РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

Отчет лабораторной работы 2

Дисциплина: Администрирование сетевых подсистем

Студент: Астахова Марина

Группа: НПИбд-02-23

Тема: Настройка DNS-сервера

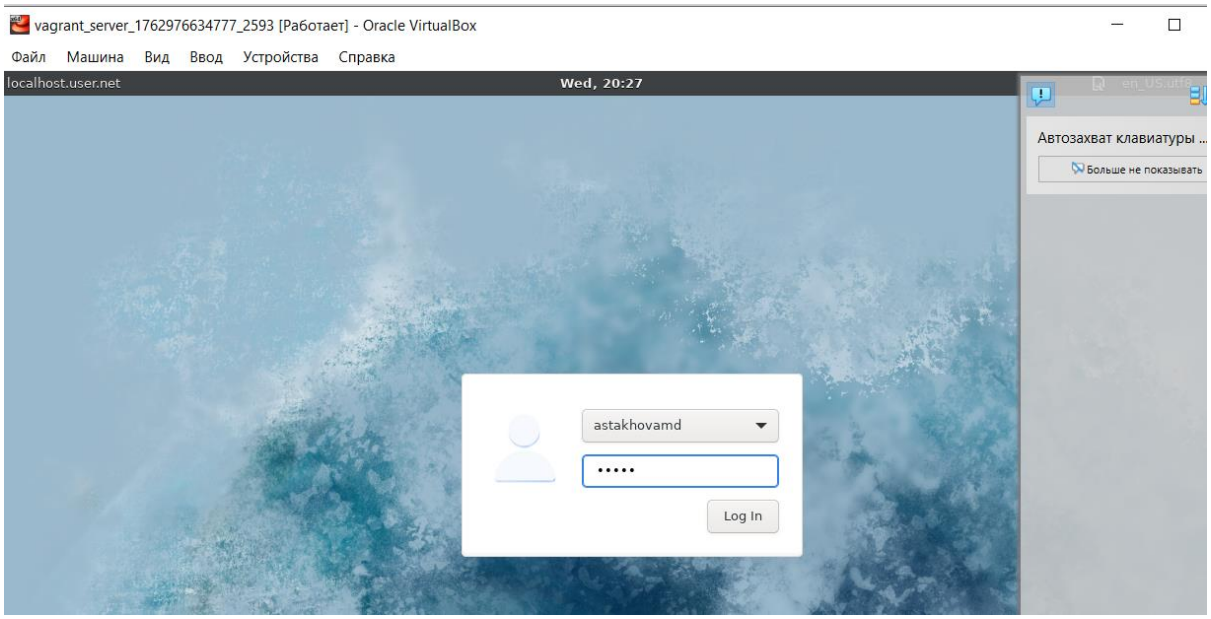
2.1. Цель работы.

Приобретение практических навыков по установке и конфигурированию DNS сервера, усвоение принципов работы системы доменных имён.

2.2. Выполнение работы

1. Установка DNS-сервера

Включаем виртуальную машину и вводим логин



Установка bind и bind-utils:

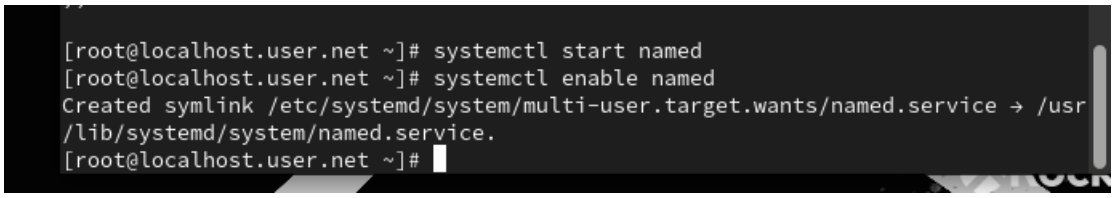
```
[sudo] password for astakhovamd:
[root@localhost.user.net ~]# dnf -y install bind bind-utils
Last metadata expiration check: 0:17:20 ago on Wed 12 Nov 2025 08:42:46 PM UTC.
Package bind-utils-32:9.16.23-18.el9_4.1.x86_64 is already installed.
Dependencies resolved.
================================================================================
 Package               Arch       Version                   Repository     Size
================================================================================
Installing:
 bind                  x86_64     32:9.16.23-31.el9_6       appstream     488 k
Upgrading:
 bind-libs             x86_64     32:9.16.23-31.el9_6       appstream     1.2 M
 bind-license          noarch     32:9.16.23-31.el9_6       appstream      12 k
 bind-utils            x86_64     32:9.16.23-31.el9_6       appstream     199 k
Installing dependencies:
 bind-dnssec-doc       noarch     32:9.16.23-31.el9_6       appstream      45 k
 python3-bind          noarch     32:9.16.23-31.el9_6       appstream      60 k
 python3-ply           noarch     3.11-14.el9.0.1           baseos        103 k
Installing weak dependencies:
 bind-dnssec-utils     x86_64     32:9.16.23-31.el9_6       appstream     112 k

Transaction Summary
================================================================================
Install  5 Packages
```

