**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра теории вероятностей и кибербезопасности**

**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

*дисциплина: Администрирование сетевых подсистем*

Студент: Чигладзе Майя Владиславовна

Студ. билет № 1132239399

Группа: НПИбд-02-23

**МОСКВА**

2025 г.

# Цель работы

# Целью данной работы является приобретение практических навыков установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

**Выполнение работы**

**2.4.1. Установка DNS-сервера**

1. Загрузила вашу операционную систему и перейдите в рабочий каталог с проектом:

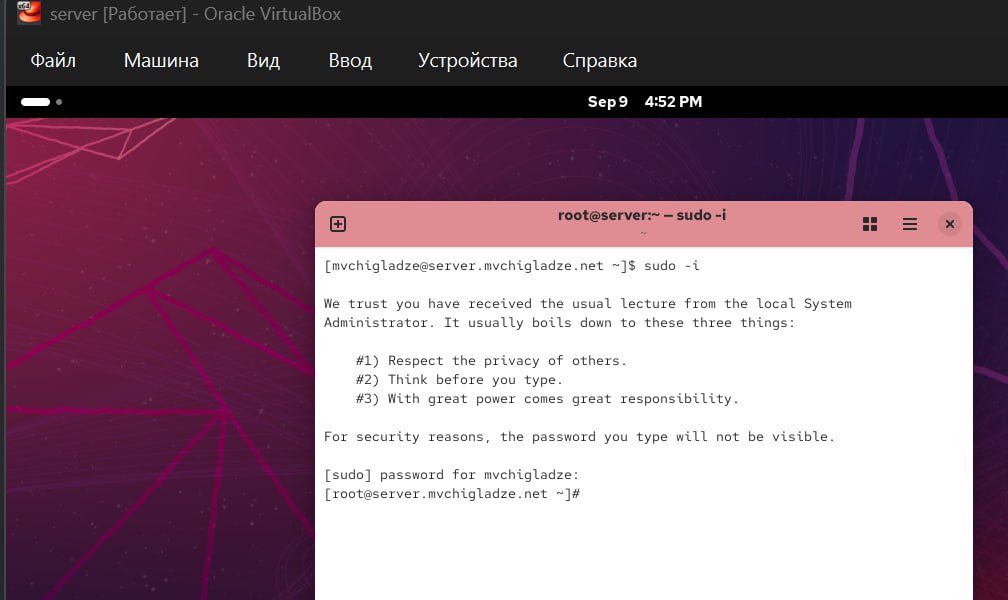
cd /var/tmp/user\_name/vagrant

2. Запустите виртуальную машину server:

make server-up

3. На виртуальной машине server войдите под созданным вами в предыдущей работе пользователем и откройте терминал. Перейдите в режим суперпользователя:

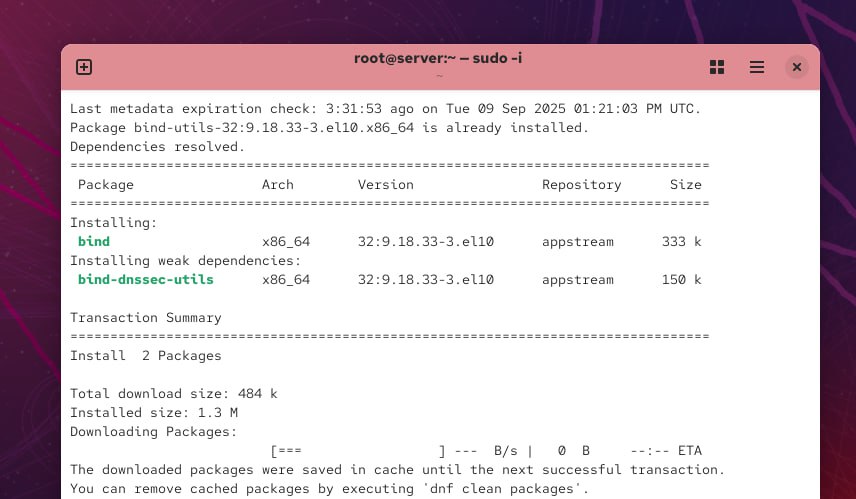
sudo -I



**Рис. 1.1**. Режим суперпользователя

4. Установите bind и bind-utils:

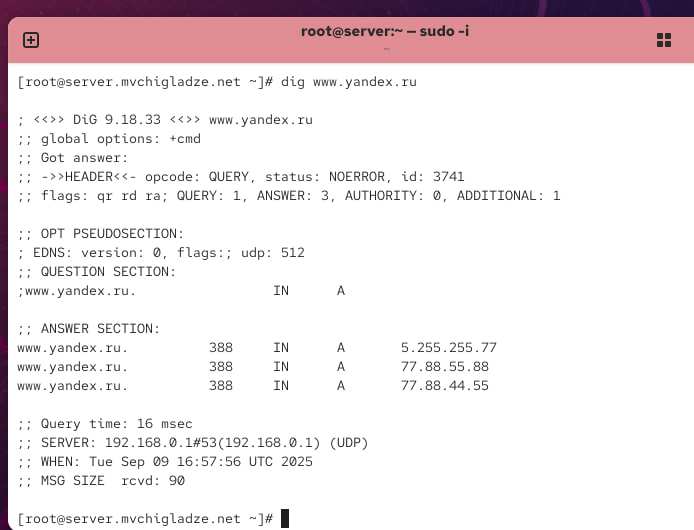
dnf -y install bind bind-utils



**Рис. 1.2**. Установите бинд

5. В качестве упражнения с помощью утилиты dig сделайте запрос, например, к DNS-адресу www.yandex.ru:

dig [www.yandex.ru](http://www.yandex.ru)



**Рис. 1.3**. Упражнения с помощью утилиты

Анализ:

1.  <<>> DiG 9.18.33 <<>> www.yandex.ru:  Версия dig и запрошенный домен.

2.  ;; Got answer:: Успешный ответ.

3.  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3741; flags: qr rd ra; ...:  Заголовок DNS-ответа: запрос типа QUERY, без ошибок (NOERROR), и флаги (ответ, рекурсия запрошена/доступна).

4.  ;; QUESTION SECTION::  Секция вопроса: запрашиваем A запись для www.yandex.ru.

5.  ;www.yandex.ru. IN A:  Конкретный вопрос: A запись для  www.yandex.ru.

6.  ;; ANSWER SECTION:: Секция ответа: содержит найденные IP-адреса.

7.  www.yandex.ru. 388 IN A 5.255.255.77 (и еще 2 строки с разными IP):  www.yandex.ru имеет IPv4 адрес 5.255.255.77 (и еще 2 других). TTL - 388 секунд.

8.  ;; Query time: 16 msec: Время запроса: 16 миллисекунд

9.  ;; SERVER: 192.168.0.1#53 (UDP): Использован DNS-сервер 192.168.0.1 по UDP.

10. ;; WHEN: Tue Sep 09 16:57:56 UTC 2025: Время запроса.

11. ;; MSG SIZE rcvd: 90: Размер полученного ответа: 90 байт.

В итоге:  dig запросил IP-адреса для www.yandex.ru, получил три разных IP-адреса от DNS-сервера 192.168.0.1 за 16 мс.

## **2.4.2. Конфигурирование кэширующего DNS-сервера**

**2.4.2. Конфигурирование кэширующего DNS-сервера**

1. В отчёте проанализируйте построчно содержание файлов /etc/resolv.conf,

/etc/named.conf, /var/named/named.ca, /var/named/named.localhost,

/var/named/named.loopback.

1. /etc/resolv.conf: Список DNS-серверов для клиента. nameserver - IP DNS, search - домены для автодополнения.

2. /etc/named.conf: Главный конфиг DNS-сервера. options - глобальные настройки (listen-on, directory, allow-transfer, recursion). zone - описание зон (корневой, localhost, reverse).

3. /var/named/named.ca: Список корневых DNS-серверов. NS и A записи корневых серверов.

4. /var/named/named.localhost: Зона localhost. SOA, NS, A (127.0.0.1), AAAA (::1) записи.

5. /var/named/named.loopback: Обратная зона для сети 127.0.0.0/8. SOA, NS, PTR (1 -> localhost) записи.

Ключевое:

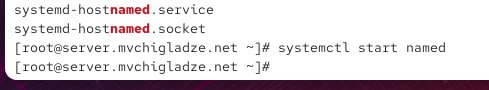
•  resolv.conf: Для кого искать DNS.

•  named.conf: Как серверу DNS работать.

•  Остальные: Базовые зоны (корень, localhost) для работы.

