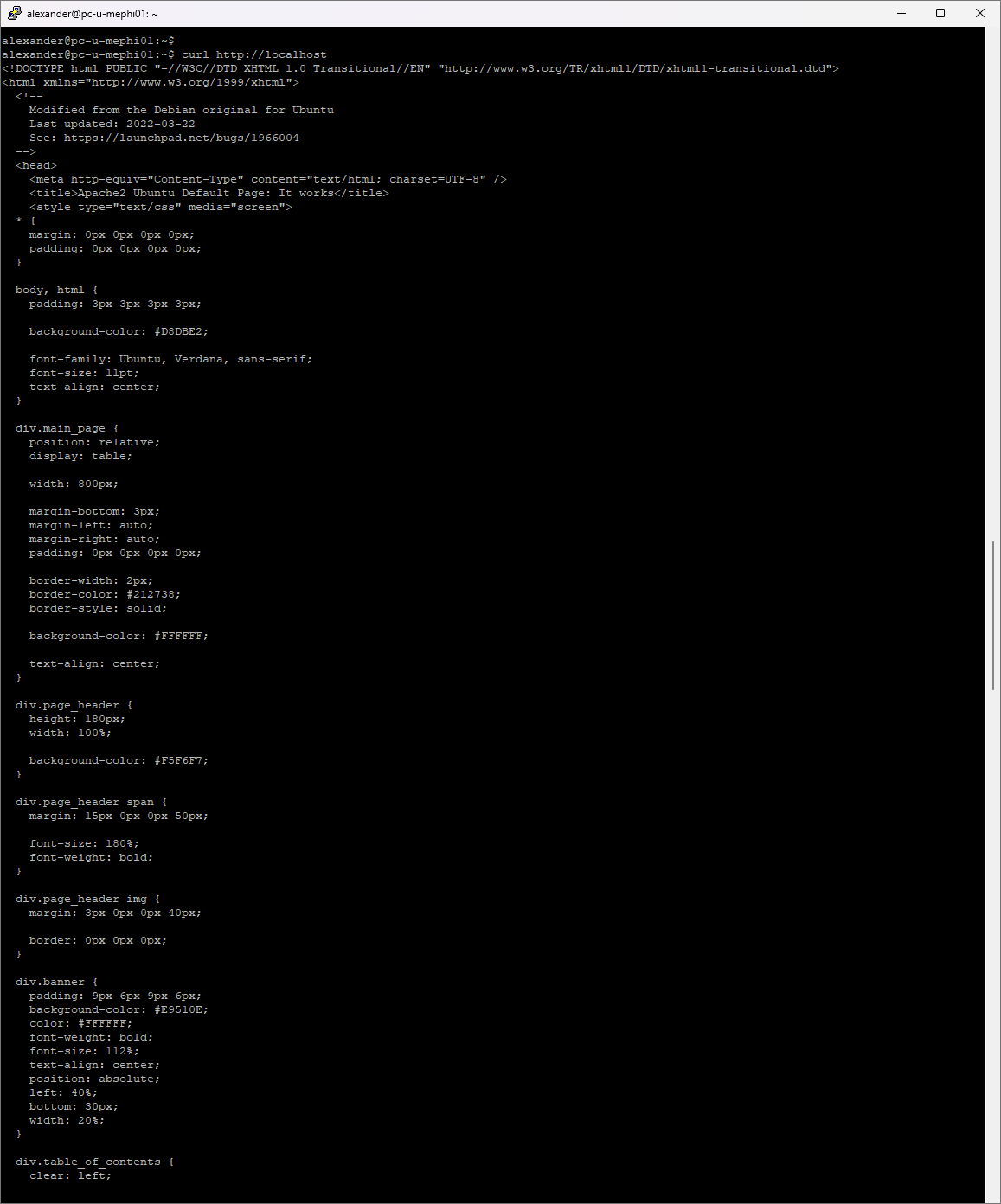


### Проверка работы через браузер или curl

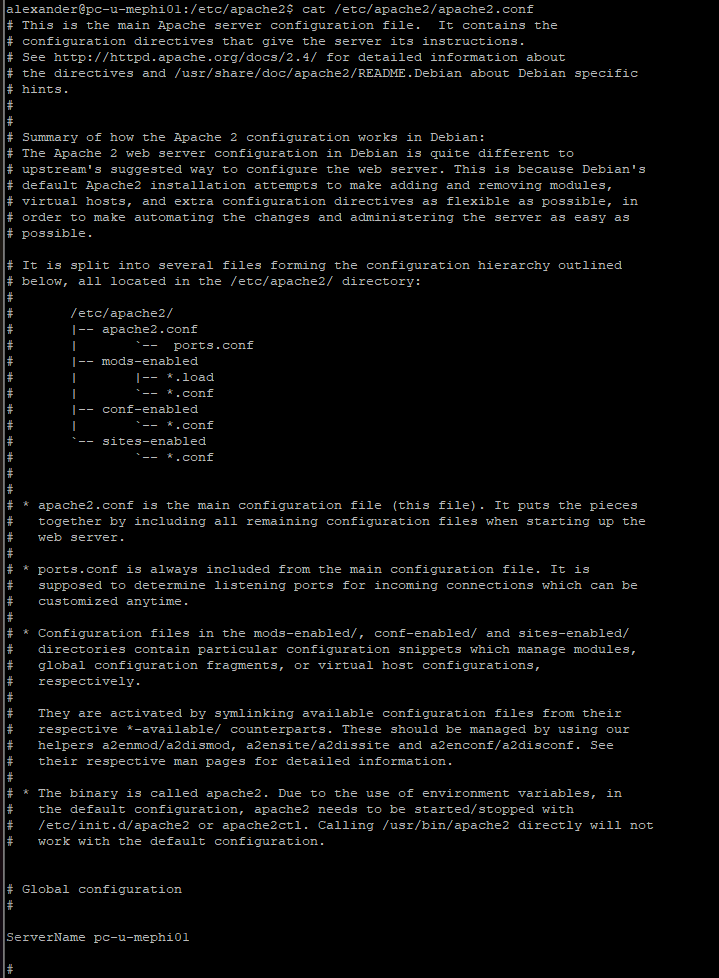
Поскольку ВМ подключена напрямую к роутеру, она доступна в локальной сети по своему статическому IP адресу. Если для данного IP добавить алиас «pc-u-mephi01» в файл C:\Windows\System32\drivers\etc\hosts , то наша виртуальная машина станет доступна по адресу <http://pc-u-mephi01> . Проверим работу Apache в браузере:



Проверим работу через утилиту curl .

