Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Российский экономический университет им. Г.В. Плеханова»
МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ

Курсовая работа

ПМ.03 «Эксплуатация сетевой инфраструктуры»

МДК 03.02 «Безопасность функционирования информационных систем»

Специальность 09.02.02 «Компьютерные сети»

Тема: «Организация защищенного канала связи Open VPN»

МПТ 09.02.02 ПЗ 04 КР

Пояснительная записка

Листа: 58

Руководитель
_____ / И.О. Фамилия /
«_____» _____ 2021 г.

Исполнитель
_____ / И.О. Фамилия/
«_____» _____ 2021 г.

2021 г.

СОДЕРЖАНИЕ

Введение.

В последние пару лет VPN всё чаще оказывается на слуху. Это происходит по разным причинам, например, из-за суровых законов некоторых стран в отношении интернет-ресурсов, а также из-за пандемии коронавируса из-за, которой сотрудники различных компании все чаще уходят на удаленную работу.

Данная технология активно используется для решения самых разных задач от обхода блокировок сайтов на территории определенных стран, до объединения филиалов компании в единую сеть. Также часто VPN применяется в компаниях для безопасного удаленного подключения сотрудника к локальной сети предприятия. Это основные сценарии использования VPN.

Также с помощью VPN можно обеспечить доступность кого-нибудь ресурса из локальной сети в глобальной сети. Например, когда провайдер выделят серый IP адрес, без возможности приобрести белый IP адрес, то можно воспользоваться VPN сервером, имеющим белый IP, который обеспечит доступность к какому-либо ресурсу локальной сети по своему белому IP адресу с помощью прокси-сервера.

В данной курсовой работе будут рассмотрены следующие темы:

1. Принцип работы VPN (туннелирование), различные VPN протоколы.
2. Подробно рассмотрен протокол Open VPN.
3. Сертификация, центры сертификации. SSL сертификат.
4. Протоколы, технологии, программное обеспечение: dns, docker, Nextcloud, nginx, IPsec, http и https,
5. Алгоритм внедрения Open VPN для решения конкретной практической задачи.

В практической части работы будет рассмотрена настройка и внедрение протокола Open VPN, а также создание облачного файлового хранилища Nextcloud на NAS сервере компании, для обеспечения его доступности в глобальное сети Интернет по протоколу HTTPS. При условии того, что в городе, где находиться офис компании провайдер не выдает белый IP адрес.

Знание принципов работы VPN протоколов и их различий, а также умение выбрать нужный протокол для решения конкретной практической задачи является крайне важным для современного системного администратора в связи с большим количеством практических задач, решаемых с помощью VPN протоколов.

Тема данной курсовой работы является актуальной и рекомендуется к прочтению специалистам в области системного администрирования.

Основная часть.

Раздел 1. Теоретическая часть.

1.1 Технологии VPN. Протоколы.

На данный момент существует множество различных VPN протоколов, различающихся по самым разным параметрам таким как: безопасность, поддерживаемость различными ОС, скорость работы и тд.[1] Для решения разных задач используются разные протоколы. Существуют, как и устаревшие протоколы “канувшие в небытие” так и новые активно разрабатывающиеся и набирающие популярность. Для успешного выбора протокола для решения определенной задачи необходимо знать, чем VPN протоколы отличаются.

Чтобы понять принцип работы VPN и понимать то, о чём будет говориться в данной работе необходимо знать основную связанную с VPN терминологию.

В данном пункте будут рассмотрены разные VPN протоколы их отличия преимущества и минусы, а также принцип работы VPN туннеля.

Терминология.

VPN (Virtual Private Network) – Виртуальная частная сеть. Эта технология позволяющая создать поверх публичной сети, такой как Интернет, другую защищенную сеть с ограниченным числом участников в сети. VPN является защищенной сетью благодаря использованию различных средств, криптографии.[11]

Туннелирование – Это процесс создание логического соединения между двумя конечными точками посредством инкапсуляции различных протоколов.[11] Туннелирование это метод построения сетей, при котором один сетевой протокол инкапсулируется в другой.

Инкапсуляция – Это процесс формирования пакета перед отправкой по сети.[9] Инкапсуляция происходит сверху вниз по модели (OSI) начиная с прикладного уровня и заканчивая физическим уровнем.[9] На каждом уровне к пакету добавляется новый блок информации, предназначенный для конкретного уровня. Также есть и обратный процесс **деинкапсуляция**.

Туннель (Point-to-point) - Зашифрованное соединение между клиентом и сервером. Туннель создаётся в незащищённой сети, в качестве которой чаще всего выступает Интернет.[11] Соединение «точка-точка» подразумевает, что оно всегда устанавливается между двумя компьютерами, которые называются «узлами» или «peers». Каждый «peer» отвечает за шифрование данных до того, как они попадут в туннель, и расшифровка этих данных произойдёт после того, как они покинут туннель.

Шифр (Cipher) – Математический алгоритм, используемый для шифрования данных.[2]

IP адрес – Уникальный номер компьютера в сети, позволяющий идентифицировать конкретный ПК в сети. Существует два вида адресов: **белый IP** адрес, который участвует в глобальной сети Интернет и **серый IP** адрес, который участвует в локальной сети и не взаимодействует с сетью Интернет.[3]

Порт – Это некоторое целое число, используемое в заголовках протоколов транспортного уровня модели OSI.[5] Он необходим для возможности взаимодействия разных протоколов (программ) с одним IP адресом.

Принцип работы VPN.

Чтобы понять принцип работы VPN, необходимо для начала понять, как инкапсулируется пакет с целью создания туннеля. Рассмотрим структуру пакета (Рис.1). Вначале идёт заголовок Ethernet (кадр) в нем указывается мак адрес отправителя и мак адрес получателя данный заголовок соответствует канальному уровню модели OSI, также в этом заголовке указывается, какой протокол используется на сетевом уровне обычно это протокол IP (IPv4 или IPv6).[10] Далее идёт IP заголовок, в нём указаны IP адреса (получателя, отправителя), а также информация о том какой протокол используется на транспортном уровне UDP или TCP.[4] После идёт заголовок TCP или UDP в нём указаны порт отправки и порт получения пакета, а также различная служебная информация.

Рисунок 1 – Структура пакета.

Для доставки пакета от одного узла с белым IP адресом до другого узла с белым IP адресом достаточно этих трёх уровней. Для доставки пакета, как правило, промежуточным устройствам нет необходимости «рассматривать» пакет «глубже». С тем что лежит внутри заголовка TCP это личное дело того, кто отправляет пакет и той программы, которая получает этот пакет.

В стандартных случаях внутри, например, заголовка TCP идёт информация, допустим о протоколе HTTP. Но при использовании туннелирования внутрь заголовка TCP записывается ещё один заголовок Ethernet, а внутри него IP заголовок, а внутри него UDP заголовок и тд.[6] Для удобства назовем эту запись «внутренним» пакетом.

Таким образом, получается пакет внутри пакета. Что это даёт? А именно то, что при деинкапсуляции пакета, отправленного по глобальной сети от отправителя с белым IP адресом, происходит доставка «внутреннего» пакета, у которого так называемый виртуальный IP адрес.[15] Благодаря этому виртуальному адресу ПК, участвующие в процессе обмена пакетами «думают» что находятся в одной сети, то есть рядом. Таким образом, появляется виртуальная сеть поверх глобальной сети Интернет между двумя узлами. Также на уровне «внутреннего» пакета путь до получателя выглядит как один шаг, хотя в процессе передачи этого пакета происходит множество шагов в сети Интернет.

При добавлении шифрования к туннелированию получается VPN.

В зависимости от требований и применяемых протоколов VPN может быть трёх видов:

- Сеть в сеть (Site-to-site) – Объединение двух и более локальных сетей в единую виртуальную сеть.
- Удаленный доступ (Point-to-site) – Подключение типа точка сеть позволяет создать безопасное соединение отдельного компьютера (узла) с виртуальной сетью.
- Client/Server VPN - Такой способ соединения служит, когда серверу нужно создать и предоставить клиентам несколько различных сетей. Таким образом, пользователи внутри одной сети подключаются к серверу и передают ему данные по двум различным внутренним сетям. [11]

Протоколы VPN.

Протокол VPN – основа любого VPN сервиса.[7] Протокол в данном случае является фундаментом, на котором выстроен сервис, ведь в нем содержатся протоколы передачи данных и стандарты шифрования, которые позволяют быстро и защищено обменивать данными с VPN-серверами

Критерии, на которые нужно обращать внимание при выборе VPN протокола:

1) Безопасность – в разных протоколах используются разные криптографические методы шифрования и другие механизмы обеспечения безопасности.[8] Стоит уделить особое внимание этому пункту при выборе протокола VPN, когда важна конфиденциальность.
2) Скорость работы – этот параметр зависит от архитектуры протокола.[6] Существуют протоколы, которые показывают высокую производительность только в больших сетях и наоборот есть протоколы, работающие быстрее при их использовании в маленьких сетях.
3) Поддерживаемые ОС – существуют достаточно специфичные протоколы, работающие только на конкретных платформах, а также есть и такие которые работают на всех существующих ОС. [8]

Существует множество VPN протоколов, основными считаются такие протоколы как: Open VPN, IKEv2/IPSec, L2TP/IPSec, SSTP, PPTP.[11] Также в рамках данной курсовой работы будет рассмотрен набирающий популярность такой VPN протокол как WireGuard.

PPTP (Point-to-Point Tunneling Protocol) – Протокол, разработанный компанией Microsoft в 1999 году. Достаточно старый протокол, который на данный момент практически не используется. [2]

PPTP – использует два соединения одно управляющие другое для инкапсуляции данных. Первое соединение работает по протоколу TCP с портом 1723, второе работает на базе протокола GRE, который является транспортным протоколом, заменяющим TCP и UDP. Использование GRE не дает возможности клиентам, находящимся в локальной сети за NAT соединением установить подключение с сервером VPN (то есть клиентам, имеющим серый IP адрес)[14]. Само использование GRE-туннеля имеет серьёзные последствия: из сети с NAT к одному серверу сможет одновременно подключиться только один клиент.[10]

PPTP сильно устарел, и сейчас разработавшая его компания Microsoft настоятельно рекомендует использовать другие VPN решения, такие как SSTP или L2TP/IPSec.[11] К причинам, почему его сейчас практически не используют можно отметить:

- Серьезные проблемы с безопасностью. Практически не защищает пользователей. Получить доступ к частной сети можно методом перебора в течение 23 часов с помощью любого профильного online сервиса. [13]
- Нестабильность соединение. При разрыве соединение потребуется куда больше времени на восстановление туннеля в сравнение с другими протоколами.
- Сложность в настройки и управление.

SSTP (Secure Socket Tunneling Protocol) — это модернизированный протокол PPTP. Хоть он и не является популярным VPN решением, но в тоже время он не имеет таких проблем с безопасностью как PPTP. Перехват трафика при использовании данного протокола намного сложнее, все благодаря SSL шифрованию.[7] Вовремя установки соединения весь трафик идёт через TCP порт 443. Огромным плюсом данного протокола является

возможность его использования в странах, где большая часть VPN сервисов заблокирована.

Множество исследование доказывают, что данный протокол способен быстро передавать данные, а также быстро восстанавливать соединение в случаи разрывов.

Главным минусом данного протокола является серьезное снижение скорости при даже незначительной загруженности канала связи.

Данный протокол хорошо поддерживается только операционными системами Windows.

Перед тем как рассматривать такие протоколы как IKEv2/IPSec, L2TP/IPSec. Стоит понять, что такое IPSec.

IPSec (Internet Protocol Security) – это набор протоколов позволяющих обеспечить конфиденциальность данных передаваемых по IP сетям.[9] Этот стек протоколов не был разработан конкретно для VPN это один из вариантов его применения.

Главным преимуществом IPSec является его упрощенность в настройки. Из-за того, что IPSec работает на сетевом уровне отпадает необходимость в подключении сторонних клиентов.[12] Поднять VPN с IPSec можно стандартными встроенными в ОС средствами.

IPSec шифрует весть IP-пакет используя:

- Authentication Header (AH). Эта технология ставит цифровую подпись на каждом пакете.[1]
- Encapsulating Security Protocol (ESP). Технология, обеспечивающая конфиденциальность, целостность и аутентификацию пакета при передаче.[1]

L2TP/IPSec (Layer 2 Tunneling Protocol) – это VPN протокол созданный на основе двух протоколов L2F от компании Cisco и PPTP от компании Microsoft.[9] Данный протокол сам по себе не поддерживает шифрование или аутентификацию, поэтому в сочетание с ним используется протокол IPSec.

Протокол L2TP/IPSec не имеет серьезных проблем с безопасностью в отличии от PPTP. Благодаря IPSec он может использовать такие протоколы шифрования как 3DES и AES, хотя в настоящие время 3DES считается слабым алгоритмом шифрования и используется редко.[10]

Использование IPSec имеет один недостаток, заключающийся в том, что многие брандмауэры часто блокируют соединения по 500 порту. [9]

Протокол L2TP/IPSec достаточно безопасен, поддерживается всеми основными ОС (Windows, Linux, Android, macOS), подходит для использования в не критически важных задачах. Однако данный протокол инкапсулирует данные дважды, что делает его медленнее других более современных протоколов.

IKEv2/IPSec (Internet Key Exchange version 2) – протокол VPN разработанный совместно компаниями Microsoft и Cisco.[9] Также, как и L2TP не поддерживает сам по себе шифрование и аутентификацию для этих целей данный протокол использует IPSec.

Поскольку данный протокол использует IPSec для шифрования данных то можно сказать, что он имеет такой же уровень безопасности, как и L2TP/IPSec.[9]

Главным плюсом данного протокола можно считать поддержку таких технологий как Mobility или Multi-homing-Protocol. Что позволяет IKEv2/IPSec быть устойчивым к смене сетей. Поддержка этой технологии делает этот протокол предпочтительным для использования мобильными

устройствами, поскольку данные устройства часто переключаются между сетями.

Multi-homing-Protocol – Множественная адресация позволяющая устройству или сети иметь присутствие в нескольких сети.

К плюсам данного протокола можно отнести его высокую скорость работы и возможность быстрого автоматического пере подключения к серверу.

К минусам стоит отнести ограниченное количество поддерживаемых платформ.

OpenVPN – протокол VPN с открытым исходным кодом выпущенный в 2002 году и активно развивающийся по сегодняшний день. На данный момент один из самых популярных VPN протоколов.[11] Благодаря открытому исходному коду протокол прошел большое количество проверок безопасности от компания занимающихся информационной безопасностью

OpenVPN считается одним из самых защищенных протоколов, этому поспособствовало множество поддерживаемых алгоритмов шифрования, поддержка SSL, а также алгоритмов AES-256-GCM.[11]

Протокол OpenVPN является одним из самых быстрых протоколов многие VPN сервисы, использующие за основу OpenVPN способны обеспечить скорость передачи шифрованных данных до 2000 Мбит в секунду.[8] Но в тоже время OpenVPN нельзя назвать самым быстрым VPN протоколом быстрее его работает такой протокол как WireGuard.

При постройке туннеля OpenVPN может использовать как UDP, так и TCP, что в свою очередь позволяет обходить блокировки VPN сервиса от провайдера благодаря использованию 443 порта на котором работает HTTPS. Провайдер не может заблокировать это порт так как иначе у пользователей на будет доступа к HTTPS сайтам.

Еще одной чертой OpenVPN является возможность его использования на всех современных ОС, но только после установки специальной программы клиента OpenVPN.

WireGuard – Абсолютно новый VPN протокол с открытым исходным кодом визитной карточкой которого стал размер в 4000 строк. Активно набирает популярность у пользователей считается что в скором времени станет заменой OpenVPN.[11]

Для обеспечения безопасности WireGuard использует современную криптографию различных протоколов, берущих каждый на себя решение конкретной задачи:

- Curve25519 – для обмена ключами;
- Poly1305 – для аутентификации;
- ChaCha20 – для шифрования;
- SipHash – для ключей хеш-таблицы;
- BLAKE2 – для хеширования.

За все время существования этого протокола не было найдено серьезных уязвимостей. В случае же нахождения уязвимости разработчики могут очень быстро и оперативно исправить её, благодаря небольшому коду. Также совсем не давно WireGuard был включен в состав ядра Linux 5.6 и прошел дополнительный аудит безопасности от независимых экспертов который не выявил не каких проблем с безопасностью.

По результату тестов производительности, которые можно посмотреть как на официальном сайте, так и у независимых тестировщиков можно прейти к выводу что WireGuard намного превосходит все ранее существующие VPN протоколы.

При построении туннеля WireGuard использует UDP с выбором любого порта.

Хотя туннель WireGuard взломать можно сказать и невозможно, но если говорить о конфиденциальность, то WireGuard не как не защищает пользователей. Это происходит из-за того, что в настройках Wireguard в явном виде указываются IP адреса пользователей. И если сервером заинтересуются правоохранительные органы, то не кто не уйдет от внимания. Но трафик, конечно, останется неизвестным. В вопросе конфиденциальности WireGuard проигрывает OpenVPN, так как OpenVPN не требует от пользователей прописывать IP адреса в явном виде.

Протокол WireGuard поддерживается всеми современными ОС.

1.2 Технология OpenVPN.

Как говорилось ранее OpenVPN на данный момент считается одним из самых популярных VPN протоколов. Для выполнения практической часть был выбран именно протокол OpenVPN причиной этому стала отличная защищенность данного протокола, а также конфиденциальность клиентов. Также значительным плюсом в пользу выбора OpenVPN стал его открытый исходный код. При выборе VPN протокола также рассматривался протокол WireGuard из-за своей высокой скорости работы, но у него есть большой минус, а именно отсутствие конфиденциальности. В этом разделе рассмотрим протокол OpenVPN со всех сторон.

Обмен данными по сети.

OpenVPN способен передавать данные по сети на любом порту TCP или UDP с применением драйверов TUN/TAP. Протокол UDP в сочетание с драйвером TUN позволяет подключиться клиенту находящемуся за NAT, то есть с серым IP адресом подключиться к серверу OpenVPN.

