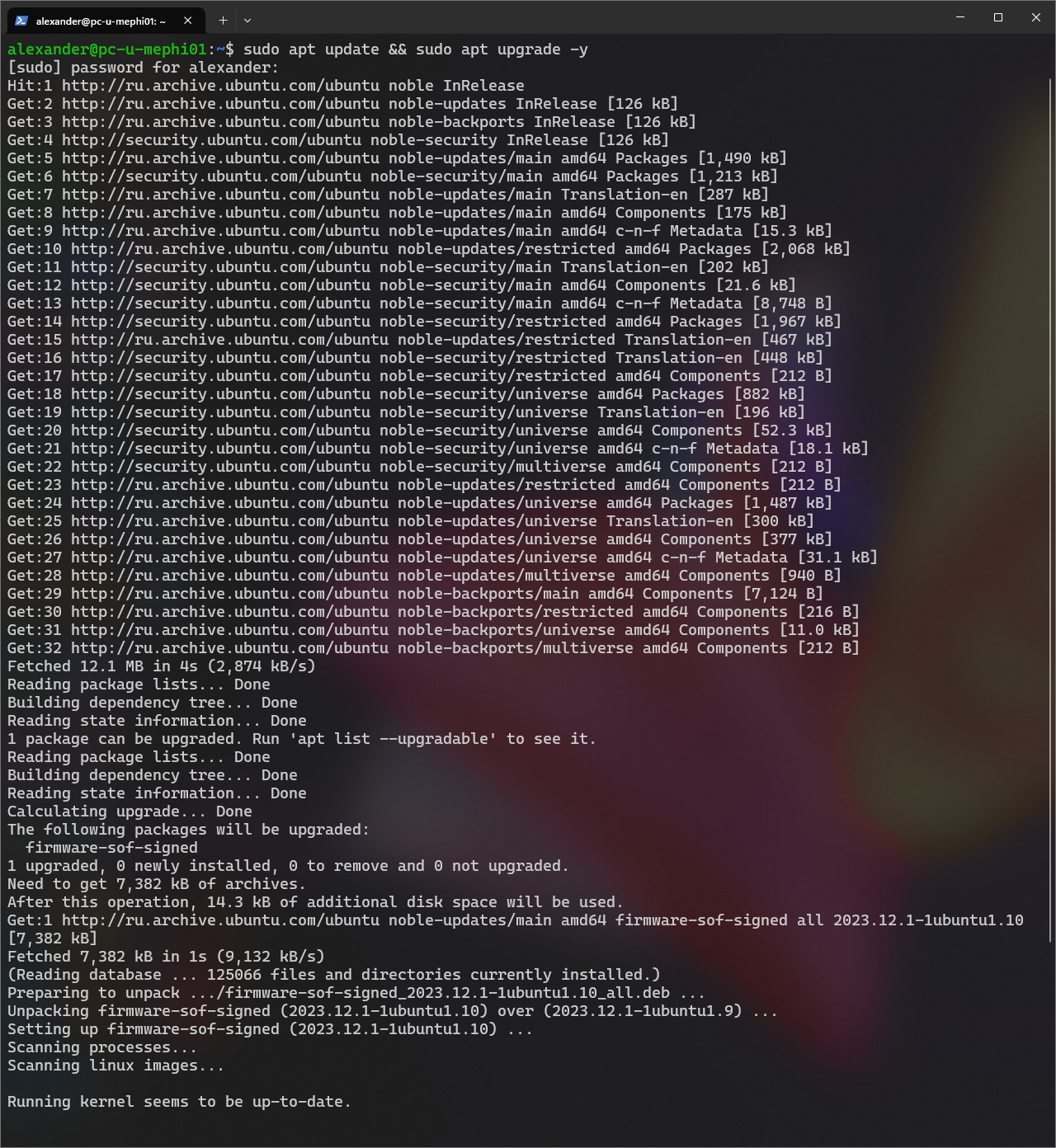
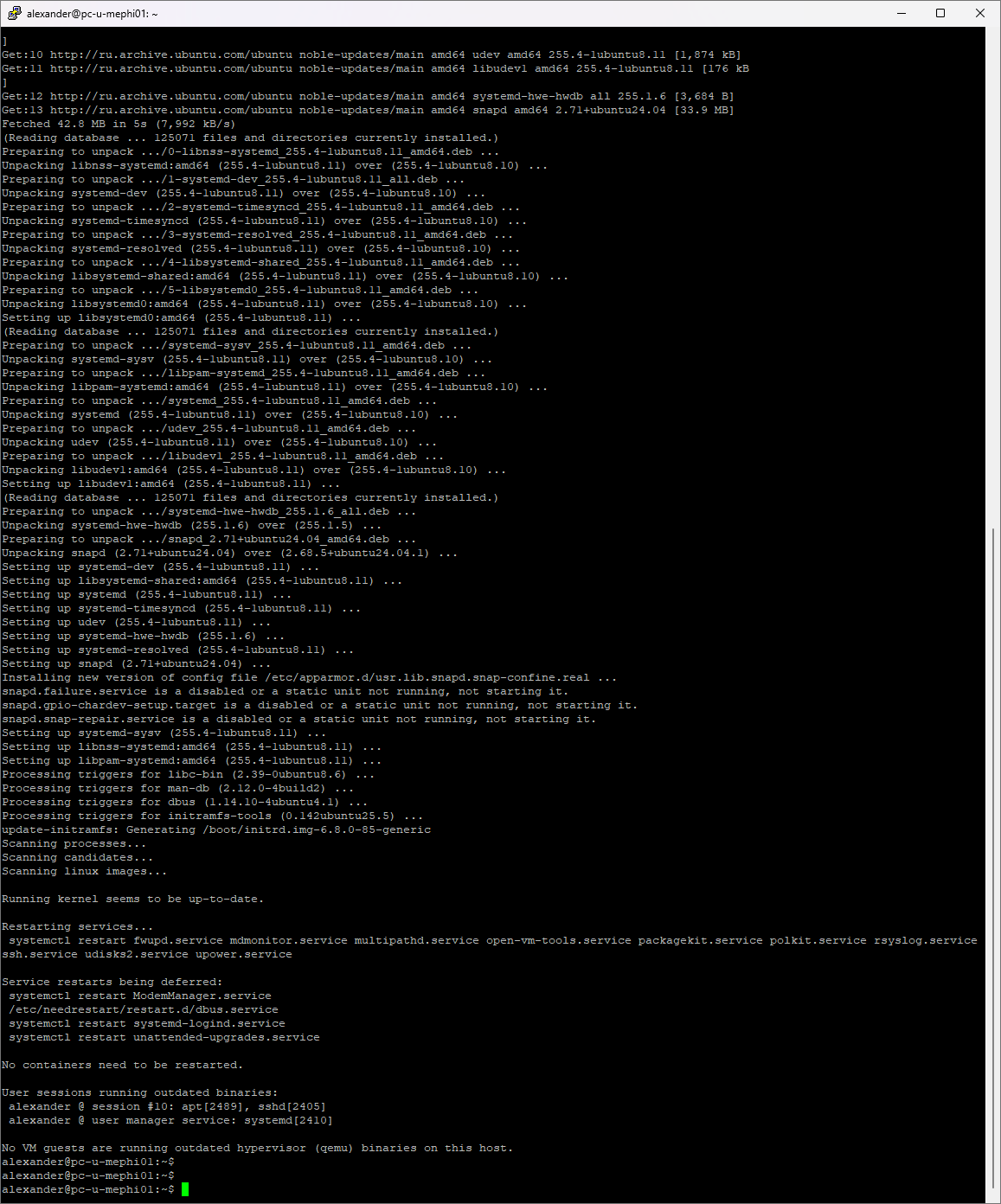
# Часть 1: Настройка Apache с HTTPS