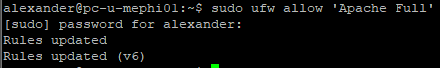


### Добавим ServerName в конфигурацию /etc/apache2/apache2.conf



## Шаг 3: Настройка брандмауэра

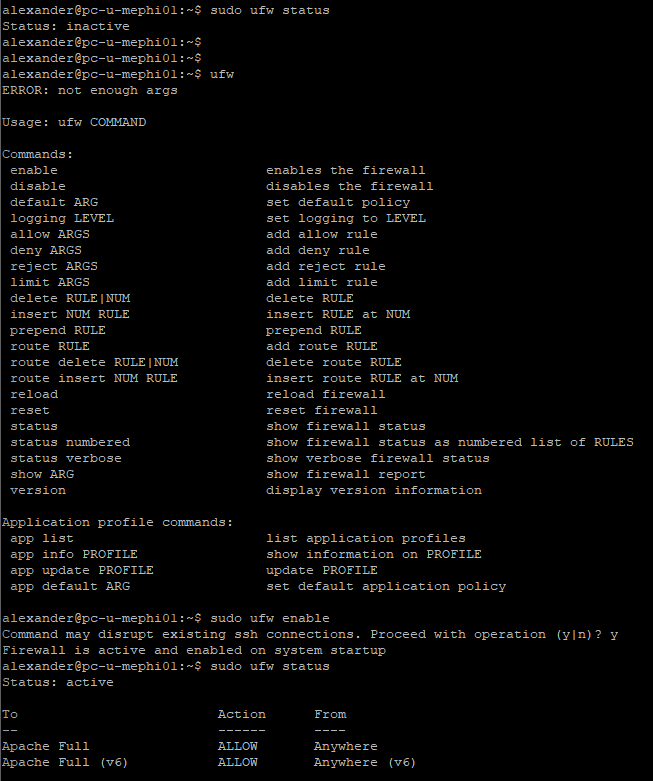
### Разрешение HTTP и HTTPS трафика



### Проверка правил брандмауэра

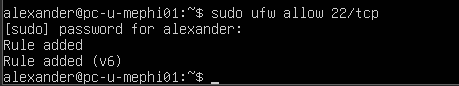


Включим брандмауэр и проверим его статус

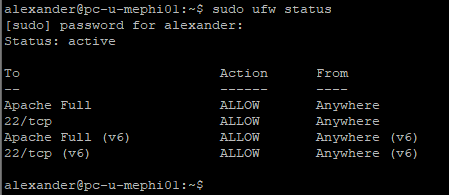


### Разрешение подключения по ssh

Поскольку мы должны взаимодействовать с ВМ по ssh, внесем соответствующее разрешение в брандмауэр.

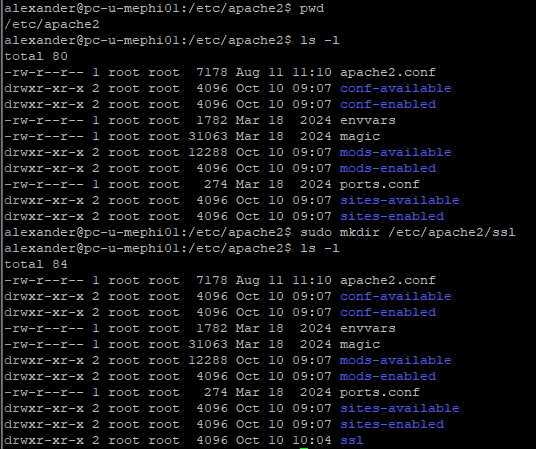


Проверим изменения:



## Шаг 4: Создание самоподписанного SSL-сертификата

### Создание директории для сертификатов

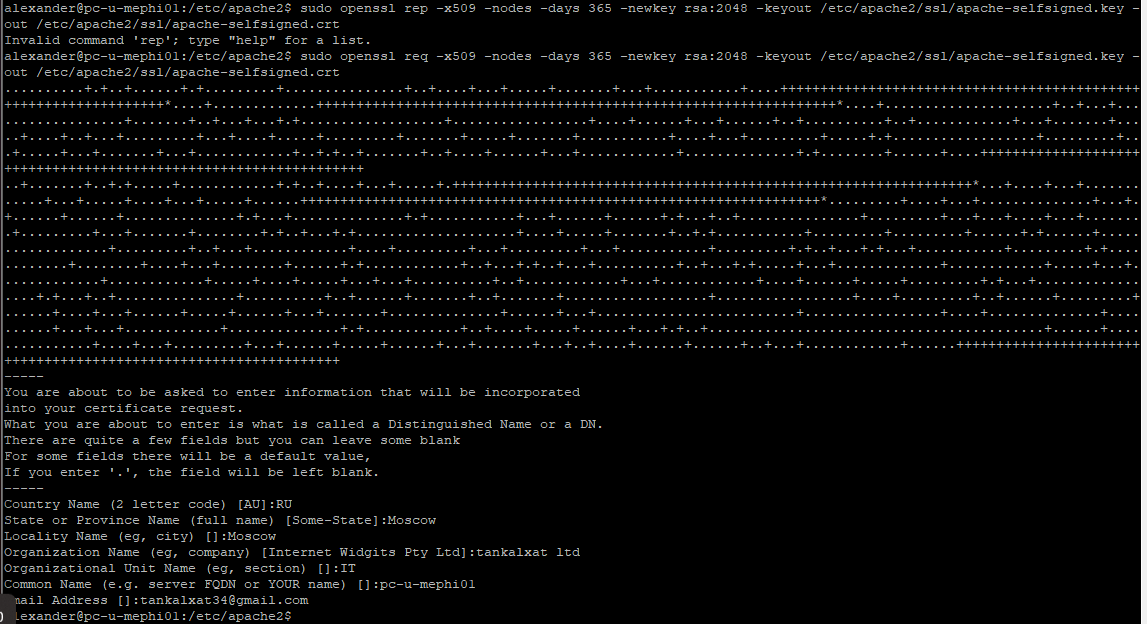


### Генерация самоподписанного сертификата

Воспользуемся командой: sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache-selfsigned.key -out /etc/apache2/ssl/apache-selfsigned.crt

**Параметры команды:**

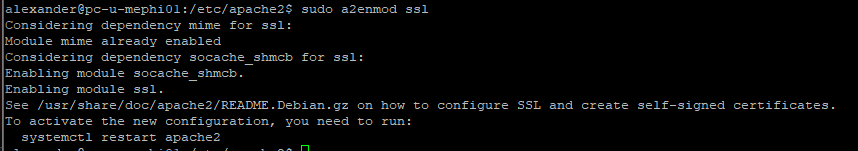
* sudo — запуск с правами администратора
* openssl req — создание запроса на подпись сертификата
* -x509 — создать самоподписанный сертификат
* -nodes — не шифровать приватный ключ
* -days 365 — срок действия сертификата (1 год)
* -newkey rsa:2048 — создать новый RSA ключ длиной 2048 бит
* -keyout — путь для сохранения приватного ключа
* -out — путь для сохранения сертификата