Драйвер TAP эмулирует Ethernet устройство работающие на канальном уровне модели OSI, а драйвер TUN (сетевой туннель) на сетевом уровне модели OSI.

Возможность OpenVPN выбирать любой порт, позволяет преодолевать ограничения маршрутизаторов и файрволлов провайдера в случае их наличия.

Безопасность и шифрование.

В OpenVPN для обеспечения безопасности и шифрования используется библиотека OpenSSL в сочетание с протоколом транспортного уровня TLS который пришел на смену протоколу SSL.

В OpenSSL есть возможность использовать как симметричное, так и ассиметричное шифрование.

Симметричное шифрование — это когда для шифрования и расшифрование используется один и тот же ключ.[4] В данном варианте ключ должен быть известен обоим сторонам в чём и зачастую заключается проблема при начальной передаче ключа одной из сторон (этап согласования ключей). Если злоумышленник получит этот ключ, то он будет иметь доступ ко всей информации передаваемой по каналу связи.

Ассиметричное шифрование — в данном случае используется два ключа закрытый и открытый.[4] Открытый ключ передается по открытым каналам связи и используется для шифрования. А для расшифровки используется закрытый ключ. Таким образом если кто-то перехватит открытый ключ, то он не сможет изменить передаваемую информацию или прочитать.

В основном при использовании OpenSSL для генерации ключей используют ассиметричное шифрование. Генерируется два ключа публичный для шифрования данных и приватный для расшифровки данных.

Приватный ключ нельзя передавать на другие компьютеры кроме того на котором он создан. Можно передавать только публичный ключ.

Для безопасной передачи данных вначале нужно идентифицировать участников обмена данными иначе есть вероятность стать жертвой атаки называемой “человек по середине”. В ходе такой атаки злоумышленник подключается к каналу связи и прослушивает его. Он может спокойно перехватывать данные и даже изменять или удалять.

Для защиты от этой атаки используется проверка подлинности пользователя (аутентификация), для этого используется протокол TLS и инфраструктура открытых ключей (PKI - Public Key Infrastructure) в сочетании с асимметричной криптографией.

Существует возможность расшифрования данных без наличия приватного ключа, например, методом перебора. Размер ключа хоть и влияет на сложность расшифровки, но в тоже время замедляет процесс обмена данными. [5]

Сертификаты и центр сертификации CA.

Для избежание подделки открытого ключа используется процесс заверения ключа. Если ключ будет создан для публичного использования, то его должна заверить коммерческая или государственная организация с чистой репутацией. В результате процедуры заверения ключа создается сертификат открытого ключа.

Созданный сертификат также должна заверить организация которой доверяют.

Есть много организаций, создающих сертификаты, к примеру для протокола HTTPS или для цифровой подписи сообщений электронной почты. Такие сертификаты стоят денег и служат ограниченный срок.

Для сети VPN используемой в конкретной компании нет необходимости покупать сертификаты. Можно создать свой собственный центр сертификации CA и создавать собственные сертификаты. Естественно доверие к таким сертификатам будет только на уровне той организации, где был создан этот сертификат. Основными плюсами использования собственного центра сертификации CA можно считать его абсолютную бесплатность и то, что этого будет вполне достаточно для создания виртуальной частной сети VPN на базе протокола OpenVPN.

Созданные сертификаты будут играть роль открытых ключей с помощью, которых клиенты сети OpenVPN будут шифровать данные. Соответственно для расшифровки данных будут использоваться приватные ключи.

Сертификаты создаются в соответствии со стандартом X.509. Этот стандарт определяет тип данных и процессы распределения открытых ключей с помощью сертификатов, снабженных электронными подписями.[2]

Сертификат X.509 – это публичный ключ, в котором содержаться следующие данные: владелец сертификата, имя узла, период действия, алгоритм и значение подписи сертификата и т.д. [2] Он должен быть подписан закрытым ключом центра сертификации.

Когда клиент OpenVPN подключается к серверу OpenVPN по протоколу TLS то происходит отправка ему с сервера сертификата X.509. Для проверки подписи используется открытый ключ центра сертификации, находящийся на клиенте.

Таким образом происходит проверка сервера, к которому подключается клиент это позволяет избежать атаки “человек по середине”.

Список отзыва сертификатов.

В OpenVPN предусмотрен список отзыва сертификатов (CRL), а также средства управления этим списком. Данный список необходим если нужно заблокировать доступ к VPN какому-либо отдельному сотруднику (клиенту), к примеру уволившемуся. [11]

Список CRL генерируется в центре сертификации CA и потом копируется на сервер OpenVPN. После изменения списка CRL его нужно заново копировать на сервер OpenVPN.

Файл Диффи-Хелмана.

Файл Диффи-Хелмана в протокол OpenVPN используется для обеспечения защиты перехваченного трафика от расшифровки, при утере ключей.[9] Под перехваченным трафиком имеется введу записанный трафик до похищения ключей.

Данный файл создается на сервере OpenVPN, а не на сервере центра сертификации.

Статический ключ HMAC.

HMAC (Hash-based Message Authentication Code) – обеспечивает проверку подлинности информации, передаваемой между сторонами. Этот ключ генерируется на сервере OpenVPN с целью защиты от DoS атак.[11]

Пример файлов конфигурации клиента и сервера OpenVPN рассматриваются в практической части.

1.3 Выбор программных средств для реализации практической части.

Для выполнения практического задания будет использоваться различное ПО и средства.

Выбор сервера OpenVPN и серверной ОС.

Для выполнения практической части был выбран хостинг провайдер vscale.io. Данный провайдер предоставляет возможность выбора VDS сервера с безлимитным трафиком в различных конфигурациях за приемлемую цену.

Для настройки OpenVPN сервера был выбран VDS сервер в минимальной конфигурации на базе ОС Linux Ubuntu 20.04 LTS.

Почему Linux? Во-первых, это надежность, во-вторых, легковесность, в-третьих, бесплатность, в-четвертых, безопасность и т.д. Linux выигрывает по всем параметрам при решение подобного рода задач.

Выбор пал на дистрибутив Ubuntu поскольку автор лучше знаком именно с этим дистрибутивом. Для решения практической части особой разницы при выборе конкретного дистрибутива нету так как настройка идентична. А сервер не планируется использовать еще для чего-то кроме как OpenVPN сервера.

Утилита Easy-RSA.

Создание ключей и сертификатов можно выполнить с помощью утилиты OpenSSL, но намного проще воспользоваться специальной созданной программой Easy-RSA, которая использует OpenSSL. [11]

Для начала необходимо установить эту утилиту, после создать инфраструктуру публичных ключей PKI. Для ее создание используется команда:”./easyrsa init-pki”. В результате выполнения данной команды будет создан каталог /pki где будут находиться публичный ключи PKI.

После создается центр сертификации с помощью команды “./easyrsa build-ca”. В результате выполнения команды будет предложено выбрать имя центра сертификации, а также ввести пароль для защиты приватного ключа. Данный пароль будет необходимо вводить для подписания сертификатов для клиентов и серверов.

Можно также обойти ввод пароля при необходимости для этого необходимо дописать в указанную раннее команду “nopass”. Но стоит учитывать, что если данный ключ украдут, то им можно будет легко воспользоваться так как он без пароля.

После выполнения команды “build-ca” создастся два файла:

1. /pki/private/ca.key – это приватный ключ центра сертификации CA. Его нельзя переносить на другие устройства в сети
2. /pki/ca.crt – этот же ключ открытый и он будет использоваться на серверах и клиентах OpenVPN. Его нужно будет перенести на сервер с OpenVPN и клиенту.

Для создания списка отзыва сертификатов необходимо ввести команду “/easyrsa gen-crl”. При выполнении команды будет запрошен пароль приватного ключа ca.key для подписи сертификата отзыва. В итоге будет создан файл “/pki/crl.pem”.

Для того чтобы заблокировать ранее выданный сертификат нужно ввести команду “./easyrsa revoke (имя сертификата)”. После необходимо скопировать новый файл CRL и перенести на сервер OpenVPN после чего перезапустить демон OpenVPN командой “systemctl reload openvpn”.

Технология DNS для практической части.

По заданию для практической части необходимо сделать файловое хранилище доступным в сети интернет по доменному имени. Для решения этой задачи можно воспользоваться услугами компании, предоставляющей услугу регистрации домена на своих DNS серверах.

DNS – Это технология позволяющая “связать” доменное имя с IP адресом.[9] Таким образом что в компьютерной сети можно будет обращаться к удаленному ресурсу не по IP адресу, а по доменному имени.

DNS сервер – Это сервер на котором хранятся домены и доменные имена. DNS сервер можно сравнить с списком контактов на смартфоне на нем хранятся соответствия IP адреса к доменному имени, также и в смартфоне в телефонной книги хранятся соответствия телефонного номера к имени контакта.[3] В сети этот сервер предоставляет информацию о IP адресе, который связан с запрошенным доменным именем.

Для реализации практической части воспользуемся услугами компании Reg.ru и приобретем доменное имя.

На DNS сервере есть различные типы записей, использующиеся для различных целей, но основной и самой часто используемой считается запись типа A это запись указывает соответствие IP адреса к доменному имени. Именно эту запись и будем прописывать в личном кабинете Reg.ru.

Выбор прокси сервера Nginx.

На данный момент существует два основных Web сервера это Apache и Nginx. Для решения практической части необходим быстрый веб сервер способный к перенаправлению запросов (проксированию). Оба Web сервера могут перенаправлять запросы, но выбор был сделан в пользу Nginx по нескольким причинам.

Во-первых, Nginx в отличии от Apache позиционирует себя как http-proxy-сервер. Что в свою очередь позволяет Nginx не выполнять не какой тяжелой работы по обработки информации. Nginx использует асинхронную событийную архитектуру.[11] Благодаря этому Nginx обрабатывает намного быстрее запросы любого другого сервера, а также потребляет меньше системных ресурсов.

В сравнение с Apache один рабочий процесс Nginx обрабатывает не один запрос пользователя, а сразу множество запросов.[11] Поскольку Nginx это proxy сервер то он может отправит запрос пользователя на backend, а пока backend занимается запросом пользователя, Nginx продолжает обрабатывать запросы пользователей.

Во-вторых, хотя Apache и является самым функциональным веб-сервером на данный момент. Он имеет огромное количество модулей для решения самых разных задач. Однако из-за этого он очень ресурсоемкий и поэтому намного лучше для небольших проектов использовать быстрый и легковесный Nginx.

В-третьих, Nginx набирает все большую популярность и на сегодняшний момент он активно разрабатывается.

Выбор протокола HTTP или HTTPS.

HTTP – протокол передачи гипертекста используется для передачи данных в сети в частности веб-сайтов.[2] Этот протокол реализует технологию клиент-сервер, где клиент является инициатором соединения отправляя запрос, а сервер, получающий запрос выполняет его и отправляет клиенту.

HTTPS – безопасный протокол передачи гипертекста. Это тот же протокол что и HTTP, но с расширением в виде поддержки шифрования с помощью SSL и TLS.[2]

Отличие протоколов заключается в том, что HTTPS использует шифровании данных и работает на 443 TCP порту, а HTTP передаёт данные не зашифрованными и работает на 80 TCP порту.

Когда необходимо использовать HTTPS, а когда можно использовать HTTP?

Протокол HTTPS нужно использовать если сайт собирает конфиденциальную информацию, производит банковские операции, использует авторизацию и тд. Более того согласно федеральному закону №152 “О персональных данных” если сайт собирает даже минимальную базовую информацию о пользователях, к примеру ФИО, то он становиться оператором персональных данных. По закону такой ресурс обязан соблюдать различные меры по защите этих данных, и использование HTTPS одна из них.

Протокол HTTP можно использовать в случаях если сайт несет только информативный характер, к примеру страница портфолио, одностраничный сайт компании (лендинг) и тд.

В практической части будет создан файловый сервер на базе сервиса Next Cloud, данный сервис использует авторизацию, также аккаунты пользователей хранят в себе персональные данные. Утечка пароля и логина является неприемлемой, поскольку тогда будут скомпрометированы данные пользователей. По этим причинам в практической части будет использоваться протокол HTTPS.

SSL сертификата, Let`s Encrypt и утилита cerbot.

SSL сертификат – это цифровая подпись сайта, которая нужна для работы протокола защищенной передачи данных в сети.[7] SSL сертификат необходим для работы HTTPS. Перехватить трафик также легко, как и при использовании HTTP протокола, но вот чтобы расшифровать данные

понадобиться много лет. Благодаря SSL информации которой обменивается пользователь с сайтом защищена от просмотра провайдером, администратором сети, хакером. Таже с помощью данного сертификата подтверждается подлинность сайта, пользователь может узнать какой компании принадлежит информационный ресурс.

Выбор центра сертификации Let`s Encrypt.

Как говорилось ранее в разделе 1.2 получить сертификат можно в удостоверяющих центрах. Одним из таких центров является Let`s Encrypt от компании Internet Security Research Group (ISRG).

Почему именно Let`s Encrypt?

1. Бесплатность. Любой пользователь может использовать Let’s Encrypt для получения SSL/TLS сертификатов, не тратя денег.
2. Программное обеспечение Let’s Encrypt, запущенное на web-сервере, само позаботится о выпуске, настройке и обновлении сертификатов.
3. Прозрачность**.** Все выданные или отозванные сертификаты будут сохранены, в том числе и для любых проверок безопасности.
4. Открытость. Протокол выпуска и обновления сертификатов имеет открытый стандарт.

Главной причиной выбора Let`s Encrypt является его бесплатность.

Утилита cerbot (ранее называемая letsencrypt) необходима в практической части для получения SSL сертификата. Cerbot является официальным клиентом Let`s Encrypt.

Сервис Nextcloud.

Nextcloud – это набор клиент-серверных программ для создание и использования облачного хранилища с веб интерфейсом его можно установить как на хостинге так и на собственном сервере. Похож на DropBox.

Имеет открытый исходный код благодаря чему к Nextcloud написано множество модулей на любой вкус.[11]

Помимо облака Nextcloud также предоставляет возможность работы с электронной почтой и видеоконференциями. Также в нем можно создавать заметки и задачи, а еще он поддерживает совместное редактирование документов.

Каких-то особых причин выбора именно Nextcloud для реализации практической части нет. Он удовлетворяет требование к практической задаче по созданию облачного файлового хранилища с веб интерфейсом.

Контейнеризация Docker.

Контейнеризация отличная альтернатива аппаратной виртуализации. Все процессы в контейнере протекают на уровне операционной системы что дает возможность эффективно использовать ресурсы.

Одним из самых популярных инструментов для программной виртуализации считается Docker – автоматизированное средство управления виртуальными контейнерами. Docker создает контейнеры, размещает в них приложения, а также управляет процессами.[6]

Docker – программное обеспечение с открытым исходным кодом, применяется для разработки, тестирования, доставки и запуска веб-приложений в средах с поддержкой контейнеризации. Он необходим для более эффективного использования ресурсов ПК, быстрого развертывания готовых программных продуктов, а также для переноса созданных

контейнеров с готовым ПО на другие ОС с гарантией сохранения стабильной работы.

Контейнеризация — это тип виртуализации, который позволяет упаковывать программное в некоторый контейнер, который изолирован от основной ОС. Каждый такой контейнер содержит все нужные элементы для работы приложения.

Docker это отдельная большая тема в рамках данной курсовой работы рассмотреть ее полностью невозможно. В практической части используется Docker для быстрого развертывания Nextcloud.

1.3 Алгоритм внедрения Open VPN.

В практической части необходимо сделать следующие:

1. Обеспечить доступ к серверу NAS с облачным сервисом Nextcloud из сети Интернет по доменному имени, при условии того, что провайдер выделяет серый IP адрес.
2. Доступ к облаку необходимо сделать по протоколу HTTPS, поскольку Nextcloud оперирует такими данными как пароли и личные данные пользователей.
3. Nextcloud необходимо поднять в Docker контейнере для избегания проблем с зависимостями и последующими обновлениями.

Для выполнения практического задания необходимо для начала выбрать хостинг VDS провайдера. Установить на выбранный хостинг серверную OS и настроить SSH. После необходимо приобрести доменное имя. Как будет куплено доменное имя можно перейти к добавлению записи типа A на DNS сервере.

Как только подготовительные моменты будут готовы и к хостингу можно будет подключиться с помощью ssh по доменному имени можно перейти к настройки Open VPN сервера.

В настройке Open VPN выделяется два этапа:

1. Настройка центра сертификации и генерация необходимых ключей и сертификатов.
2. Настройка конфигураций Open VPN сервера и клиента.

Как только Open VPN будет настроен и между NAS сервером и хостингом появиться VPN туннель можно перейти к настройке Nginx прокси для перенаправления запросов пришедших на белый IP адрес хостинга на адрес tun интерфейса NAS сервера.

Последним шагом поднимаем контейнер docker с Nextcloud таким образом что бы запросы пришедшие на tun интерфейс NAS сервера перенаправлялись на внутренний адрес контейнера в сети docker.

Логическая топология рассматриваемой сети (Рис.2).