В качестве упражнения с помощью утилиты dig сделали запрос к DNS адресу www.yandex.ru:

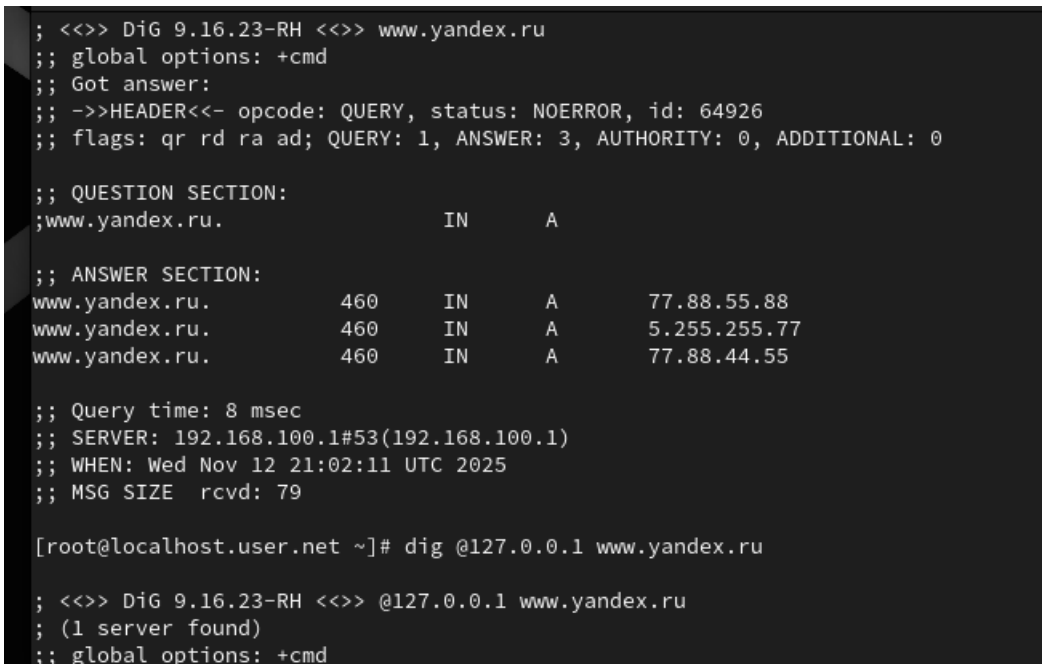
```
[root@localhost.user.net ~]# dig www.yandex.ru

; <<>> DiG 9.16.23-RH <<>> www.yandex.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64645
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.yandex.ru.                 IN      A

;; ANSWER SECTION:
www.yandex.ru.          537     IN      A       77.88.55.88
www.yandex.ru.          537     IN      A       77.88.44.55
www.yandex.ru.          537     IN      A       5.255.255.77

;; Query time: 72 msec
;; SERVER: 192.168.100.1#53(192.168.100.1)
;; WHEN: Wed Nov 12 21:00:55 UTC 2025
;; MSG SIZE  rcvd: 90

[root@localhost.user.net ~]#
```

2. Конфигурирование кэширующего DNS-сервера

2.1 Запустим сервер



2.2 Проанализируем в отчёте отличие в выведенной на экран информации при выполнении команд



```
; <<>> DiG 9.16.23-RH <<>> @127.0.0.1 www.yandex.ru
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46366
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 86bb01a04255ef3c010000006914f5e3d0fe7b4a04478128 (good)
;; QUESTION SECTION:
;www.yandex.ru.                 IN      A

;; ANSWER SECTION:
www.yandex.ru.          600     IN      A       5.255.255.77
www.yandex.ru.          600     IN      A       77.88.44.55
www.yandex.ru.          600     IN      A       77.88.55.88

;; Query time: 4962 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Nov 12 21:02:27 UTC 2025
;; MSG SIZE  rcvd: 118

[root@localhost.user.net ~]#
```

2.3 Сделаем DNS-сервер сервером по умолчанию для хоста server и внутренней виртуальной сети. Для этого требуется изменить настройки сетевого соединения eth0 в NetworkManager, переключив его на работу с внутренней сетью и указав для него в качестве DNS-сервера по умолчанию адрес 127.0.0.1:

```
[root@localhost.user.net ~]# nmcli connection edit eth0

===| nmcli interactive connection editor |===

Editing existing '802-3-ethernet' connection: 'eth0'

Type 'help' or '?' for available commands.
Type 'print' to show all the connection properties.
Type 'describe [<setting>.<prop>]' for detailed property description.

You may edit the following settings: connection, 802-3-ethernet (ethernet), 802-
1x, dcb, sriov, ethtool, match, ipv4, ipv6, hostname, link, tc, proxy
nmcli> remove ipv4.dns
nmcli> set ipv4.ignore-auto-dns yes
nmcli> set ipv4.dns 127.0.0.1
nmcli> save
Connection 'eth0' (52b757e4-3833-4753-b0eb-de0af7f26f57) successfully updated.
nmcli> quit
[root@localhost.user.net ~]#
```

