

# **Отчет по лабораторной работе 2**

Генералов Даниил, НПИбд-01-21, 1032202280

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	27
5	Контрольные вопросы	28

## Список иллюстраций

3.1	vagrant	7
3.2	dnf	8
3.3	dig	9
3.4	resolv	9
3.5	named	10
3.6	named	11
3.7	named	12
3.8	dig	13
3.9	named	14
3.10	named	15
3.11	wireshark	16
3.12	named	16
3.13	dig	18
3.14	nmcli	19
3.15	named	20
3.16	named	20
3.17	named	21
3.18	named	21
3.19	named	22
3.20	named	22
3.21	dig	23
3.22	host	24
3.23	vagrant	24
3.24	vagrant	25
3.25	vagrant	26

## Список таблиц

# 1 Цель работы

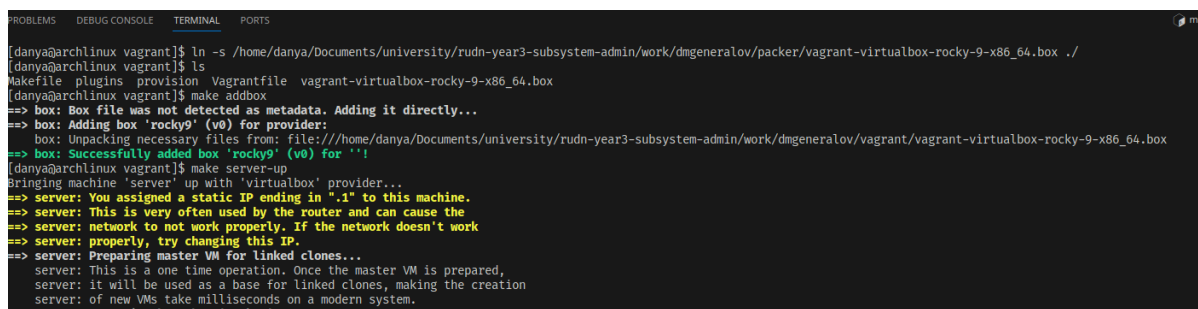
Приобретение практических навыков по установке и конфигурированию DNS-сервера, усвоение принципов работы системы доменных имён.

## 2 Задание

1. Установите на виртуальной машине server DNS-сервер bind и bind-utils (см. раздел 2.4.1).
2. Сконфигурируйте на виртуальной машине server кэширующий DNS-сервер (см. раздел 2.4.2).
3. Сконфигурируйте на виртуальной машине server первичный DNS-сервер (см. раздел 2.4.3).
4. При помощи утилит dig и host проанализируйте работу DNS-сервера (см. раздел 2.4.4).
5. Напишите скрипт для Vagrant, фиксирующий действия по установке и конфигурированию DNS-сервера во внутреннем окружении виртуальной машины server. Соответствующим образом внесите изменения в Vagrantfile (см. раздел 2.4.5).

### 3 Выполнение лабораторной работы

Сначала я запустил сервер через Vagrant. Чтобы его файлы находились в правильной папке, я сначала сделал `make addbox`.



```
PROBLEMS  DEBUG CONSOLE  TERMINAL  PORTS
[danya@archlinux vagrant]$ ln -s /home/danya/Documents/university/rudn-year3-subsystem-admin/work/dmgeneralov/packer/vagrant-virtualbox-rocky-9-x86_64.box ./
[danya@archlinux vagrant]$ ls
Makefile  plugins  provision  Vagrantfile  vagrant-virtualbox-rocky-9-x86_64.box
[danya@archlinux vagrant]$ make addbox
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'rocky9' (v0) for provider:
box: Unpacking necessary files from: file:///home/danya/Documents/university/rudn-year3-subsystem-admin/work/dmgeneralov/vagrant/vagrant-virtualbox-rocky-9-x86_64.box
==> box: Successfully added box 'rocky9' (v0) for ''
[danya@archlinux vagrant]$ make server-up
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: Preparing master VM for linked clones...
server: This is a one time operation. Once the master VM is prepared,
server: it will be used as a base for linked clones, making the creation
server: of new VMs take milliseconds on a modern system.
```