2. Запустила DNS-сервер:

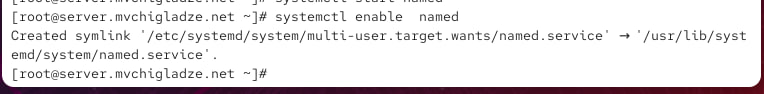
systemctl start named



**Рис. 2.2**. Запустить днс сервер

3. Включила запуск DNS-сервера в автозапуск при загрузке системы:

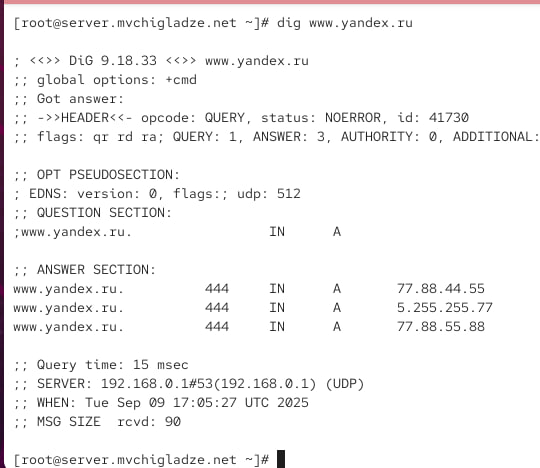
systemctl enable named



**Рис. 2.3**. Запуск днс сервера при автозапуске

4. Проанализировала в отчёте отличие в выведенной на экран информации при выполнении команд

dig [www.yandex.ru](http://www.yandex.ru) и dig @127.0.0.1 [www.yandex.ru](http://www.yandex.ru)



**Рис. 2.4**. Выполнение команд

Результат 1: dig www.yandex.ru (без указания DNS-сервера)

DNS-сервер: Использован DNS-сервер, настроенный по умолчанию в системе: 192.168.0.1. Это, скорее всего, ваш роутер.

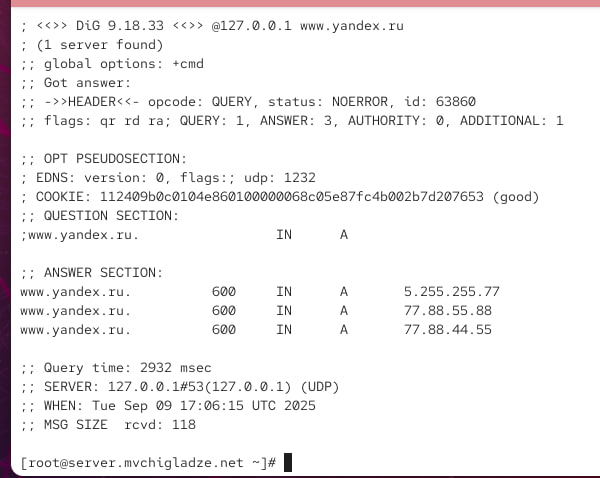
IP-адреса: Получены IP-адреса: 77.88.44.55, 5.255.255.77, 77.88.55.88.

TTL: 444 секунды.

Время запроса: 15 мсек.

Размер ответа: 90 байт.

EDNS: Версия 0, udp: 512



**Рис. 2.5**. С собакой

Результат 2: dig @127.0.0.1 www.yandex.ru (с указанием DNS-сервера 127.0.0.1)

DNS-сервер: Явно указан DNS-сервер: 127.0.0.1 (localhost). Это говорит о том, что на этой машине запущен DNS-сервер.

IP-адреса: Получены те же IP-адреса: 5.255.255.77, 77.88.55.88, 77.88.44.55. Порядок может отличаться.

TTL: 600 секунд.

Время запроса: 2932 мсек (значительно дольше!).

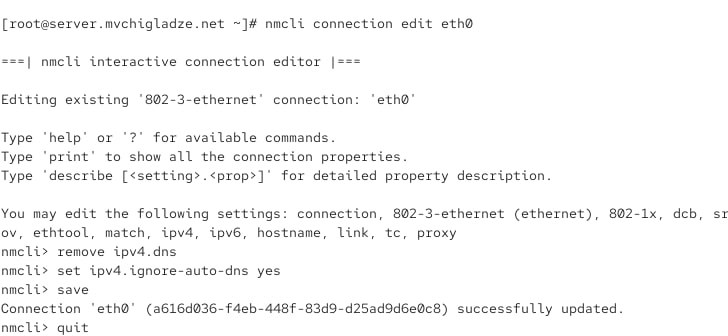
Размер ответа: 118 байт.

EDNS: Версия 0, udp: 1232, COOKIE: (что указывает на поддержку EDNS COOKIE, что служит защитой от spoofing атак)

Общий вывод:

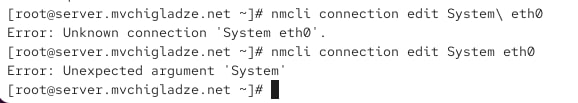
Оба запроса успешно разрешили имя www.yandex.ru. Однако, использование локального DNS-сервера (127.0.0.1) заняло значительно больше времени, чем использование DNS-сервера, настроенного по умолчанию (роутера). Это может указывать на проблемы с производительностью локального DNS-сервера, отсутствием записи в кэше, или на его более медленное соединение с другими DNS-серверами. Поддержка EDNS COOKIE свидетельствует о более продвинутой настройке и безопасности локального сервера.

1. Сделала DNS-сервер сервером по умолчанию для хоста server и внутренней виртуальной сети.



**Рис. 2.6**. Сделала днс сервер по умолчанию

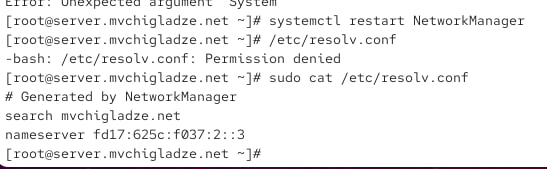
1. Сделайте тоже самое для соединения System eth0 (если оно активно)



**Рис. 2.7**. Не активно

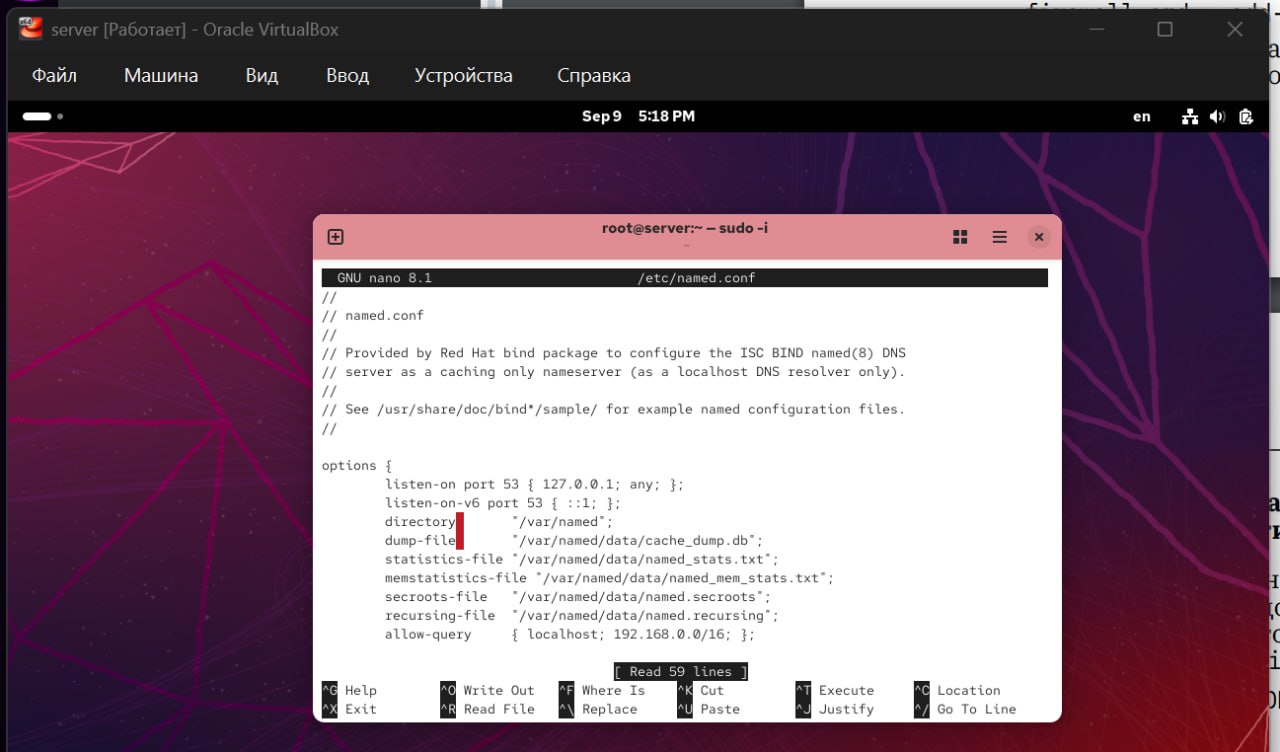
1. Перезапустите NetworkManager:

systemctl restart NetworkManager



**Рис. 2.8**. Перезапустила нетворк

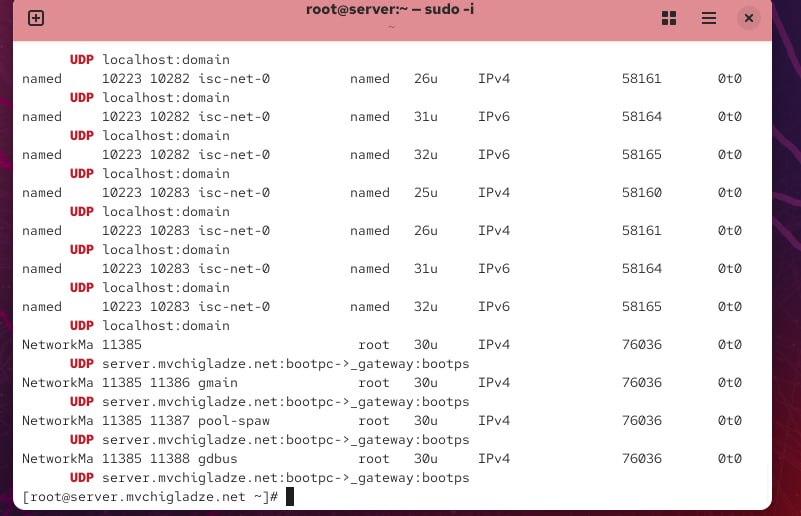
1. Требуется настроить направление DNS-запросов от всех узлов внутренней сети, включая запросы от узла server, через узел server. Для этого внесите изменения в файл /etc/named.conf, заменив строку



**Рис. 2.9**. Заменяем строку

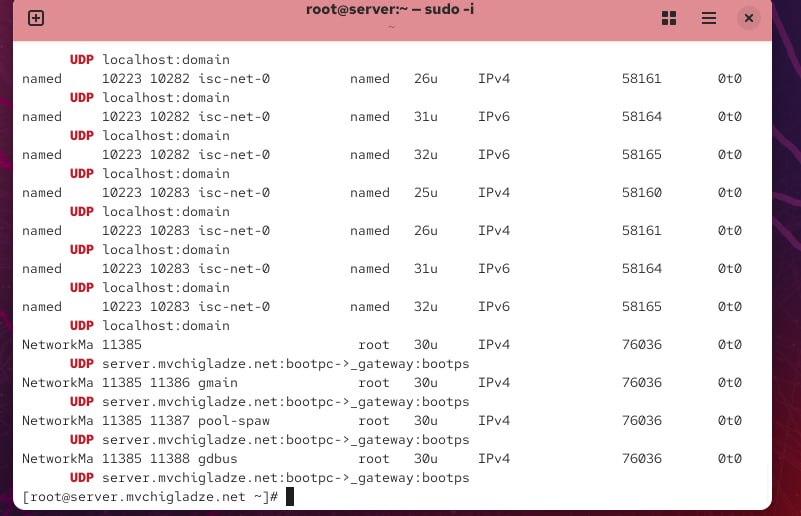
Внесите изменения в настройки межсетевого экрана узла server, разрешив работу с DNS

1. Внесите изменения в настройки межсетевого экрана узла server, разрешив работу с DNS



**Рис. 2.10**. Вносим изменения в настройки

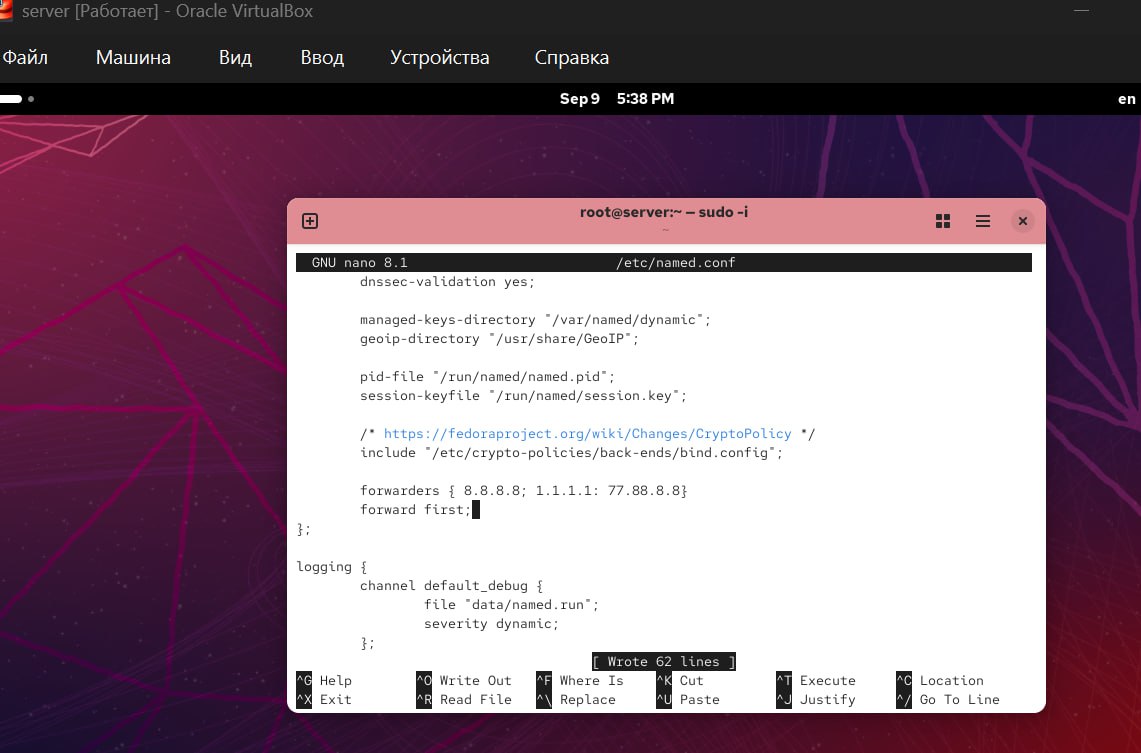
1. Для этого на данном этапе используйте команду lsof



**Рис. 2.11**. Команда лсоф

**2.4.2.2. Конфигурирование кэширующего DNS-сервера при наличии фильтрации DNS-запросов маршрутизаторами**

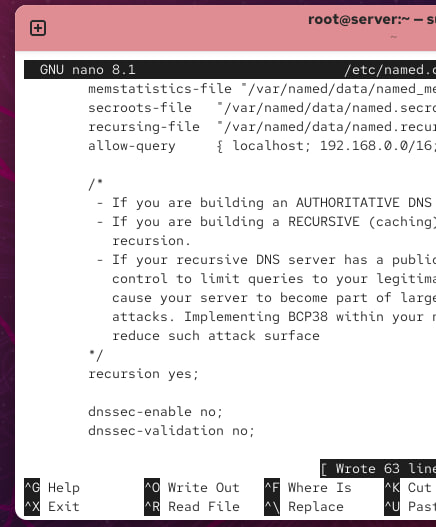
В случае возникновения в сети ситуации, когда DNS-запросы от сервера фильтруются сетевым оборудованием, следует добавить перенаправление DNS-запросов на конкретный вышестоящий DNS-сервер.

****

**Рис. 3.1**. Перенаправление днс запросов 1

****

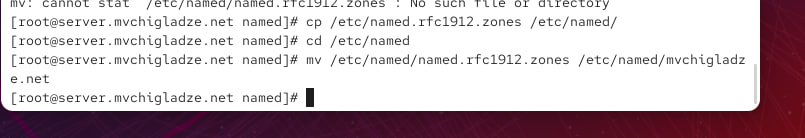
**Рис. 3.2**. Перенаправление днс запросов 2