2.4 Сделаем то же самое для соединения System eth0 (если оно активно):

```
Connection 'eth0' (52b757e4-3833-4753-b0eb-de0af7f26f57) successfully updated.
nmcli> quit
[root@localhost.user.net ~]# systemctl restart NetworkManager
```

2.5 Проверим изменения в resolv.conf

```
  GNU nano 5.6.1
# Generated by NetworkManager
search user.net
nameserver 127.0.0.1
nameserver fd17:625c:f037:2::3
```

(При настройке dns я исправляла этот файл и сейчас он выглядит по другому)

```
root@localhost:/etc
  GNU nano 5.6.1                    resolv.conf
# Generated by NetworkManager
search user.net
nameserver 192.168.1.10
nameserver 192.168.100.1
```

2.6 Настройка направление DNS-запросов от всех узлов внутренней сети, включая запросы от узла server, через узел server.

```
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
        listen-on port 53 { 127.0.0.1; any; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        secroots-file   "/var/named/data/named.secroots";
        recursing-file  "/var/named/data/named.recursing";
        allow-query { localhost; 192.168.0.0/16; };

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

2.7 Внесем изменения в настройки межсетевого экрана узла server, разрешив работу с DNS:

```
[root@localhost.user.net etc]# firewall-cmd --add-service=dns
success
[root@localhost.user.net etc]# firewall-cmd --add-service=dns --permanent
success
[root@localhost.user.net etc]#
```

2.8 Убедитесь, что DNS-запросы идут через узел server, который прослушивает порт 53. Для этого на данном этапе используйте команду lsof:

```
[root@localhost.user.net etc]# lsof | grep UDP
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs
      Output information may be incomplete.
avahi-dae   593                         avahi   12u     IPv4              19438       0t0
     UDP *:mdns
avahi-dae   593                         avahi   13u     IPv6              19439       0t0
     UDP *:mdns
avahi-dae   593                         avahi   14u     IPv4              19440       0t0
     UDP *:44539
avahi-dae   593                         avahi   15u     IPv6              19441       0t0
     UDP *:60757
chronyd     619                        chrony    5u     IPv4              19827       0t0
     UDP localhost:323
chronyd     619                        chrony    6u     IPv6              19828       0t0
     UDP localhost:323
named     45474                         named   32u     IPv4              63718       0t0
     UDP localhost:domain
```

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

3. Конфигурирование кэширующего DNS сервера при наличии фильтрации DNS-запросов маршрутизаторами

Посмотрим на содержание файла resolv.conf и добавить в список в named.conf

```
[root@localhost.user.net etc]# sudo nano resolv.conf
[root@localhost.user.net etc]# cat resolv.conf
# Generated by NetworkManager
search astakhovamd.net
nameserver 192.168.1.10
nameserver 192.168.100.1
[root@localhost.user.net etc]#
```

```
//
// See /usr/share/doc/bind*/sample/ for example named con
//

options {
        forwarders {192.168.1.10;192.168.100.1;};
        forward first;
        listen-on port 53 { 127.0.0.1;192.168.1.10; };
```

Кроме того, возможно вышестоящий DNS-сервер может не поддерживать технологию DNSSEC, тогда следует в конфигурационном файле named.conf указать следующие настройки:

```
dnssec-enable no;
dnssec-validation no;
```

Но из-за того, что у меня все работает и без дополнительных настроек, то я комментирую эти записи.

```
// dnssec yes;
// dnssec-enable yes;
// dnssec-validation yes;
```

4. Конфигурирование первичного DNS-сервера

1. Скопировали шаблон описания DNS-зон named.rfc1912.zones из каталога /etc в каталог /etc/named и переименовали его в astakhovamd.net :

```
[root@localhost.user.net etc]# cp /etc/named.rfc1912.zones /etc/named/
[root@localhost.user.net etc]# cd /etc/named
[root@localhost.user.net named]#
mv /etc/named/named.rfc1912.zones /etc/named/astakhovamd.net
mv: overwrite '/etc/named/astakhovamd.net'? Y
[root@localhost.user.net named]#
```

2. Включили файл описания зоны /etc/named/user.net в конфигурационном файле DNS /etc/named.conf, добавив в нём в конце строку:

```
include "/etc/named/astakhovamd.net";
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
include "/etc/named/keys/dhcp_updater.key";
```

3. Открыли файл /etc/named/astakhovamd.net и прописали:

```
//

zone "astakhovamd.net" IN {
      type master;
      file "master/fz/astakhovamd.net";
      allow-update { none; };
};
zone "1.168.192.in-addr.arpa" IN {
      type master;
      file "master/rz/192.168.1";
      allow-update { none; };
};
```

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