Рис. 3.1: vagrant

После запуска и входа в систему я установил `bind` и `bind-utils`.

```
root@server:~  
[vagrant@server ~]$ sudo -i  
[root@server ~]# dnf install -y bind bind-utils  
Rocky Linux 9 - BaseOS 4.3 kB/s | 4.1 kB 00:00  
Rocky Linux 9 - AppStream 7.3 kB/s | 4.5 kB 00:00  
Rocky Linux 9 - Extras 5.4 kB/s | 2.9 kB 00:00  
Package bind-utils-32:9.16.23-11.el9_2.2.x86_64 is already installed.  
Dependencies resolved.  
=====
```

Package	Arch	Version	Repository	Size
---------	------	---------	------------	------

```
=====
```

Installing:

bind	x86_64	32:9.16.23-11.el9_2.2	appstream	487 k
------	--------	-----------------------	-----------	-------

Installing dependencies:

bind-dnssec-doc	noarch	32:9.16.23-11.el9_2.2	appstream	44 k
python3-bind	noarch	32:9.16.23-11.el9_2.2	appstream	60 k
python3-ply	noarch	3.11-14.el9.0.1	baseos	103 k

Installing weak dependencies:

bind-dnssec-utils	x86_64	32:9.16.23-11.el9_2.2	appstream	112 k
-------------------	--------	-----------------------	-----------	-------

Transaction Summary

```
=====
```

Install 5 Packages

Total download size: 806 k

Рис. 3.2: dnf

Можно проверить, что интернет и DNS работают.



```
Complete!
[root@server ~]# dig www.yandex.ru

; <<>> DiG 9.16.23-RH <<>> www.yandex.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12266
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.yandex.ru.                IN      A

;; ANSWER SECTION:
www.yandex.ru.                3600    IN      A      77.88.55.88
www.yandex.ru.                3600    IN      A      5.255.255.70
www.yandex.ru.                3600    IN      A      77.88.55.60
www.yandex.ru.                3600    IN      A      5.255.255.77

;; Query time: 40 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Sat Nov 11 11:22:07 UTC 2023
;; MSG SIZE rcvd: 95

[root@server ~]#
```

Рис. 3.3: dig

В /etc/resolv.conf находятся две строки: IP-адрес DNS-сервера и search-домен для данной сети.

```
;; MSG SIZE rcvd: 95

[root@server ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search user.net
nameserver 10.0.2.3
[root@server ~]#
```

Рис. 3.4: resolv

В /etc/named.conf хранятся настройки DNS-сервера: по каким портам и адресам стоит слушать (по умолчанию – localhost и порт 53), каким хостам стоит разрешать

делать запросы (опять же, только localhost), в каких папках и файлах находится информация, которую использует сервер, вроде статистики и кеша рекурсии (все в /var/named), а также путь, куда стоит писать логи, и какие зоны следует публиковать (по умолчанию – только корневую зону в named.ca как zone hints).

```
options {
    listen-on port 53 { 127.0.0.1; };
    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; };

    /*
     - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
     - If you are building a RECURSIVE (caching) DNS server, you need to enable
       recursion.
     - If your recursive DNS server has a public IP address, you MUST enable access
       control to limit queries to your legitimate users. Failing to do so will
       cause your server to become part of large scale DNS amplification
       attacks. Implementing BCP38 within your network would greatly
       reduce such attack surface
    */
    recursion yes;

    dnssec-validation yes;

    managed-keys-directory "/var/named/dynamic";
    geoip-directory "/usr/share/GeoIP";

    pid-file "/run/named/named.pid";
    session-keyfile "/run/named/session.key";
}
```