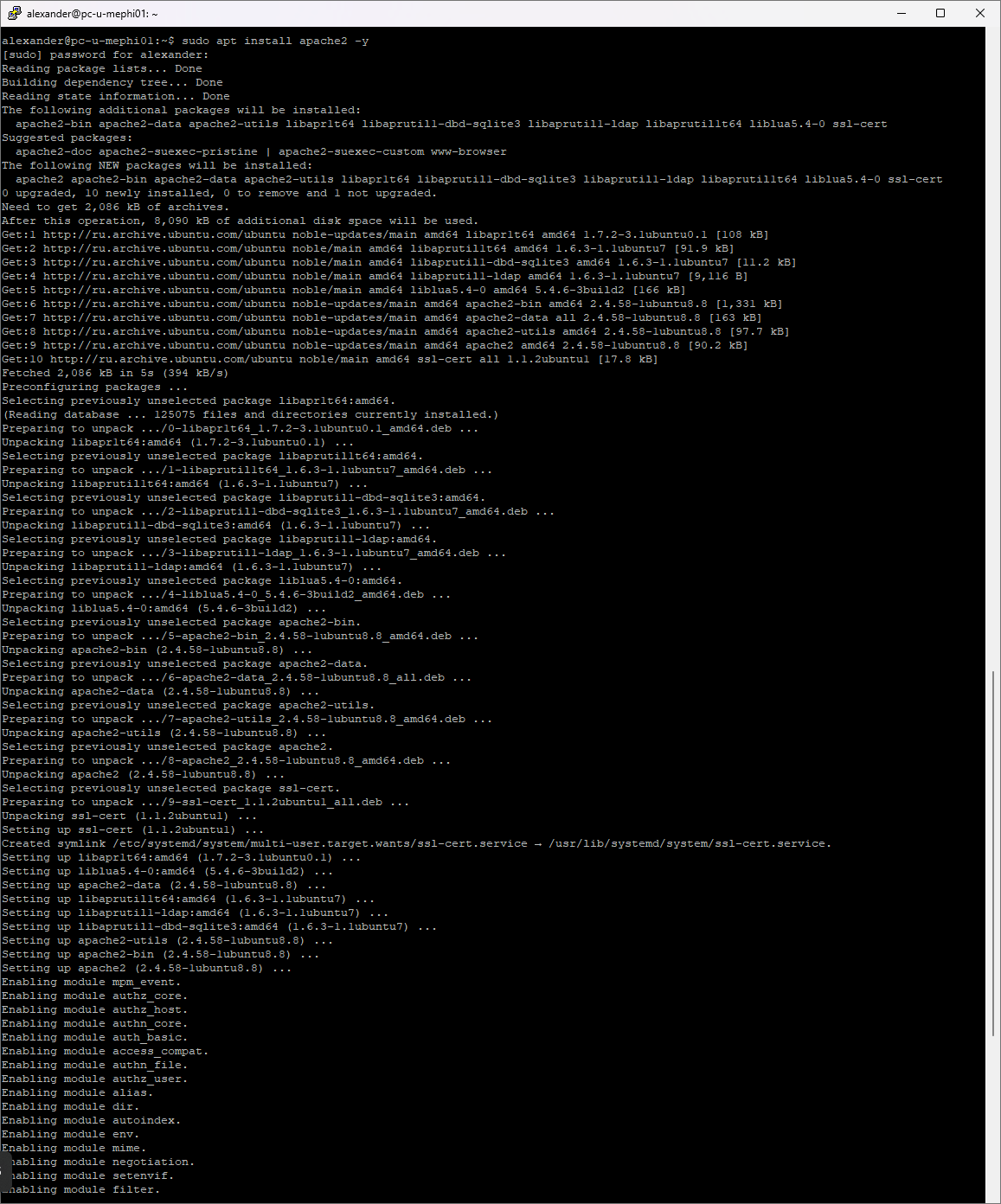
1. Установка Apache

### Обновление пакетов системы



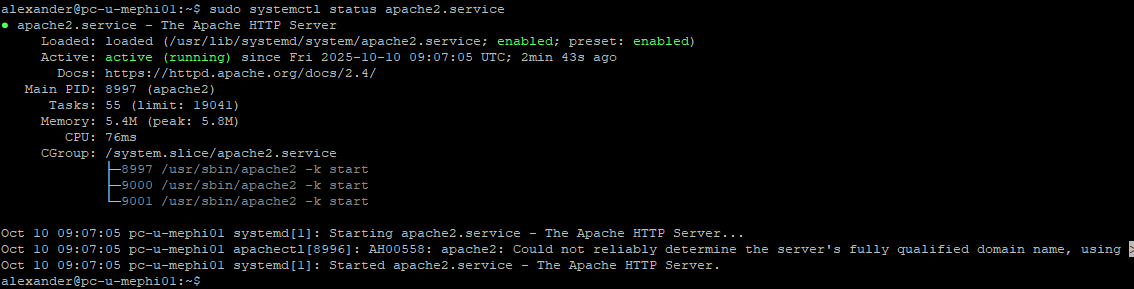


### Установка Apache



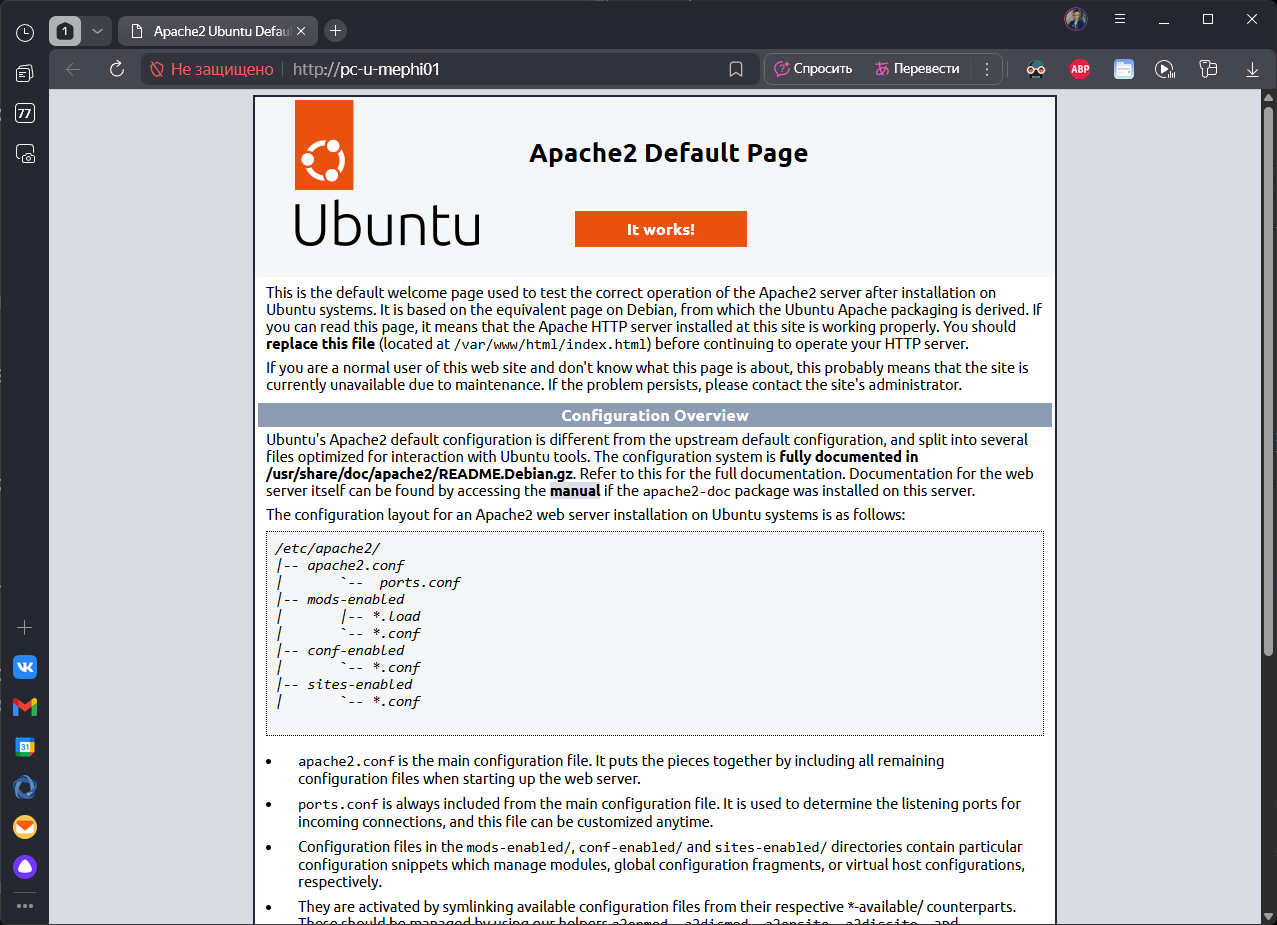
1. Проверка работы Apache

### Проверка статуса службы Apache

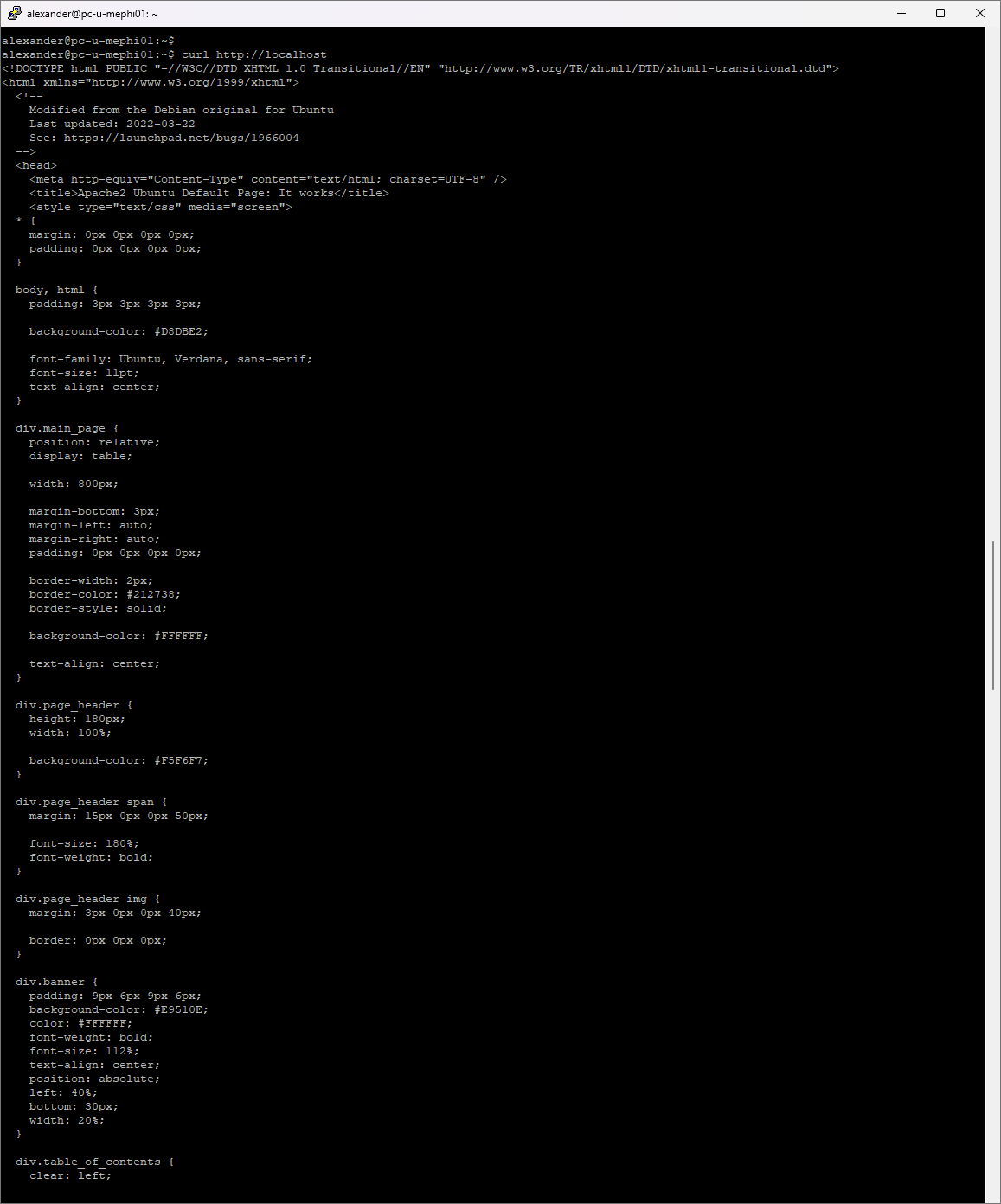


### Проверка работы через браузер или curl

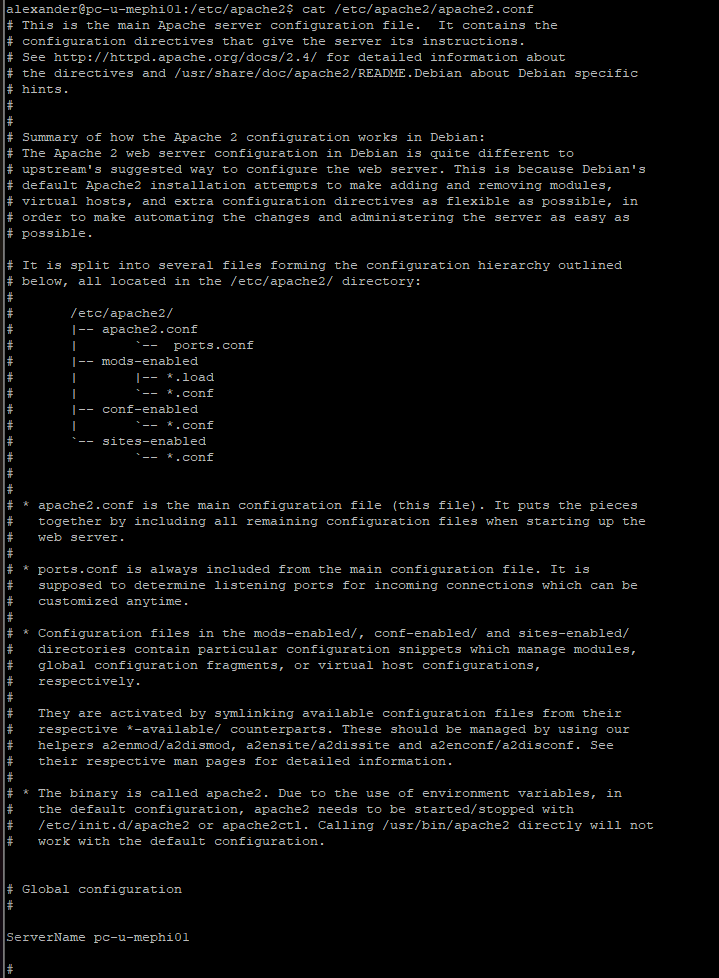
Поскольку ВМ подключена напрямую к роутеру, она доступна в локальной сети по своему статическому IP адресу. Если для данного IP добавить алиас «pc-u-mephi01» в файл C:\Windows\System32\drivers\etc\hosts , то наша виртуальная машина станет доступна по адресу <http://pc-u-mephi01> . Проверим работу Apache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