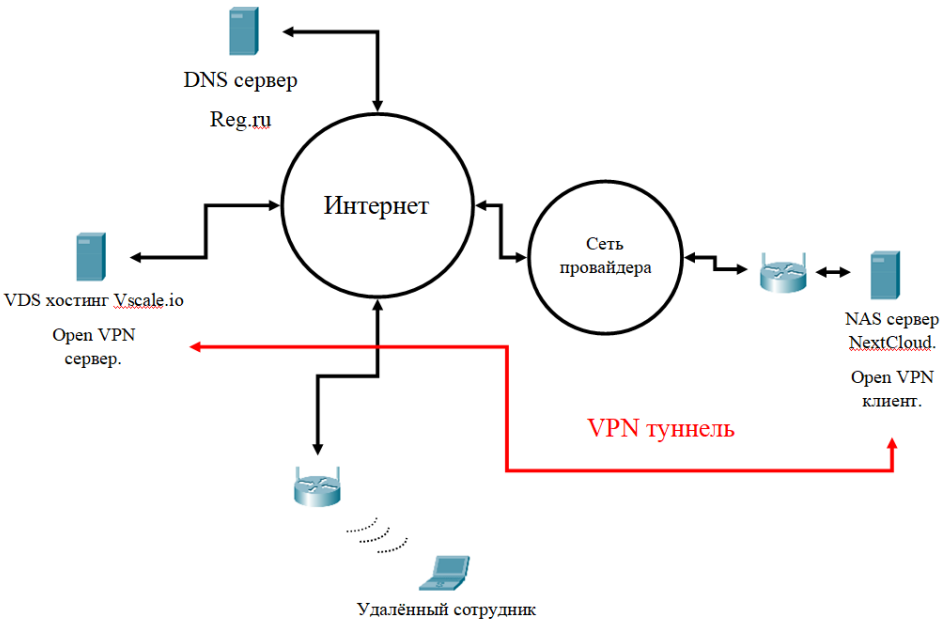

Рисунок 2 – Логическая топология сети.

Раздел 2. Практическая часть.

В практической части будет произведена настройка Open VPN сервера для выведения в глобальную сеть такого облачного решения как NextCloud расположенного на NAS сервере в локальной сети. Локальный роутер подключен к серому IP-адресу провайдера из-за этого напрямую, вывести сервис в глобальную сеть Интернет невозможно. Также сервис должен быть доступен по доменному имени. Для повышения безопасности необходимо сделать так чтобы NextCloud открывался по HTTPS протоколу. Для того чтобы не было проблем с зависимостями при обновлении NextCloud, необходимо развернуть данное приложение в Docker контейнере.

2.1 Выбор доменного имени, хостинга и настройка удаленного доступа по доменному имени на хостинг.

Для начала необходимо выбрать и приобрести доменное имя для будущего хостинга с Open VPN. Перейдем на сайт reg.ru в разделе домены выведен список популярных доменов (Рис.3).

Рисунок 3 – Список доменов.

Для выполнения данной практической задачи заранее было выбрано доменное имя star-time.space (Рис.4).

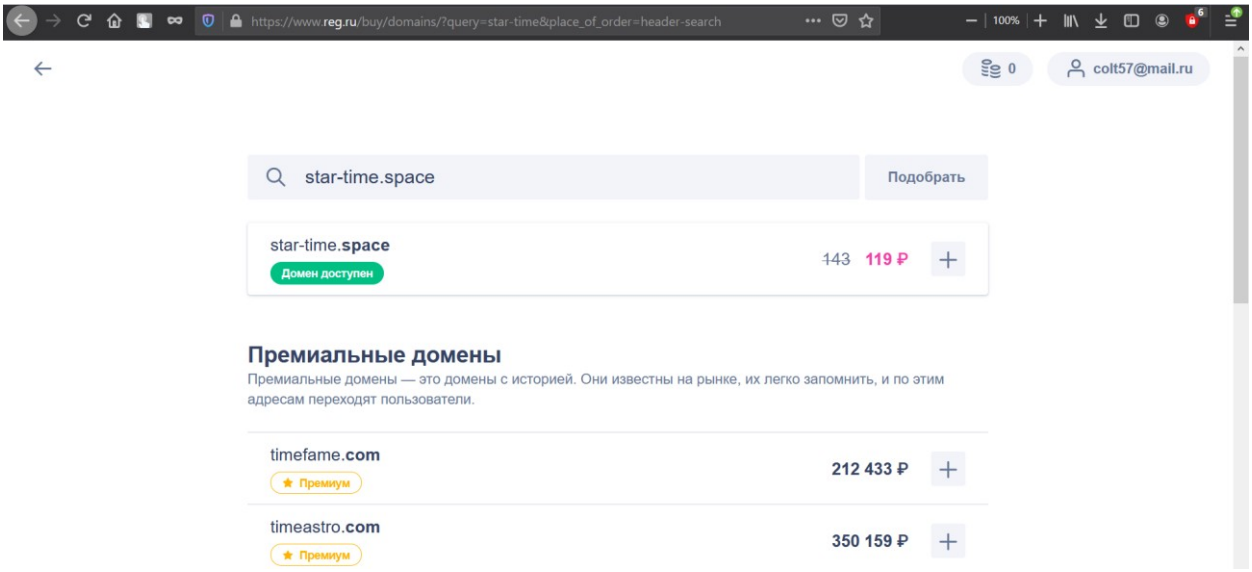

Рисунок 4 – Выбор доменного имени.

После прохождения регистрации на сайте и выбора доменного имени и дополнительных услуг производим оплату (Рис.5).

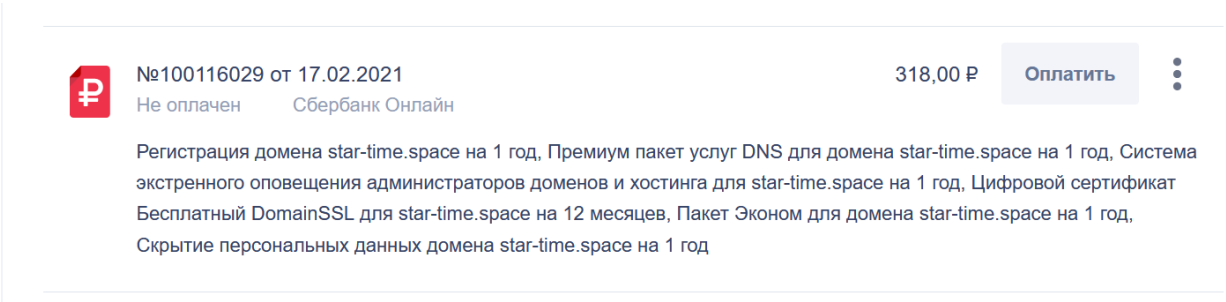

Рисунок 5 – Покупка доменного имени.

Так выглядит личный кабинет на reg.ru (Рис.6).

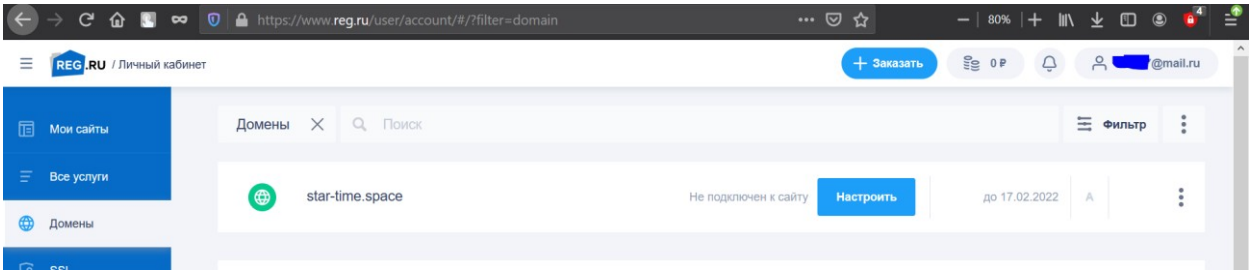

Рисунок 6 – Личный кабинет.

Дальше необходимо приобрести хостинг. Для этого можно воспользоваться услугами vscale.ru. Для успешного прохождения регистрации необходимо пополнить кошелек на минимальную сумму 200 рублей. Данная сумы хватит на поддержания сервера в течение 28 дней, после необходимо будет пополнить кошелек.

Выбор дистрибутива (Рис.7). Для решения задачи практической части курсовой работы необходимости выбирать, конкретный дистрибутив нет, поэтому выбираем Ubunto 20.04.

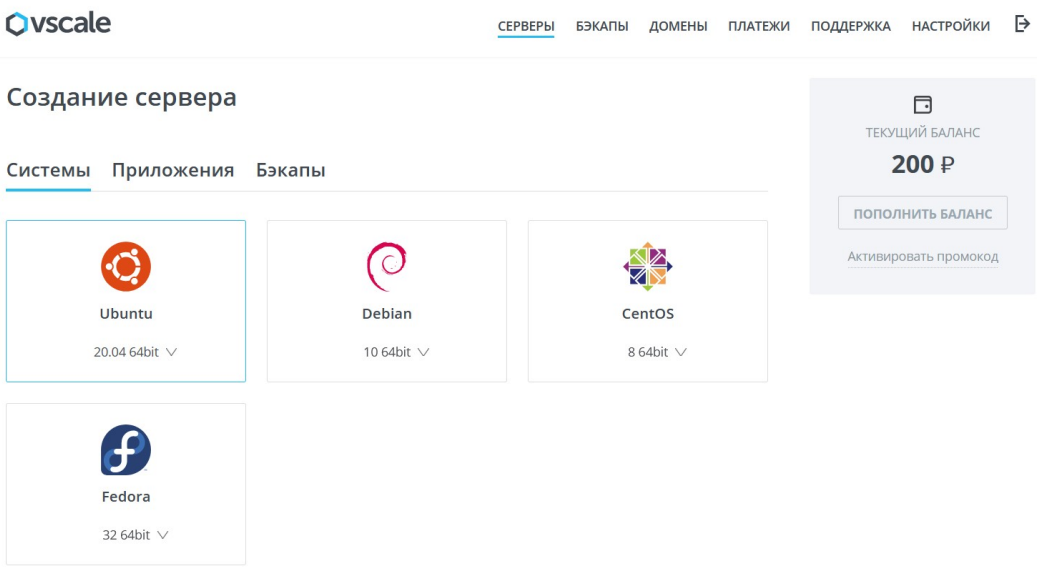

Рисунок 7 – Выбор дистрибутива.

Выбор конфигурации сервера. Для решения задачи практической части хватит минимальной конфигурации (Рис.8).

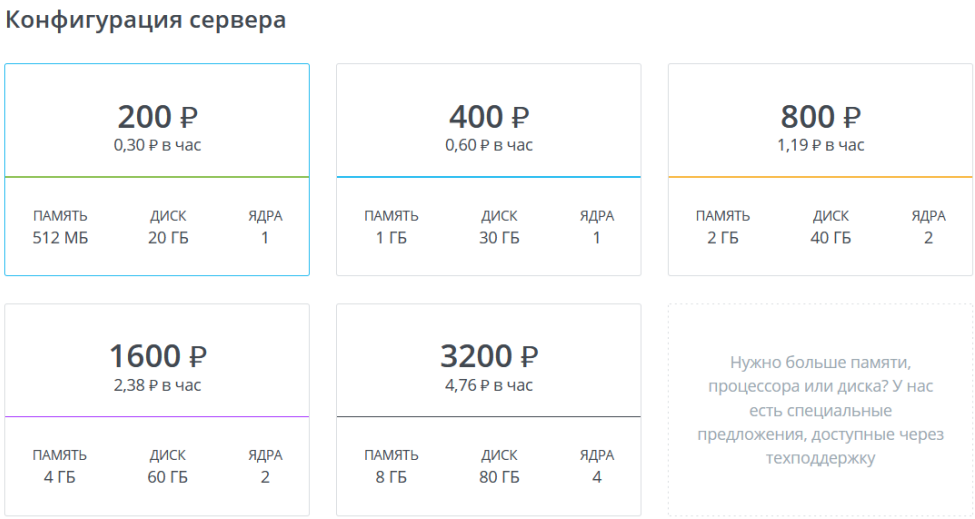

Рисунок 8 – Конфигурации серверов.

На следующем этапе необходимо выбрать физическое расположение сервера на выбор vscale.io дает города Москва и Санкт-Петербург (Рис.9).

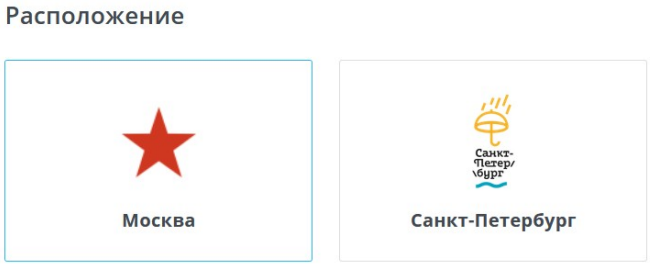

Рисунок 9 – Выбор расположения сервера - Москва.

Далее необходимо задать имя сервера. Задаем имя supernova (Рис.10).

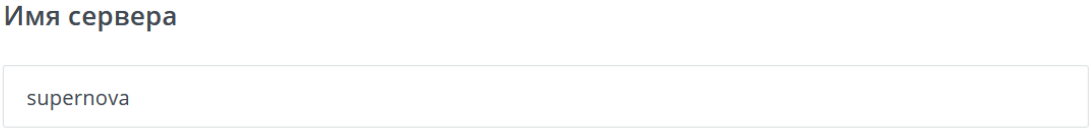

Рисунок 10 – Имя сервера.

Последним этапом начальной настройки хостинга является добавление ssh ключа. Для начала необходимо сгенерировать ssh ключ. Что бы это сделать, можно воспользоваться в ОС Windows программой Putty. В нашем случае сгенерируем ключ на сервере NAS, для этого подключимся к нему по ssh и выполним команду ssh-keygen –t rsa вводим необходимые утилите данные и генерируем ключ, после командой cat ~/.ssh/id_rsa.pub выводим открытый ключ (Рис.11). Не забываем скопировать ключ.

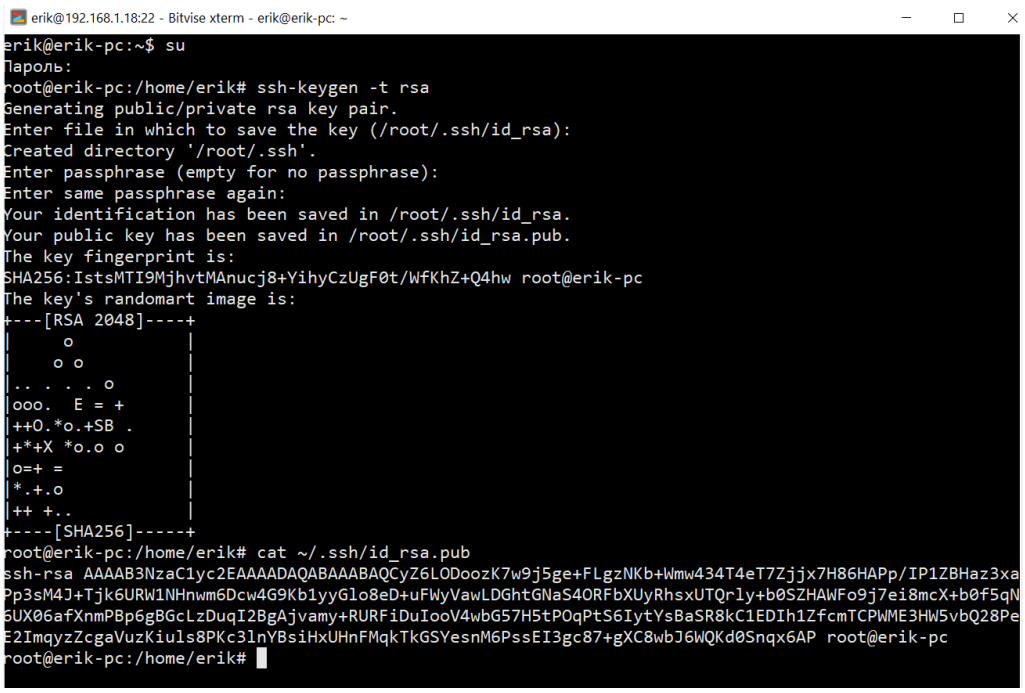

Рисунок 11 – Генерация ssh ключа.

Добавляем ключ на host машину. Добавить ключ можно двумя способами: первым способом можно добавить ключ в процессе выбора конфигурации сервера, вторым с помощью консоли в Linux командой ssh-copy-id user@ip or domain name. Поскольку хостинг еще не запущен, добавляем ключ первым способом (Рис.12).

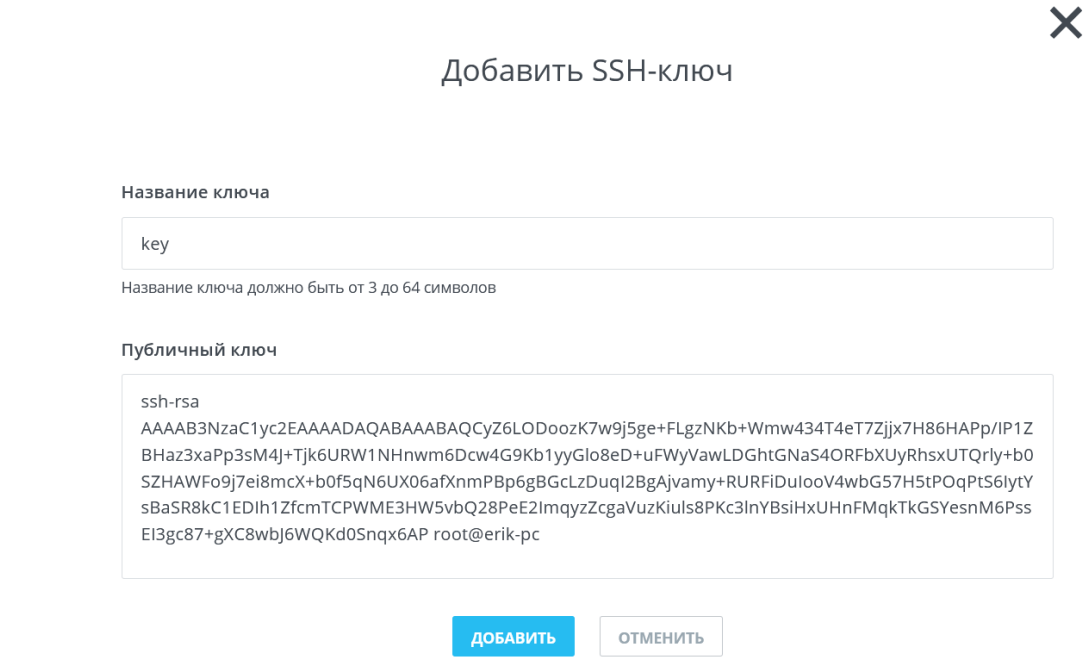

Рисунок 12 – Добавление ssh ключа.