Рис. 3.5: named

Файл /var/named/named.ca – это лог команды dig, которая спрашивает информацию о корневой зоне. Этот формат файла является также форматом зоны для named. Здесь находятся А-записи для серверов корневой зоны, которые почти никогда не меняются и поэтому могут быть кешированы на такое долгое время, чтобы стать частью стандартной конфигурации сервера.

```
root@server ~]# cat /var/named/named.conf
; <<>> DiG 9.11.3-RedHat-9.11.3-3.fc27 <<>> +bufsize=1200 +norec @a.root-servers.net
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46900
;; flags: qr aa; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 27

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1472
;; QUESTION SECTION:
; .                IN      NS

;; ANSWER SECTION:
.                518400 IN      NS      a.root-servers.net.
.                518400 IN      NS      b.root-servers.net.
.                518400 IN      NS      c.root-servers.net.
.                518400 IN      NS      d.root-servers.net.
.                518400 IN      NS      e.root-servers.net.
.                518400 IN      NS      f.root-servers.net.
.                518400 IN      NS      g.root-servers.net.
.                518400 IN      NS      h.root-servers.net.
.                518400 IN      NS      i.root-servers.net.
.                518400 IN      NS      j.root-servers.net.
.                518400 IN      NS      k.root-servers.net.
.                518400 IN      NS      l.root-servers.net.
.                518400 IN      NS      m.root-servers.net.

;; ADDITIONAL SECTION:
a.root-servers.net. 518400 IN      A      192.41.0.4
```

Рис. 3.6: named

Файл `/var/named/named.localhost` определяет зону, для которой TTL равен одному дню, и зона называется `@`, что является ссылкой на название этой зоны (то есть зона – `localhost`). Описывается, что для этих зон сервером имен является сам `localhost`, а IP-адресом – `127.0.0.1`. Эта зона существует, чтобы отвечать на запрос `localhost`-адреса через DNS. Файл `/var/named/named.loopback` определяет обратную зону: он говорит, что зона `@` – а именно `1.0.0.127.in-addr.arpa` – связана PTR-записью с `localhost`.

```

[root@server ~]# cat /var/named/named.localhost
$TTL 1D
@      IN SOA  @ rname.invalid. (
                                0      ; serial
                                1D     ; refresh
                                1H     ; retry
                                1W     ; expire
                                3H )   ; minimum

    NS      @
    A       127.0.0.1
    AAAA    ::1
[root@server ~]# cat /var/named/named.loopback
$TTL 1D
@      IN SOA  @ rname.invalid. (
                                0      ; serial
                                1D     ; refresh
                                1H     ; retry
                                1W     ; expire
                                3H )   ; minimum

    NS      @
    A       127.0.0.1
    AAAA    ::1
    PTR     localhost.
[root@server ~]# █

```

Рис. 3.7: named

Теперь я запустил named и попробовал сделать запрос через системный resolver и через локальный. Но локальный сервер отвечал ошибкой.

```

; <<>> DiG 9.16.23-RH <<>> www.yandex.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62747
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.yandex.ru.                IN      A

;; ANSWER SECTION:
www.yandex.ru.                3600    IN      A      77.88.55.60
www.yandex.ru.                3600    IN      A      5.255.255.70
www.yandex.ru.                3600    IN      A      5.255.255.77
www.yandex.ru.                3600    IN      A      77.88.55.88

;; Query time: 9 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Sat Nov 11 11:46:09 UTC 2023
;; MSG SIZE rcvd: 95

[root@server ~]# dig @127.0.0.1 www.yandex.ru

; <<>> DiG 9.16.23-RH <<>> @127.0.0.1 www.yandex.ru
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 9483
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

... OPT PSEUDOSECTION:

```

Рис. 3.8: dig

Если посмотреть в лог, то можно увидеть ошибки, связанные с тем, что сервер пытается использовать IPv6-DNS-сервера, чтобы получить ответ, а IPv6 недоступен.

```

RMERR resolving './NS/IN': 198.41.0.4#53
RMERR resolving './NS/IN': 199.7.83.42#53
network unreachable resolving './NS/IN': 2001:503:c27::2:30#53
RMERR resolving './NS/IN': 198.97.190.53#53
RMERR resolving './NS/IN': 192.33.4.12#53
RMERR resolving './NS/IN': 192.5.5.241#53
RMERR resolving './NS/IN': 199.7.91.13#53
resolver priming query complete
network unreachable resolving '_._yandex.ru/A/IN': 2001:dc3::35#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:7fd::1#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:503:ba3e::2:30#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:500:9f::42#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:500:1::53#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:500:2::c#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:500:2f::f#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:500:2d::d#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:7fe::53#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:500:200::b#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:500:a8::e#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:500:12::d0d#53
network unreachable resolving '_._yandex.ru/A/IN': 2001:503:c27::2:30#53
network unreachable resolving './NS/IN': 2001:dc3::35#53
network unreachable resolving './NS/IN': 2001:7fe::53#53
network unreachable resolving './NS/IN': 2001:7fd::1#53
network unreachable resolving './NS/IN': 2001:500:a8::e#53
network unreachable resolving './NS/IN': 2001:500:12::d0d#53
network unreachable resolving './NS/IN': 2001:503:ba3e::2:30#53

```

Рис. 3.9: named

Чтобы исправить это, я настроил в /etc/sysconfig/named, чтобы серверу передавался параметр -4, который должен сделать так, чтобы использовался только IPv4.

```
GNU nano 5.6.1 /etc/sysconfig/named
# BIND named process options
# ~~~~~
#
# OPTIONS="whatever" -- These additional options will be passed
#                       at startup. Don't add -t here, enable
#                       -chroot.service unit file.
#
# NAMEDCONF=/etc/named/alternate.conf
#                       -- Don't use -c to change configuration
#                       Extend systemd named.service instead
#                       variable.
#
# DISABLE_ZONE_CHECKING -- By default, service file calls named-
#                           utility for every zone to ensure all
#                           valid before named starts. If you set
#                           to 'yes' then service file doesn't pe
#                           checks.
OPTIONS="-4"
```

Рис. 3.10: named

Теперь остались только сообщения о FORMERR – только сейчас я понял, что это значит “format error”. Поэтому я попробовал использовать Wireshark на внешнем компьютере, чтобы понять, что происходит с трафиком DNS. BIND спрашивает каждый из корневых серверов про NS для корня, и получает ответ, но этот ответ не нравится BIND.

No.	Time	Source	Destination	Protocol	Length	Info
330	10.094236630	10.0.0.128	192.203.230.10	DNS	82	Standard query 0x3ed2 NS <Root> OPT
331	10.095727649	10.0.0.128	192.203.230.10	DNS	88	Standard query 0x7f43 A _._com OPT
332	10.097396773	192.203.230.10	10.0.0.128	DNS	567	Standard query response 0x3ed2 NS <Root> NS a.root-
336	10.101903922	10.0.0.128	192.203.230.10	DNS	108	Standard query 0x02fb NS <Root> OPT
338	10.108817832	192.203.230.10	10.0.0.128	DNS	593	Standard query response 0x02fb NS <Root> NS a.root-
341	10.113610900	10.0.0.128	199.7.83.42	DNS	82	Standard query 0xe5e3 NS <Root> OPT
344	10.116859236	199.7.83.42	10.0.0.128	DNS	567	Standard query response 0xe5e3 NS <Root> NS a.root-
348	10.119022791	10.0.0.128	199.7.83.42	DNS	108	Standard query 0x636e NS <Root> OPT
350	10.126079380	199.7.83.42	10.0.0.128	DNS	593	Standard query response 0x636e NS <Root> NS a.root-
353	10.128974950	10.0.0.128	193.0.14.129	DNS	82	Standard query 0x1668 NS <Root> OPT
356	10.132486182	193.0.14.129	10.0.0.128	DNS	567	Standard query response 0x1668 NS <Root> NS a.root-
360	10.134420206	10.0.0.128	193.0.14.129	DNS	108	Standard query 0x5f90 NS <Root> OPT
362	10.141785210	193.0.14.129	10.0.0.128	DNS	593	Standard query response 0x5f90 NS <Root> NS a.root-
365	10.143150131	10.0.0.128	192.112.36.4	DNS	82	Standard query 0x53fc NS <Root> OPT
368	10.147149520	192.112.36.4	10.0.0.128	DNS	567	Standard query response 0x53fc NS <Root> NS a.root-
372	10.149173234	10.0.0.128	192.112.36.4	DNS	108	Standard query 0x613e NS <Root> OPT