Проверим работу через утилиту curl .

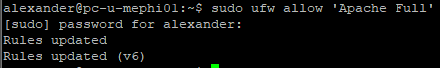


### Добавим ServerName в конфигурацию /etc/apache2/apache2.conf



1. Настройка брандмауэра

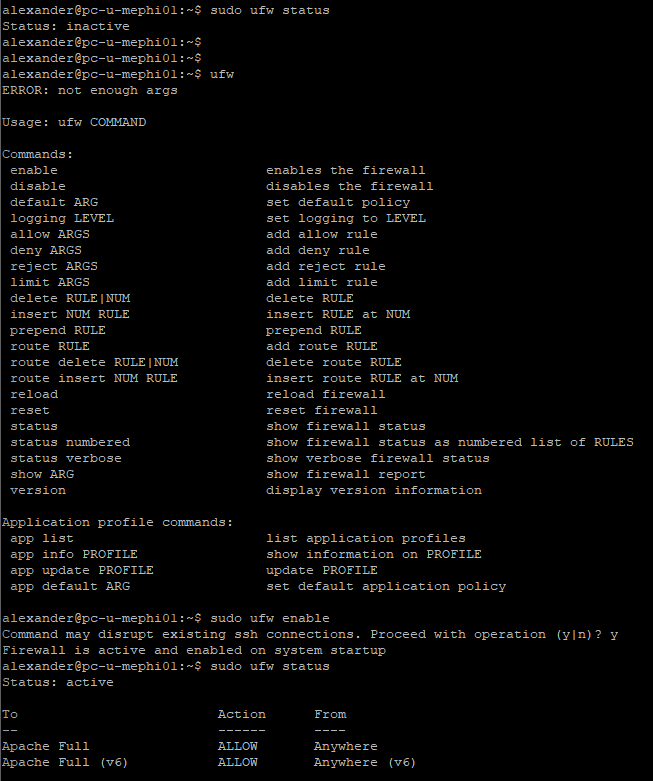
### Разрешение HTTP и HTTPS трафика



### Проверка правил брандмауэра

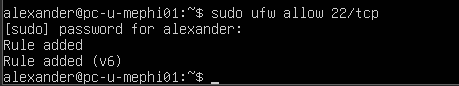


Включим брандмауэр и проверим его статус

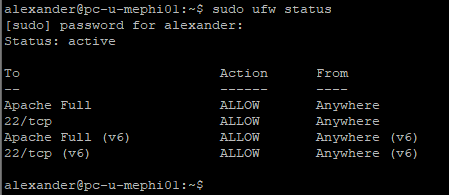


### Разрешение подключения по ssh

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

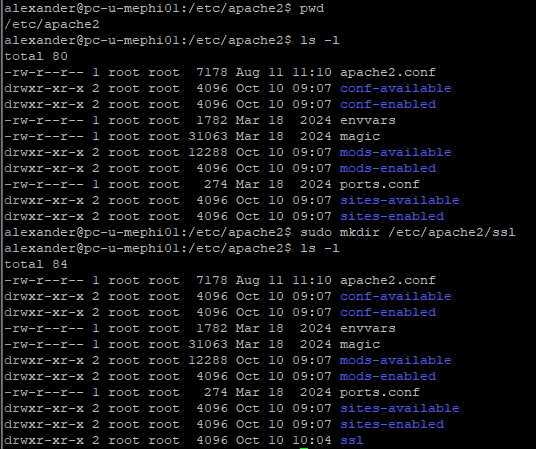


Проверим изменения:



1. Создание самоподписанного SSL-сертификата

### Создание директории для сертификатов

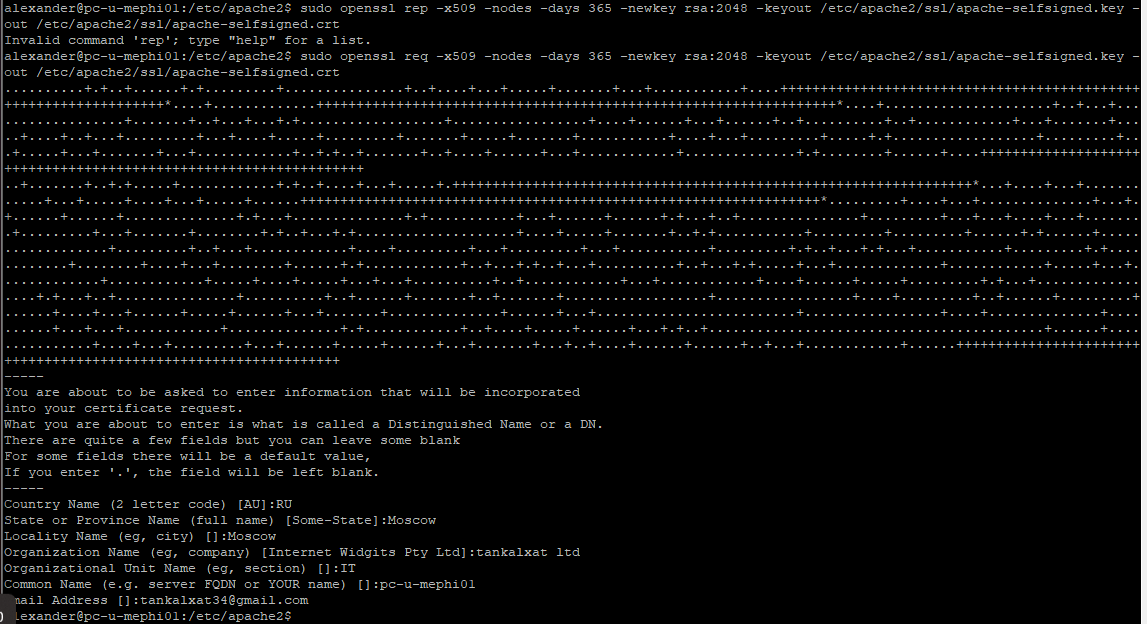


### Генерация самоподписанного сертификата

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:**

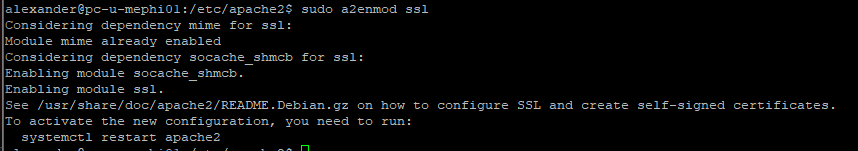
* sudo — запуск с правами администратора
* openssl req — создание запроса на подпись сертификата
* -x509 — создать самоподписанный сертификат
* -nodes — не шифровать приватный ключ
* -days 365 — срок действия сертификата (1 год)
* -newkey rsa:2048 — создать новый RSA ключ длиной 2048 бит
* -keyout —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



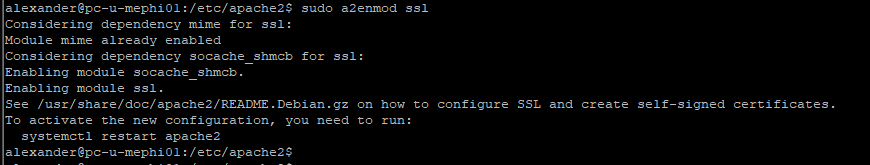
1. Настройка SSL в Apache

### Включение модуля SSL

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

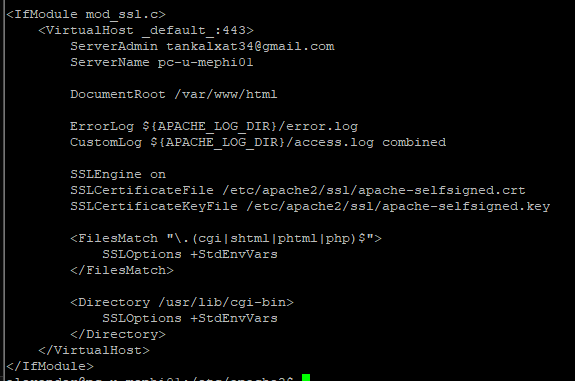


### Включение модуля rewrite для перенаправления HTTP на HTTPS



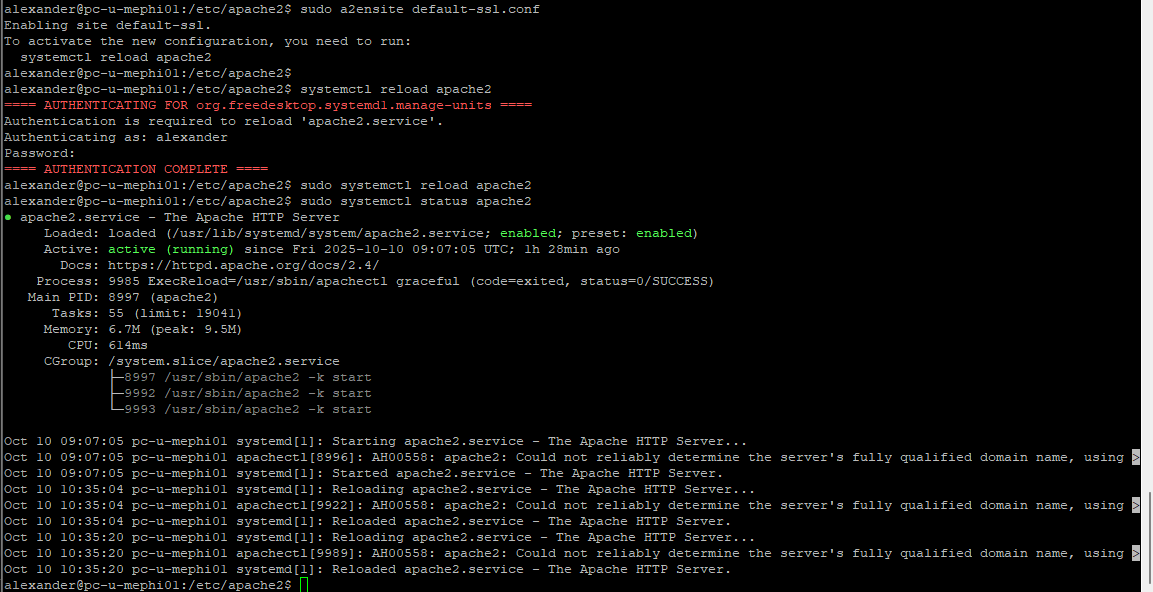
### Создание виртуального хоста для HTTPS

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



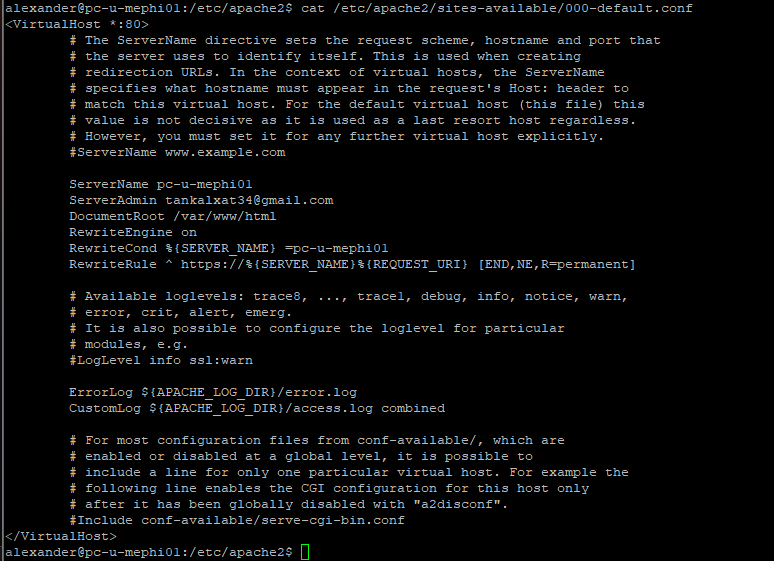
1. Активация SSL сайта и перенаправление HTTP на HTTPS