****

**Рис. 3.3**. Перенаправление днс запросов 3

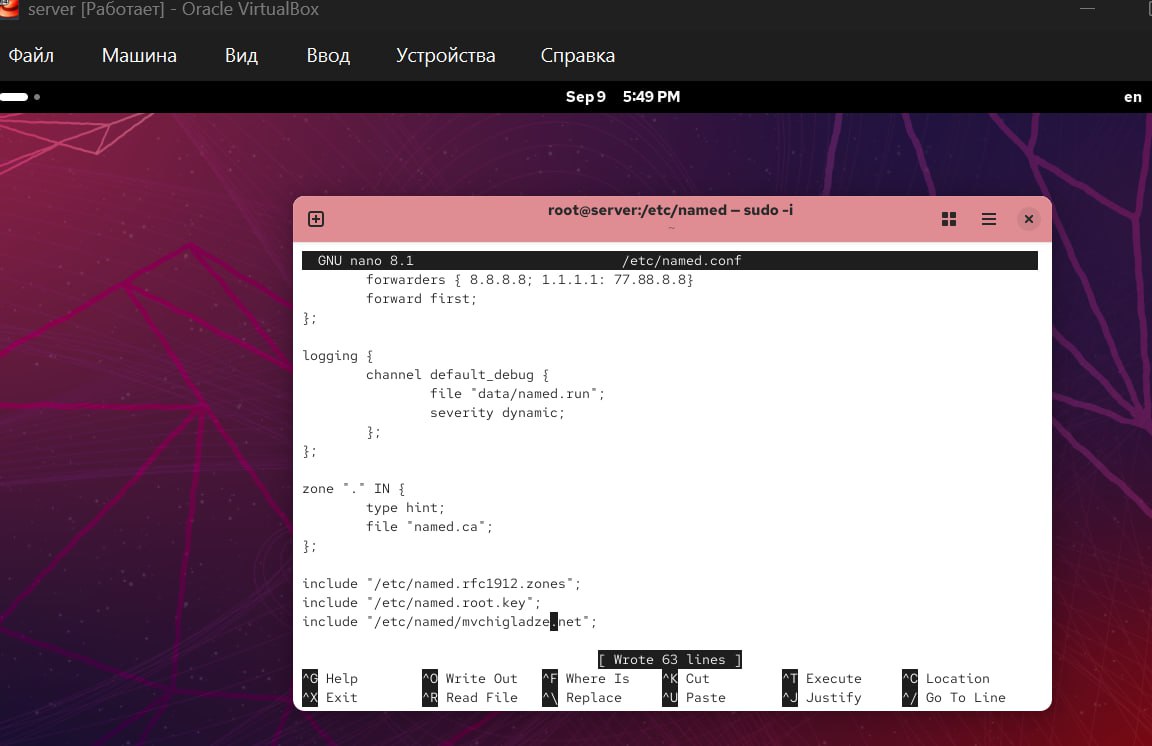
**2.4.3. Конфигурирование первичного DNS-сервера**

1. Скопируйте шаблон описания DNS-зон named.rfc1912.zones из каталога /etc в каталог /etc/named и переименуйте его в user.net (вместо user укажите свой логин)



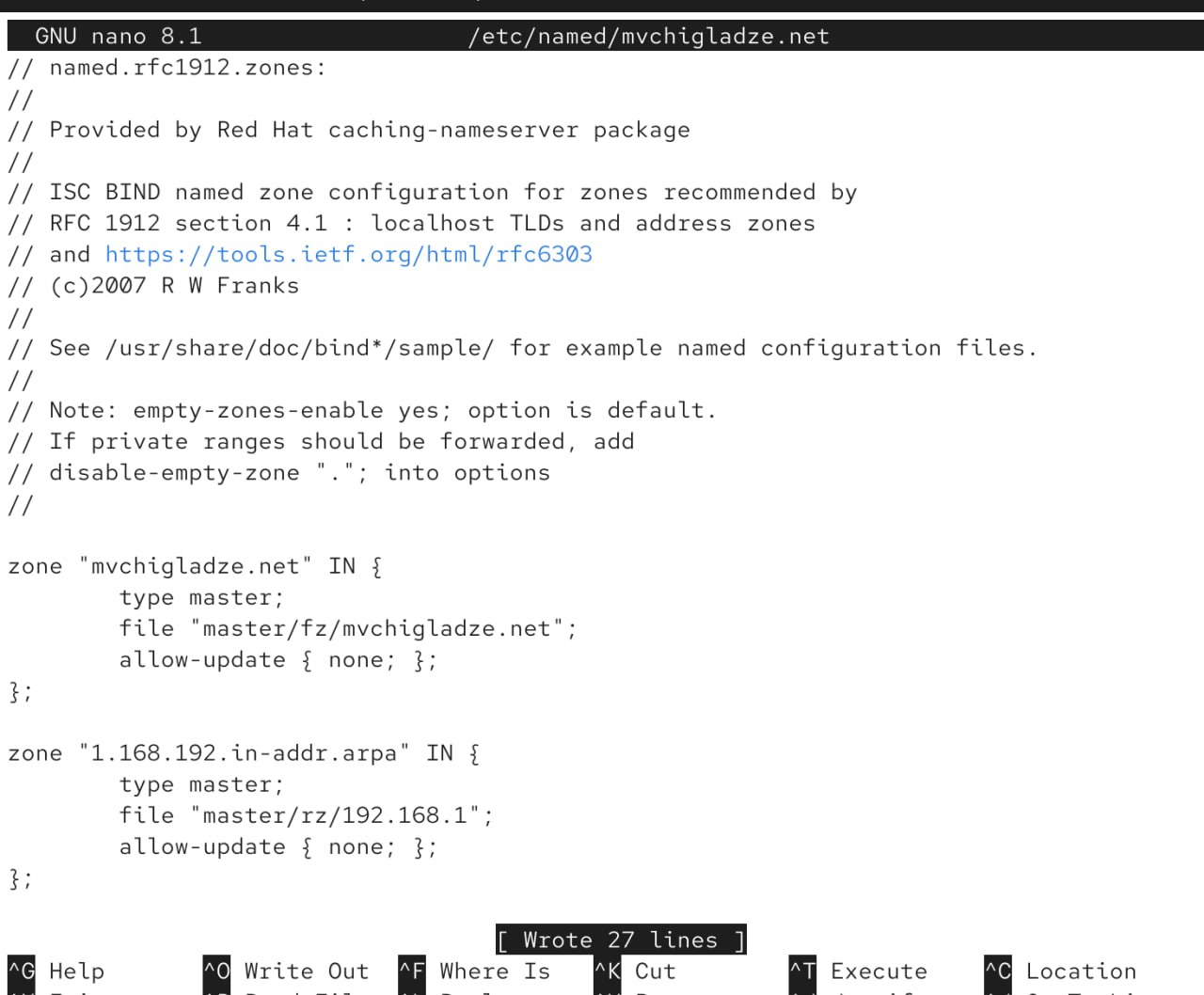
**Рис. 4.1**. Скопируйте шаблон

2. Включите файл описания зоны /etc/named/user.net в конфигурационном файле



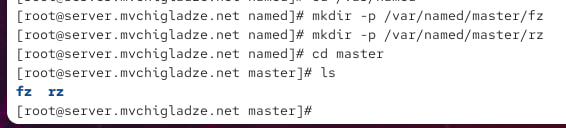
**Рис. 4.2** Включите файл

1. Откройте файл /etc/named/user.net на редактирование и вместо зоны пропишите свою прямую зону



**Рис. 4.3**. Прописываем свою прямую зону

1. В каталоге /var/named создайте подкаталоги master/fz и master/rz, в которых будут располагаться файлы прямой и обратной зоны соответственно

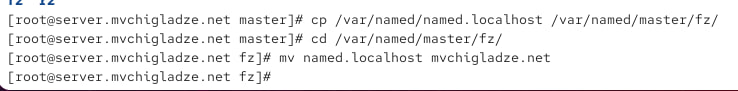


**Рис. 4.4**. Создаем подкаталоги

1. Скопируйте шаблон прямой DNS-зоны named.localhost из каталога

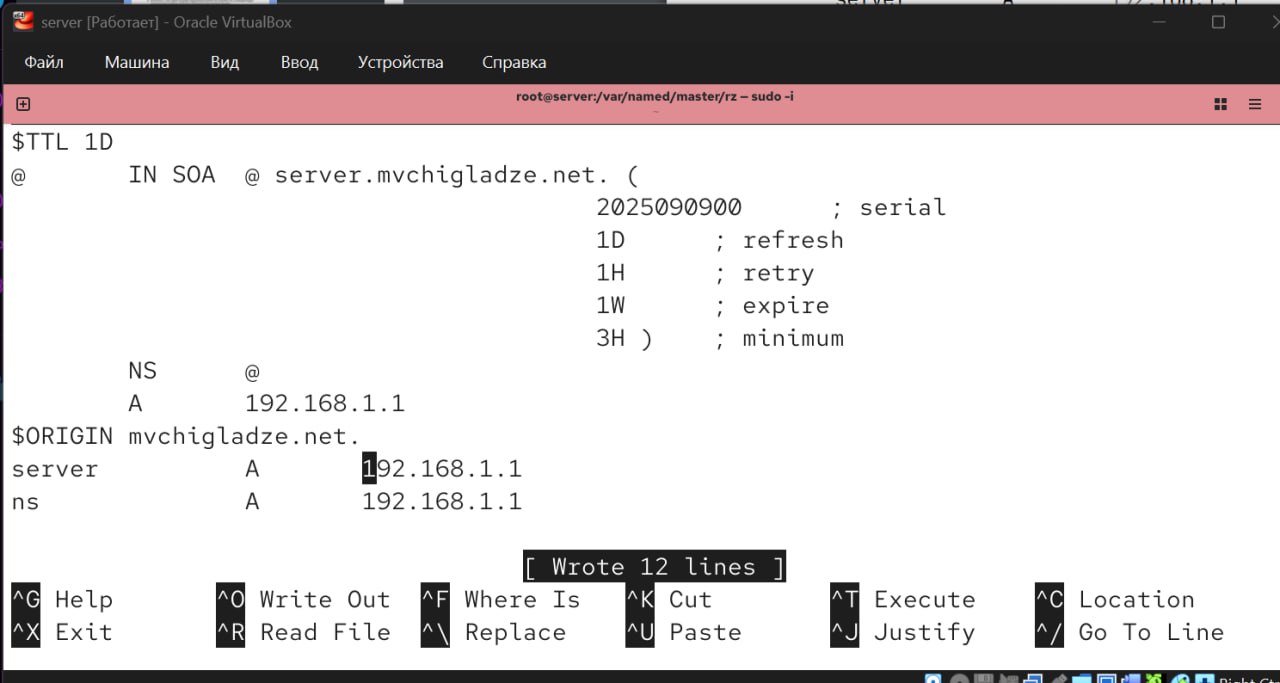
/var/named в каталог /var/named/master/fz и переименуйте его в user.net (вместо

user укажите свой логин)