Additional RRs: 1  
Queries  
Answers  
<Root>: type NS, class IN, ns a.root-servers.net  
<Root>: type NS, class IN, ns b.root-servers.net  
Name: <Root>  
Type: NS (authoritative Name Server) (2)  
Class: IN (0x0001)  
Time to live: 512864 (5 days, 22 hours, 27 minutes, 44 seconds)  
Data length: 4  
Name Server: b.root-servers.net  
<Root>: type NS, class IN, ns a.root-servers.net

0000 4c cc 6a e2 4a f6 50 ff 20 8c 34 53 08 00  
0010 02 29 c7 b2 40 00 40 11 bf bb c0 cb e6 00  
0020 00 80 00 35 a1 a4 02 15 13 13 3e d2 82 a1  
0030 00 0e 00 00 00 01 00 00 02 00 01 00 00 00  
0040 00 07 d3 60 00 14 01 61 0c 72 6f 6f 74 21  
0050 72 76 65 72 73 03 6e 65 74 00 00 00 02 00  
0060 07 d3 60 00 04 01 62 c0 1e 00 00 02 00 00  
0070 d3 60 00 04 01 63 c0 1e 00 00 02 00 01 00  
0080 d3 60 00 04 01 64 c0 1e 00 00 02 00 01 00  
0090 00 04 01 65 c0 1e 00 00 02 00 01 00 07 d3  
00a0 04 01 66 c0 1e 00 00 02 00 01 00 07 d3 6f  
00b0 01 67 c0 1e 00 00 02 00 01 00 07 d3 60 00  
00c0 68 c0 1e 00 00 02 00 01 00 07 d3 60 00 00  
00d0 c0 1e 00 00 02 00 01 00 07 d3 60 00 04 01  
00e0 1e 00 00 02 00 01 00 07 d3 60 00 04 01

Domain Name System: Protocol

Packets: 2293 · Displayed: 164 (7.2%)

Profile: Default

Рис. 3.11: wireshark

В этот момент я решил перейти к шагу, когда рекурсия заменяется forwarding, и перенаправлять все запросы на сервер 1.1.1.1, который работает для меня. Для этого в настройках BIND я добавил forwarders и выключил DNSSEC.

```

- If your recursive DNS server has a public IP address, you
control to limit queries to your legitimate users. Failing
cause your server to become part of large scale DNS amplif
attacks. Implementing BCP38 within your network would grea
reduce such attack surface

*/
recursion yes;

forwarders { 1.1.1.1; 1.0.0.1 };
forward first;

dnssec-enable no;
dnssec-validation no;

managed-keys-directory "/var/named/dynamic";
geoip-directory "/usr/share/GeoIP";