### 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

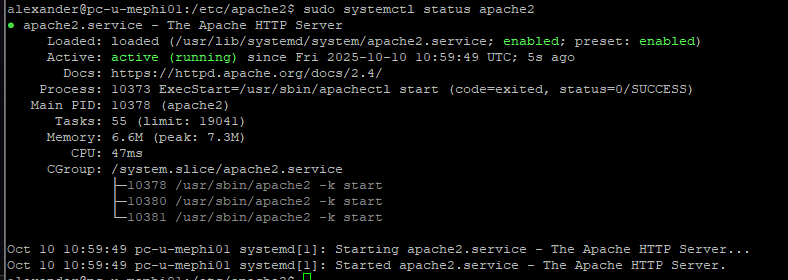


1. Перезапуск Apache

### Проверка конфигурации



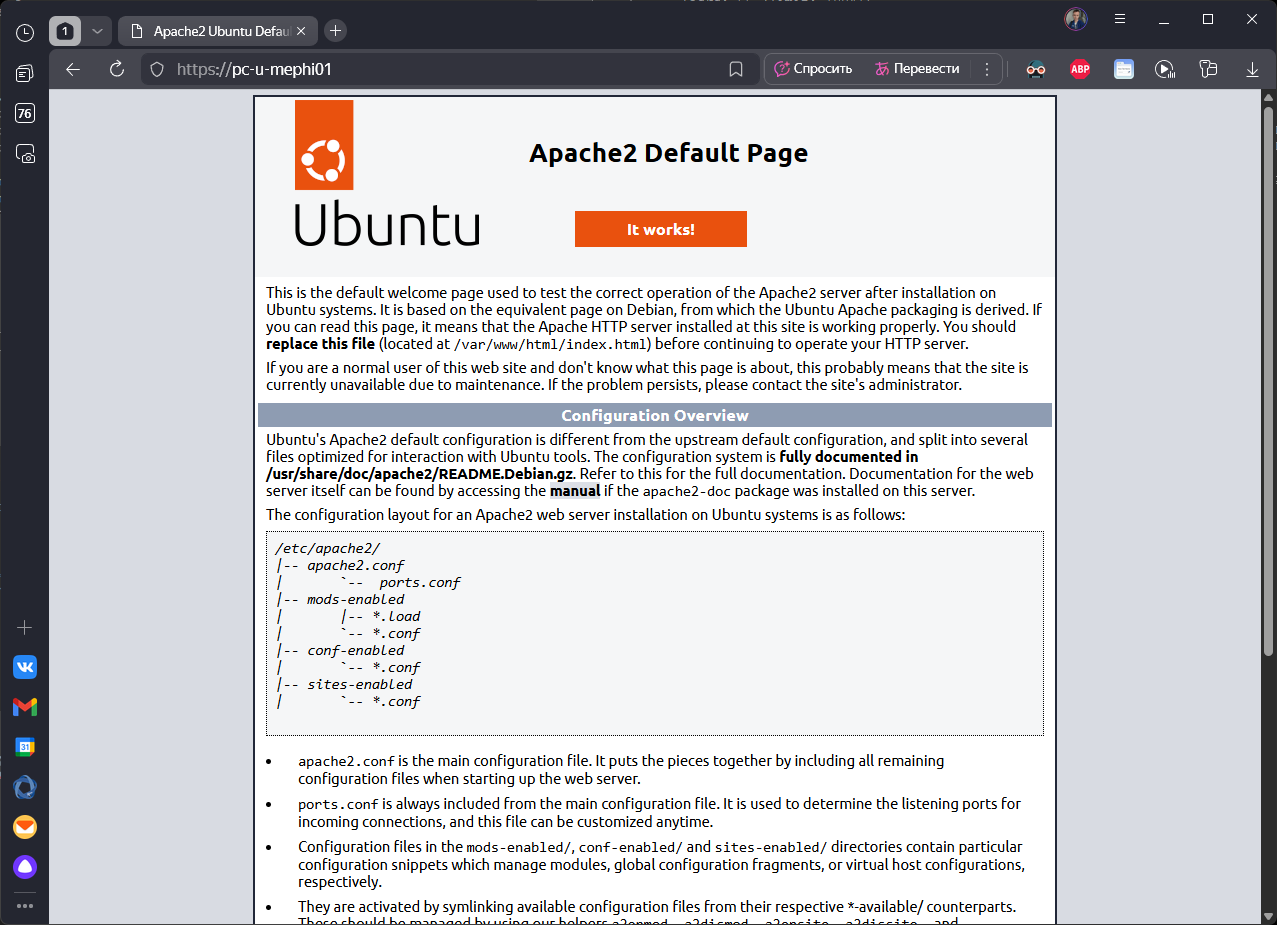
### Перезапуск Apache



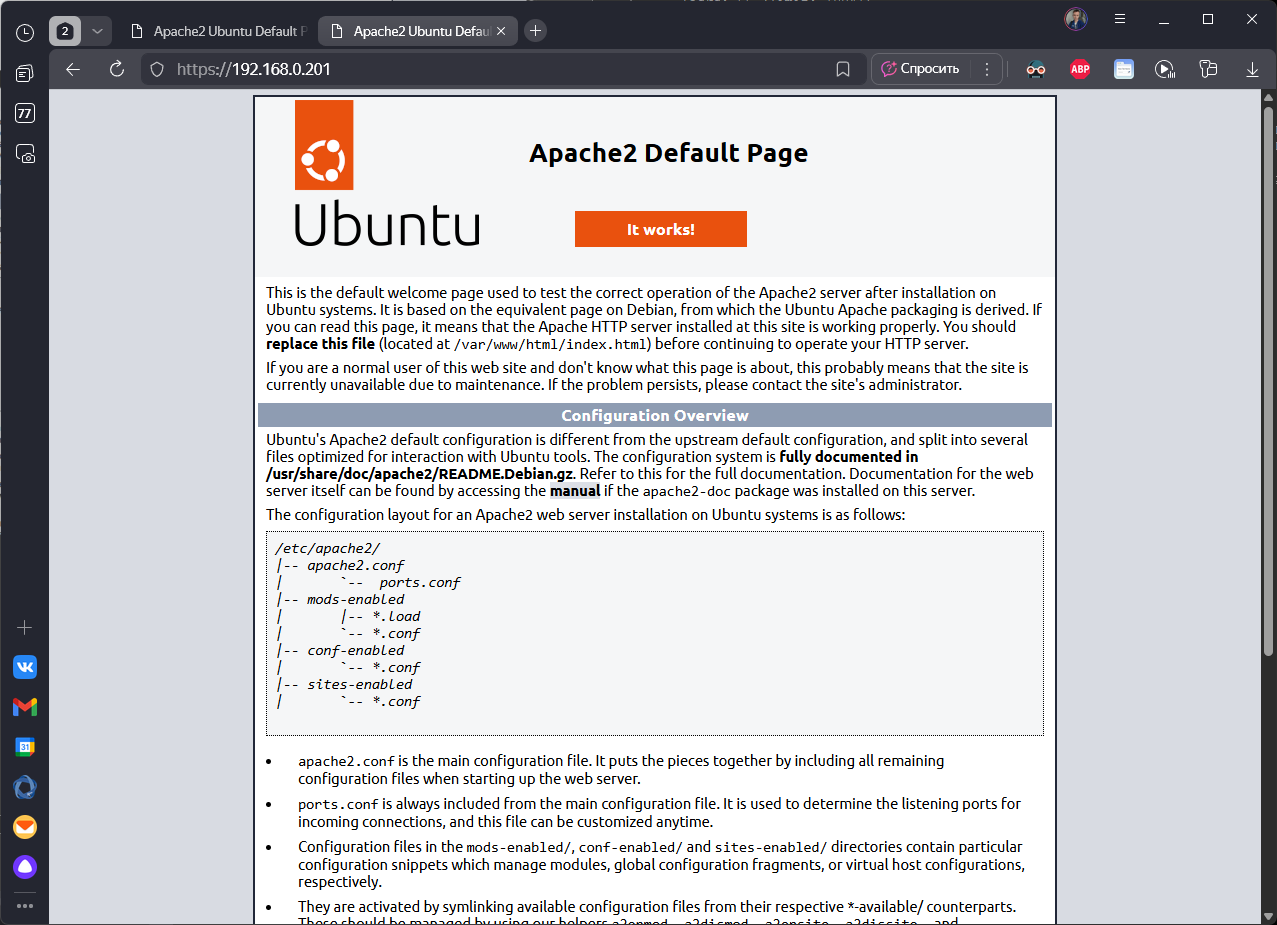
### Проверка работы HTTPS

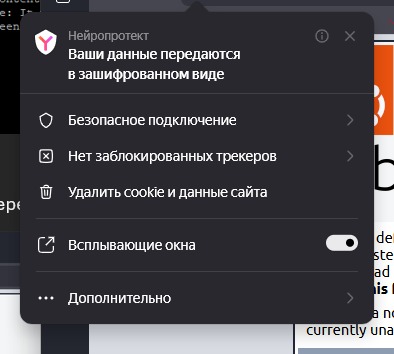


Проверка работы в веб-браузере. Видим что происходит успешное подключение по https.