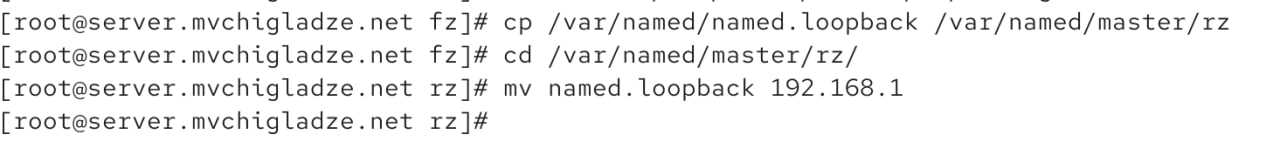
**Рис. 4.5**. Скопируйте шаблон

1. Измените файл /var/named/master/fz/user.net, указав необходимые DNS-записи для прямой зоны.



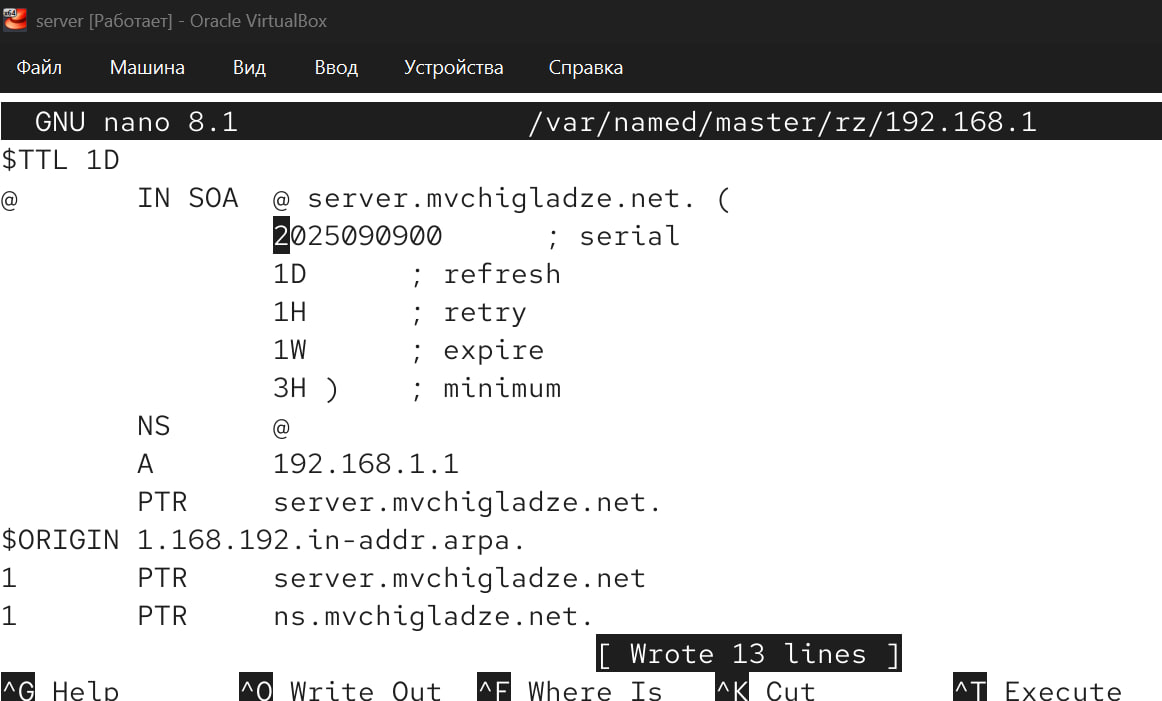
**Рис. 4.6**. Измените файл

1. Скопируйте шаблон обратной DNS-зоны named.loopback из каталога /var/named в каталог /var/named/master/rz и переименуйте его в 192.168.1



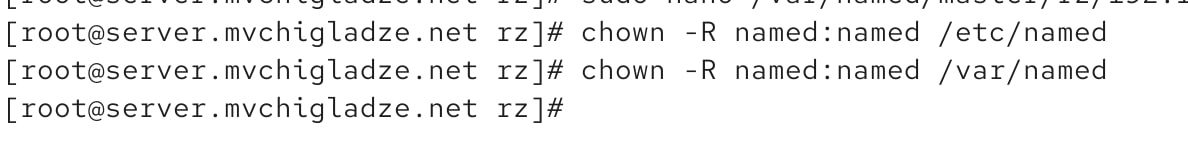
**Рис. 4.7**. Скопируйте шаблон

1. Измените файл /var/named/master/rz/192.168.1, указав необходимые DNS-записи для обратной зоны



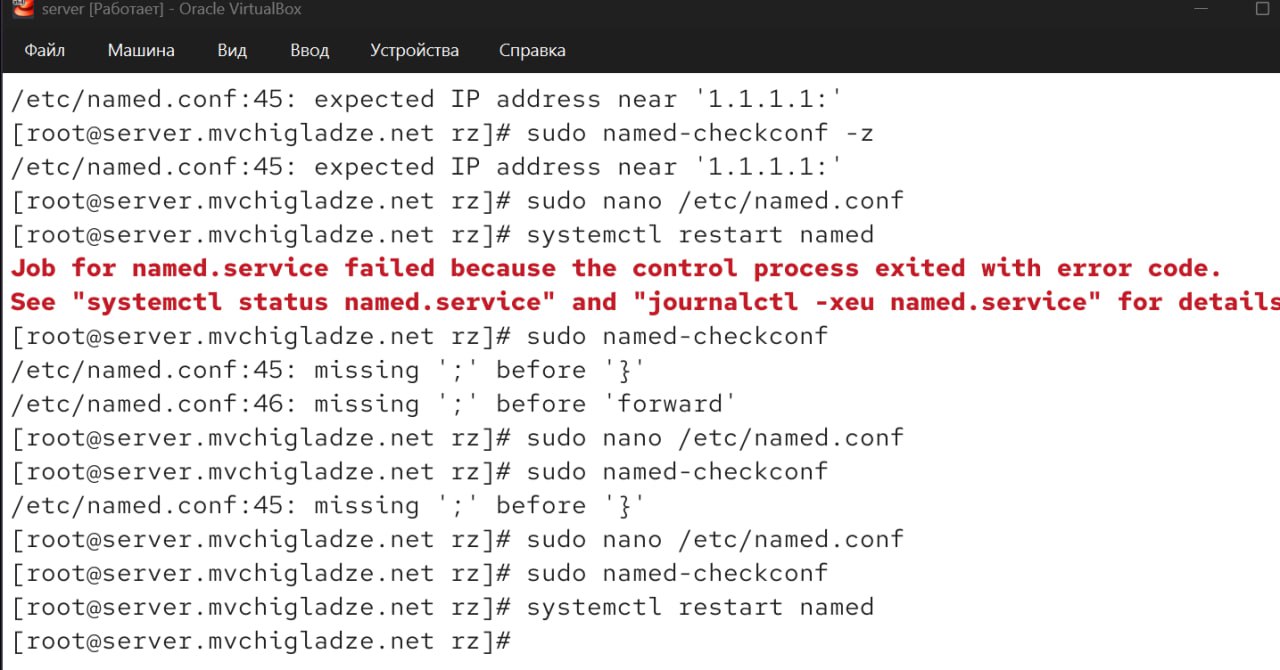
**Рис. 4.8**. Изменяю файл

1. Далее требуется исправить права доступа к файлам в каталогах /etc/named и /var/named, чтобы демон named мог с ними работать



**Рис. 4.9**. Исправить права

1. В системах с запущенным SELinux все процессы и файлы имеют специальные метки безопасности (так называемый «контекст безопасности»), используемые системой для принятия решений по доступу к этим процессам и файлам. После изменения доступа к конфигурационным файлам named требуется корректно восстановить их метки в SELinux.