После начинается стадия установки (Рис.13).

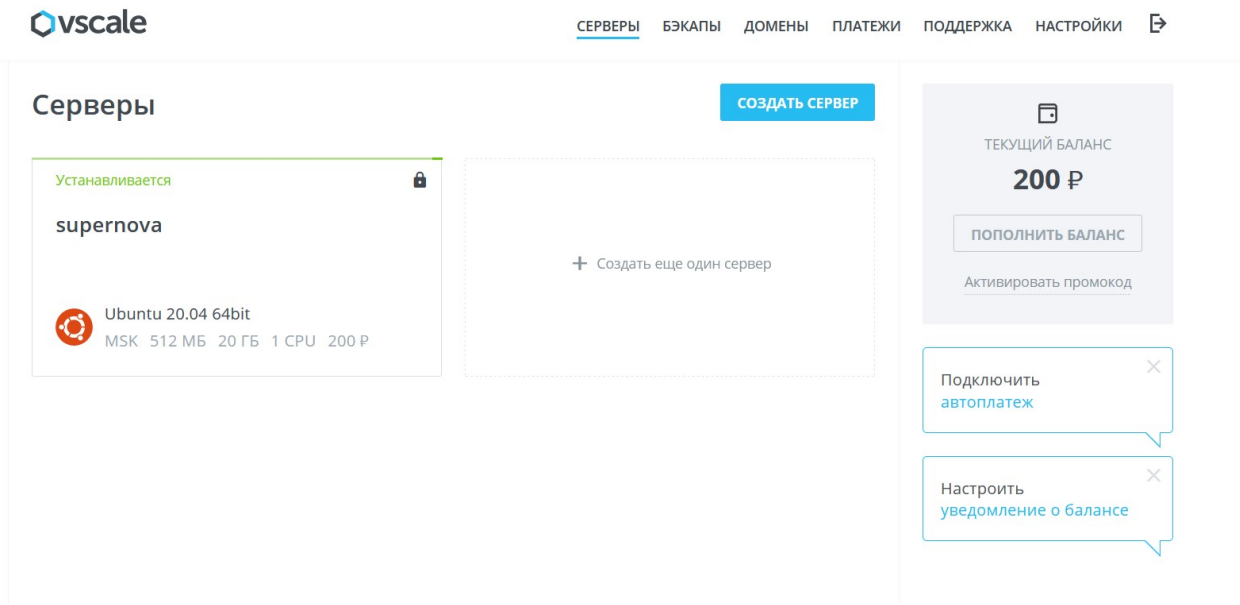

Рисунок 13 – Установка.

Как только установка закончена можно, перейти посмотреть параметры сервера (Рис.14). Здесь отображается IP-адрес маска подсети шлюз имя хоста и название используемого ssh ключа. В низу также есть команда для подключения по ssh к хостингу через консоль. Скопируем ip адрес.

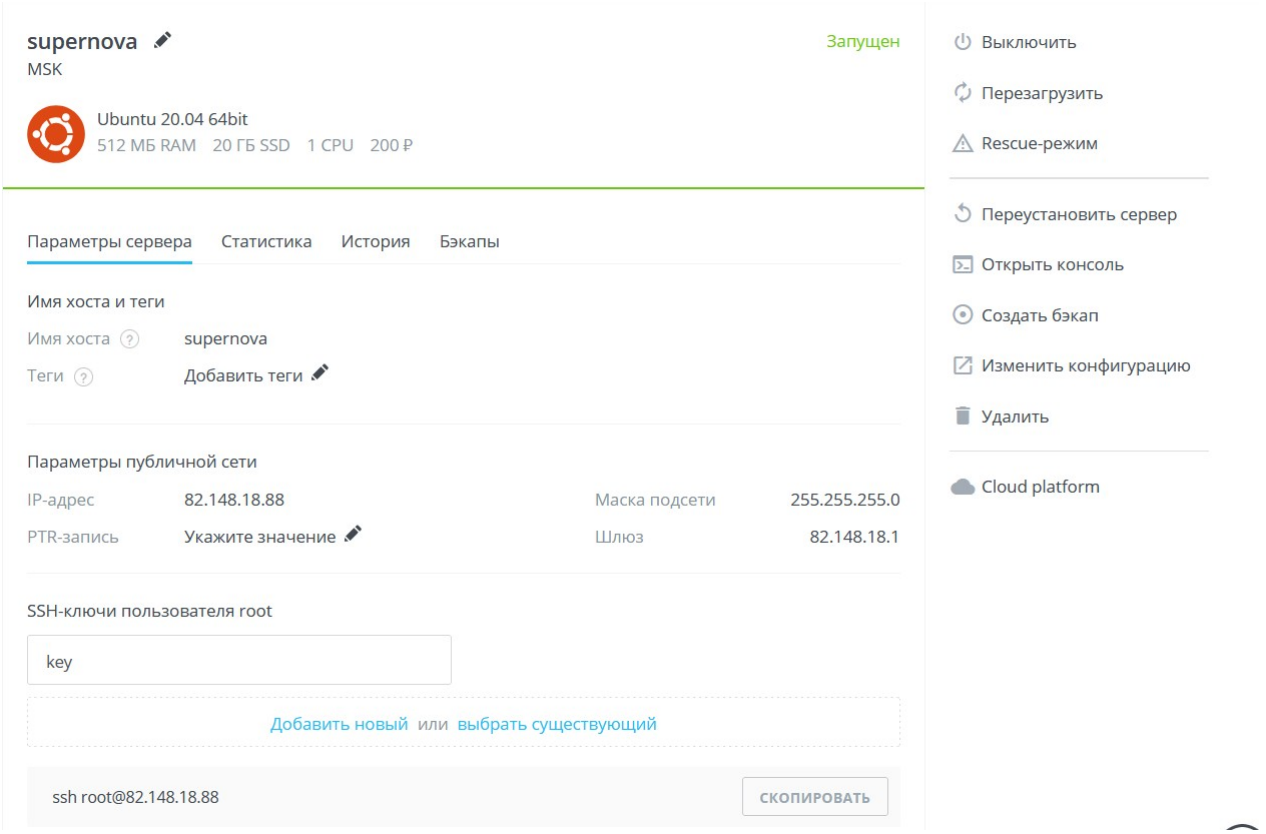

Рисунок 14 – Параметры сервера.

Перед тем как подключаться по ssh перейдем в браузере в терминал хостинга это можно сделать, нажав в правой панели на пункт “открыть консоль” (Рис.12), для того чтобы задать пароль для root пользователя. Пароль можно задать командой passwd (Рис.15).

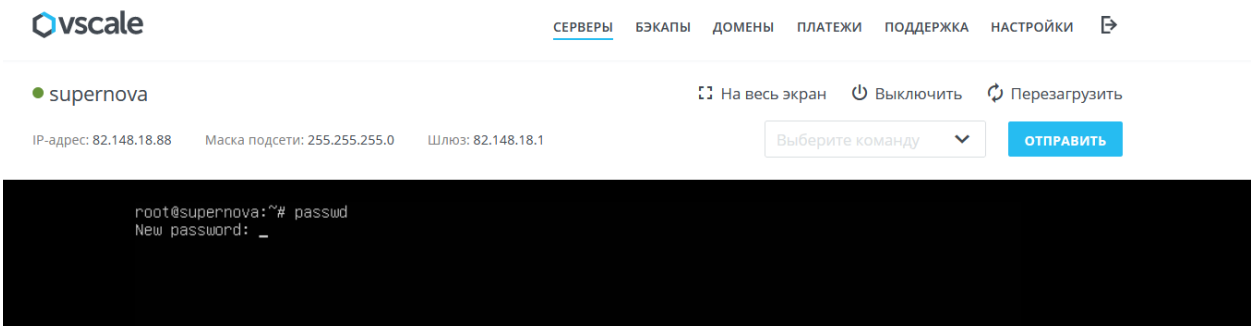

Рисунок 15 – Установка пароля.

Теперь можно подключиться по ssh. Для удаленной работы с хостингом будем использовать программу Bitvise SSH Client. Что удобно в этой программе также встроен SFTP клиент в дальнейшем он нам понадобится, для переноса конфигурация Open VPN клиента на NAS сервер. Вводим необходимые данные (Рис.16).

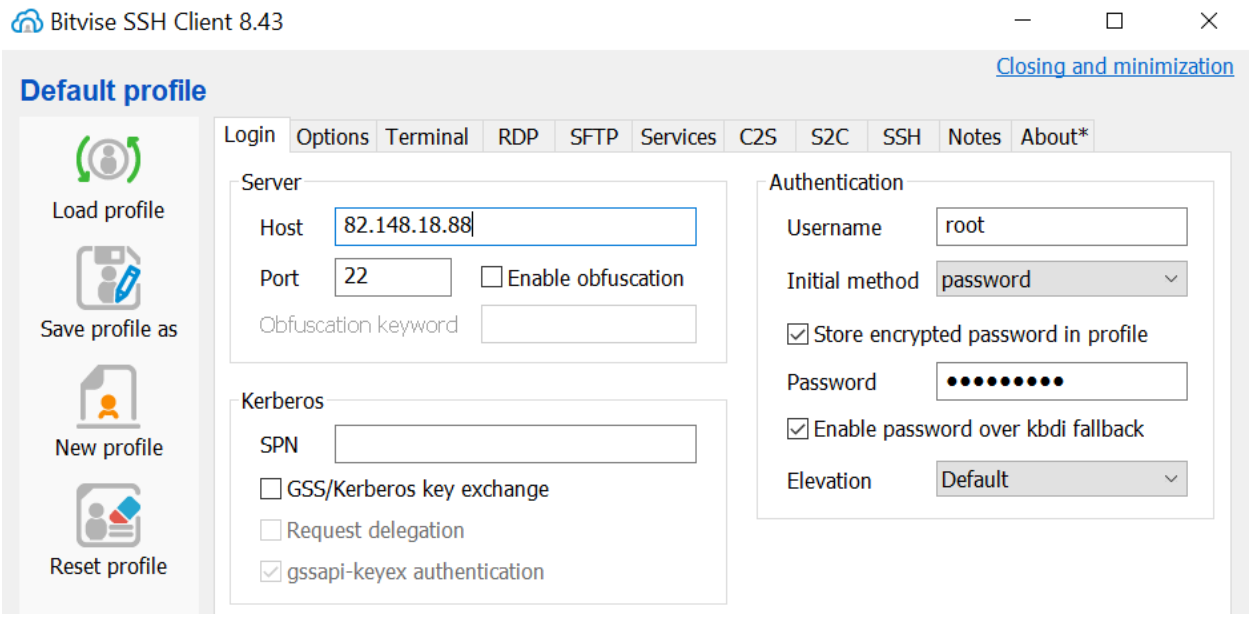

Рисунок 16 – Bitvice SSH Client.

Подключение (Рис.17).

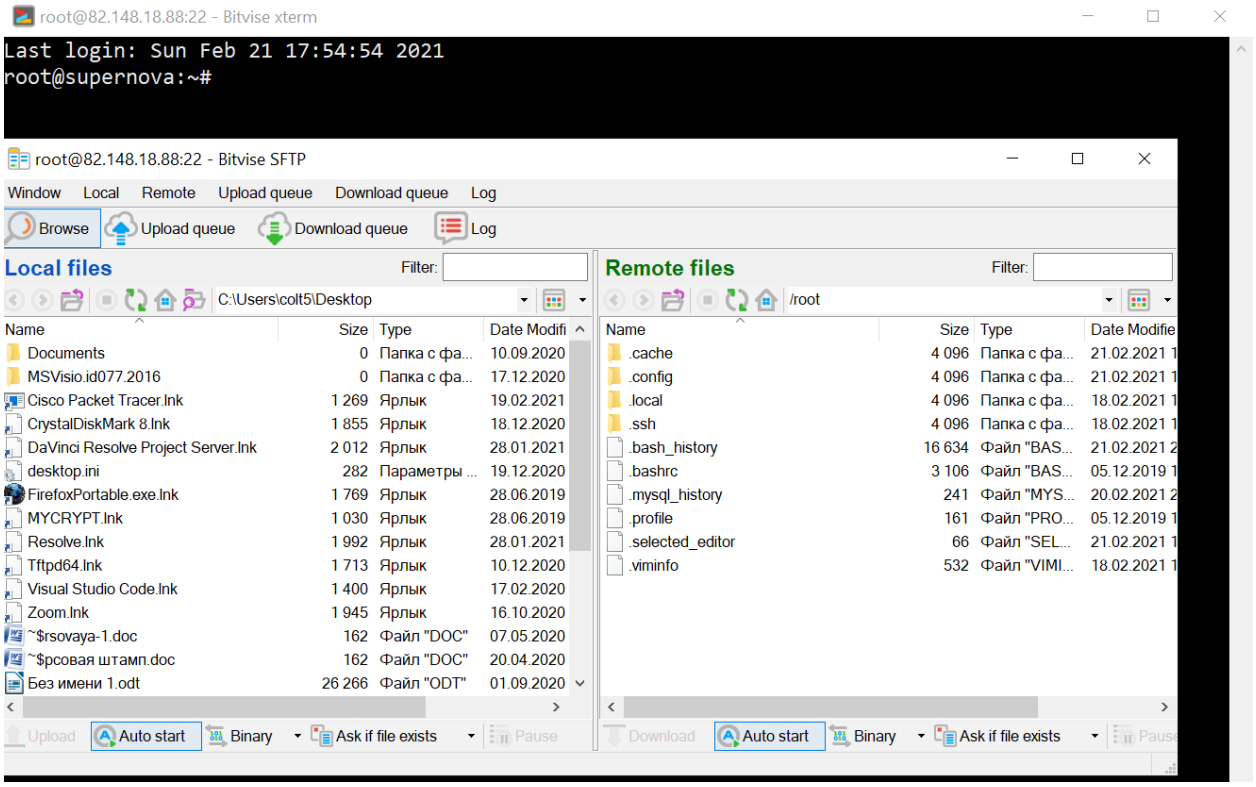

Рисунок 17 – Подключение к хостингу.

Теперь необходимо сделать так чтобы доступ до хостинга был возможен по доменному имени для этого перейдем в личном кабинете в раздел “DNS-серверы и управление зоной” и создадим запись типа A. Данная запись типа A “свяжет” ip 82.148.18.88 с доменным именем supernova.star-time.space (Рис.18).

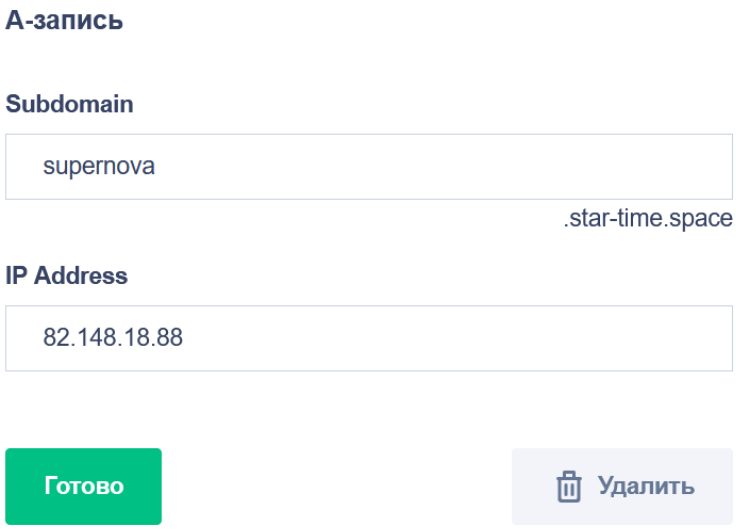

Рисунок 18 – Добавление записи типа A.

После необходимо добавить доменное имя для сервера это можно сделать, перейдя на сайт vscale.io в раздел домены и нажать кнопку добавить домен (Рис.19).

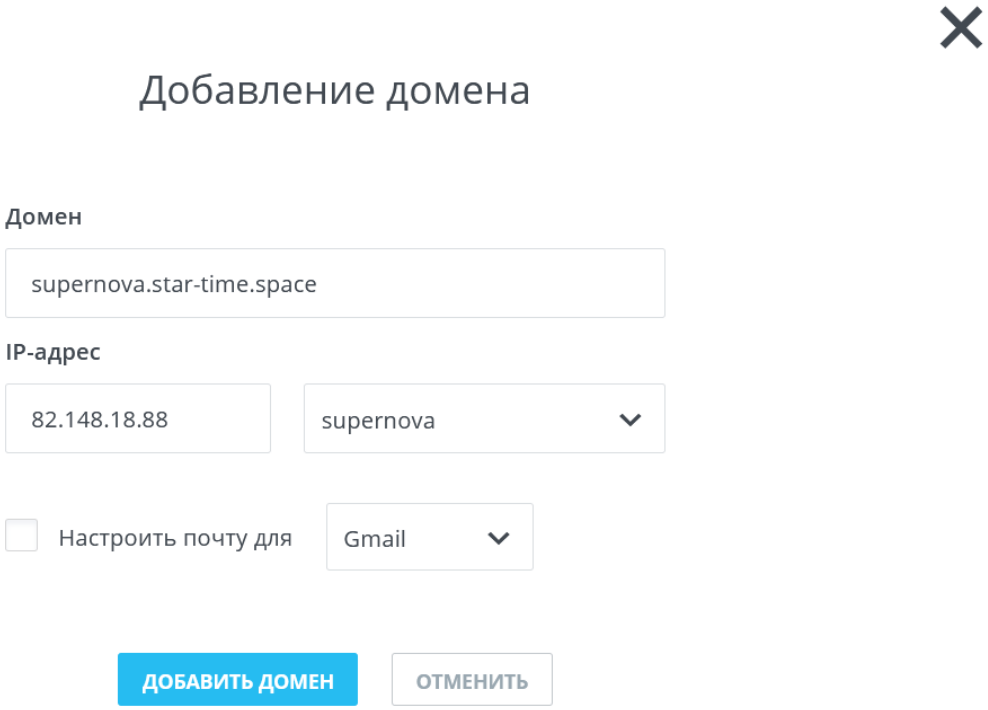

Рисунок 19 – Добавление домена.