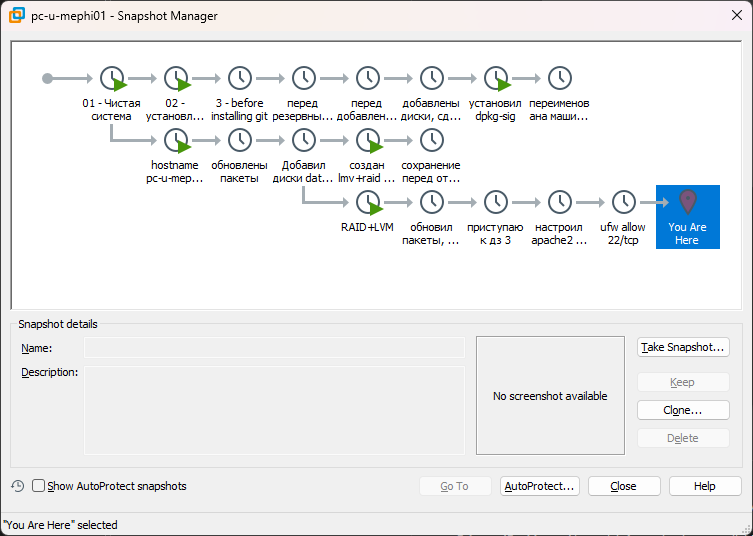


Проверим доступ по ip адресу:



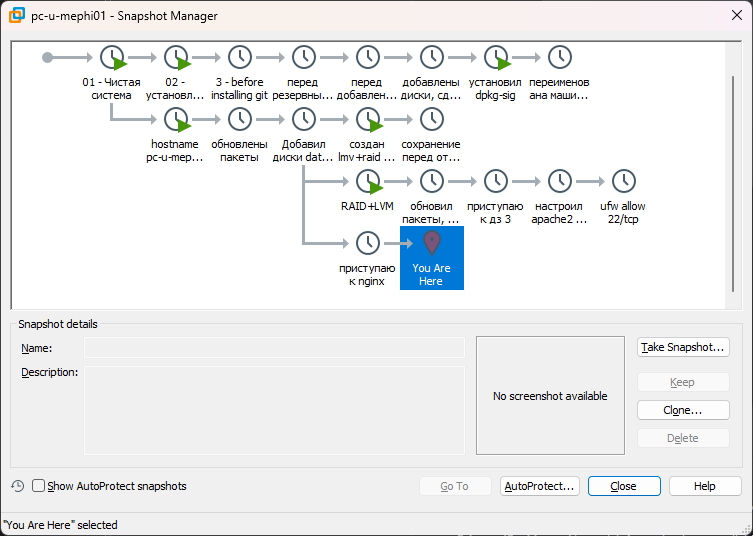


Настройка Apache успешно выполнена. Получившиеся снапшоты виртуальной машины приведены ниже:



# Часть 2: Настройка Nginx с прокси и ModSecurity

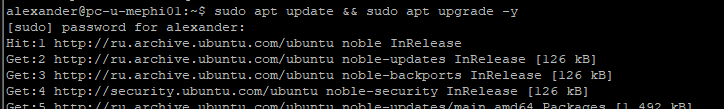
Для выполнения задания откатимся на снапшот «Добавил диски dat…», чтобы не возникло конфликтов при работе Apache и Nginx.



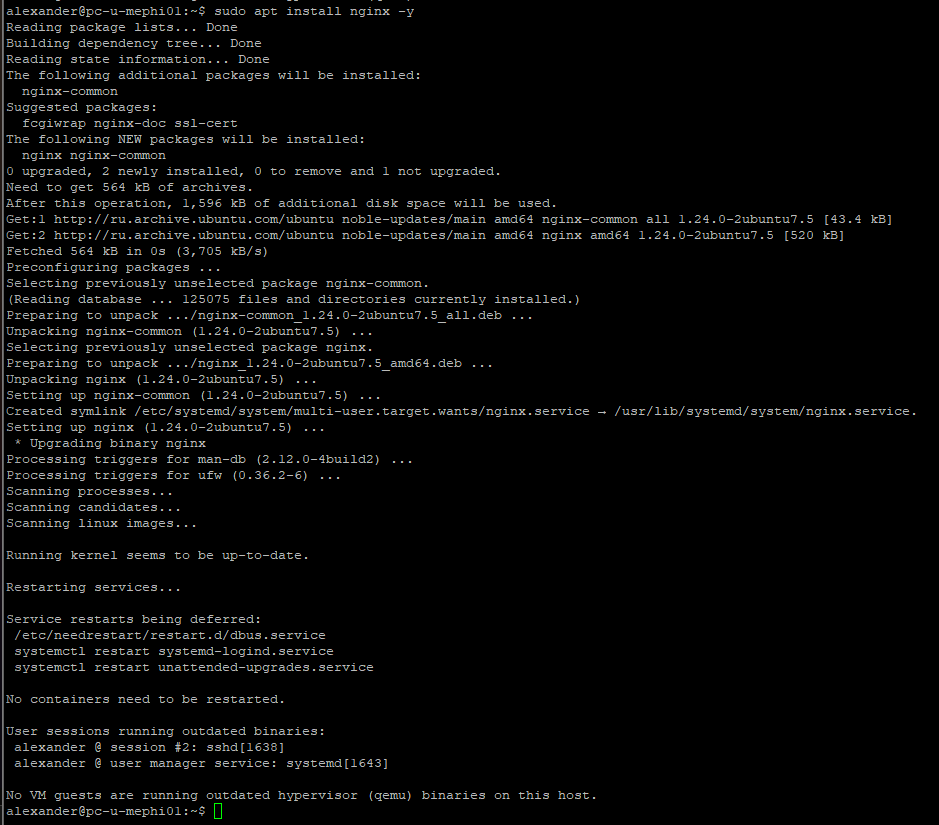
Для тестирования прямого и обратного прокси дополнительно воспользуемся приложением Postman.

1. Установка Nginx

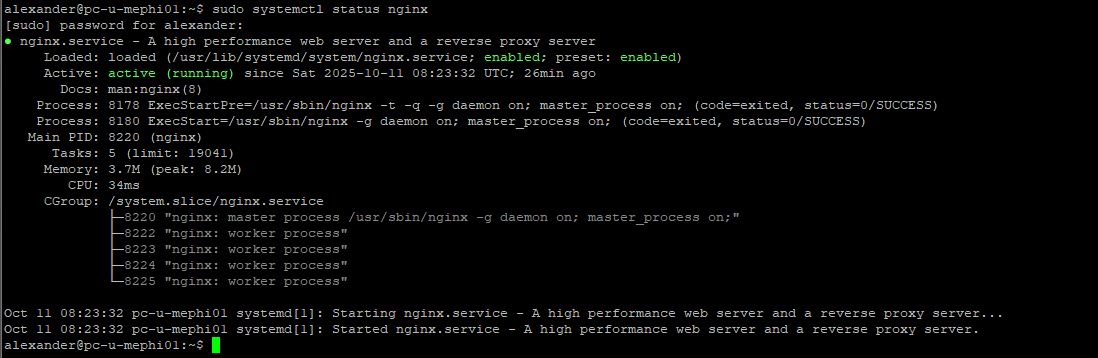
### Обновление пакетов



### Установка Nginx



### Проверка статуса службы

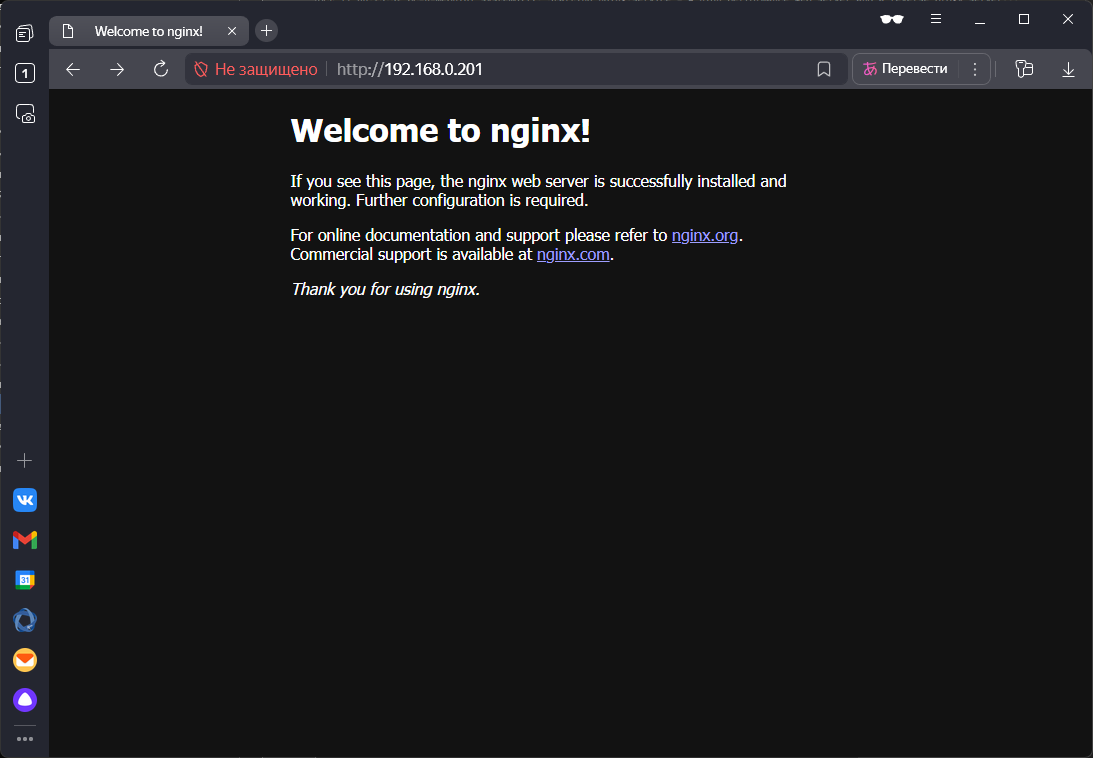


### Включение автозапуска



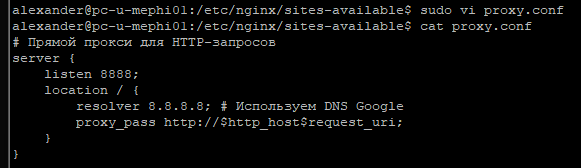
### Проверка в браузере на локальной машине

Проверка пройдена, страница nginx отображается в браузере.