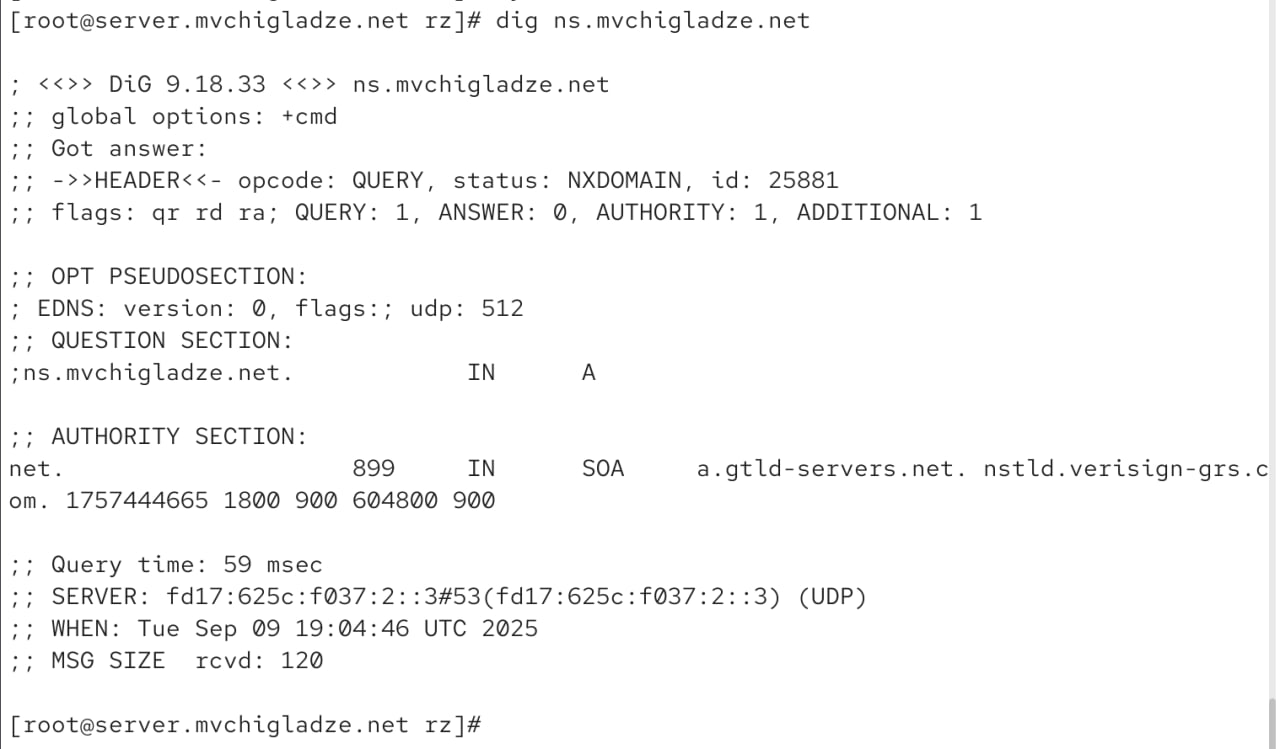


**Рис. 4.10**. Метки безопасности

1. Во дополнительном терминале запустите в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы

**2.4.4. Анализ работы DNS-сервера**

1. При помощи утилиты dig получите описание DNS-зоны с сервера ns.user.net (вместо user должен быть указан ваш логин)

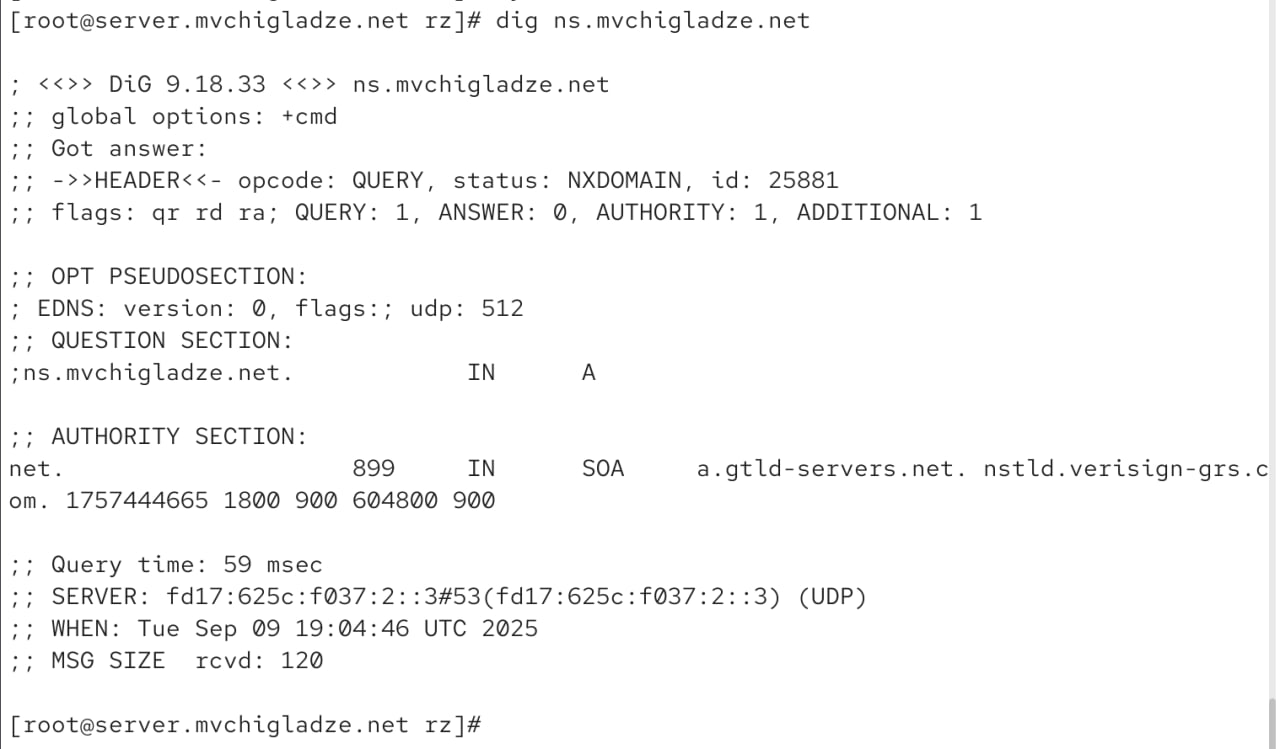


**Рис. 5.1**. Утилиты диг

1. При помощи утилиты host проанализируйте корректность работы DNS-сервера

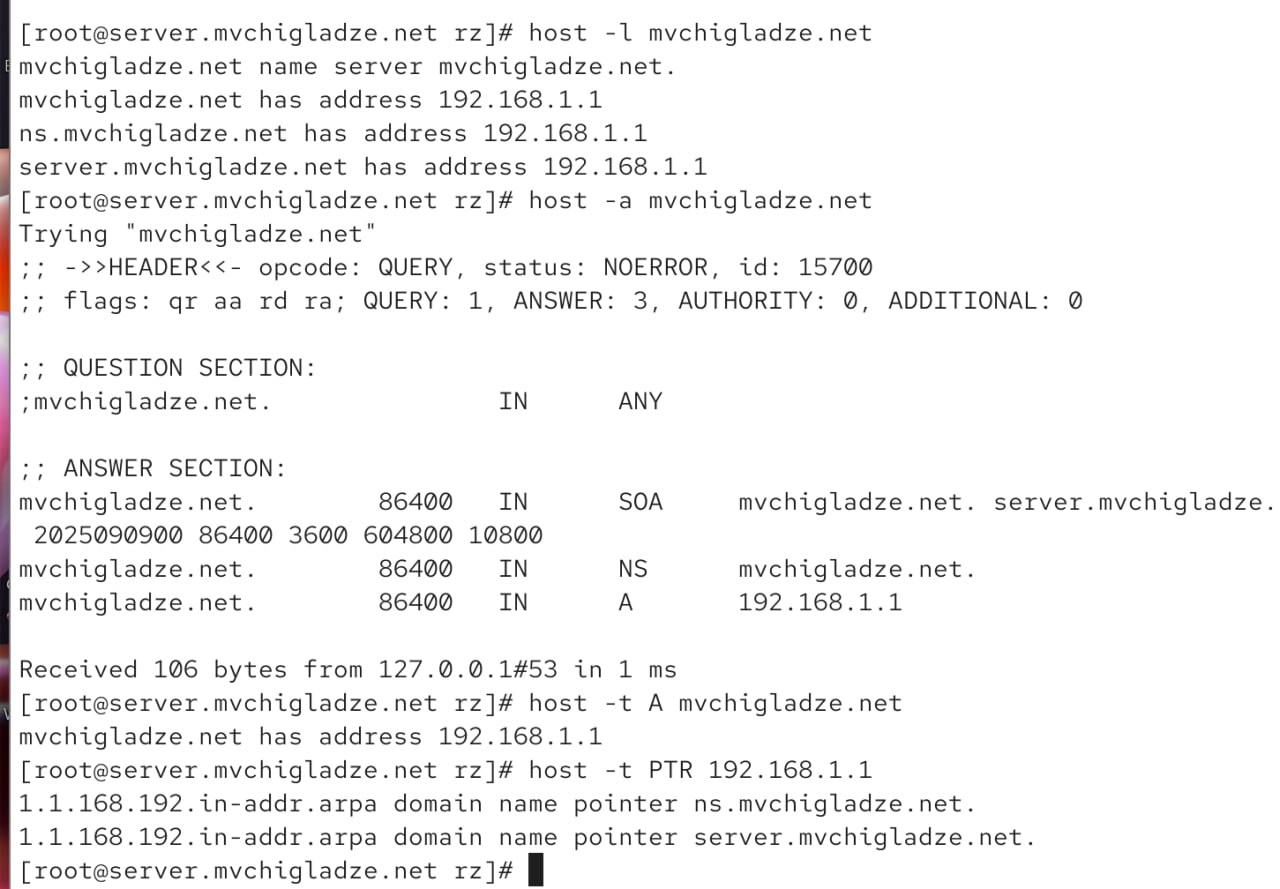
**2.4.5. Внесение изменений в настройки внутреннего окружения виртуальной машины**