```
[root@localhost.user.net etc]# cd /etc/named
[root@localhost.user.net named]# sudo nano astakhovamd.net
[root@localhost.user.net named]# cd /var/named
[root@localhost.user.net named]# mkdir -p /var/named/master/fz
[root@localhost.user.net named]# mkdir -p /var/named/master/rz
[root@localhost.user.net named]#
```

5. Скопировали шаблон прямой DNS-зоны named.localhost из каталога /var/named в каталог /var/named/master/fz и переименовали файл

```
[root@localhost.user.net named]# mkdir -p /var/named/master/fz
[root@localhost.user.net named]# mkdir -p /var/named/master/rz
[root@localhost.user.net named]# cp /var/named/named.localhost /var/named/master/fz/
[root@localhost.user.net named]# cd /var/named/master/fz/
[root@localhost.user.net fz]# mv named.localhost astakhovamd.net
mv: overwrite 'astakhovamd.net'? Y
[root@localhost.user.net fz]#
```

6. Изменим файл /var/named/master/fz/astakhovamd.net, указав необходимые DNS-записи для прямой зоны.

```
root@localhost:/var/named/master/fz
 GNU nano 5.6.1                                  user.net
$TTL 1D
@       IN SOA          @ server.astakhovamd.net. (
                2024072700          ; serial
                1D                  ; refresh
                1H                  ; retry
                1W                  ; expire
                3H )                ; minimum
        NS      @
        A       192.168.1.1
$ORIGIN astakhovamd.net.
server      A       192.168.1.1
ns      A       192.168.1.1
```

7. Скопировали шаблон обратной DNS-зоны named.loopback из каталога /var/named в каталог /var/named/master/rz и переименовали его в 192.168.1:

```
[root@localhost.user.net fz]# cp /var/named/named.loopback /var/named/master/rz/
[root@localhost.user.net fz]# cd /var/named/master/rz/
[root@localhost.user.net rz]# mv named.loopback 192.168.1
[root@localhost.user.net rz]#
```

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

```
 GNU nano 5.6.1                                  192.168.1
$TTL 1D
@       IN SOA          @ server.astakhovamd.net. (
                2024072700          ; serial
                1D                  ; refresh
                1H                  ; retry
                1W                  ; expire
                3H )                ; minimum
        NS      @
        A       192.168.1.1
        PTR     server.astakhovamd.net.
$ORIGIN 1.168.192.in-addr.arpa.
1       PTR     server.astakhovamd.net.
1       PTR     ns.astakhovamd.net.
```

9. Далее требуется исправить права доступа к файлам и восстановить их метки в SELinux:

```
[root@localhost.user.net rz]# chown -R named:named /etc/named
[root@localhost.user.net rz]# chown -R named:named /var/named
[root@localhost.user.net rz]# restorecon -vR /etc
[root@localhost.user.net rz]# restorecon -vR /var/named
```

10. Для проверки состояния переключателей SELinux, относящихся к named, введем:

```
[root@localhost.user.net rz]# restorecon -vR /var/named
[root@localhost.user.net rz]# getsebool -a | grep named
named_tcp_bind_http_port --> off
named_write_master_zones --> on
```

```
[root@localhost.user.net rz]# setsebool named_write_master_zones 1
[root@localhost.user.net rz]# setsebool -P named_write_master_zones 1
```

11. Во дополнительном терминале запустим в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы:

```
[root@localhost.user.net rz]# systemctl restart named
Job for named.service failed because the control process exited with error code.
See "systemctl status named.service" and "journalctl -xeu named.service" for details.
[root@localhost.user.net rz]#
```

Исправили файлы и получилось запустить

```
Subject: A start job for unit named.service has finished successfully
Defined-By: systemd
Support: https://wiki.rockylinux.org/rocky/support

A start job for unit named.service has finished successfully.

The job identifier is 3116.
Nov 18 14:49:00 localhost.user.net systemd[1]: systemd-hostnamed.service: Deactivated successfully.
Subject: Unit succeeded
Defined-By: systemd
Support: https://wiki.rockylinux.org/rocky/support

The unit systemd-hostnamed.service has successfully entered the 'dead' state.
Nov 18 14:49:06 localhost.user.net named[3750]: managed-keys-zone: Unable to fetch DNSKEY set '.': timed out
Nov 18 14:49:06 localhost.user.net named[3750]: resolver priming query complete
```

5. Анализ работы DNS-сервера

```
[root@localhost.user.net ~]# dig ns.astakhovamd.net

; <<>> DiG 9.16.23-RH <<>> ns.astakhovamd.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 34754
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ns.astakhovamd.net.            IN      A

;; AUTHORITY SECTION:
net.                    900     IN      SOA     a.gtld-servers.net. nstld.verisign-grs.com. 1763477590 1800 900 604800 900