```

Рис. 3.12: named



После перезагрузки BIND у меня наконец получилось сделать запрос к локальному серверу и получить ответ. У этого ответа низкий TTL по сравнению с запросом к системному resolver, потому что мы делаем forward на кеширующий сервер, который возвращает нам ответ, который он получил давно.

```

; www.yandex.ru.          IN      A

;; ANSWER SECTION:
www.yandex.ru.          3600    IN      A      77.88.55.88
www.yandex.ru.          3600    IN      A      5.255.255.77
www.yandex.ru.          3600    IN      A      5.255.255.70
www.yandex.ru.          3600    IN      A      77.88.55.60

;; Query time: 15 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Sat Nov 11 12:13:48 UTC 2023
;; MSG SIZE rcvd: 95

[root@server ~]# dig @127.0.0.1 www.yandex.ru

; <>> 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: 3257
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: cf7fbe24c08bf68101000000654f6ffd3e1e57979559645e (good)
;; QUESTION SECTION:
; www.yandex.ru.          IN      A

;; ANSWER SECTION:
www.yandex.ru.          158     IN      A      5.255.255.77
www.yandex.ru.          158     IN      A      77.88.55.60
www.yandex.ru.          158     IN      A      77.88.55.88
www.yandex.ru.          158     IN      A      5.255.255.70

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sat Nov 11 12:13:49 UTC 2023
;; MSG SIZE rcvd: 134

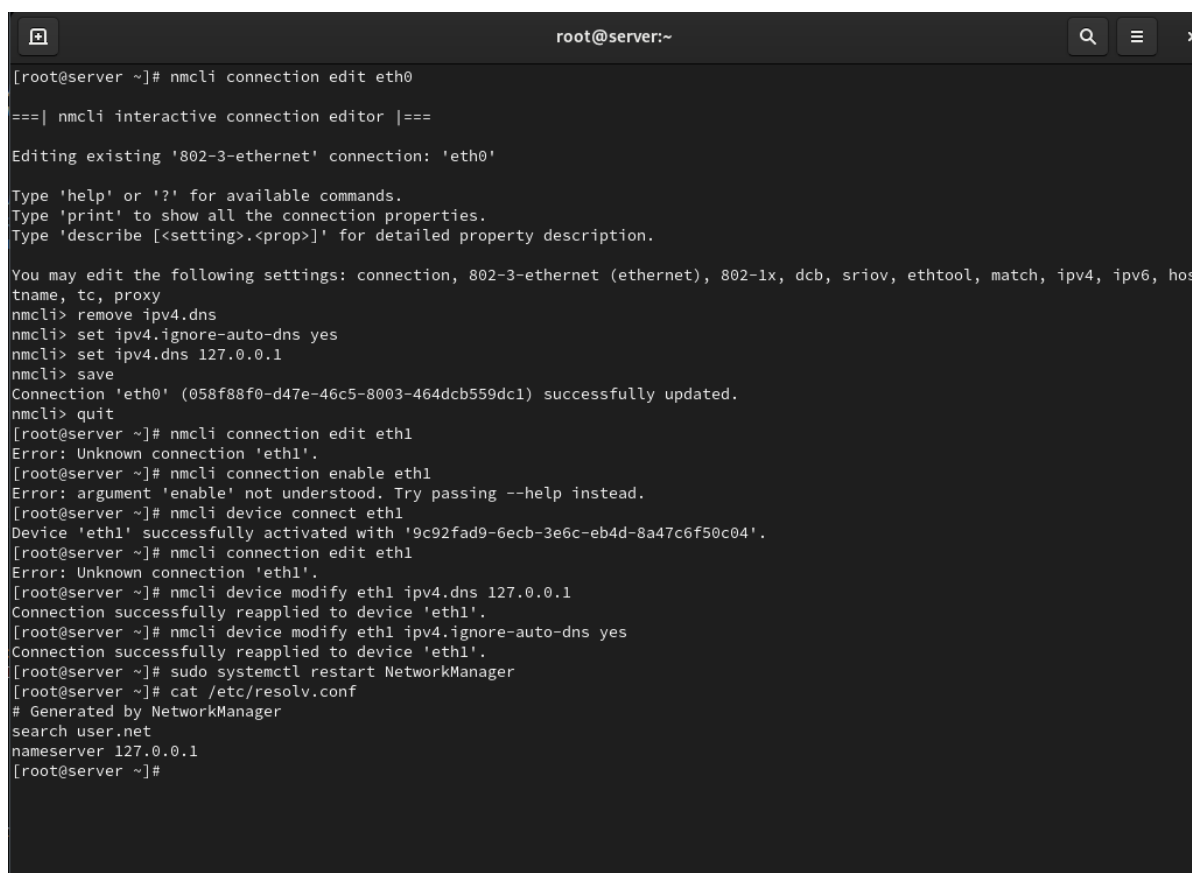
[root@server ~]# █