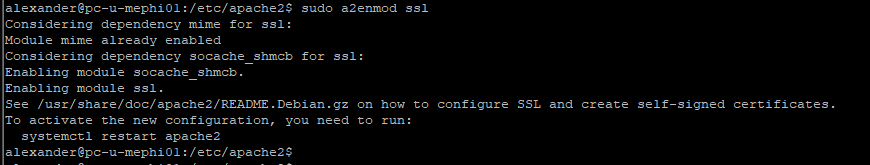
## Шаг 5: Настройка SSL в Apache

### Включение модуля SSL

Команда sudo a2enmod ssl активирует модуль SSL для веб-сервера Apache

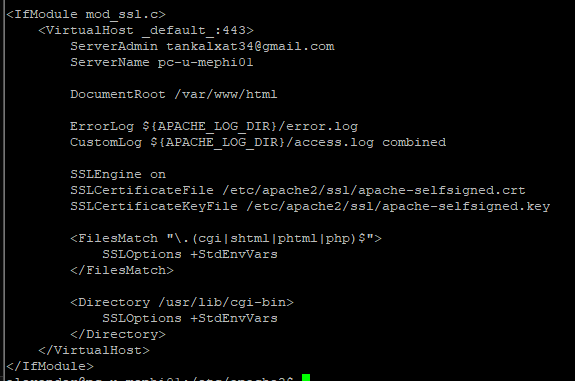


### Включение модуля rewrite для перенаправления HTTP на HTTPS



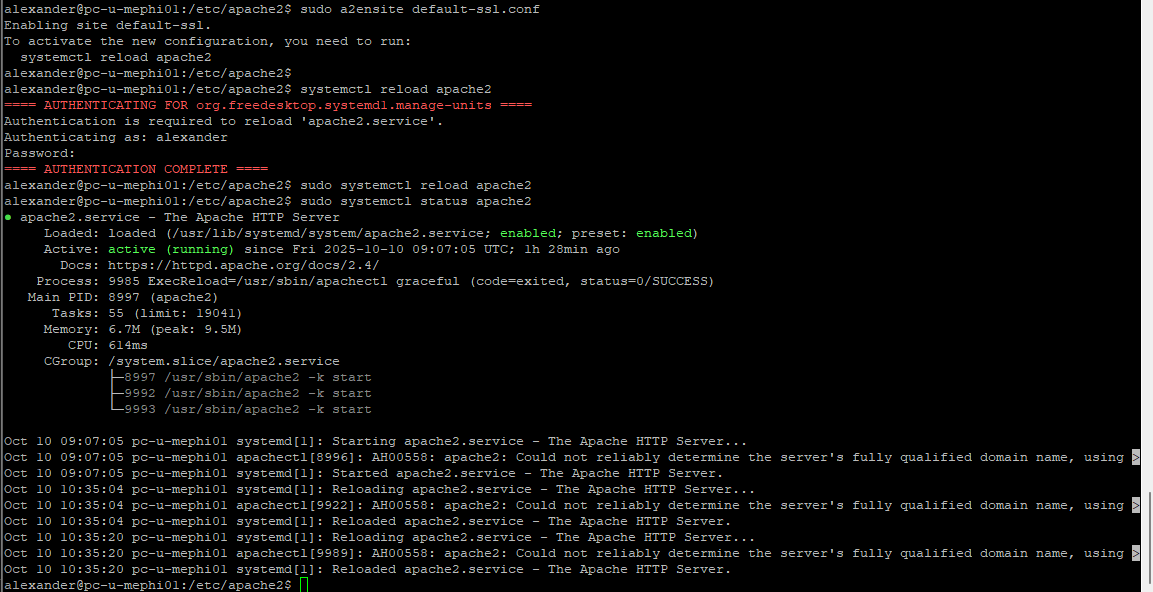
### Создание виртуального хоста для HTTPS

В конец конфигурации «/etc/apache2/sites-available/default-ssl.conf» добавим новую секцию:



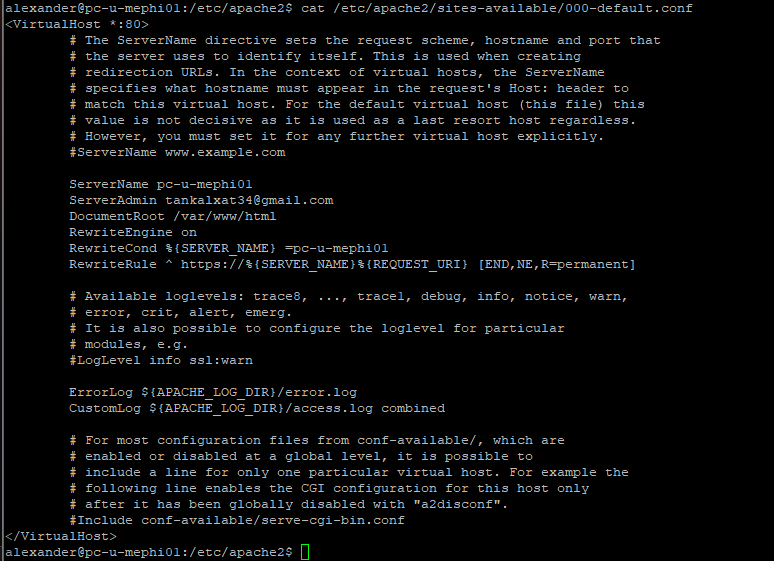
## Шаг 6: Активация SSL сайта и перенаправление HTTP на HTTPS

### Активация SSL виртуального хоста



### Настройка перенаправления HTTP на HTTPS

В конфигурацию «/etc/apache2/sites-available/000-default.conf» после директивы DocumentRoot добавим RewriteEngine, RewriteCond, RewriteRule