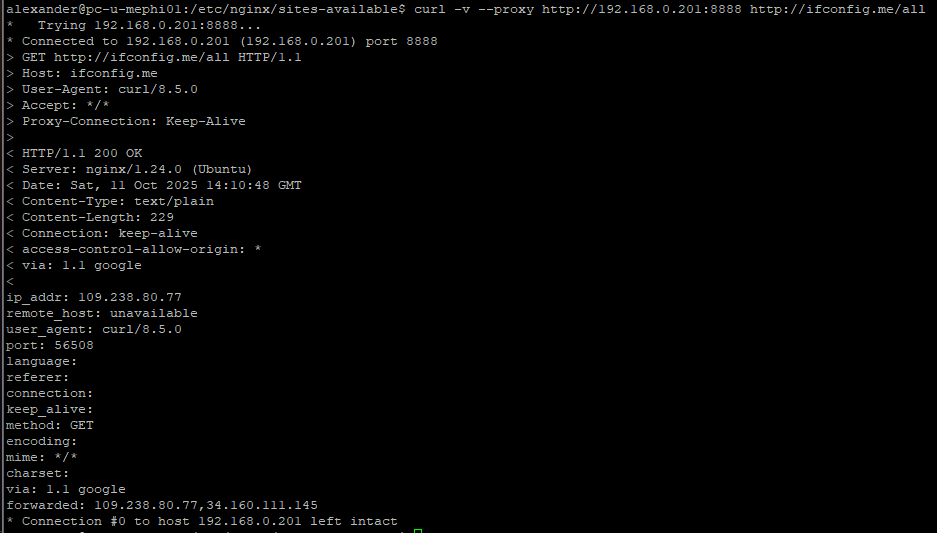
1. Настройка прямого прокси на Nginx

### Создадим файл proxy.conf в директории /etc/nginx/sites-available/.



### Проверим работу прямого прокси

#### Проверка через curl на ВМ:

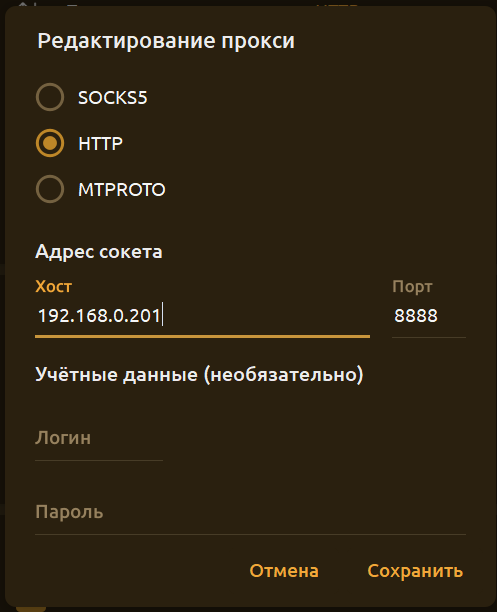


#### Проверка через ПК клиент Telegram:

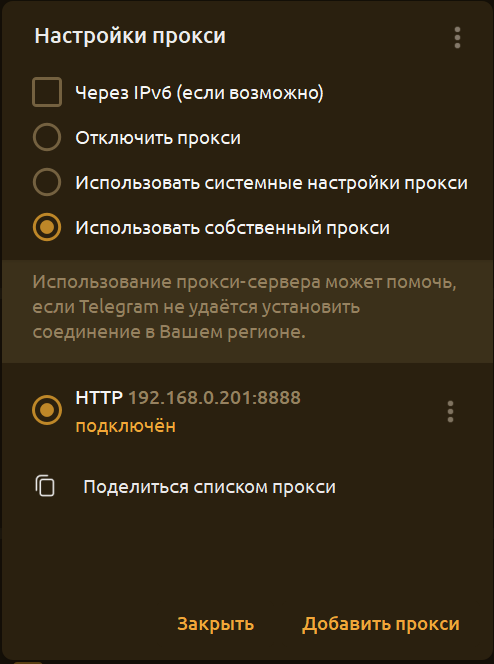
Настройки – Продвинутые настройки – Тип соединения – Добавить прокси

Хост = 192.168.0.201

Порт = 8888



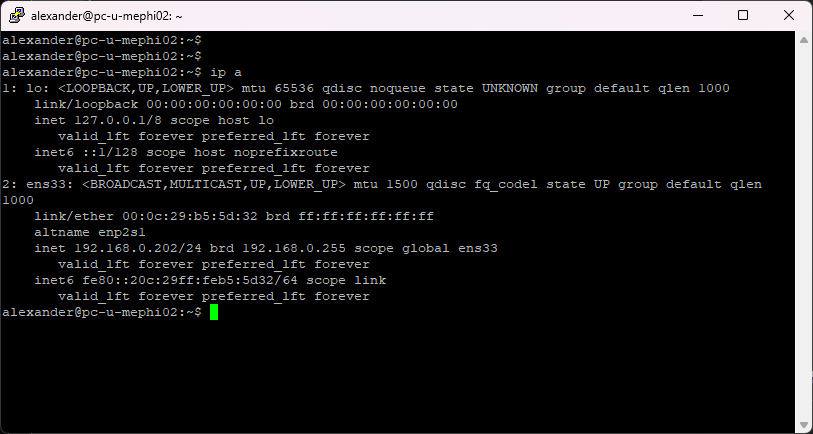
Прокси работает, клиент Telegram успешно подключен к прокси (https://t.me/proxy?server=192.168.0.201&port=8888):



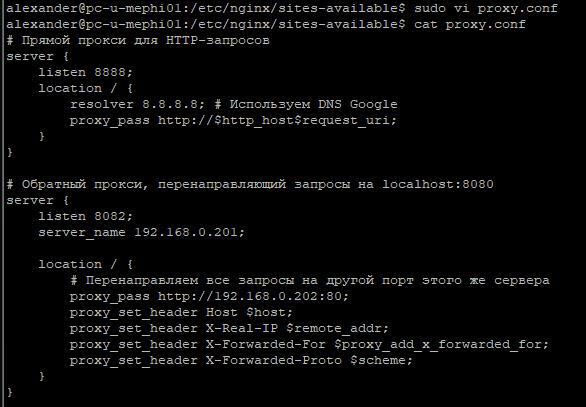
1. Настройка обратного прокси на Nginx

Добавим в конфигурацию proxy.conf настройки для обратного прокси. Для проверки обратного прокси поднимем вторую виртуальную машину на ip адресе 192.168.0.202 (pc-u-mephi02) (вторая машина), на которой также будет находиться Nginx. Аналогично шагам выше обновим пакеты, установим nginx.