```

Рис. 3.13: dig

Теперь мы меняем настройки networkmanager, чтобы использовать этот сервер для всех сетевых соединений. После перезагрузки networkmanager изменения

применились к /etc/resolv.conf.



```
root@server:~  
[root@server ~]# 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, host-  
name, 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' (058f88f0-d47e-46c5-8003-464dcb559dc1) successfully updated.  
nmcli> quit  
[root@server ~]# nmcli connection edit eth1  
Error: Unknown connection 'eth1'.  
[root@server ~]# nmcli connection enable eth1  
Error: argument 'enable' not understood. Try passing --help instead.  
[root@server ~]# nmcli device connect eth1  
Device 'eth1' successfully activated with '9c92fad9-6ecb-3e6c-eb4d-8a47c6f50c04'.  
[root@server ~]# nmcli connection edit eth1  
Error: Unknown connection 'eth1'.  
[root@server ~]# nmcli device modify eth1 ipv4.dns 127.0.0.1  
Connection successfully reapplied to device 'eth1'.  
[root@server ~]# nmcli device modify eth1 ipv4.ignore-auto-dns yes  
Connection successfully reapplied to device 'eth1'.  
[root@server ~]# sudo systemctl restart NetworkManager  
[root@server ~]# cat /etc/resolv.conf  
# Generated by NetworkManager  
search user.net  
nameserver 127.0.0.1  
[root@server ~]#
```

Рис. 3.14: nmcli

В настройках /etc/named.conf мы разрешаем доступ с всех устройств виртуальной сети.

```

с // See /usr/share/doc/bind*/sample/ for example named configuratio
щи //
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; 192.0.0.0/16; };

    /*

```

Рис. 3.15: named

Теперь мы начинаем настройку авторитетной зоны для этого сервера. Для этого сначала подключаем файл, где будет находится настройка этой зоны: `dmgeneralov.net`.

```

and.     type hint;
        file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";

id: include "/etc/named/dmgeneralov.net";
HORI

```

Рис. 3.16: named

Затем в этом файле указываем прямую и обратную зоны, и где они будут находиться.

```
GNU nano 5.6.1 /etc/named/dmgeneralov.net
zone "dmgeneralov.net" IN {
    type master;
    file "master/fz/dmgeneralov.net";
    allow-update { none; };
};

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

Рис. 3.17: named

После этого создаем папки и файл прямой зоны.

```
GNU nano 5.6.1 /var/named/master/fz/dmgeneralov.net
$TTL 1D
@      IN SOA  @ server.dmgeneralov.net. (
                                2023111100      ; serial
                                1D      ; refresh
                                1H      ; retry
                                1W      ; expire
                                3H )   ; minimum
      NS      @
      A       192.168.1.1
$ORIGIN dmgeneralov.net.
server A      192.168.1.1
ns     A      192.168.1.1
```

Рис. 3.18: named

А затем файл обратной зоны.

```
GNU nano 5.6.1 /var/named/master/rz/192.168.1
$TTL 1D
@      IN SOA  @ server.dmgeneralov.net. (
                                0      ; serial
                                1D      ; refresh
                                1H      ; retry
                                1W      ; expire
                                3H )    ; minimum

      NS      @
      A      192.168.1.1
      PTR     server.dmgeneralov.net.

$ORIGIN 1.168.192.in-addr.arpa.
1      PTR     server.dmgeneralov.net.
1      PTR     ns.dmgeneralov.net.
```

Рис. 3.19: named

Теперь нужно разрешить BIND доступ к этим файлам. Для этого нужно сначала сменить владельца этих файлов, а затем с помощью `restorecon` вернуть стандартный SELinux-контекст всем файлам в папке. После этого мы запускаем BIND, и он запускается без ошибок.