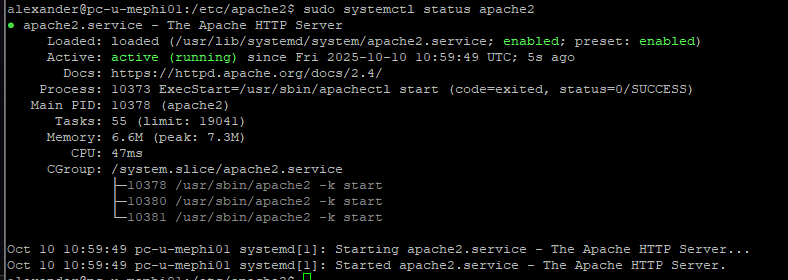


## Шаг 7: Перезапуск Apache

### Проверка конфигурации



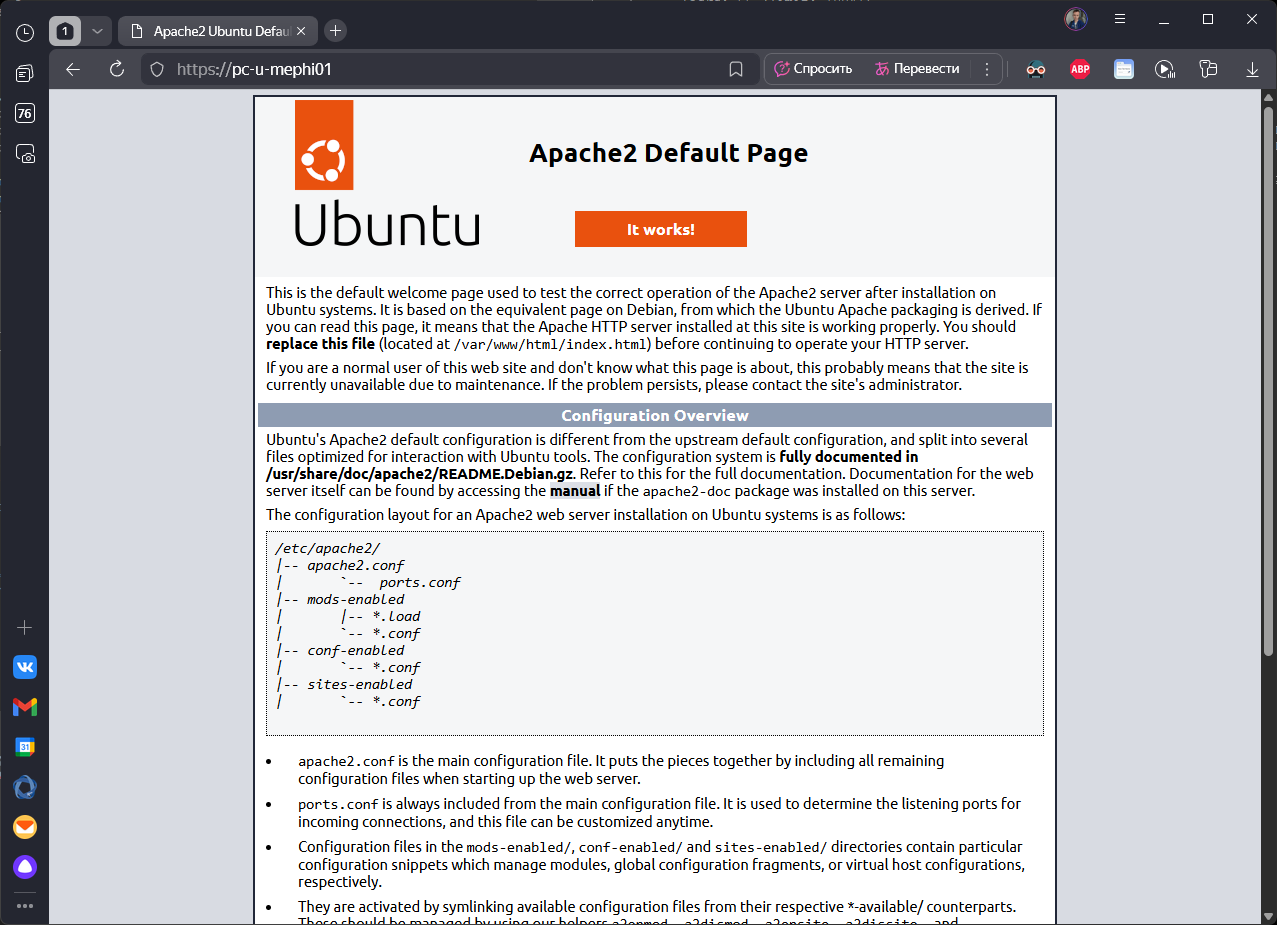
### Перезапуск Apache



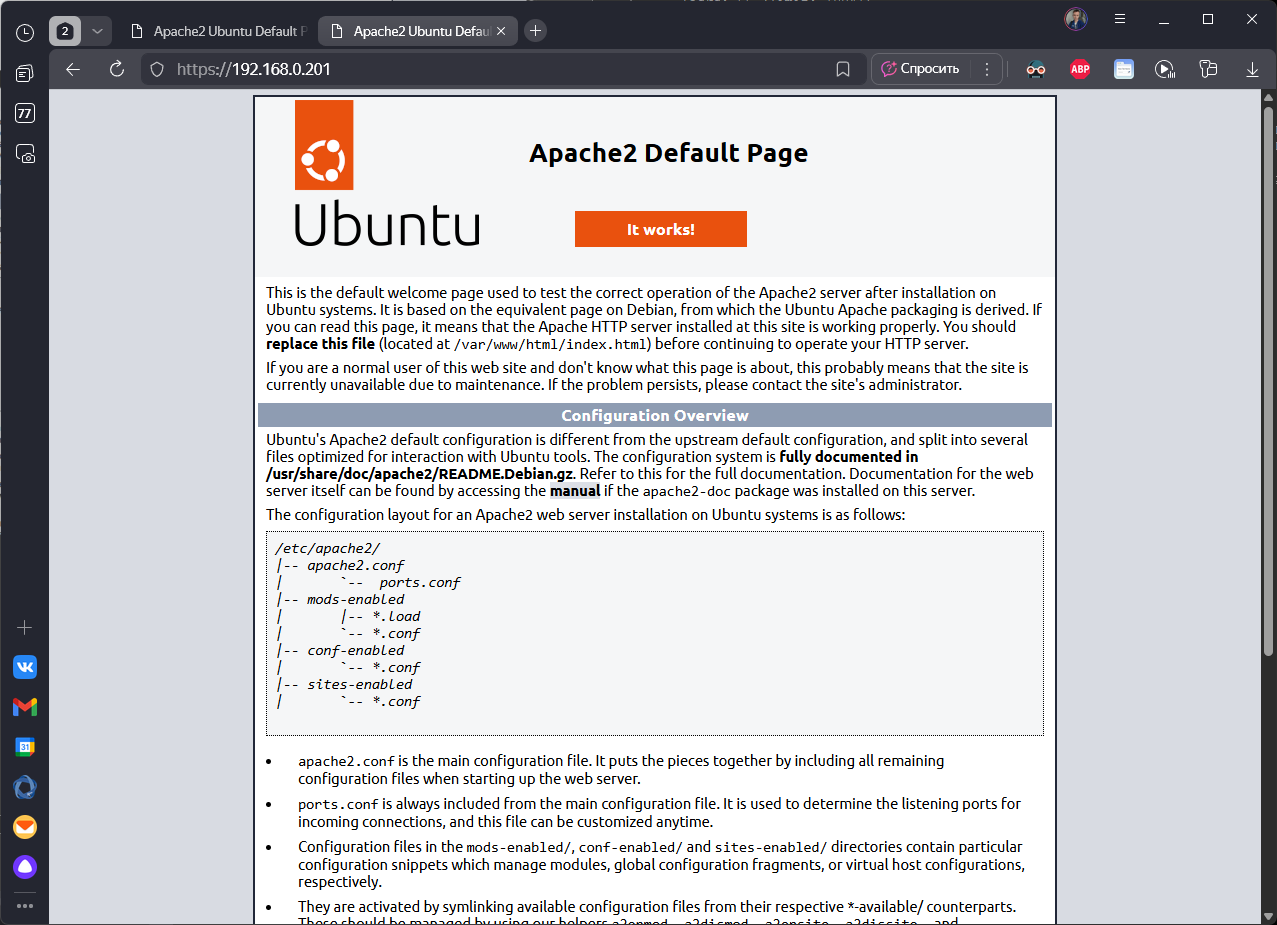
### Проверка работы HTTPS

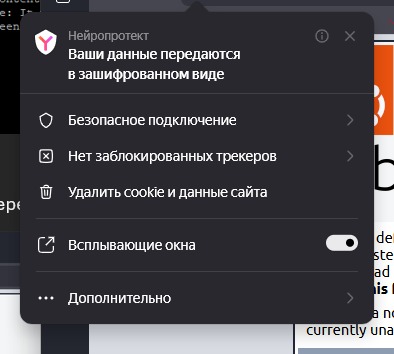


Проверка работы в веб-браузере. Видим что происходит успешное подключение по https.

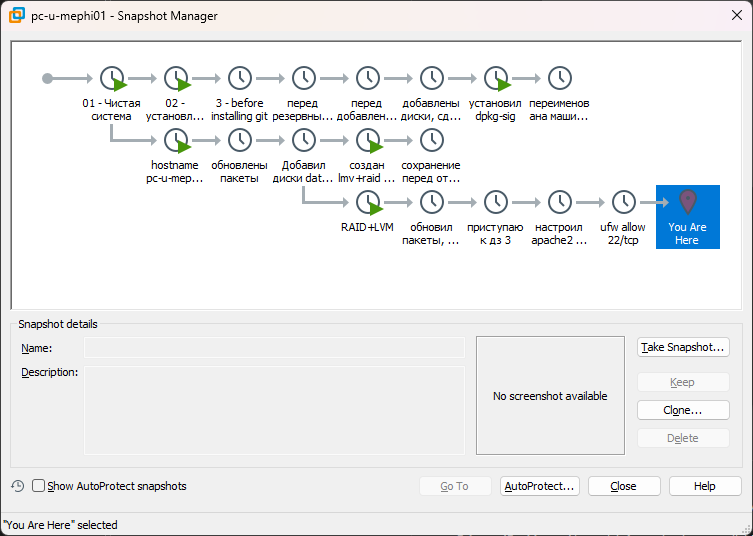


Проверим доступ по ip адресу:



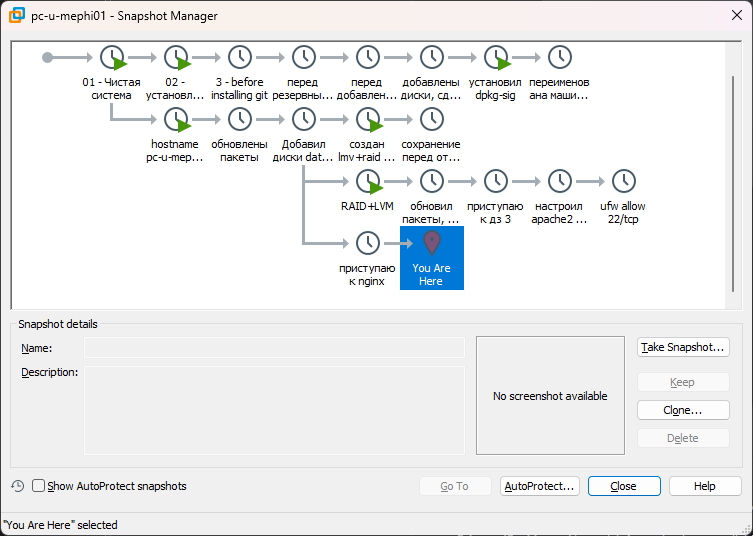


Настройка Apache успешно выполнена. Получившиеся снапшоты виртуальной машины приведены ниже:



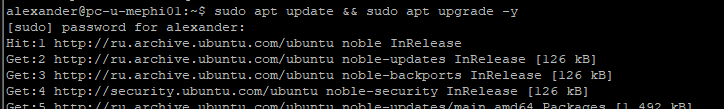
# Часть 2: Настройка Nginx с прокси и ModSecurity

Для выполнения задания откатимся на снапшот «Добавил диски dat…», чтобы не возникло конфликтов при работе Apache и Nginx.