Теперь попробуем подключиться по DNS имени. Вводим в Bitvise SSH Client все те же данные, но уже вместо IP адреса указываем DNS имя supernova.star-time.space. Подключение (Рис.20). Можно заметить, что теперь сверху вместо IP адреса указанно DNS имя.

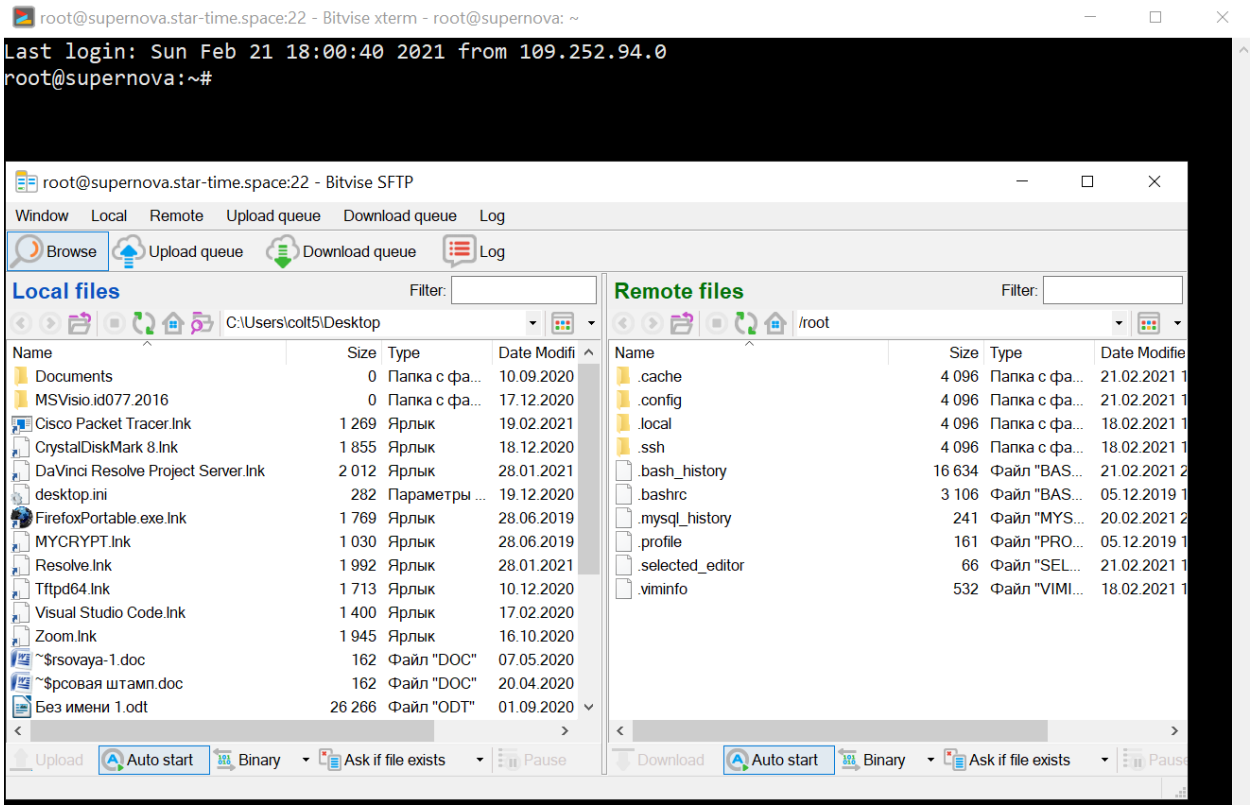

Рисунок 20 – Подключение по доменному имени.

2.2 Настройка Open VPN.

Подключаемся по ssh к серверу (Рис.21).

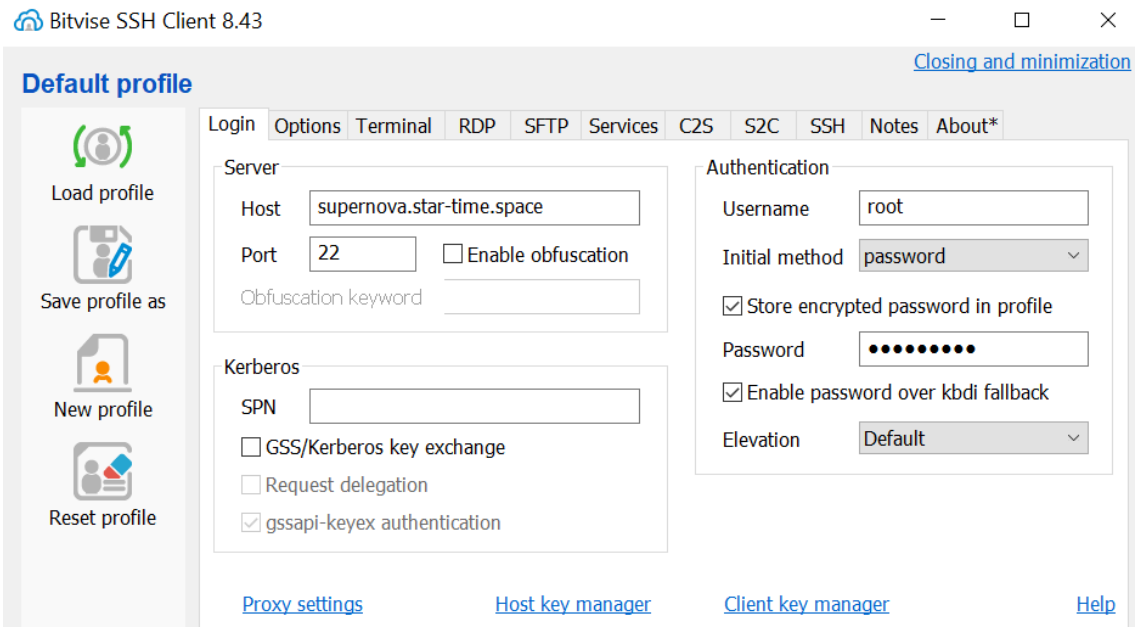

Рисунок 21 – Подключение.

Для развертывания Open VPN необходимо установить следующие пакеты, а именно openvpn, easy-rsa iptables и bash-completion, в данном случае нужные пакеты были установлены заранее (Рис.22). Пакет openvpn это и есть сам vpn. Пакет easy-rsa необходим для генерации ключей и создания центра сертификации. Пакет iptables необходим для создания необходимых для работы нашего сервера разрешающих правил “для открытия портов”. Пакет bash-copletion нужен для того, чтобы дописывались команды по нажатию на “Tab” по умолчанию на хостинг серверах его часто нет. Для установки пакетов в дистрибутиве Ubuntu есть утилита apt “пакетный менеджер”.

```
root@supernova:~# apt install openvpn easy-rsa iptables bash-completion
Reading package lists... Done
Building dependency tree
Reading state information... Done
bash-completion is already the newest version (1:2.10-1ubuntu1).
iptables is already the newest version (1.8.4-3ubuntu2).
openvpn is already the newest version (2.4.7-1ubuntu2).
easy-rsa is already the newest version (3.0.6-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@supernova:~#
```

Рисунок 22 – Установка необходимых пакетов.

Создание центра сертификации. Для начала необходимо создать директорию, где будет находиться центр сертификации в нашем случае центр сертификации находиться на одном хостинге с Open VPN, поэтому для удобства расположим центр сертификации в директории, где будут находиться конфиги Open VPN. Хотя с точки зрения безопасности настоятельно рекомендуется использовать два выделенных сервера один для Open VPN другой для центра сертификации, но в тоже время нужно придерживаться принципа рациональности, для решения данной задачи не имеет смысла переплачивать за дополнительный хостинг. После создания директории копируем в созданную директорию папку easy-rsa со всем содержимым. Выведем списком содержимое директории чтобы убедиться, что копирование произошло (Рис.23).

```
root@supernova:~# mkdir /etc/openvpn/easy-rsa
root@supernova:~# cd /etc/openvpn/easy-rsa
root@supernova:/etc/openvpn/easy-rsa# cp -R /usr/share/easy-rsa /etc/openvpn/
root@supernova:/etc/openvpn/easy-rsa# ls -l
total 72
-rwxr-xr-x 1 root root 48730 Feb 18 10:38 easyrsa
-rw-r--r-- 1 root root  4651 Feb 18 10:38 openssl-easyrsa.cnf
-rw-r--r-- 1 root root  8576 Feb 18 10:38 vars.example
drwxr-xr-x 2 root root  4096 Feb 18 10:38 x509-types
root@supernova:/etc/openvpn/easy-rsa#
```

Рисунок 23 – Создание директории для центра сертификации.

Теперь произведем инициализацию центра сертификации, использую команду “./easyrsa init-pki”. А также создадим корневую пару ключей командой ./easyrsa buid-ca. При создании закрытого ключа центра сертификации предлагается ввести пароль. Стоит ввести пароль, так как если закрытый ключ попадет злоумышленнику, то это может привести к компрометации всех открытых ключей и сертификатов. Также указываем в поле “Common Name” наименование центра сертификации в данном случае назовем центр сертификации именем сервера, а именно “supernova” (Рис.24).

После выполнения данных команд произойдет создание директории центра сертификации. Публичный сертификат будет расположен в директории pki, а закрытый ключ в директории pki/private. Приватный ключ, не стоит не при каких обстоятельствах передавать по открытым каналам связи, также по-хорошему стоит ограничить права на этот ключ.

```
root@supernova:/etc/openvpn/easy-rsa# ./easyrsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/easy-rsa/pki

root@supernova:/etc/openvpn/easy-rsa# ./easyrsa build-ca

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020

Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................+++++
.......+++++
e is 65537 (0x010001)
Can't load /etc/openvpn/easy-rsa/pki/.rnd into RNG
139817897375040:error:2406F079:random number generator:RAND_load_file:Cannot open file:../cry
pto/rand/randfile.c:98:Filename=/etc/openvpn/easy-rsa/pki/.rnd
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:supernova

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/etc/openvpn/easy-rsa/pki/ca.crt
```

Рисунок 24 – Инициализации центра сертификации.

После командой “./easyrsa gen-dh” создадим файл параметров “Диффи-Хеллмана” dh.pem (Рис.25). Данный файл также будет располагаться в директории pki.

```
root@supernova:/etc/openvpn/easy-rsa# ./easyrsa gen-dh

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.....
.....+.....+.....
.....+...
.....
.....+.....+.....+...
.....
.....
.....+.....
....+...+.....+.....
```

Рисунок 25 – Создание файла параметров Диффи-Хеллмана.

Поскольку будет использоваться TLS авторизация то для предотвращения DoS атак по протоколу UDP необходимо сгенерировать ключ Hash-based Message Authentication Code (HMAC) (Рис.26). Данный ключ будет расположен также в директории pki.

```
root@supernova:/etc/openvpn/easy-rsa# openvpn --genkey --secret pki/ta.key
root@supernova:/etc/openvpn/easy-rsa#
```

Рисунок 26 – Создание ключа HMAC.

Для отзыва уже сгенерированных сертификатов необходимо создать сертификат отзыва. В нашем случае он не нужен, но мы его создадим на всякий случай (Рис.27). Сертификат отзыва необходим для возможности принудительного отключение от частной сети клиента, к примеру, уволенных сотрудников компании. Ключ crl.pem создается в директории pki.

```
root@supernova:/etc/openvpn/easy-rsa# sudo ./easyrsa gen-crl

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/easy-rsa/pki/private/ca.key:

An updated CRL has been created.
CRL file: /etc/openvpn/easy-rsa/pki/crl.pem

root@supernova:/etc/openvpn/easy-rsa#
```

Рисунок 27 – Создание сертификата отзыва.

Теперь можно перейти к созданию сертификатов используемых Open VPN сервером. Командой “./easyrsa build-server-full supernva nopass” создаются все необходимы ключи для настройки Open VPN (Рис.28). Название supernova в команде это название сервера, а параметр nopass отключает введение пароля.

```
root@supernova:/etc/openvpn/easy-rsa# ./easyrsa build-server-full supernova nopass

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Generating a RSA private key
..................+++++
.................................+++++
writing new private key to '/etc/openvpn/easy-rsa/pki/private/supernova.key.yWQQ6tgsE1'
-----
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'supernova'
Certificate is to be certified until Feb  3 11:35:26 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated
```

Рисунок 28 – Создание сертификатов для сервера.

На данном этапе все ключи и сертификаты для настройки сервера созданы и можно перейти, к конфигурированию сервера, но для начала скопируем ключи в директорию, где будет лежать конфигурация сервера для удобства прописывания путей до ключей и сертификатов (Рис.29-Рис.30).

```
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/ca.crt /etc/openvpn/ca.crt
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/dh.pem /etc/openvpn/dh.pem
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/crl.pem /etc/openvpn/crl.pem
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/ta.key /etc/openvpn/ta.key
```

Рисунок 29 – Копирование ключей и сертификатов.

```
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/issued/supernova.crt /etc/openvpn/supernova.crt
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/private/supernova.key /etc/openvpn/supernova.key
```

Рисунок 30 – Копирование ключа и сертификата сервера.

Теперь перейдем к с созданию конфигурации сервера. Для этого с помощью команды nano в директории /etc/openvpn создадим файл server.conf (Рис.31).

Вначале указываем поле port и порт, на котором будет работать наш Open VPN сервер, в данном случае указываем порт 1194. Дальше необходимо указать протокол в нашем случае указываем протокол UDP. После в полу dev необходимо указать тип используемого интерфейса указываем интерфейс типа tun. Далее подряд указываем сгенерированные сертификаты и ключ в также параметры Диффи-Хаффмана. В поле server указывается сеть, используемую Open VPN, указываем сеть 10.10.10.0 255.255.255.0.

Далее указываем расположение до файла, где будут храниться соответствия между IP адресом и хостом, для того чтобы одним и тем же клиентам выдавать один и тот же IP адрес. Эта директива очень важна для решения нашей практической задачи, так как NAS сервер, на котором будет расположен NextCloud, не должен иметь динамический IP адрес для правильной переадресации запросов на него.

С помощью директивы push указываем те параметры, которые должны передаться клиенту в первом push указываем так чтобы все запросы шли через сервер, а также, чтобы динамически клиенту выдавался IP адрес. Вторым push указываем адрес DNS сервера. После в поле keepalive указываем через пробел два значения в первом значение как часто отправлять ping во втором значение через, сколько секунд перезапускать туннель.

После указываем ключ HMAC, который генерировался для защиты от DoS атак по протоколу UDP, после названия файла ставиться цифра 0 она указывает, что это конфигурация сервера. В поле cipher указываем тип шифрования, используем тип шифрования AES-256-CBC.

Ниже идут два параметра persist-key и persist-tun эти два параметра позволяют в случае разрыва соединения восстановиться соединению без повторного запуска Open VPN сервера. Эти параметры также позволяют хранить ключи в оперативной памяти, и это позволяет при смене пользователя, на не root пользователя, нормально функционировать Open VPN серверу.

Параметр copm-lzo позволяет использовать в туннеле сжатие трафика, что положительно сказывается на скорости работы сервера.

Далее двумя строками задаем параметры логов первой строкой директорию, в которой будут храниться логи о состоянии Open VPN, второй строкой уровень логирования в нашем случае это уровень 3.

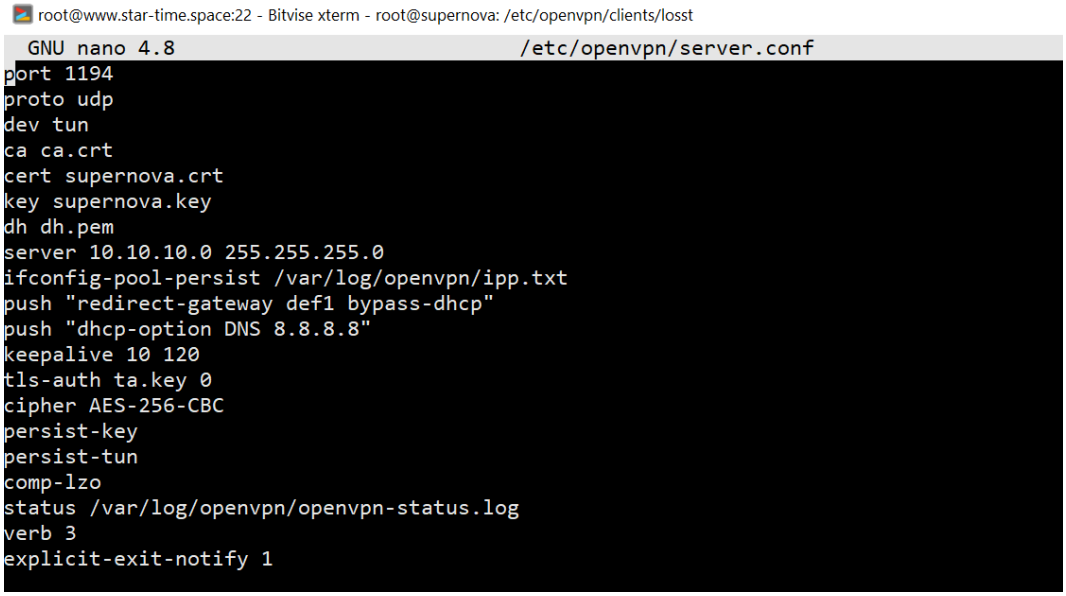

Рисунок 31 – Конфигурация Open VPN сервера.