;; Query time: 39 msec
;; SERVER: 192.168.100.1#53(192.168.100.1)
;; WHEN: Tue Nov 18 14:51:22 UTC 2025
;; MSG SIZE  rcvd: 120
```

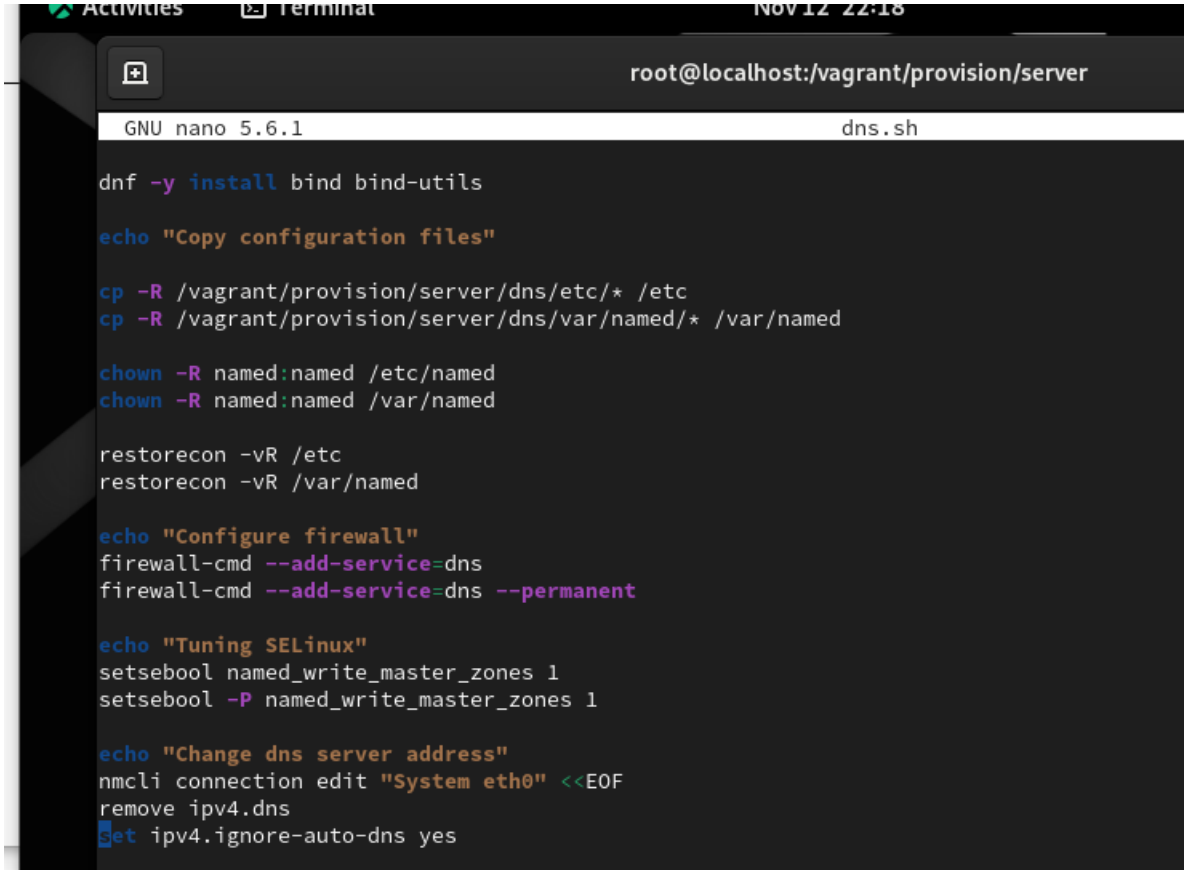
```
root@localhost:/var/named

root@localhost:~                    root@localhost:/var/named

 GNU nano 5.6.1                  named.astakhovamd.net.zone
TTL 1D
       IN      SOA     server.astakhovamd.net. admin.astakhovamd.net. (
                           2023111302 ;
                           3H
                           1H
                           1W
                           1D )

       IN      NS      server.astakhovamd.net.

erver  IN      A       192.168.1.1
lient  IN      A       192.168.1.2
ww     IN      A       192.168.1.1
hcp    IN      A       192.168.1.3  ;
```

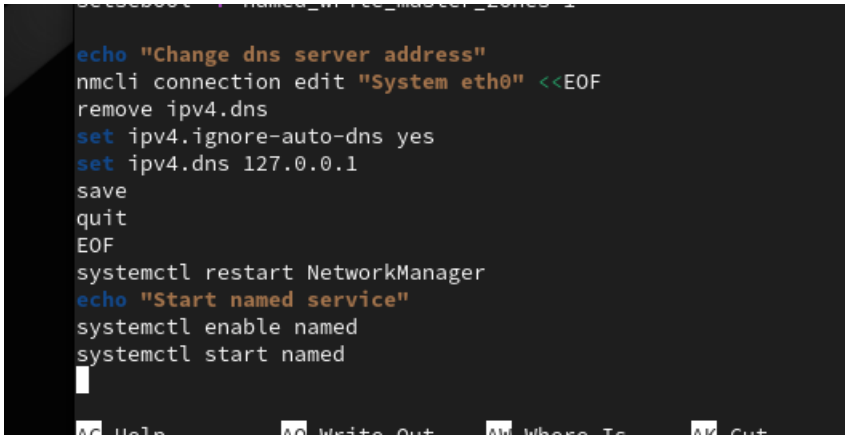
6. Внесение изменений в настройки внутреннего окружения виртуальной машины