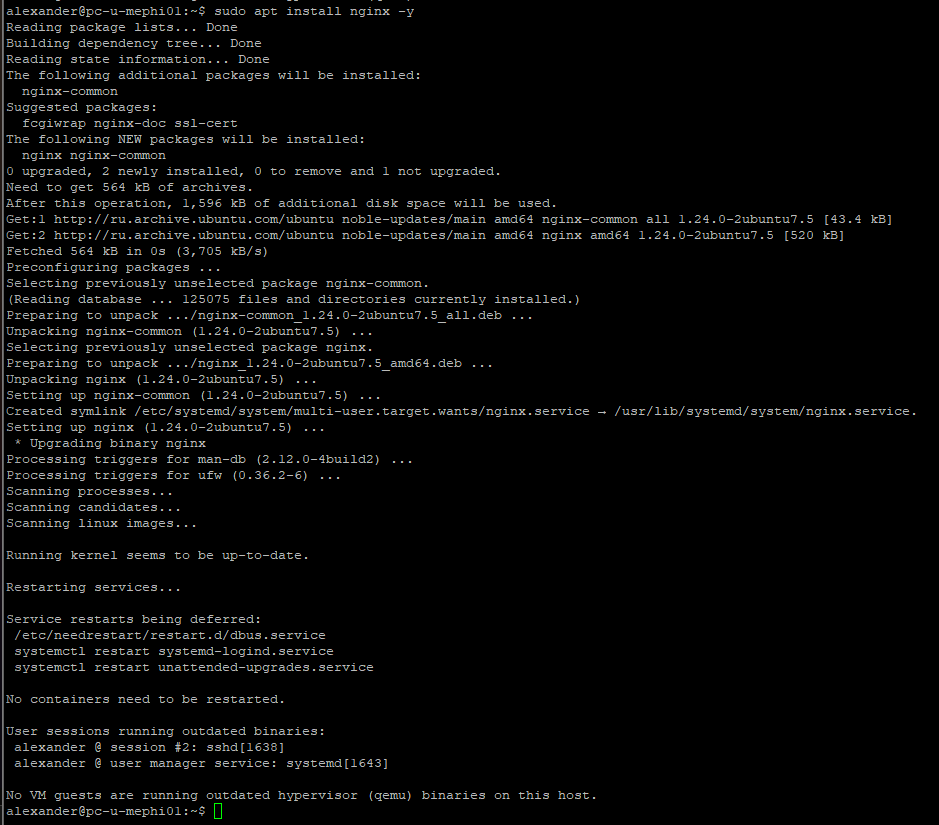


## Шаг 1: Установка Nginx

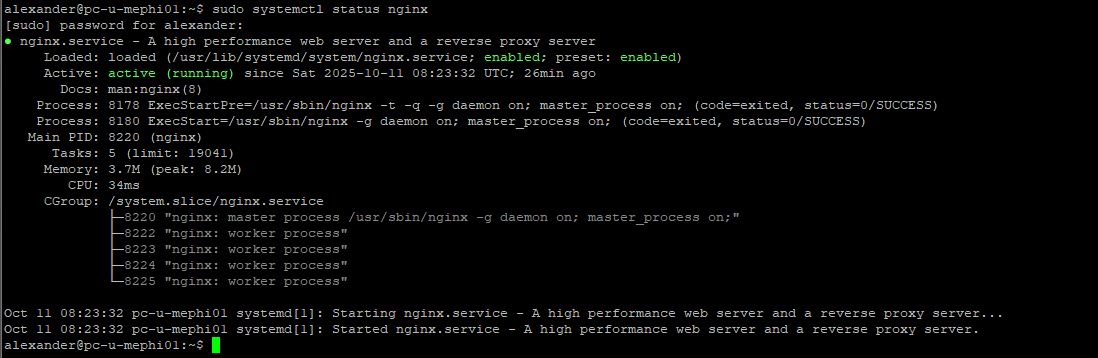
### Обновление пакетов



### Установка Nginx



### Проверка статуса службы



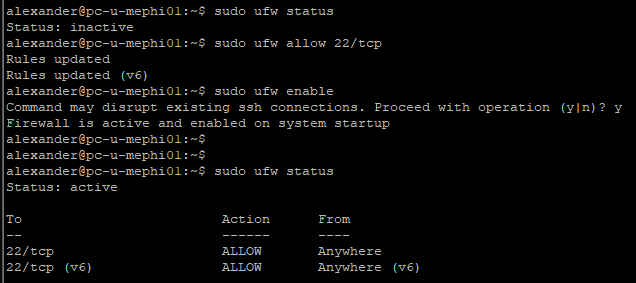
### Включение автозапуска



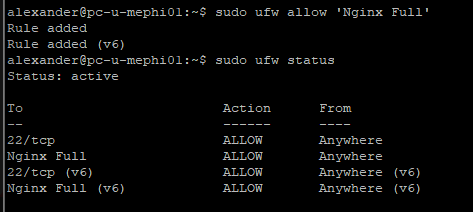
## Шаг 2: Настройка брандмауэра для Nginx