Проверить конфигурацию можно запустив Open VPN сервер. Как видно из скриншота все работает (Рис.32)

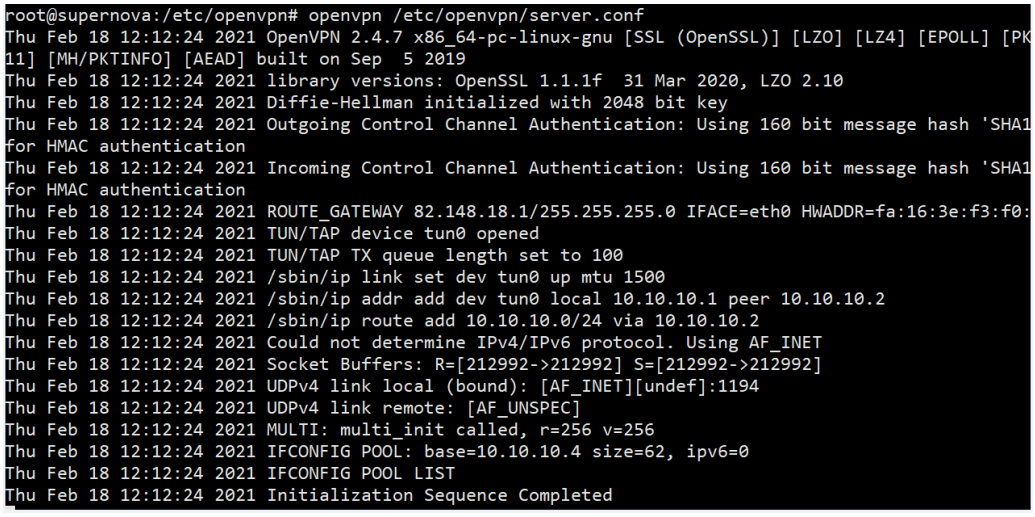

Рисунок 32 – Запуск Open VPN сервера.

Для того чтобы убедиться, что сервер поднялся, посмотрим командой 'ip a' конфигурацию интерфейсов (Рис.33). Как видно из скриншота появился интерфейс tun0 с IP адресом из указанной сети в конфигурационном файле.

```
root@supernova:/etc/openvpn# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether fa:16:3e:f3:f0:a6 brd ff:ff:ff:ff:ff:ff
    inet 82.148.18.88/24 brd 82.148.18.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fef3:f0a6/64 scope link
       valid_lft forever preferred_lft forever
10: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group defa
ult qlen 100
    link/none
    inet 10.10.10.1 peer 10.10.10.2/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::464e:78ec:81e6:3488/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
root@supernova:/etc/openvpn#
```

Рисунок 33 – Конфигурация интерфейсов.

Подними Open VPN с помощью systemd, для этого введем команду systemctl start openvpn@server. Где server название конфигурационного файла, расширение .conf можно опускать. После проверим статус сервера командой systemctl status openvpn@server (Рис.34).

```
root@supernova:/etc/openvpn# systemctl start openvpn@server
root@supernova:/etc/openvpn# systemctl status openvpn@server
● openvpn@server.service - OpenVPN connection to server
     Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)
     Active: active (running) since Thu 2021-02-18 12:14:38 UTC; 35s ago
       Docs: man:openvpn(8)
             https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
             https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 44500 (openvpn)
     Status: "Initialization Sequence Completed"
      Tasks: 1 (limit: 504)
     Memory: 1.0M
     CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
             └─44500 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 1

Feb 18 12:14:38 supernova ovpn-server[44500]: /sbin/ip addr add dev tun0 local 10.10.10.1 peer 10
Feb 18 12:14:38 supernova ovpn-server[44500]: /sbin/ip route add 10.10.10.0/24 via 10.10.10.2
Feb 18 12:14:38 supernova ovpn-server[44500]: Could not determine IPv4/IPv6 protocol. Using AF_IN
Feb 18 12:14:38 supernova ovpn-server[44500]: Socket Buffers: R=[212992->212992] S=[212992->21299
Feb 18 12:14:38 supernova ovpn-server[44500]: UDPv4 link local (bound): [AF_INET][undef]:1194
Feb 18 12:14:38 supernova ovpn-server[44500]: UDPv4 link remote: [AF_UNSPEC]
Feb 18 12:14:38 supernova ovpn-server[44500]: MULTI: multi_init called, r=256 v=256
Feb 18 12:14:38 supernova ovpn-server[44500]: IFCONFIG POOL: base=10.10.10.4 size=62, ipv6=0
Feb 18 12:14:38 supernova ovpn-server[44500]: IFCONFIG POOL LIST
Feb 18 12:14:38 supernova ovpn-server[44500]: Initialization Sequence Completed
```

Рисунок 34 – Состояние Open VPN.

Теперь необходимо создать правила в Iptables для корректной работы сервера. Первым правилом прописываем настройку NAT для того, чтобы предоставить сети VPN доступ к Internet. Далее создаем правила для открытия порта для ssh и портов для HTTP и HTTPS. После сохраняем созданные правила в файл /etc/iptables.rules (Рис.35).

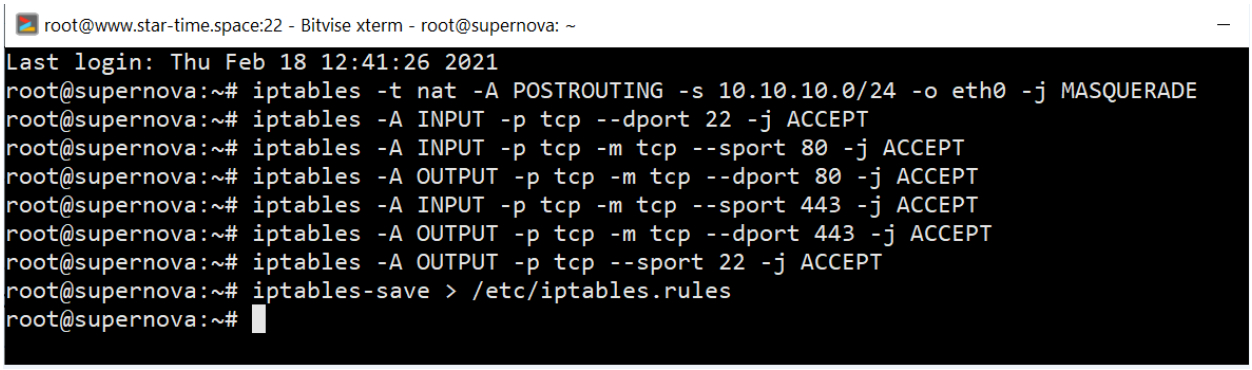

Рисунок 35 – Создание правил.

После перезагрузки по умолчанию правила Iptables отчищаются для того чтобы званого не прописывать данные правила поставим их в автозагрузку. Для этого перейдем в конфигурационный файл интерфейсов, и в конце файла напишем команду pre-up данная команда говорит о том, что в момент загрузки ОС выполнить следующую команду (Рис.36). Первым pre-up указываем, чтобы в Iptables загружались правила из файла /etc/iptables.rules. Во второй pre-up прописываем команду, которая позволит включить ipv4 маршрутизации в ОС Linux это необходимо для корректной работы NAT, эта настройка также после перезагрузки ОС сбрасывается.

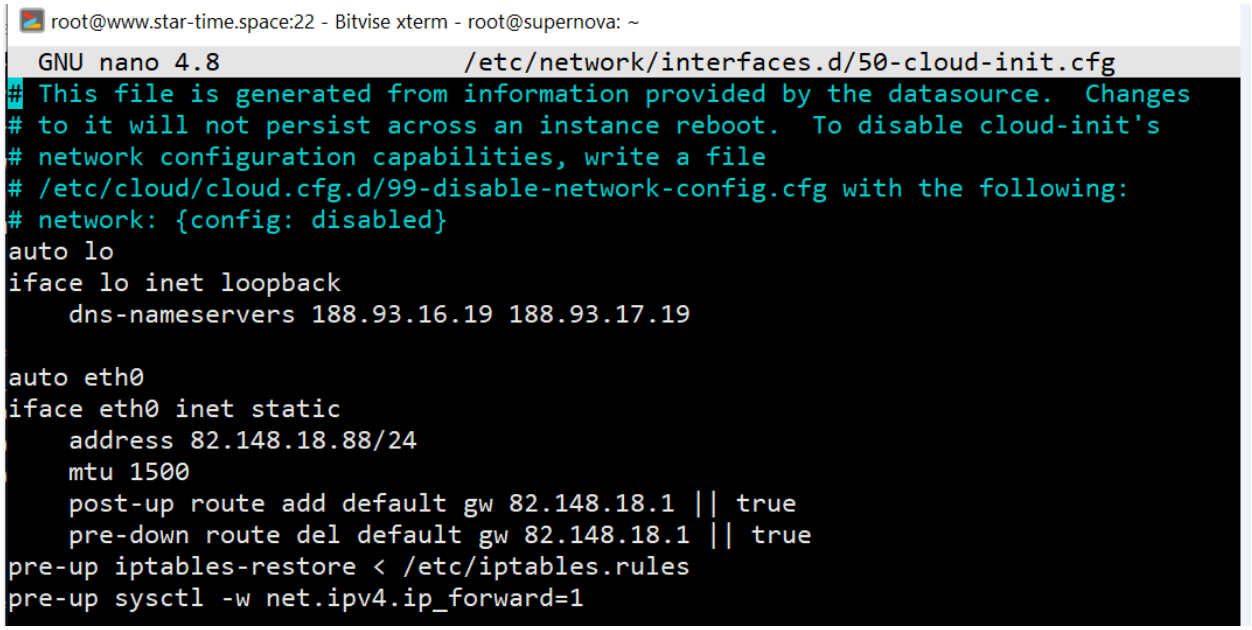

Рисунок 36 – Автоматический запуск команд перезагрузки ОС.

На данном этапе все готово для генерации ключей и сертификатов клиента. Перейдем в директорию центра сертификации и запустим команду ./easyrsa build-client-full CLINextCloud nopass. Этой командой мы обращаемся к центру сертификации для генерации ключей и сертификатов для клиента CLINextCloud. Параметр nopass опускает пароль. После генерации выводятся пути, куда сохранены сертификаты и ключи (Рис.37).

```
root@supernova:~# cd /etc/openvpn/easy-rsa/
root@supernova:/etc/openvpn/easy-rsa# ./easyrsa build-client-full CLINextCloud nopass

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Generating a RSA private key
.....................................................................+++++
.....................................................................+++++
writing new private key to '/etc/openvpn/easy-rsa/pki/private/CLINextCloud.key.B8OvIHz7ws'
-----
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'CLINextCloud'
Certificate is to be certified until Feb  7 12:27:26 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated
root@supernova:/etc/openvpn/easy-rsa#
```

Рисунок 37 – Генерация ключей клиента.

Для удобства создадим директорию clients, в ней будем хранить директории с названием клиентов, в которых будут все нужные ключи и сертификаты. Создаем директорию CLINextCloud. Переходим в директорию и копируем туда необходимые ключи и сертификаты. После командой ls выводим содержимое директории (Рис.38).

```
root@supernova:/etc/openvpn# mkdir clients
root@supernova:/etc/openvpn# cd clients/
root@supernova:/etc/openvpn/clients# mkdir CLINextCloud
root@supernova:/etc/openvpn/clients# cd CLINextCloud/
root@supernova:/etc/openvpn/clients/CLINextCloud# cp /etc/openvpn/easy-rsa/pki/ca.crt /etc/openvpn/clients/CLINextCloud/
root@supernova:/etc/openvpn/clients/CLINextCloud# cp /etc/openvpn/easy-rsa/pki/ta.key /etc/openvpn/clients/CLINextCloud/
root@supernova:/etc/openvpn/clients/CLINextCloud# cp /etc/openvpn/easy-rsa/pki/issued/CLINextCloud.crt /etc/openvpn/clients/CLINex
tCloud/
root@supernova:/etc/openvpn/clients/CLINextCloud# cp /etc/openvpn/easy-rsa/pki/private/CLINextCloud.key /etc/openvpn/clients/CLINe
xtCloud/
root@supernova:/etc/openvpn/clients/CLINextCloud# ls
CLINextCloud.crt  CLINextCloud.key  ca.crt  ta.key
root@supernova:/etc/openvpn/clients/CLINextCloud#
```

Рисунок 38 – Директория для хранения конфигураций клиентов.

Теперь необходимо создать конфигурацию клиента. Для этого командой nano CLINextCloud.conf создадим файл конфигурации (Рис.39).

Первой директивой client мы указываем программе Open VPN что дальше идет конфигурация именно клиента. После указываем тип интерфейса tun. Протокол UDP.

Далее указываем адрес сервера Open VPN, так как уже есть связанное доменное имя с хостингом, то укажем именно его. Дальше конфигурация в большей степени аналогично конфигурации сервера.

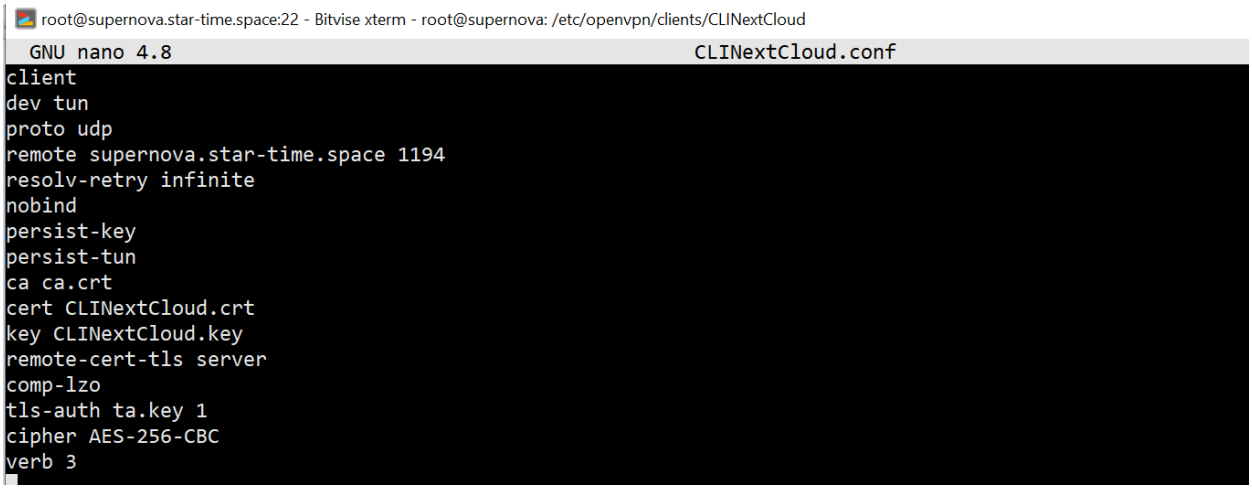

Рисунок 39 – Конфигурация клиента.

С помощью протокола SFTP перенесем конфигурационный файл и файлы ключей и сертификатов на локальную машину. Потом подключимся по Betvice SSH Client к NAS серверу и с помощью SFTP перенесем на него файлы конфигурации (Рис.40 – Рис.41).

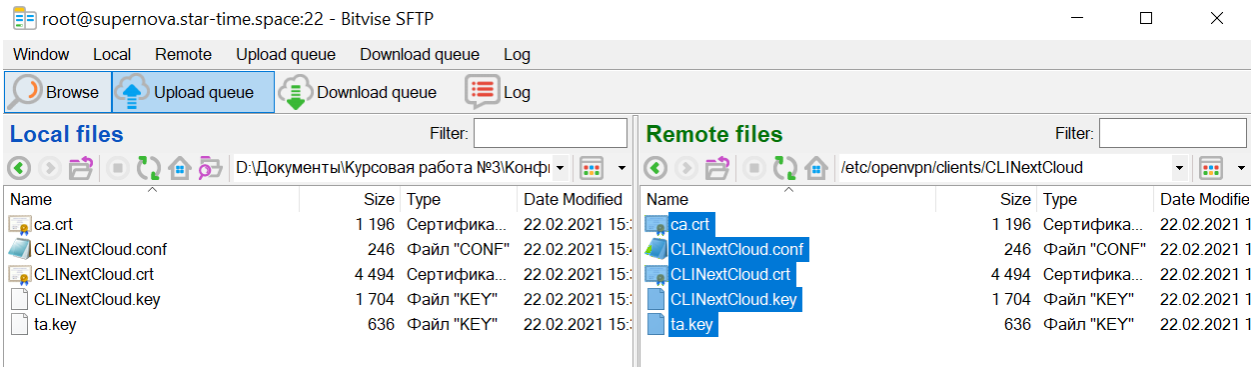

Рисунок 40 – Перенос конфигурации на локальную машину.

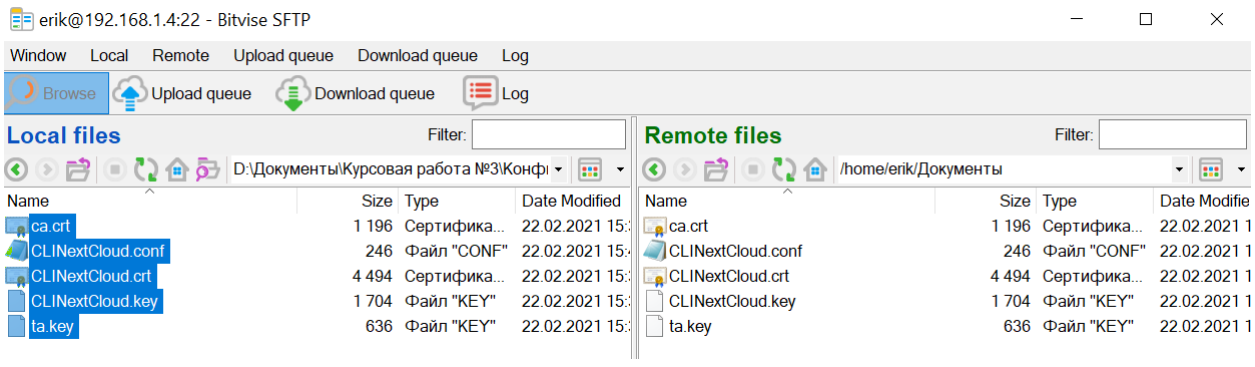

Рисунок 41 – Перенос конфигурации на NAS.