```
[root@localhost.user.net ~]# sudo nano /var/named/named.user.net.zone
[root@localhost.user.net ~]# sudo nano /var/named/named.astakhovamd.net.zone
[root@localhost.user.net ~]# sudo chown root:named /var/named/named.astakhovamd
.net.zone
[root@localhost.user.net ~]# sudo chmod 640 /var/named/named.astakhovamd.net.zo
ne
[root@localhost.user.net ~]#     sudo systemctl restart named
```

```
-bash: cd: /vagrant: No such file or directory
[root@localhost.user.net named]# mkdir -p /vagrant/provision/server/dns/etc/named
[root@localhost.user.net named]# mkdir -p /vagrant/provision/server/dns/var/named/master/
[root@localhost.user.net named]# cp -R /etc/named.conf /vagrant/provision/server/dns/etc/
[root@localhost.user.net named]# cp -R /etc/named/* /vagrant/provision/server/dns/etc/named/
[root@localhost.user.net named]# cp -R /var/named/master/*/vagrant/provision/server/dns/var/named/master/
cp: missing destination file operand after '/var/named/master/*/vagrant/provision/server/dns/var/named/master/'
Try 'cp --help' for more information.
[root@localhost.user.net named]# cp -R /var/named/master/* /vagrant/provision/server/dns/var/named/master/
[root@localhost.user.net named]# cd /vagrant/provision/server
[root@localhost.user.net server]# touch dns.sh
[root@localhost.user.net server]# chmod +x dns.sh
[root@localhost.user.net server]#
```

Создание файла dns.sh





Добавляем изменения в vagrantfile

```
                        preserve_order: true,
                        path: "provision/server/01-dummy.sh"
  server.vm.provision "server dns",
                        type: "shell",
                        preserve_order: true,
                        path: "provision/server/dns.sh"
  end
end