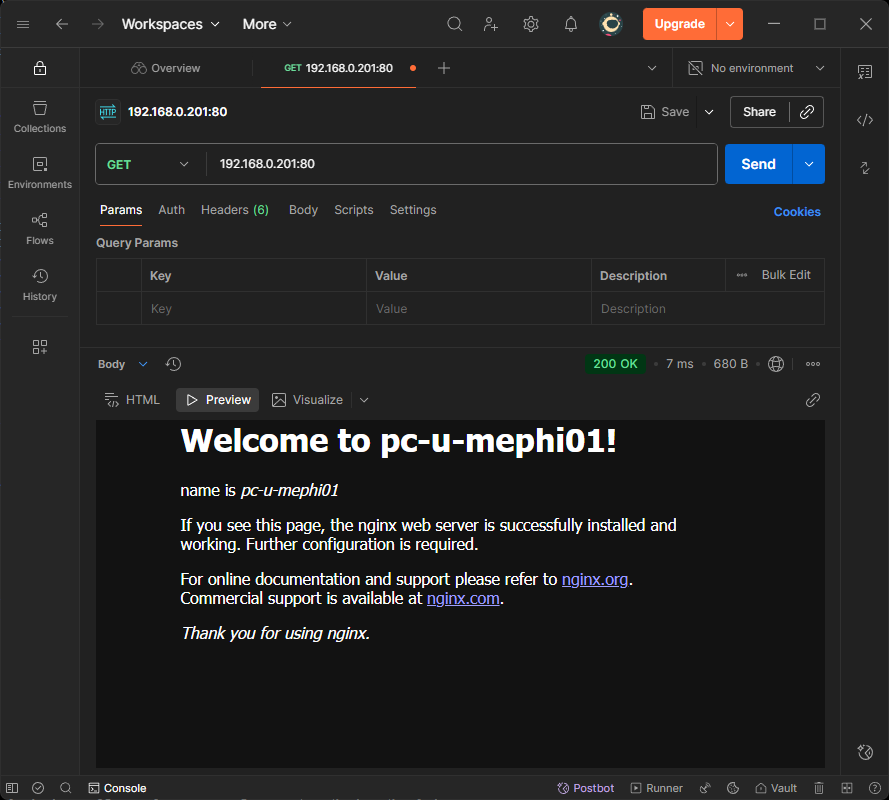
Адрес второй машины:



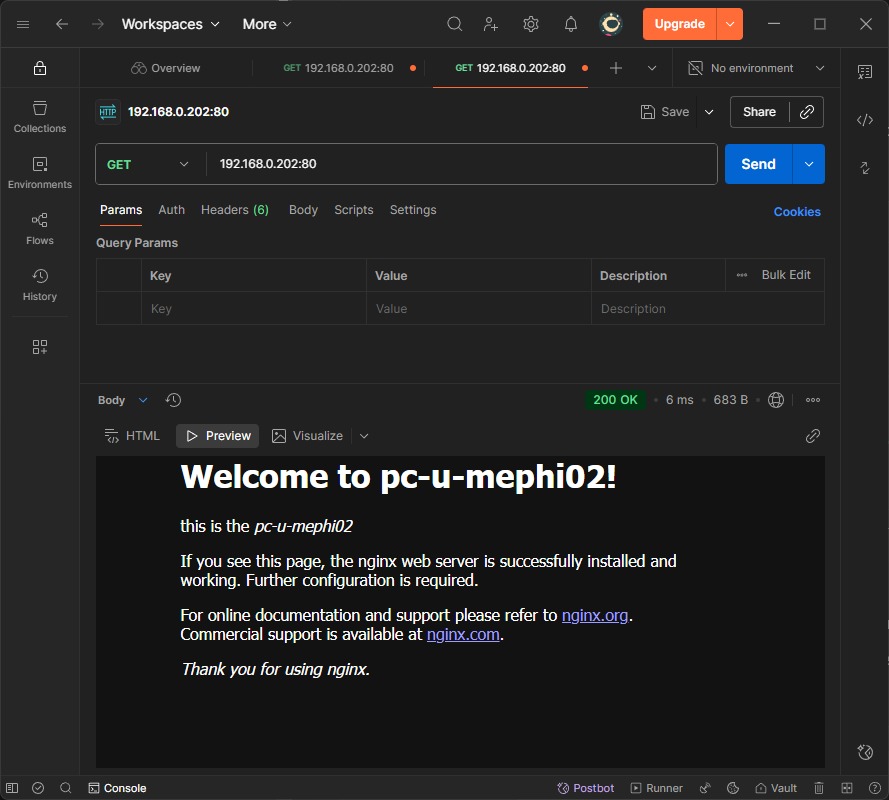
Конфигурация прокси на pc-u-mephi01 (главной машине):



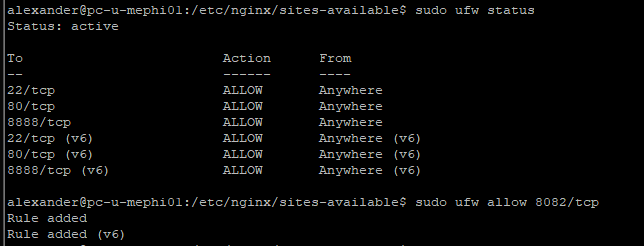
В страницу по умолчанию на машине pc-u-mephi01 внесем информацию об имени этой машины, чтобы отличить ее от второй машины:



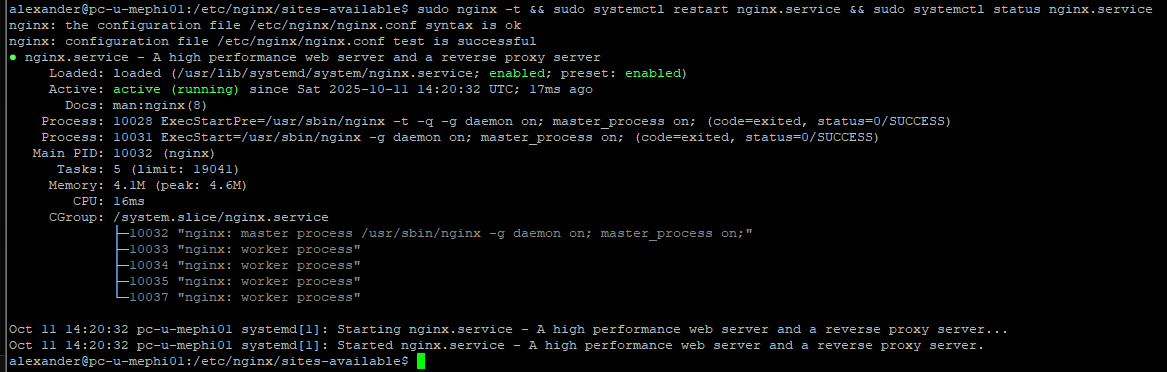
На странице для pc-u-mephi02 также внесем имя этой машины на страницу по умолчанию:



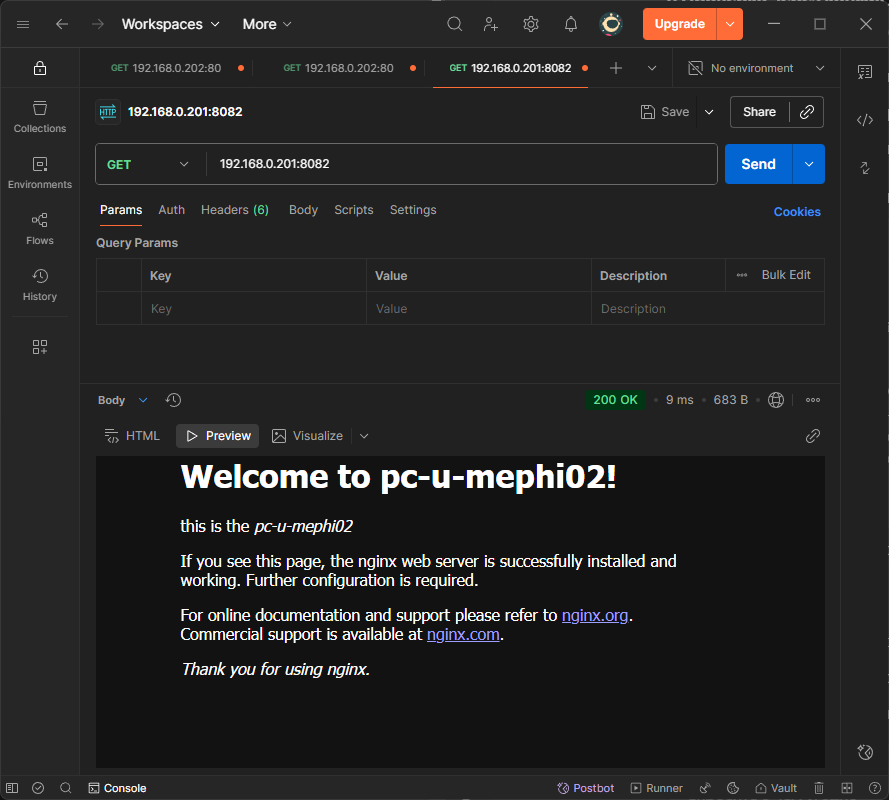
Добавим исключение в брандмауэр на первой машине по порту 8082:



Перезапустим службу nginx на первой машине.

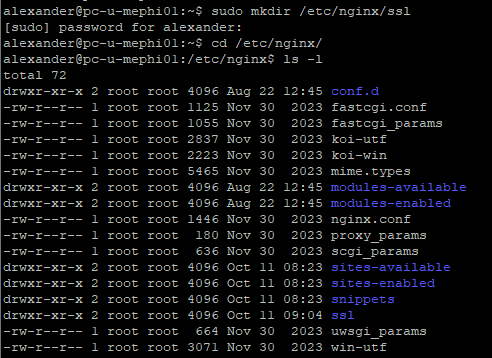


Проверим работу обратного прокси. Для этого сделаем запрос на первую машину на порт 8082. Результат – возвращается страница со второй машины:



1. 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