Устанавливаем пакет openvpn на NAS сервер (Рис.42).

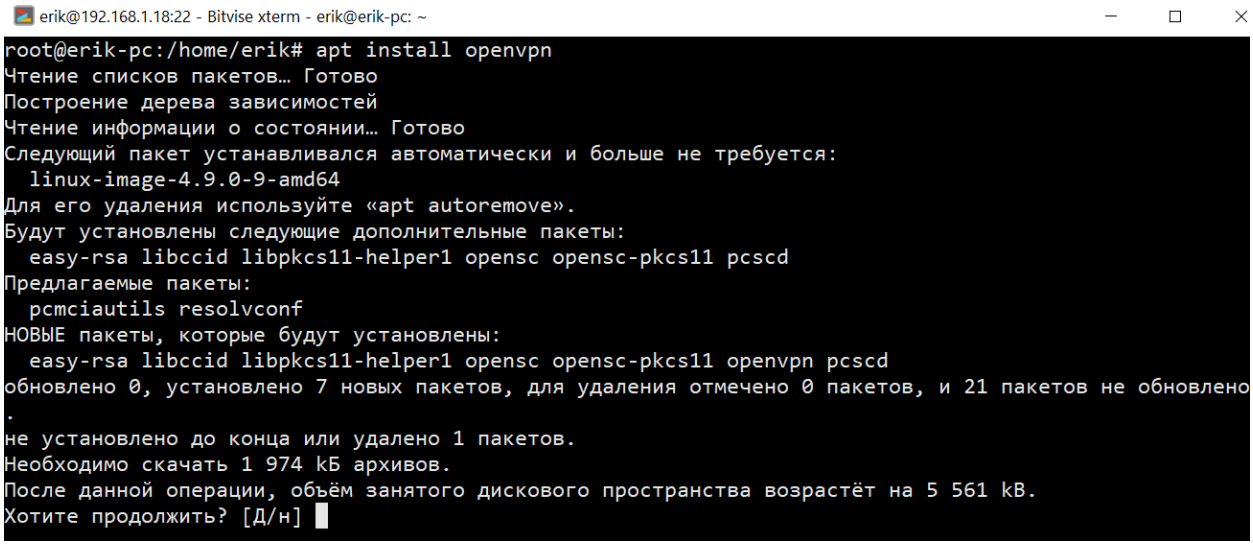

Рисунок 42 – Установка.

Копируем перенесенную конфигурацию в директорию /etc/openvpn/client (Рис.43)

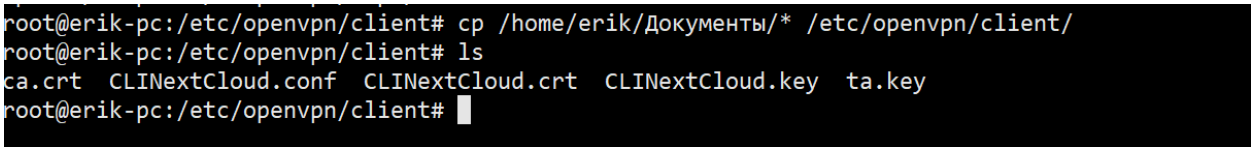

Рисунок 43 – Копирование конфигурации клиента.

Запустим Open VPN на клиенте (Рис.44).

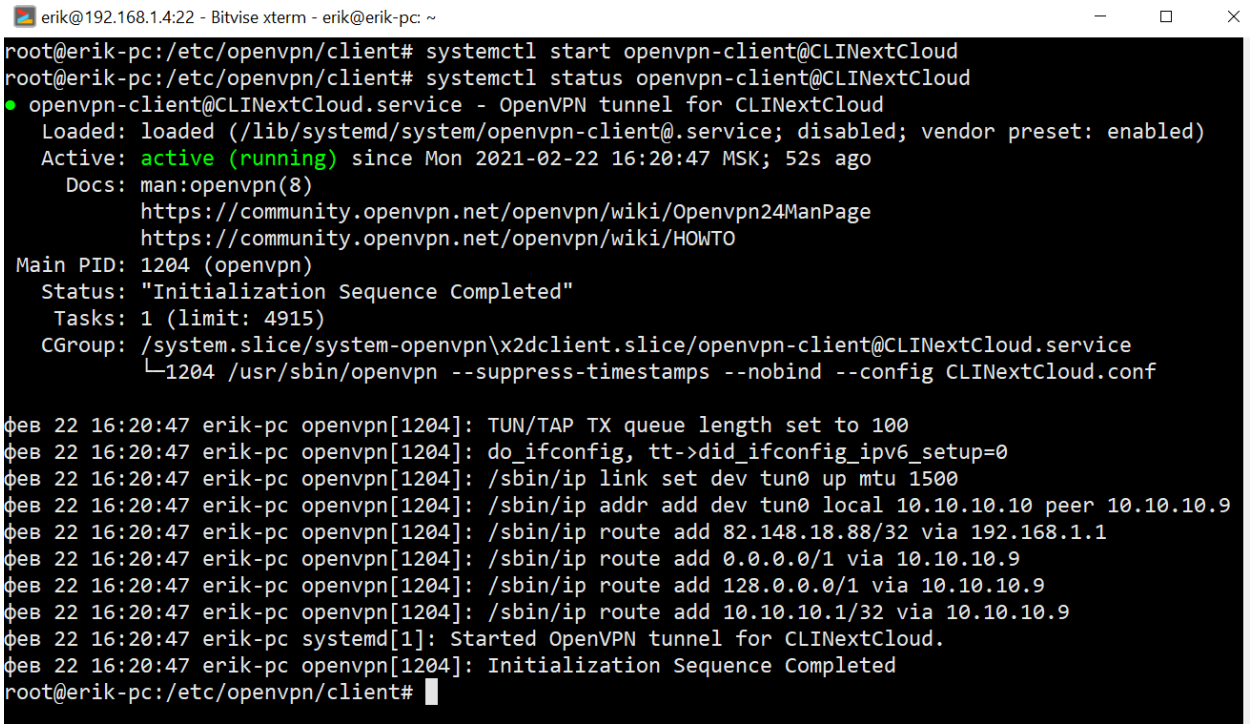

Рисунок 44 – Запуск Openvpn-client.

Проверяем поднятый интерфейс командой ip a. Также проверим доступность Open VPN сервера командой ping (Рис.45).

```
7: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group def
ault qlen 100
    link/none
    inet 10.10.10.10 peer 10.10.10.9/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::de57:7720:81d:a6ab/64 scope link flags 800
       valid_lft forever preferred_lft forever
root@erik-pc:/etc/openvpn/client# ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
64 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=6.92 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=6.89 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=5.78 ms
64 bytes from 10.10.10.1: icmp_seq=4 ttl=64 time=5.86 ms
64 bytes from 10.10.10.1: icmp_seq=5 ttl=64 time=14.2 ms
64 bytes from 10.10.10.1: icmp_seq=6 ttl=64 time=5.70 ms
^C
--- 10.10.10.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5024ms
rtt min/avg/max/mdev = 5.706/7.567/14.238/3.026 ms
root@erik-pc:/etc/openvpn/client#
```

Рисунок 45 – Работа VPN соединения.

2.3 Настройка Nginx. Получение ssl сертификата.

Первым делом установим на хостинг веб сервер nginx воспользовавшись пакетным менеджером apt (Рис.46).

```
root@supernova:~# apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libtiff5 libwebp6 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxpm4
  libxslt1.1 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libtiff5 libwebp6 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxpm4
  libxslt1.1 nginx nginx-common nginx-core
0 upgraded, 23 newly installed, 0 to remove and 6 not upgraded.
Need to get 3332 kB of archives.
After this operation, 11.6 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Рисунок 46 – Установка nginx.

Создадим новую DNS запись типа A для хостинга. Данное DNS имя будет нужно для создания виртуального хоста в nginx (Рис.47).

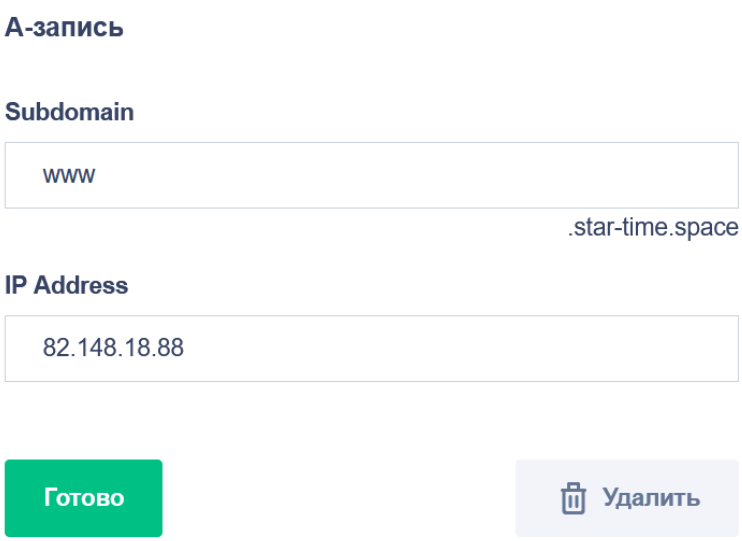

Рисунок 47 – Создание новой записи.

Далее перейдем в директорию /etc/nginx/sites-available и создадим конфигурационный файл www.star-time.space (Рис.48). В поле server_name указываем прослушиваемое доменное имя www.star-time.space. В поле proxy_pass http://10.10.10.10:8080 данная директива будет перенаправлять запросы, поступающие на хостинг на адрес NAS сервера на порт 8080.

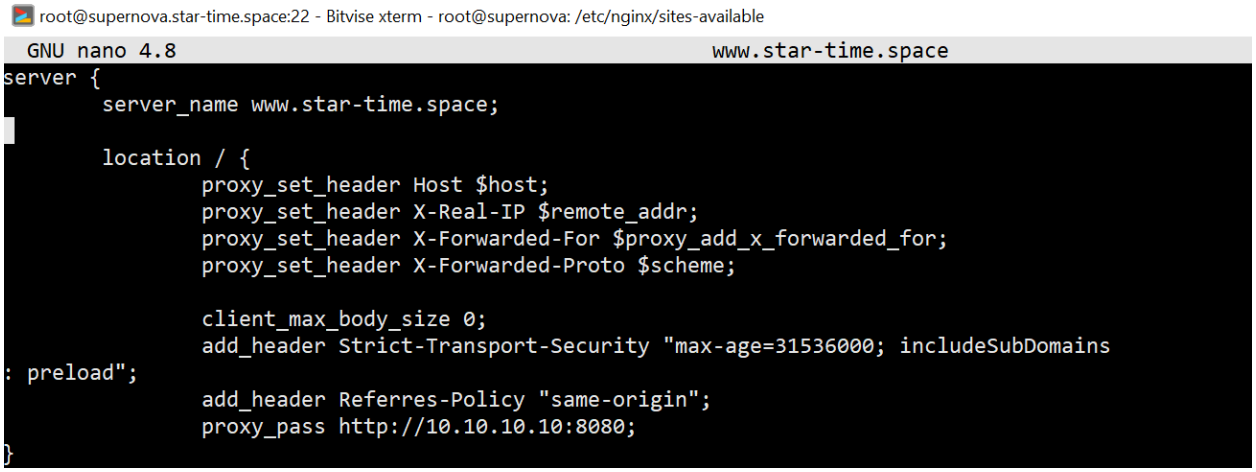

Рисунок 48 – Конфигурация.

Для получения SSL сертификата установим утилиту cerbot (Рис.49).

```
root@supernova:/etc/nginx/sites-available# apt install python3-certbot-nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  certbot python3-acme python3-certbot python3-cffi-backend python3-configargparse
  python3-configobj python3-cryptography python3-distutils python3-future python3-josepy
  python3-lib2to3 python3-mock python3-openssl python3-parsedatetime python3-pbr
  python3-pyparsing python3-requests-toolbelt python3-rfc3339 python3-setuptools python3-tz
  python3-zope.component python3-zope.event python3-zope.hookable python3-zope.interface
Suggested packages:
  python3-certbot-apache python-certbot-doc python-acme-doc python-certbot-nginx-doc
  python-configobj-doc python-cryptography-doc python3-cryptography-vectors python-future-doc
  python-mock-doc python-openssl-doc python3-openssl-dbg python-pyparsing-doc
  python-setuptools-doc
Recommended packages:
  python3-icu
The following NEW packages will be installed:
  certbot python3-acme python3-certbot python3-certbot-nginx python3-cffi-backend
  python3-configargparse python3-configobj python3-cryptography python3-distutils python3-future
  python3-josepy python3-lib2to3 python3-mock python3-openssl python3-parsedatetime python3-pbr
  python3-pyparsing python3-requests-toolbelt python3-rfc3339 python3-setuptools python3-tz
  python3-zope.component python3-zope.event python3-zope.hookable python3-zope.interface
0 upgraded, 25 newly installed, 0 to remove and 6 not upgraded.
Need to get 2003 kB of archives.
```

Рисунок 49 – Установка.

Данная утилита автоматически правит конфигурационный файл нужного нам виртуального хоста. Запуск утилиты выглядит, следующем образом (Рис.50). Для начала утилита проводит первичную регистрацию, после спрашивает, делать ли “жесткий” ридерект на https. Выбираем, чтоб был ридерект. Благодаря, данной надстройки будет перенос с http на https и открыть http будет нельзя.

```
root@supernova:/etc/nginx/sites-available# certbot --nginx -d www.star-time.space
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): colt57@mail.ru

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(A)gree/(C)ancel: A

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: N
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for www.star-time.space
Waiting for verification...
Cleaning up challenges
Deploying Certificate to VirtualHost /etc/nginx/sites-enabled/default

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
```

Рисунок 50 – Получение ssl сертификата.

Посмотрим, как теперь выглядит файл конфигурации виртуального хоста (Рис.51). В файл конфигурации добавился прослушиваемый порт 443 соответствующий протоколу HTTPS, ниже добавились данные о ssl сертификате. Также добавился еще один конфиг сервера слушающий порт 80 с условием, что при получение информации о прослушиваемом доменном имени www.star-time.space сделать перенаправление на https.

```
server {
        server_name www.star-time.space;

        location / {
                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;

                client_max_body_size 0;
                add_header Strict-Transport-Security "max-age=31536000; includeSubDomains
: preload";
                add_header Referres-Policy "same-origin";
                proxy_pass http://10.10.10.10:8080;
}
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/www.star-time.space/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/www.star-time.space/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}

server {
    if ($host = www.star-time.space) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


        server_name www.star-time.space;
    listen 80;
    return 404; # managed by Certbot
```

Рисунок .51 – Конфигурационный файл.

Перезапустим nginx и проверим status (Рис.52).