## Client configuration
```

Ответы на вопросы

1. Что такое DNS?

DNS (Domain Name System) - это распределенная система доменных имен, которая преобразует понятные человеку доменные имена (например, google.com) в IP-адреса (например, 172.217.160.142), которые компьютеры используют для идентификации друг друга в сети. DNS работает как телефонная книга для Интернета.

2. Каково назначение кэширующего DNS-сервера?

Кэширующий DNS-сервер (также называемый рекурсивным DNS-сервером) хранит ответы на DNS-запросы в своей памяти (кэше). Когда клиент запрашивает разрешение доменного имени, кэширующий сервер сначала проверяет, есть ли ответ в его кэше. Если есть, он возвращает этот ответ клиенту, не обращаясь к другим DNS-серверам. Это значительно ускоряет процесс разрешения имен и снижает нагрузку на корневые и авторитативные DNS-серверы.

3. Чем отличается прямая DNS-зона от обратной?

• **Прямая DNS-зона:** Отображает доменные имена в IP-адреса. Например, она преобразует www.example.com в 192.0.2.1. • **Обратная DNS-зона:** Отображает IP-адреса в доменные имена. Например, она преобразует 192.0.2.1 в www.example.com. Используется для обратного DNS-поиска (reverse DNS lookup) или rDNS, часто применяемого для проверки подлинности и предотвращения спама.

4. В каких каталогах и файлах располагаются настройки DNS-сервера? Кратко охарактеризуйте, за что они отвечают.

В Linux (и других Unix-подобных системах) с использованием BIND9 (наиболее распространенный DNS-сервер) обычно используется следующая структура:

• **/etc/bind/**: Основной каталог конфигурации BIND. * **named.conf**: Главный конфигурационный файл BIND. Содержит общие параметры сервера, определения зон, параметры журналирования и т.д.

* **named.conf.options**: Определяет глобальные опции DNS-сервера, такие как рекурсия, перенаправление запросов (forwarders), списки контроля доступа (ACL).

* **named.conf.local**: Определяет локальные зоны (прямые и обратные).

* **named.conf.default-zones**: Определяет стандартные корневые зоны, необходимые для разрешения доменных имен.

* **db.example.com, db.192.0.2**: Файлы зон. Содержат записи ресурсов (RR) для конкретных доменов и подсетей. Например, db.example.com - записи для зоны example.com, а db.192.0.2 - для обратной зоны 192.0.2.in-addr.arpa.

• **/var/cache/bind/**: Директория, где хранятся файлы зон.

5. Что указывается в файле resolv.conf?

Файл /etc/resolv.conf содержит информацию о DNS-серверах, которые клиентская система должна использовать для разрешения доменных имен. Обычно включает:

• nameserver <IP-адрес>:

Указывает IP-адрес DNS-сервера. Можно указать несколько серверов, которые будут использоваться по порядку.

• domain <имя_домена>:

Устанавливает локальный домен для поиска неполных доменных имен.

• search <список_доменов>:

Устанавливает список доменов для поиска неполных доменных имен.

6. Какие типы записи описания ресурсов есть в DNS и для чего они используются?

Некоторые из наиболее распространенных типов записей ресурсов (Resource Records - RR) в DNS:

• **A (Address record):** Отображает доменное имя в IPv4-адрес.

• **AAAA (Address record IPv6):** Отображает доменное имя в IPv6-адрес.

• **CNAME (Canonical Name record):** Создает псевдоним для доменного имени. Например, www.example.com может быть CNAME для example.com.

• **MX (Mail eXchange record):** Указывает почтовый сервер, отвечающий за прием электронной почты для домена. Содержит приоритет для определения порядка использования серверов.

• **NS (Name Server record):** Указывает авторитативный DNS-сервер для домена.

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

• **SOA (Start of Authority record):** Определяет параметры зоны, такие как главный DNS-сервер, адрес электронной почты администратора и параметры обновления зоны.

• **TXT (Text record):** Содержит произвольный текст. Используется для различных целей, таких как проверка домена (SPF, DKIM).

• **SRV (Service record):** Определяет местоположение служб (например, LDAP, SIP) по домену.

7. Для чего используется домен in-addr.arpa?

Домен in-addr.arpa используется для обратных DNS-зон для IPv4-адресов. IP-адрес записывается в обратном порядке, а затем добавляется к домену in-addr.arpa. Например, для IP-адреса 192.0.2.1 будет создана запись PTR в зоне 2.0.192.in-addr.arpa.

8. Для чего нужен демон named?

named (name daemon) - это демон DNS-сервера BIND (Berkeley Internet Name Domain). Это основная программа, которая отвечает за обработку DNS-запросов, управление зонами, кэширование ответов и взаимодействие с другими DNS-серверами.

9. В чем заключаются основные функции slave-сервера и master-сервера?

• **Master-сервер (Primary DNS server):** Хранит основную копию DNS-зоны и отвечает за внесение изменений в эту зону.

• **Slave-сервер (Secondary DNS server):** Получает копию DNS-зоны с master-сервера и использует ее для ответа на DNS-запросы. Slave-серверы

обеспечивают избыточность и повышают доступность DNS-сервиса. Slave-серверы обновляют свои зоны с master-сервера через механизм zone transfer.

10. Какие параметры отвечают за время обновления зоны?

Параметры, контролирующие обновление зоны, обычно находятся в записи SOA (Start of Authority):

• **serial:** Серийный номер зоны. Slave-серверы используют этот номер, чтобы определить, нужно ли обновлять зону. Каждый раз, когда в зону вносятся изменения, серийный номер должен быть увеличен.

• **refresh:** Как часто slave-сервер должен проверять у master-сервера, изменился ли серийный номер зоны.

• **retry:** Если slave-сервер не может связаться с master-сервером при обновлении, как часто он должен пытаться повторить.

• **expire:** Если slave-сервер не может связаться с master-сервером в течение этого времени, он должен прекратить отвечать на запросы для этой зоны.

• **minimum (negative cache TTL):** Как долго кэширующие серверы должны кэшировать отрицательные ответы (например, "домен не существует").

11. Как обеспечить защиту зоны от скачивания и просмотра?

• **Списки контроля доступа (ACLs):** Определите ACL в named.conf.options для ограничения доступа к DNS-серверу. Например, можно разрешить zone transfer только доверенным slave-серверам.

• **Разрешить zone transfer только для slave-серверов:** В конфигурации зоны укажите, каким серверам разрешено запрашивать zone transfer. Пример: allow-transfer { slave1_ip; slave2_ip; };

• **DNSSEC (DNS Security Extensions):** DNSSEC добавляет криптографические подписи к DNS-записям, чтобы предотвратить подделку и гарантировать целостность данных.

• **Отключить рекурсию для внешних клиентов (если это авторитативный сервер):** Если DNS-сервер является авторитативным (а