### Разрешение подключения по ssh

Включим брандмауэр, разрешим подключения по ssh

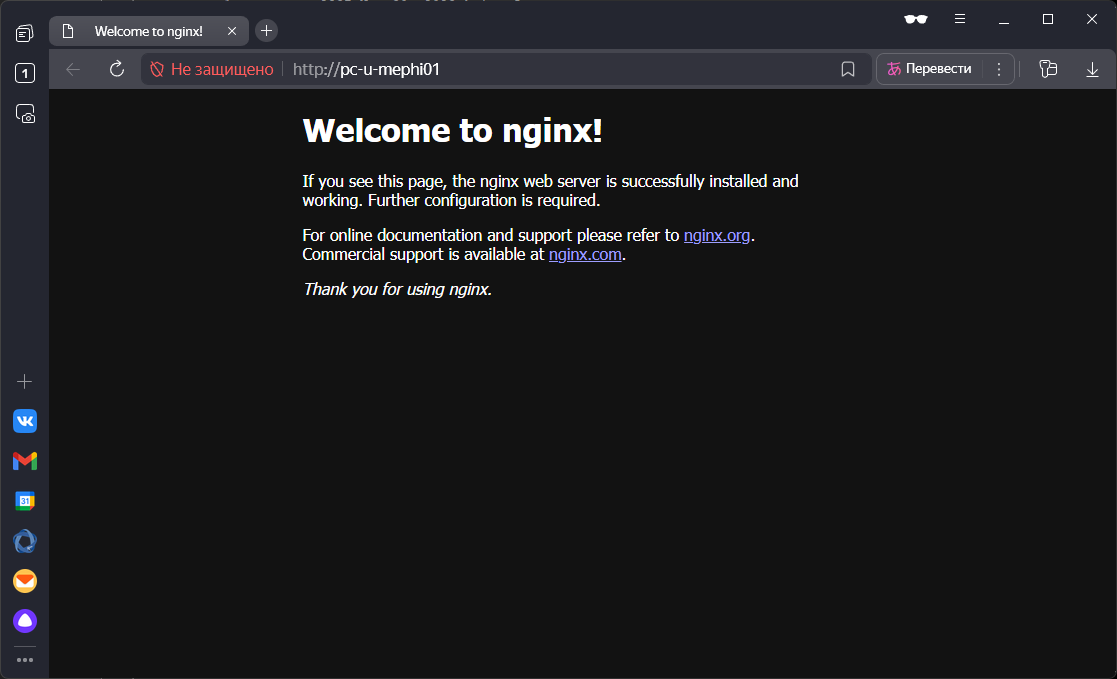


### Разрешение Nginx трафика & проверка статуса

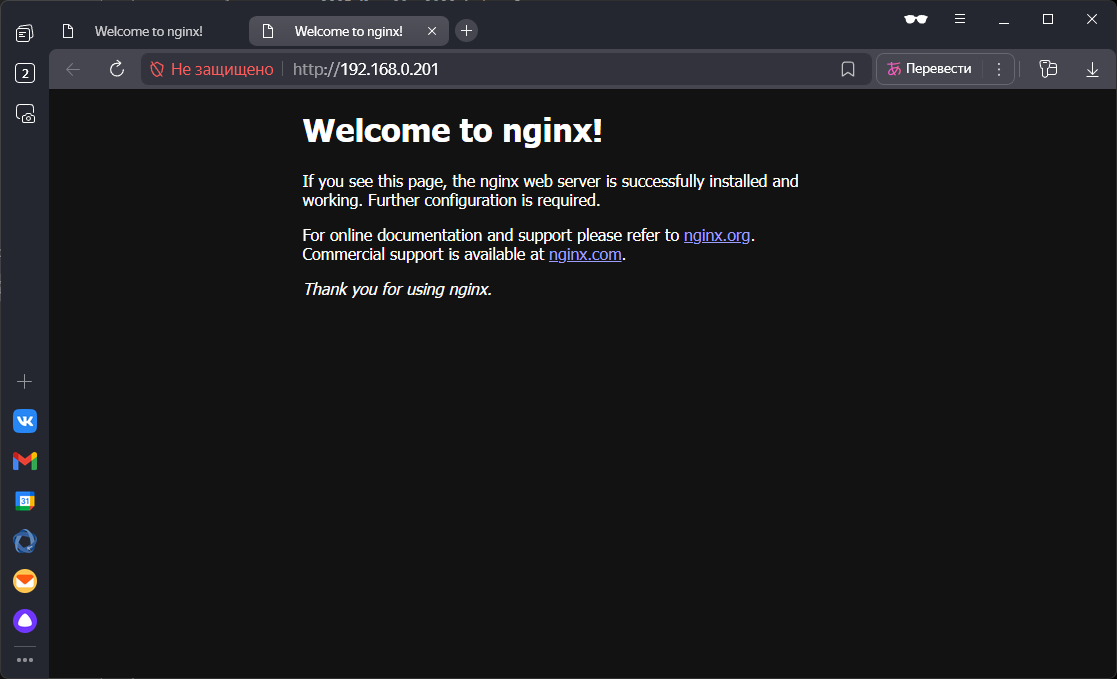


### Проверка работы Nginx

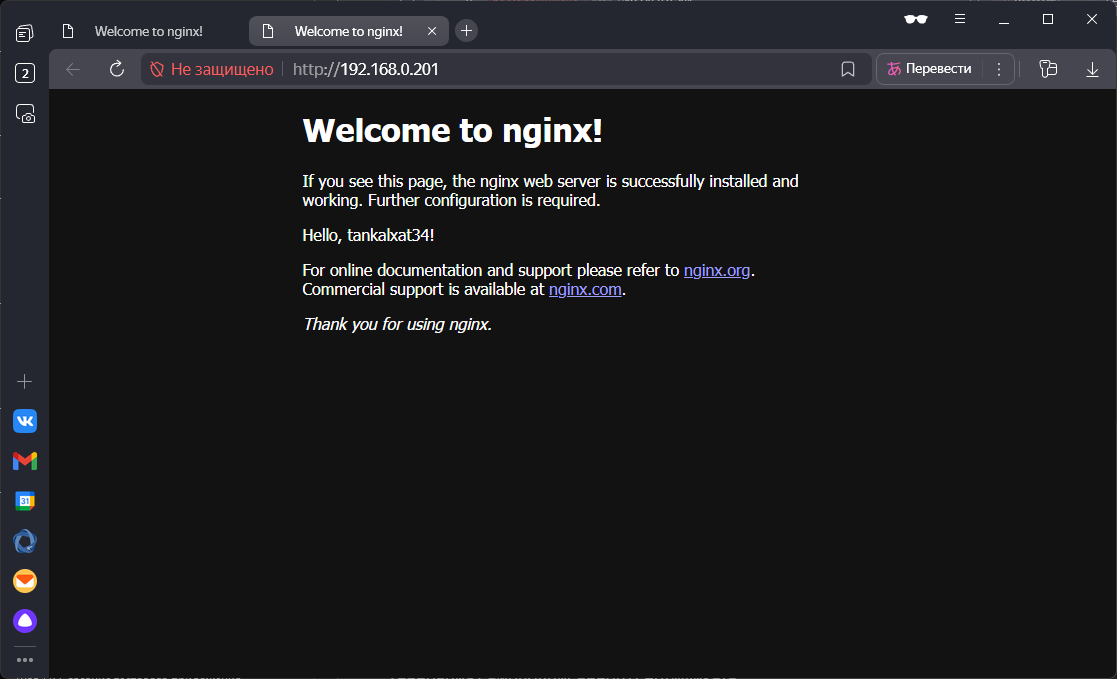
Доступ по hostname из файла hosts:



Доступ по ip адресу:

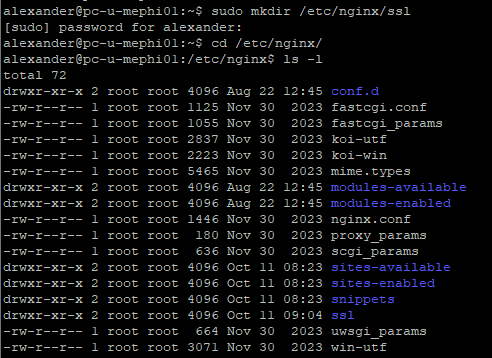


При желании, можем отредактировать html страницу в /var/www/html:



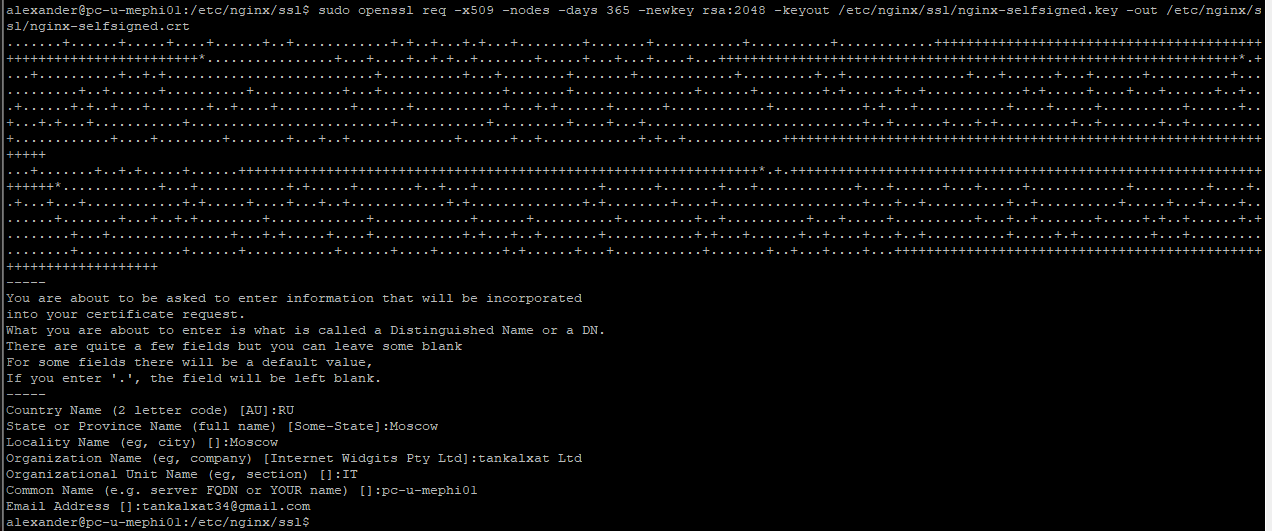
## Шаг 3: Создание SSL-сертификата для Nginx

### Создание директории для сертификатов Nginx



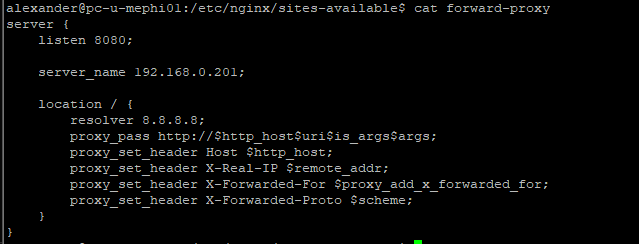
### Генерация самоподписанного сертификата

Аналогичным [способом](#_Генерация_самоподписанного_сертифик) сгенерируем самоподписанный сертификат с помощью утилиты openssl

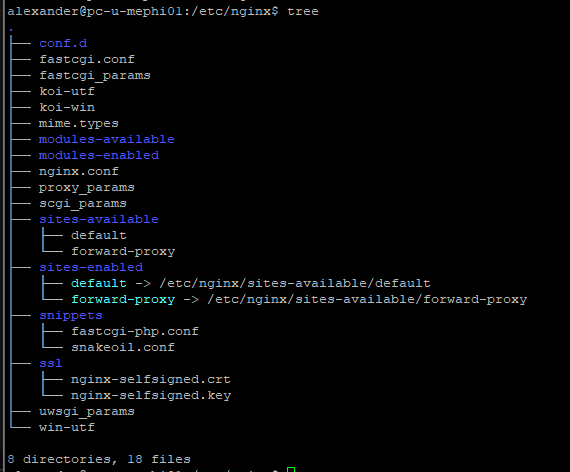


## Шаг 4: Настройка прямого прокси в Nginx

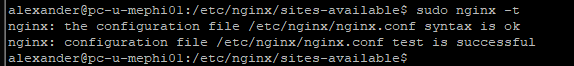
### Создадим конфигурацию прямого прокси



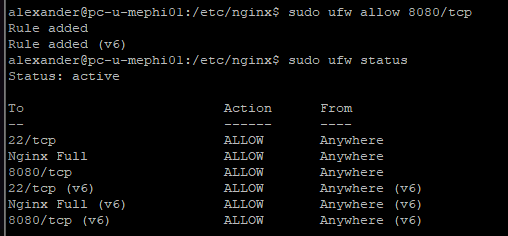
### Сделаем символическую ссылку командой `sudo ln -s /etc/nginx/sites-available/forward-proxy /etc/nginx/sites-enabled/`



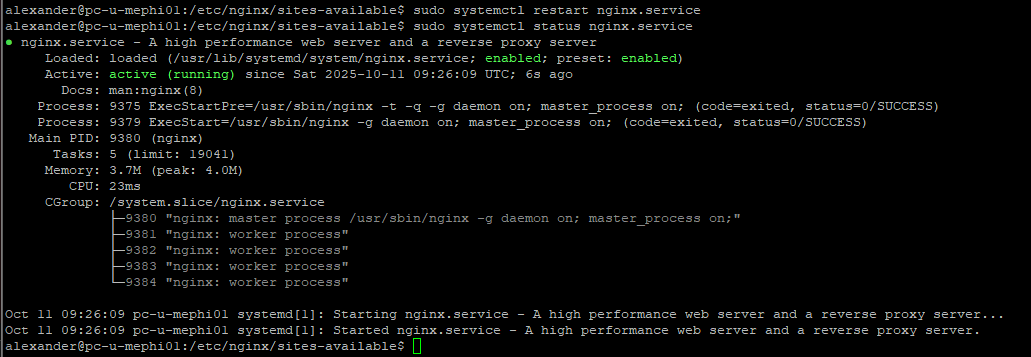
### Проверим конфигурацию на корректность:



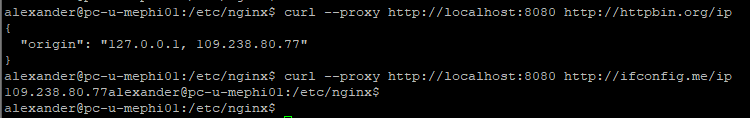
### Добавим разрешение в брандмауэр:



Перезапустим nginx:



Проверим корректность работы:



Видим, что запрашиваемые ресурсы с локальной машины доступны и возвращают ответ.

Проверим работу прямого прокси с помощью десктоп версии клиента Telegram на локальной машине на Windows.

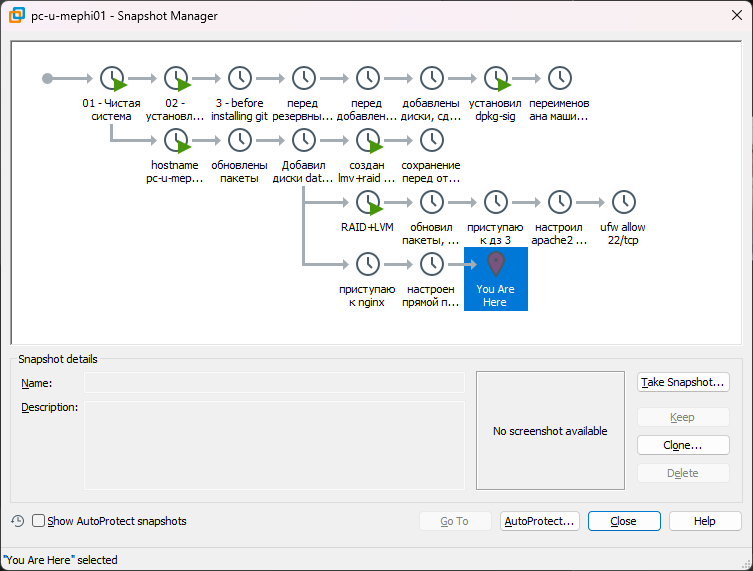
Настройки → Продвинутые настройки → Тип соединения – выбрать «Использовать собственный прокси». Добавим прокси с настройками, как показано ниже. При успешности подключения Telegram сообщит об этом. Все чаты будут доступны через прокси сервер.

|  |  |
| --- | --- |
|  |  |

Прямой прокси работает, Telegram дает ответ.