```
root@supernova:/etc/nginx/sites-available# systemctl reload nginx
root@supernova:/etc/nginx/sites-available# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2021-02-20 20:51:33 UTC; 1 day 17h ago
       Docs: man:nginx(8)
    Process: 70116 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 70128 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 116957 ExecReload=/usr/sbin/nginx -g daemon on; master_process on; -s reload (code=exited, status=0/SUCCESS)
   Main PID: 70129 (nginx)
      Tasks: 2 (limit: 504)
     Memory: 6.5M
     CGroup: /system.slice/nginx.service
             ├─ 70129 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─116958 nginx: worker process
```

Рисунок 52 – Состояние nginx.

Перейдем по доменному имени в браузере и проверим ридерект. Исходя из скриншота, ридерект работает видно соединение https (Рис.53).

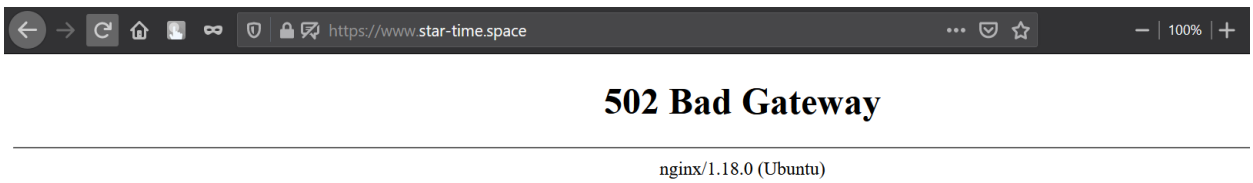

Рисунок 53 – Https соединение.

2.4 Создание контейнера Docker с Nextcloud.

Для поднятия NextCloud будет использоваться Docker контейнер. Для удобного запуска контейнера воспользуемся docker-compose. Для этого установим его (Рис.54).

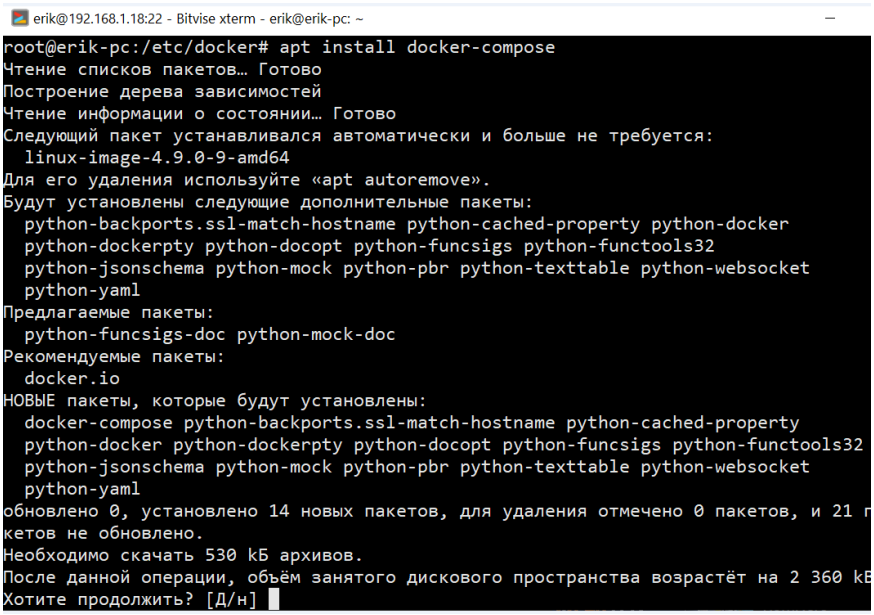

Рисунок 54 – Установка docker compose.

Создадим директорию /etc/docker, внутри директории создадим файл docker-compose.yml со следующим содержимым (Рис.55). Тут все просто данный файл говорит, что нужно поднять сервис nextcloud на адресе 10.10.10.10 с перенаправлением запросов с порта NAS сервера 8080 на порт 80 docker контейнера. А также указано, что использовать сеть docker m-network, которую позже создадим, она необходима для нормальной работы контейнера.

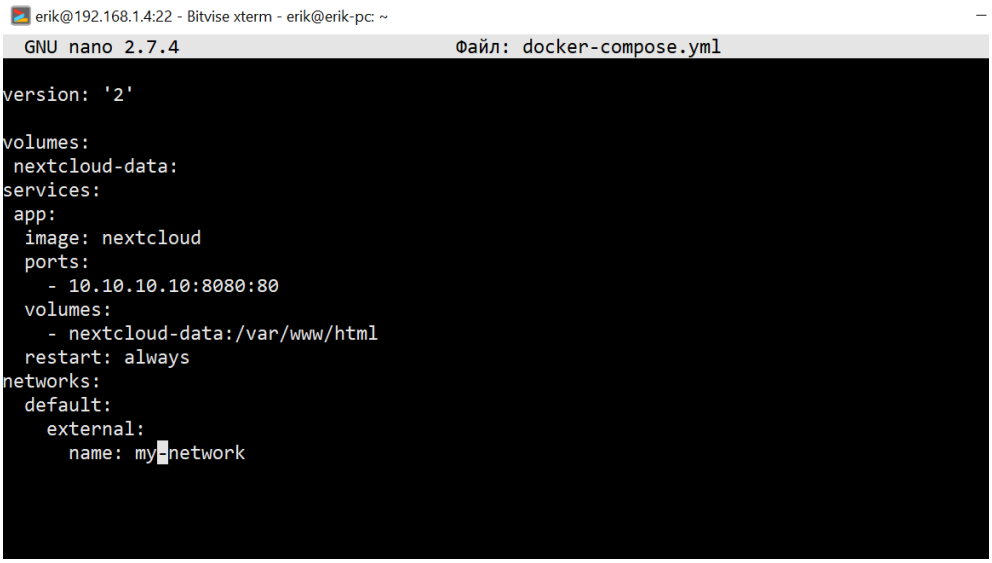

Рисунок 55 – Создание файла.

Создание сети docker (Рис.56).

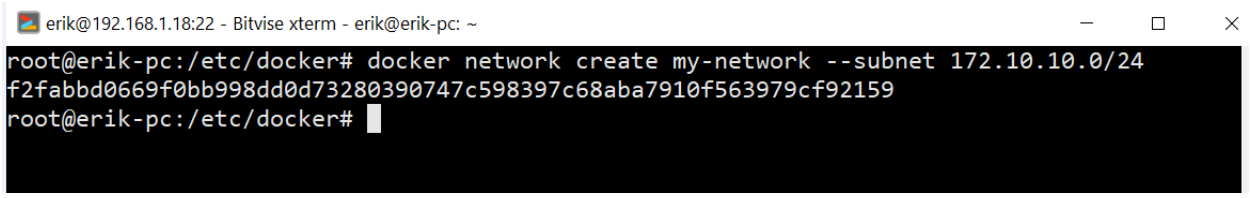

Рисунок 56 – Сеть docker.

Для запуска процесса создание контейнера с помощью docler-compose необходимо находиться в директории с конфигурационным файлом и использовать команду docker-compose up –d (Рис.57).

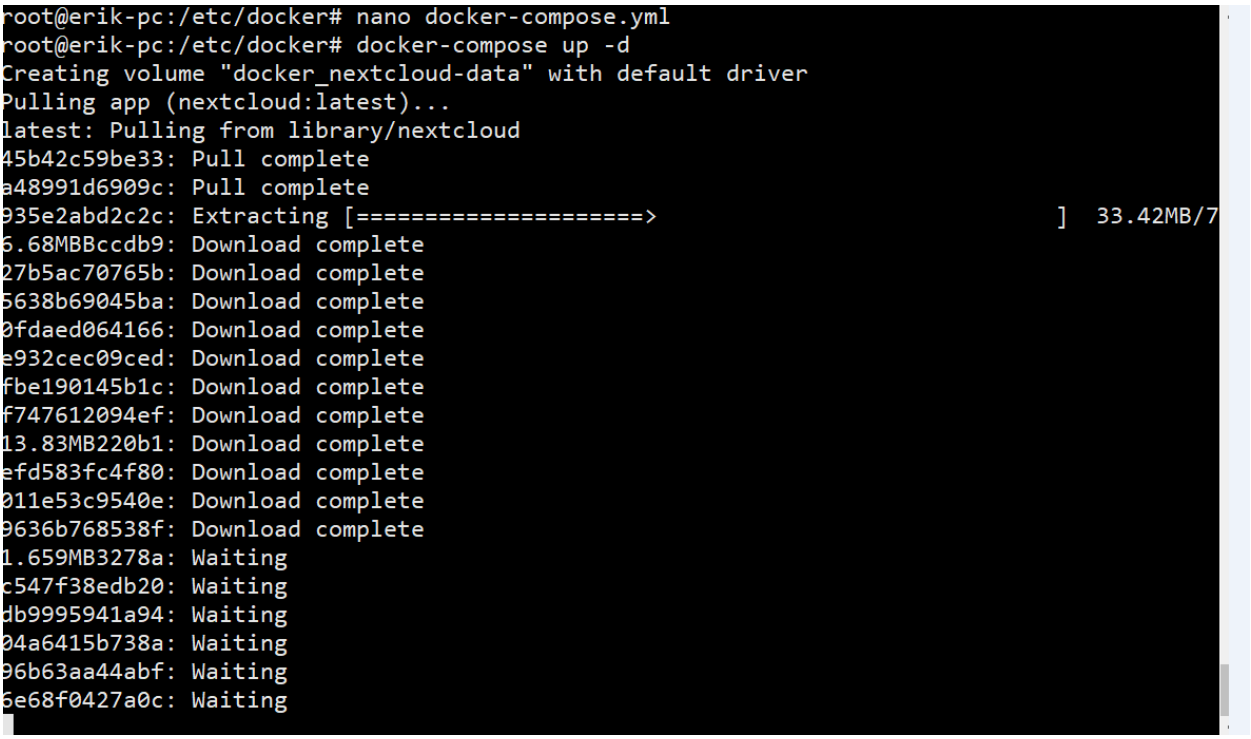

Рисунок 57 – Создание контейнера.

Посмотреть работающий контейнер можно с помощью команды docker ps (Рис.58).

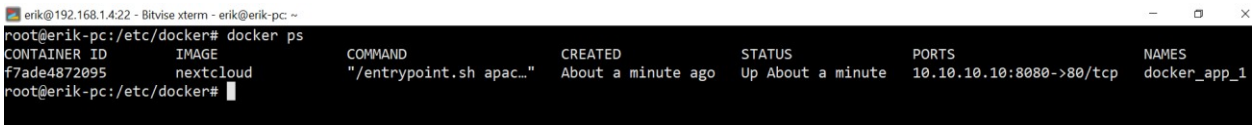

Рисунок 58 – Работающий контейнер.

Перейдем по доменному имени в браузере и увидим страничку регистрации административного пользователя NextCloud (Рис.59).

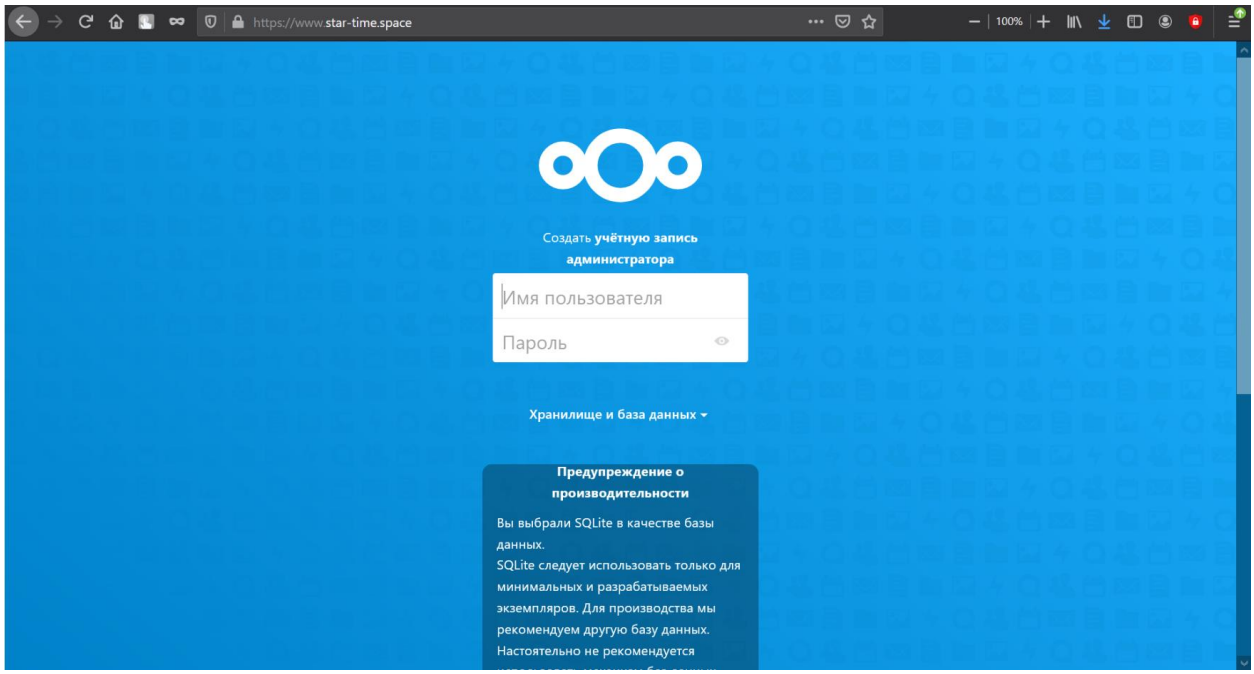

Рисунок 59 – Страница регистрации.

После прохождения регистрации (Рис.60).

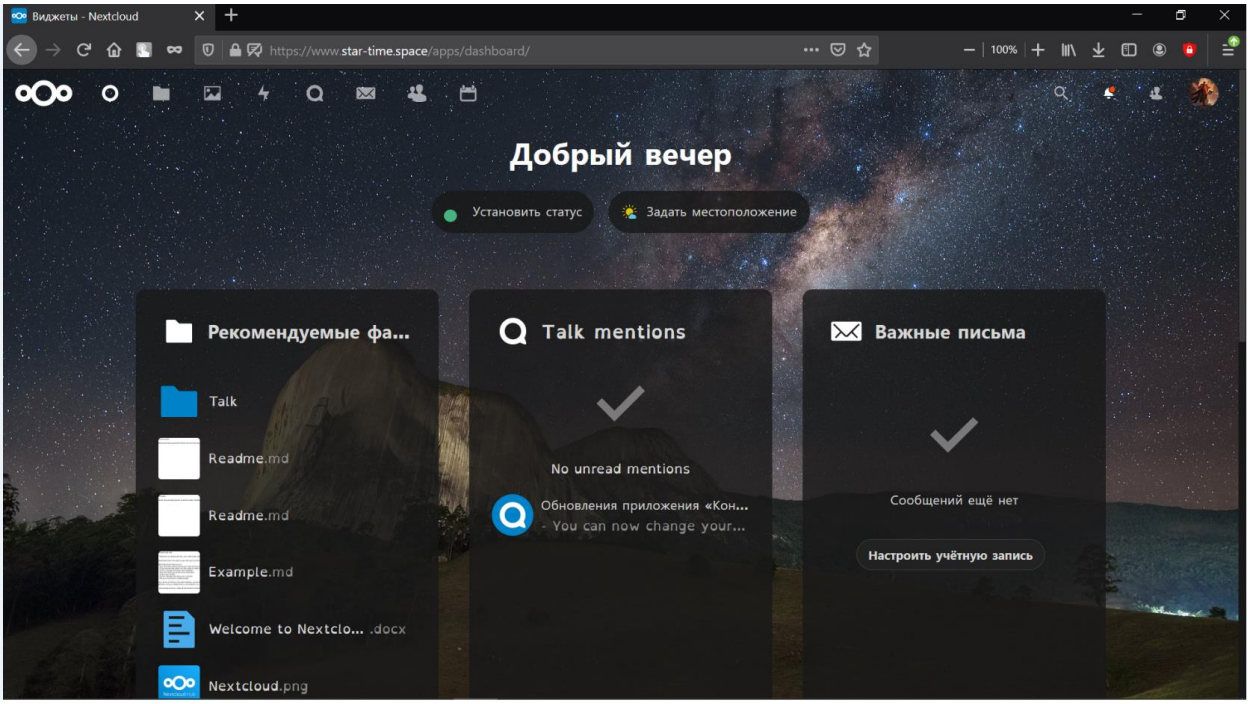

Рисунок 60 – Стартовая страница NextCloud.

На (рис.60) показана приветственная страница NextCloud.

Заключение.

В ходе выполнения курсовой работы в теоретической части были рассмотрены следующие темы: принцип работы VPN (туннелирование), различные VPN протоколы, подробно рассмотрен протокол Open VPN, тема сертификации (SSL сертификат, центры сертификации). Также были описаны разные протоколы, технологии и программное обеспечение такое как: dns, docker, Nextcloud, nginx, IPsec, http и https. Был описан алгоритм внедрения Open VPN для решения конкретной практической задачи.

В ходе решения практической задачи была произведена покупка DNS имени и хостинга. Настройка удаленного доступа к хостингу по доменному имени по протоколу ssh. Был создан центр сертификации и сгенерированы необходимые для настройки Open VPN сертификаты и ключи. Также произведена настройка Open VPN сервера и клиента. Был создан VPN туннель между NAS сервером в локальной сети и приобретенным хостингом. Для перенаправления запросов на NAS сервер был настроен Nginx. На NAS сервере был поднят docker контейнер с облаком Nextcloud. Поскольку Nextcloud оперирует такими данными как персональные данные пользователей и пароли то доступ к нему был настроен по протоколу HTTPS для этого был получен SSL сертификат в центре сертификации Let`s Encrypt.

Можно с уверенностью сказать, что поставленная задача данной курсовой работы была выполнена.

К сожалению, ограничение по размеру курсовой работы не позволяли рассмотреть и изучить тему курсовой работы глубже. Поэтому в будущем обязательно изучу эту тему еще лучше.

Подводя итог, хочется сказать, что в современном мире с активно развивающимися сетевыми технологиями специалисты, разбирающиеся области компьютерных сетей всегда будут актуальны.

Список используемых источников и литературы.

1. Авторы: Максимов Николай Вениаминович Попов Игорь Иванович Компьютерные сети (2017) — Текст: электронный — URL: https://znanium.com/catalog/document?pid=792685

2. Авторы: Кузин Александр Владимирович Кузин Дмитрий Александрович Компьютерные сети (2017) — Текст: электронный — URL: https://znanium.com/catalog/document?pid=938938

3. Компьютерные сети: расширенный начальный курс Авторы:А. А. Букатов, С. А. Гуда (2019) — Текст: электронный — URL: https://www.litres.ru/aleksandr-pyhalov-un/komputernye-seti-rasshirennyy-nachalnyy-kurs-48613245/

4. Олифер, Олифер: Компьютерные сети. Принципы, технологии, протоколы. Юбилейное издание (2020) — Текст: электронный — URL: https://www.labirint.ru/books/737421/

5. Олифер, Олифер: Компьютерные сети. Принципы, технологии, протоколы. (2019) — Текст: электронный — URL: https://www.labirint.ru/books/511422/

6. С. Грингард «Интернет вещей. «Будущее уже здесь» (2017)) — Текст: электронный — URL: https://www.alpinabook.ru/catalog/book-75330/

7. Компьютерные сети: Принципы, технологии, протоколы (2018)) — Текст: электронный — URL: https://www.bsuir.by/m/12_100229_1_85460.pdf

8. Интернет изнутри. Экосистема глобальной сети. (2017)) — Текст: электронный — URL: https://www.ozon.ru/context/detail/id/33354294/

9. Официальное руководство Cisco по подготовке к сертификационным экзаменам CCNA ICND2 200-105: маршрутизация и коммутация) — Текст: электронный — URL: https://www.ozon.ru/context/detail/id/147417590/

10. Компьютерные сети. Учебник (2018)) — Текст: электронный — URL: https://www.chitai-gorod.ru/catalog/book/1168613/

11. Linux в действии (2019). Автор - Дэвид Клинтон — Текст: электронный — URL: http://it-ebooks.ru/publ/unix/linux_in_action/14-1-0-1204

12. Is Wireguard faster than OpenVPN? We tested 114 VPN servers., 2021 — Текст: электронный — URL: https://vladtalks.tech/vpn/is-wireguard-faster-than-o

13. Reference manual for OpenVPN 2.4, 2021 2021 — Текст: https://openvpn.net/community-resources/reference-man

14. Настройка OpenVPN в Ubuntu 20.04, 2021 2021 — Текст: https://losst.ru/nastrojka-openvpn-v-ubuntu

15. Keijser J.J., OpenVPN Книга рецептов - второе издание, 2017 - https://www.packtpub.com/product/openvpn-cookbook-sec