не рекурсивным), отключите рекурсию для запросов из внешних сетей, чтобы предотвратить его использование для DDoS-атак.

12. Какая запись RR применяется при создании почтовых серверов?

Запись **MX (Mail eXchange)** используется для указания почтовых серверов, принимающих почту для домена. MX-запись содержит имя почтового сервера и приоритет (число), определяющий порядок использования серверов.

13. Как протестировать работу сервера доменных имён?

• **nslookup:** Интерактивный инструмент для запроса DNS-серверов. Позволяет запрашивать различные типы записей для доменных имен. Пример: nslookup www.example.com

• **dig (domain information groper):** Более мощный инструмент, чем nslookup, для запроса DNS-серверов. Позволяет выполнять более сложные запросы и анализировать ответы.

• **TXT (Text record):** Содержит произвольный текст. Используется для различных целей, таких как проверка домена (SPF, DKIM).

• **SRV (Service record):** Определяет местоположение служб (например, LDAP, SIP) по домену.

• **ping:** Хотя ping обычно используется для проверки доступности хоста, его можно использовать для проверки разрешения имен. Если ping работает с доменным именем, это означает, что DNS работает правильно.

• **Проверка файла /etc/resolv.conf:** Убедитесь, что файл /etc/resolv.conf содержит правильные IP-адреса DNS-серверов.

14. Как запустить, перезапустить или остановить какую-либо службу в системе?

В современных Linux-системах (с использованием systemd) используется команда systemctl:

• **Запустить:** sudo systemctl start <имя_службы> (например, sudo systemctl start named)

• **Перезапустить:** sudo systemctl restart <имя_службы> (например, sudo systemctl restart named)

• **Остановить:** sudo systemctl stop <имя_службы> (например, sudo systemctl stop named)

• **Посмотреть статус:** sudo systemctl status <имя_службы> (например, sudo systemctl status named)

• **Включить автозапуск при загрузке системы:** sudo systemctl enable <имя_службы> (например, sudo systemctl enable named)

• **Отключить автозапуск при загрузке системы:** sudo systemctl disable <имя_службы> (например, sudo systemctl disable named)

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

• **Журналы systemd:** journalctl -u <имя_службы> (например, journalctl -u named). Эта команда покажет журналы для указанной службы. Можно использовать опции, такие как -f для отслеживания в реальном времени, -n <количество_строк> для просмотра последних строк.

• **Проверить вывод службы при запуске вручную:** Иногда полезно запустить службу вручную из командной строки, чтобы увидеть любые ошибки или отладочные сообщения. Это может потребовать знания исполняемого файла службы и его параметров.

• **Использовать отладочные параметры службы:** Некоторые службы имеют параметры командной строки или конфигурационные опции для включения отладочного режима. Проверьте документацию службы.

16. Где хранится отладочная информация по работе системы и служб? Как ее посмотреть?

В современных Linux-системах используется systemd-journald для ведения системных журналов. Журналы хранятся в бинарном формате. Основные способы просмотра журналов:

• **journalctl:** Основная команда для просмотра журналов.

* journalctl: Показать все журналы.

* journalctl -b: Показать журналы с текущей загрузки.

* journalctl -u <имя_службы>: Показать журналы для указанной службы.

* journalctl -f: Отслеживать журналы в реальном времени.

* journalctl -n <количество_строк>: Показать последние N строк. * journalctl --since "YYYY-MM-DD HH:MM:SS": Показать журналы с указанной даты и времени.

• **/var/log/**: Некоторые службы все еще могут вести текстовые логи в каталоге /var/log/.

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

Использовать команду lsof (List Open Files) или proc filesystem:

• **lsof:** * sudo lsof -p : Показать файлы, открытые процессом с указанным PID (Process ID). Например, sudo lsof -p 1234.

* sudo lsof -c <имя_команды>: Показать файлы, открытые процессами с указанным именем команды. Например, sudo lsof -c named.

• **/proc//fd:** Каталог /proc//fd содержит символические ссылки на файлы, открытые процессом с указанным PID.

ChatGPT4 | Midjourney, [25.09.2025 15:04]

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

nmcli (NetworkManager Command Line Interface) - это инструмент командной строки для управления сетевыми соединениями в Linux.

Примеры:

1. Показать список сетевых соединений:

nmcli con show

2. Подключиться к Wi-Fi сети:

nmcli dev wifi connect <имя_сети> password <пароль>

19. Что такое SELinux?

SELinux (Security-Enhanced Linux) - это модуль безопасности ядра Linux, обеспечивающий принудительный контроль доступа (Mandatory Access Control - MAC). Он позволяет определять политику безопасности, которая контролирует доступ процессов к файлам, каталогам, сетевым ресурсам и другим процессам. SELinux выходит за рамки традиционной дискреционной модели контроля доступа (DAC), основанной на UID/GID.

20. Что такое контекст (метка) SELinux?

Контекст SELinux (также называемый меткой) - это атрибут, который присваивается каждому процессу, файлу, каталогу и сетевому ресурсу в системе. Контекст SELinux состоит из трех основных частей:

• user: Идентификатор пользователя SELinux.

• role: Роль, которую выполняет процесс или ресурс.

• type: Тип процесса или ресурса.

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

Используйте команду restorecon:

sudo restorecon -v <путь_к_файлу_или_каталогу>

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

Найдите сообщения о запрете (AVC messages) в журналах: Используйте journalctl или grep для поиска сообщений AVC (Access Vector Cache) в журналах. AVC messages указывают на то, что SELinux заблокировал операцию.

23. Что такое булевый переключатель в SELinux?

Булевы переключатели (SELinux booleans) - это переменные, которые позволяют изменять поведение политики SELinux во время работы системы без необходимости перекомпилировать или перезагружать всю политику. Булевы переключатели можно включать или выключать, чтобы

разрешать или запрещать определенные действия. Они предоставляют гибкий способ настройки SELinux для конкретных потребностей.

24. Как посмотреть список переключателей SELinux и их состояние?

Используйте команду getsebool: getsebool -a

25. Как изменить значение переключателя SELinux?

Используйте команду setsebool:

•Включить переключатель: sudo setsebool <имя_переключателя> 1

2.3. Итог работы

Были приобретены практические навыки по установке и конфигурированию DNS сервера.