1. На виртуальной машине server перейдите в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создайте в нём каталог dns, в который поместите в соответствующие каталоги конфигурационные файлы DNS



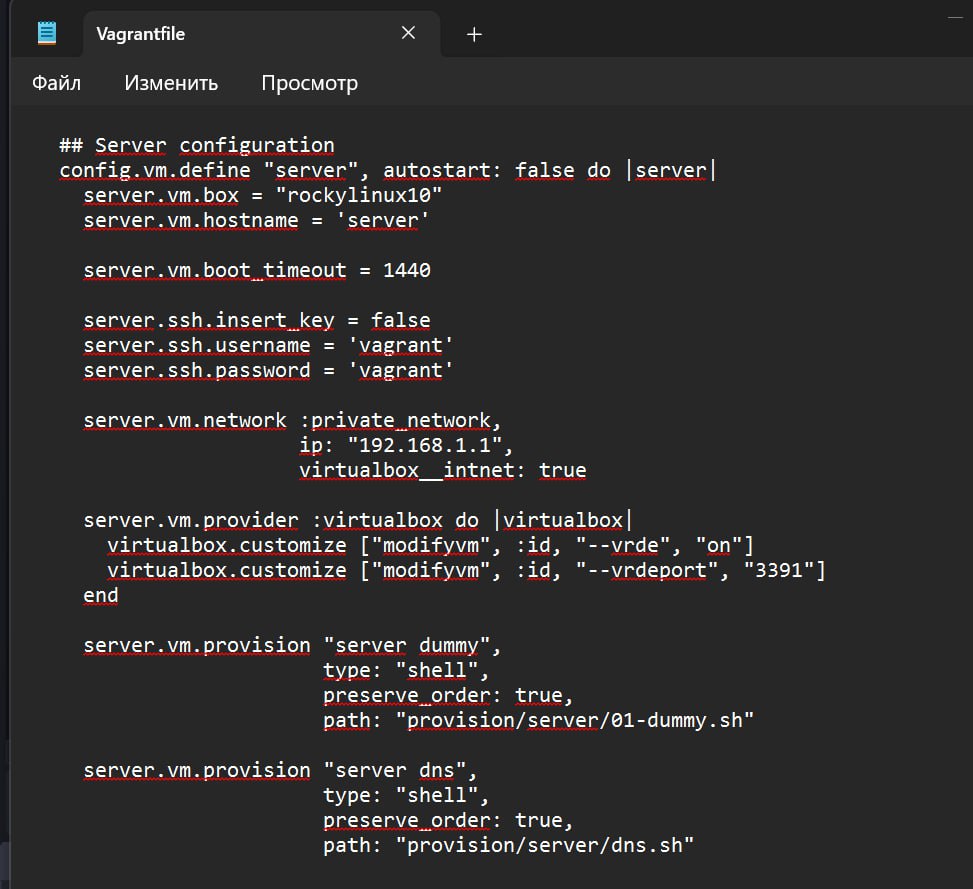
**Рис. 6.1**. Создаем каталог днс

1. В каталоге /vagrant/provision/server создайте исполняемый файл dns.sh



**Рис. 6.2**. Создаем исполняем файл

3. Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile необходимо добавить в разделе конфигурации для сервера



**Рис. 6.3**. Добавляем в файл вагрантвайл

**Вывод:**

В ходе выполнения лабораторной работы были приобретены практические навыки установки и конфигурированию DNS-сервера, усвоение принципов работы системы доменных имён.

**Ответы на контрольные вопросы:**

1. Что такое DNS? DNS (Domain Name System) - это распределенная система, которая преобразует доменные имена (например, google.com) в IP-адреса (например, 172.217.160.142). Это позволяет пользователям использовать запоминающиеся доменные имена вместо сложных IP-адресов для доступа к веб-сайтам и другим сетевым ресурсам.
2. Каково назначение кэширующего DNS-сервера? Кэширующий DNS-сервер сохраняет ответы на DNS-запросы, чтобы ускорить разрешение имен для последующих запросов. Когда клиент запрашивает доменное имя, кэширующий сервер сначала проверяет свой кэш. Если ответ найден в кэше (и срок его действия еще не истек), он немедленно возвращает его клиенту, минуя необходимость отправки запроса авторитативному DNS-серверу. Это значительно снижает задержки и нагрузку на сеть.

3. Чем отличается прямая DNS-зона от обратной?

  •  Прямая зона (Forward Zone): Содержит записи, преобразующие доменные имена в IP-адреса (A, AAAA, CNAME записи). Используется для поиска IP-адреса по имени хоста.

  •  Обратная зона (Reverse Zone): Содержит записи, преобразующие IP-адреса в доменные имена (PTR записи). Используется для поиска имени хоста по IP-адресу. Особенно важна для работы почтовых серверов (проверка подлинности).

4. В каких каталогах и файлах располагаются настройки DNS-сервера? Кратко охарактеризуйте, за что они отвечают. Обычно для BIND (самый распространенный DNS-сервер):

  •  /etc/named.conf (или /etc/bind/named.conf): Основной конфигурационный файл DNS-сервера. Определяет общие настройки сервера, такие как listen-on (IP-адреса для прослушивания), allow-transfer (разрешения на передачу зон), директории для файлов зон и определяет зоны (forward и reverse).

  •  /var/named/ (или /etc/bind/zones/): Директория, содержащая файлы зон. Файлы зон содержат записи ресурсов (RR), определяющие соответствие между доменными именами и IP-адресами (и другими типами ресурсов).

  •  /etc/resolv.conf: Файл, указывающий DNS-серверы, которые будет использовать локальная машина для разрешения имен. Этот файл используется клиентами (например, веб-браузерами) для отправки DNS-запросов.

5. Что указывается в файле resolv.conf? В файле /etc/resolv.conf указываются:

  •  nameserver: IP-адреса DNS-серверов, которые будут использоваться для разрешения имен. Можно указать несколько серверов, они будут использоваться в порядке указания.

  •  search: Список доменов, которые будут автоматически добавляться к неполным доменным именам при разрешении.

  •  options: Различные опции конфигурации, например, timeout (время ожидания ответа) и attempts (количество попыток).

6. Какие типы записи описания ресурсов есть в DNS и для чего они используются? Некоторые из наиболее распространенных типов записей ресурсов (RR):

  •  A (Address): Преобразует доменное имя в IPv4-адрес.

  •  AAAA (Address IPv6): Преобразует доменное имя в IPv6-адрес.

  •  CNAME (Canonical Name): Создает псевдоним для другого доменного имени.

  •  MX (Mail Exchange): Определяет почтовые серверы для домена.

  •  NS (Name Server): Определяет авторитативные DNS-серверы для домена.

  •  PTR (Pointer): Преобразует IP-адрес в доменное имя (используется в обратных зонах).

  •  TXT (Text): Содержит произвольный текст, который может использоваться для различных целей, например, для SPF (Sender Policy Framework) или DKIM (DomainKeys Identified Mail).

  •  SOA (Start of Authority): Определяет параметры зоны, такие как первичный DNS-сервер, адрес электронной почты администратора и серийный номер зоны.