## Шаг 5: Настройка обратного прокси с HTTPS

Перед настройкой обратного прокси выключим машину и сделаем ее снапшот:



### Создание конфигурации обратного прокси

sudo nano /etc/nginx/sites-available/reverse-proxy

Добавьте конфигурацию:

HTTP перенаправление на HTTPS

server {

listen 80;

server\_name your-server-ip-or-domain;

return 301 https://$server\_name$request\_uri;

}

HTTPS сервер с обратным прокси

server {

listen 443 ssl;

server\_name your-server-ip-or-domain;

ssl\_certificate /etc/nginx/ssl/nginx-selfsigned.crt;

ssl\_certificate\_key /etc/nginx/ssl/nginx-selfsigned.key;

Безопасность SSL

ssl\_protocols TLSv1.2 TLSv1.3;

ssl\_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384;

ssl\_prefer\_server\_ciphers off;

location / {

Проксирование на другой сервер (например, Apache на порту 8080)

proxy\_pass http://localhost:8080;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

Дополнительные настройки безопасности

proxy\_hide\_header X-Powered-By;

add\_header X-Content-Type-Options nosniff;

add\_header X-Frame-Options DENY;

add\_header X-XSS-Protection "1; mode=block";

}

Статические файлы

location /static/ {

alias /var/www/html/static/;

expires 1y;

add\_header Cache-Control "public, immutable";

}

}

## Шаг 6: Активация конфигураций Nginx

Активация сайтов

sudo ln -s /etc/nginx/sites-available/forward-proxy /etc/nginx/sites-enabled/

sudo ln -s /etc/nginx/sites-available/reverse-proxy /etc/nginx/sites-enabled/

Удаление дефолтной конфигурации

sudo rm /etc/nginx/sites-enabled/default

Проверка конфигурации

sudo nginx -t

Перезагрузка Nginx

sudo systemctl reload nginx

## Шаг 7: Установка и настройка ModSecurity

Установка ModSecurity для Nginx

sudo apt install libnginx-mod-http-modsecurity -y

Проверка установки модуля

nginx -V 2>&1 | grep -o with-http\_modsecurity\_module

Скачивание правил OWASP ModSecurity Core Rule Set (CRS)

sudo apt install git -y

sudo git clone https://github.com/coreruleset/coreruleset /etc/nginx/modsecurity/coreruleset

Создание конфигурации ModSecurity

sudo mkdir -p /etc/nginx/modsecurity

sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/nginx/modsecurity/modsecurity.conf

## Шаг 8: Настройка ModSecurity

Редактирование основного конфигурационного файла

sudo nano /etc/nginx/modsecurity/modsecurity.conf

Измените следующие параметры:

SecRuleEngine On

SecAuditEngine RelevantOnly

SecAuditLog /var/log/nginx/modsec\_audit.log

SecDebugLog /var/log/nginx/modsec\_debug.log

SecAuditLogType Serial

SecAuditLogStorageDir /var/log/nginx/modsec\_audit/

## Шаг 9: Настройка правил CRS

Копирование необходимых файлов правил

sudo cp /etc/nginx/modsecurity/coreruleset/crs-setup.conf.example /etc/nginx/modsecurity/coreruleset/crs-setup.conf

sudo cp /etc/nginx/modsecurity/coreruleset/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf /etc/nginx/modsecurity/coreruleset/

Создание конфигурации для включения ModSecurity в Nginx

sudo nano /etc/nginx/conf.d/modsecurity.conf

Добавьте содержимое:

modsecurity on;

modsecurity\_rules\_file /etc/nginx/modsecurity/modsecurity.conf;

Включение основных правил CRS

Include /etc/nginx/modsecurity/coreruleset/crs-setup.conf

Include /etc/nginx/modsecurity/coreruleset/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf

## Шаг 10: Настройка виртуального хоста с ModSecurity

Обновите конфигурацию обратного прокси:

sudo nano /etc/nginx/sites-available/reverse-proxy

Добавьте в server блок:

Включение ModSecurity

modsecurity on;

modsecurity\_rules\_file /etc/nginx/conf.d/modsecurity.conf;

Логи ModSecurity

location /modsec\_logs {

alias /var/log/nginx/modsec\_audit;

autoindex on;

}

## Шаг 11: Создание тестового приложения

Создание тестовой PHP страницы для демонстрации SQLi

sudo apt install php-fpm -y

sudo mkdir -p /var/www/html/test

sudo nano /var/www/html/test/index.php

Добавьте простой PHP код:

<?php

// Простая форма для тестирования

if ($\_POST) {

$input = $\_POST['search'];

echo "Вы ввели: " . htmlspecialchars($input);

}

?>

<form method="post">

<input type="text" name="search" placeholder="Введите поисковый запрос">

<input type="submit" value="Поиск">

</form>

## Шаг 12: Демонстрация работы ModSecurity

Тест 1: Проверка блокировки SQL-инъекции

Отправка SQL-инъекции - должен вернуть 403 Forbidden

curl -X POST http://your-server-ip/test/index.php \

-d "search=' OR 1=1--" \

-H "Content-Type: application/x-www-form-urlencoded"

Или с использованием HTTPS

curl -k -X POST https://your-server-ip/test/index.php \

-d "search=' UNION SELECT username, password FROM users--" \

-H "Content-Type: application/x-www-form-urlencoded"

Тест 2: Проверка нормального запроса

Нормальный запрос - должен вернуть 200 OK

curl -X POST http://your-server-ip/test/index.php \

-d "search=normal query" \

-H "Content-Type: application/x-www-form-urlencoded"

## Шаг 13: Отключение правила и проверка

Найти ID правила SQL-инъекции:

Просмотр логов ModSecurity

sudo tail -f /var/log/nginx/modsec\_audit.log

Или поиск в файлах правил

sudo grep -r "SQL" /etc/nginx/modsecurity/coreruleset/rules/

Закомментировать правило:

Редактирование файла правил SQL-инъекций

sudo nano /etc/nginx/modsecurity/coreruleset/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf

Найдите правило, которое блокирует запрос (обычно по ID, например 942100)

и закомментируйте его, добавив в начале строки:

SecRule REQUEST\_COOKIES|!REQUEST\_COOKIES:/\_\_utm/|REQUEST\_COOKIES\_NAMES|REQUEST\_FILENAME|REQUEST\_HEADERS:User-Agent|REQUEST\_HEADERS:Referer|ARGS\_NAMES|ARGS|XML:/\* "@detectSQLi" \

"id:942100,...

Перезагрузка Nginx и повторная проверка:

Перезагрузка Nginx

sudo systemctl reload nginx

Повторная отправка SQL-инъекции - теперь должен вернуть 200 OK

curl -X POST http://your-server-ip/test/index.php \

-d "search=' OR 1=1--" \

-H "Content-Type: application/x-www-form-urlencoded"

## Шаг 14: Просмотр логов

Просмотр логов Nginx

sudo tail -f /var/log/nginx/access.log

sudo tail -f /var/log/nginx/error.log

Просмотр логов ModSecurity

sudo tail -f /var/log/nginx/modsec\_audit.log

Финальная проверка

Проверка всех служб:

Проверка статуса всех служб

sudo systemctl status apache2

sudo systemctl status nginx

sudo systemctl status php8.1-fpm или ваша версия PHP

Проверка открытых портов

sudo netstat -tulpn | grep -E '(80|443|8080)'

Проверка SSL сертификатов

openssl x509 -in /etc/apache2/ssl/apache-selfsigned.crt -text -noout

openssl x509 -in /etc/nginx/ssl/nginx-selfsigned.crt -text -noout

Этот план предоставляет полное пошаговое руководство по настройке Apache и Nginx с HTTPS, прокси-серверами и ModSecurity для защиты от SQL-инъекций на Ubuntu Server 24.04 LTS.