7. Для чего используется домен in-addr.arpa? Домен in-addr.arpa используется для обратного разрешения DNS. Он позволяет преобразовывать IP-адреса IPv4 в доменные имена. Для IPv6 используется домен ip6.arpa.

8. Для чего нужен демон named? named (Name Daemon) -

10. Какие параметры отвечают за время обновления зоны?  В записи SOA (Start of Authority) в файле зоны содержатся параметры, определяющие время обновления:

    \*   refresh:  Интервал, через который slave-сервер запрашивает у master-сервера проверку на наличие изменений в зоне.

    \*   retry: Интервал, через который slave-сервер пытается повторно запросить зону у master-сервера, если предыдущая попытка не удалась.

    \*   expire: Интервал, по истечении которого slave-сервер перестает считать зону актуальной, если не может связаться с master-сервером.

    \*   minimum (TTL):  Минимальное время жизни (Time To Live) для отрицательных ответов (например, если доменное имя не существует).

11. Как обеспечить защиту зоны от скачивания и просмотра?

    \*   Ограничить передачу зоны (zone transfer):  В файле named.conf использовать директиву allow-transfer для указания IP-адресов или сетей, которым разрешено скачивать зону.  Например: allow-transfer { 192.168.1.0/24; }; разрешит передачу зоны только серверам из сети 192.168.1.0/24.

    \*   Использовать TSIG (Transaction Signatures):  Для аутентификации запросов на передачу зоны между master- и slave-серверами.  TSIG позволяет убедиться, что запрос на передачу зоны исходит от авторизованного сервера.

    \*   View: Использовать views в BIND для предоставления различным клиентам разных представлений зоны.  Можно создать view, который предоставляет полную информацию о зоне только доверенным серверам, а для остальных клиентов - ограниченную информацию или вообще запрещает доступ.

12. Какая запись RR применяется при создании почтовых серверов?  Запись MX (Mail Exchange).  Она указывает, какие серверы отвечают за прием почты для данного домена.  MX-запись содержит приоритет (preference), который определяет порядок использования серверов (чем меньше число, тем выше приоритет).

13. Как протестировать работу сервера доменных имён?

    \*   nslookup:  Утилита для запроса DNS-серверов.  Позволяет узнать IP-адрес доменного имени, запросить определенные типы записей (например, MX, NS) и проверить работу сервера.  Пример: nslookup google.com.  Можно указать конкретный DNS-сервер: nslookup google.com 8.8.8.8.

    \*   dig: Более мощная утилита для запроса DNS-серверов, чем nslookup.  Предоставляет больше информации о процессе разрешения имен. Пример: dig google.com.  Можно указать конкретный DNS-сервер: dig @8.8.8.8 google.com.  Можно запросить определенный тип записи: dig MX google.com.

    \*   host:  Простая утилита для поиска IP-адресов по доменному имени и наоборот.  Пример: host google.com.

    \*   ping:  Использовать ping для проверки доступности сервера по IP-адресу, полученному через DNS.

14. Как запустить, перезапустить или остановить какую-либо службу в системе?  (Предполагается systemd)

    \*   Запустить: systemctl start <имя\_службы> (например, systemctl start named)

    \*   Перезапустить: systemctl restart <имя\_службы>

    \*   Остановить: systemctl stop <имя\_службы>

    \*   Проверить статус: systemctl status <имя\_службы>

    \*   Перезагрузить конфигурацию без остановки: systemctl reload <имя\_службы>

15. Как посмотреть отладочную информацию при запуске какого-либо сервиса или службы?

    \*

journalctl -u <имя\_службы>:  Показывает логи службы, управляемой systemd.  Пример: journalctl -u named.

    \*   Проверить конфигурационный файл службы на наличие опций отладки: Некоторые службы имеют свои собственные опции для включения более подробной отладочной информации.

...

16. Где храниться отладочная информация по работе системы и служб? Как её посмотреть?

    \*   journalctl: (systemd) Хранит логи в бинарном формате.  Используется journalctl для просмотра.

    \*   /var/log/: Традиционное место хранения логов в Linux.  Внутри могут быть разные файлы, например, syslog, messages, логи конкретных служб (например, /var/log/named/ для BIND).  Используются текстовые редакторы (less, more, nano, vim) или утилиты (cat, tail, grep) для просмотра.

17. Как посмотреть, какие файлы использует в своей работе тот или иной процесс? Приведите несколько примеров.

    \*   lsof -p <PID>: Показывает список открытых файлов для процесса с указанным PID.  Пример: lsof -p 1234.

    \*   strace -p <PID>: Отслеживает системные вызовы, выполняемые процессом, включая открытие и закрытие файлов.  Предоставляет очень подробную информацию.  Пример: strace -p 1234.

    \*   /proc/<PID>/fd/: Директория, содержащая файловые дескрипторы процесса.  Можно посмотреть, куда указывают эти дескрипторы.  Пример: ls -l /proc/1234/fd/.

18. Приведите несколько примеров по изменению сетевого соединения при помощи командного интерфейса nmcli.

    \*   Подключиться к Wi-Fi сети: nmcli device wifi connect <SSID> password <пароль>

    \*   Включить сетевое соединение: nmcli connection up <имя\_соединения>

    \*   Выключить сетевое соединение: nmcli connection down <имя\_соединения>

    \*   Показать список сетевых соединений: nmcli connection show

    \*   Создать новое Ethernet-соединение с фиксированным IP-адресом:

        nmcli connection add type ethernet con-name static-eth ifname eth0 ip4 192.168.1.100/24 gw4 192.168.1.1 dns4 8.8.8.8

\*  Изменить IP-адрес существующего соединения:

nmcli connection modify <имя\_соединения> ipv4.addresses 192.168.1.150/24

  nmcli connection up <имя\_соединения> # Перезапустить соединение, чтобы изменения вступили в силу

19. Что такое SELinux?  SELinux (Security-Enhanced Linux) - это модуль безопасности ядра Linux, который обеспечивает более строгий контроль доступа, чем традиционные разрешения UNIX.  Он использует модель принудительного контроля доступа (MAC), в отличие от дискреционного контроля доступа (DAC) UNIX.  SELinux позволяет определить, какие процессы могут получать доступ к каким ресурсам на основе политик безопасности.

20. Что такое контекст (метка) SELinux?  Контекст (или метка) SELinux - это атрибут, привязанный к каждому процессу, файлу и другим ресурсам в системе.  Контекст включает в себя информацию о роли, типе и уровне безопасности.  Он используется политикой SELinux для определения, разрешен ли доступ между субъектом (процессом) и объектом (файлом).  Пример контекста: system\_u:object\_r:httpd\_sys\_content\_t:s0.

21. Как восстановить контекст SELinux после внесения изменений в конфигурационные файлы?  Использовать команду restorecon.

    \*   Для восстановления контекста одного файла: restorecon <путь\_к\_файлу>

    \*   Для восстановления контекста всех файлов в директории: restorecon -v -R <путь\_к\_директории>  (-v для вывода подробной информации, -R для рекурсивного применения к поддиректориям).

    \*   Если изменения были внесены в политику SELinux, нужно перезагрузить политику: semodule -B (перекомпилировать и перезагрузить все модули).

22. Как создать разрешающие правила политики SELinux из файлов журналов, содержащих сообщения о запрете операций?

    1.  Проанализировать логи SELinux:  Использовать ausearch или просмотреть файл /var/log/audit/audit.log, чтобы найти сообщения о запрете (AVC - Access Vector Cache) доступа.

    2.  Использовать audit2allow для создания правил:  audit2allow - утилита, которая создает правила SELinux на основе сообщений из логов аудита.

        \*   Создать правило из конкретного сообщения в логе: audit2allow -a -M mypolicy < /var/log/audit/audit.log (выбрать нужные строки из лога).

        \*   Создать правило из всех сообщений в логе: audit2allow -a -M mypolicy

    3.  Установить политику: semodule -i mypolicy.pp (mypolicy.pp - скомпилированный модуль политики).

23. Что такое булевый переключатель в SELinux?  Булевый переключатель (SELinux boolean) - это переменная, которая позволяет изменять поведение политики SELinux без необходимости перекомпиляции и перезагрузки всей политики.  Булевы переключатели позволяют включать и выключать определенные функции или разрешения, например, разрешение HTTP-серверу подключаться к сети.

24. Как посмотреть список переключателей SELinux и их состояние?

    \*   getsebool -a:  Выводит список всех булевых переключателей и их текущее состояние (on/off).

25. Как изменить значение переключателя SELinux?

    \*   Временно (до перезагрузки): setsebool <имя\_переключателя> <on|off> (например, setsebool httpd\_enable\_cgi on)

    \*   Постоянно (сохраняется после перезагрузки): setsebool -P <имя\_переключателя> <on|off> (например, setsebool -P httpd\_enable\_cgi on) -P означает permanent.