```
[root@server ~]# chown -R named:named /etc/named
[root@server ~]# chown -R named:named /var/named
[root@server ~]# restorecon -vR /etc
Relabeled /etc/sysconfig/network-scripts/ifcfg-eth1 from unconfined_u:object_r:user_tmp_t:s0 to unconfined_u:object_r:net_conf_t:s0
[root@server ~]# restorecon -vR /var/named
[root@server ~]# getsebool -a | grep named
named_tcp_bind_http_port --> off
named_write_master_zones --> on
[root@server ~]# journalctl -f
Nov 11 12:21:37 server.user.net named[19962]: FORMERR resolving './NS/IN': 192.33.4.12#53
Nov 11 12:21:37 server.user.net named[19962]: FORMERR resolving './NS/IN': 192.112.36.4#53
Nov 11 12:21:37 server.user.net named[19962]: FORMERR resolving './NS/IN': 192.36.148.17#53
Nov 11 12:21:37 server.user.net named[19962]: FORMERR resolving './NS/IN': 199.9.14.201#53
Nov 11 12:21:37 server.user.net named[19962]: FORMERR resolving './NS/IN': 192.203.230.10#53
Nov 11 12:21:37 server.user.net named[19962]: FORMERR resolving './NS/IN': 193.0.14.129#53
Nov 11 12:21:37 server.user.net named[19962]: FORMERR resolving './NS/IN': 202.12.27.33#53
Nov 11 12:21:37 server.user.net named[19962]: resolver priming query complete
Nov 11 12:36:01 server.user.net anacron[19177]: Job 'cron.weekly' started
Nov 11 12:36:01 server.user.net anacron[19177]: Job 'cron.weekly' terminated
Nov 11 12:36:52 server.user.net systemd[16875]: Started VTE child process 20196 launched by gnome-terminal-server process 17756
.
Nov 11 12:37:01 server.user.net PackageKit[17216]: search-file transaction /18_bbbabcbdb from uid 1000 finished with success after 282ms
Nov 11 12:37:03 server.user.net sudo[20243]: vagrant : TTY=pts/1 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/systemctl restart named
Nov 11 12:37:03 server.user.net sudo[20243]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Nov 11 12:37:03 server.user.net systemd[1]: Stopping Berkeley Internet Name Domain (DNS)...
Nov 11 12:37:03 server.user.net named[19962]: received control channel command 'stop'
```

Рис. 3.20: named

После этого можно проверить работу DNS-сервера. Мы видим, что для dmgeneralov.net есть IP-адрес 192.168.1.1, и тот же самый адрес имеет server.dmgeneralov.net и ns.dmgeneralov.net.

```
[root@server ~]#
[root@server ~]# dig ns.dmgeneralov.net

; <<>> DiG 9.16.23-RH <<>> ns.dmgeneralov.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11599
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 949fe95fa2e1865401000000654f75ecf880d8bbe7e4d695 (good)
;; QUESTION SECTION:
;ns.dmgeneralov.net.          IN      A

;; ANSWER SECTION:
ns.dmgeneralov.net.  86400  IN      A      192.168.1.1

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sat Nov 11 12:39:08 UTC 2023
;; MSG SIZE rcvd: 91

[root@server ~]# host -l dmgeneralov.net
dmgeneralov.net name server dmgeneralov.net.
dmgeneralov.net has address 192.168.1.1
ns.dmgeneralov.net has address 192.168.1.1
server.dmgeneralov.net has address 192.168.1.1
```

Рис. 3.21: dig

Отдельно говоря, dmgeneralov.net действительно указывает на 192.168.1.1, и тот же самый адрес имеет server.dmgeneralov.net и ns.dmgeneralov.net.

```
[root@server ~]# host -t A dmgeneralov.net
dmgeneralov.net has address 192.168.1.1
[root@server ~]# host -t PTR 192.168.1.1
1.1.168.192.in-addr.arpa domain name pointer server.dmgeneralov.net.
1.1.168.192.in-addr.arpa domain name pointer ns.dmgeneralov.net.
[root@server ~]#
```

Рис. 3.22: host

Теперь мы экспортировали из виртуальной машины все файлы конфигурации.

```
[root@server ~]# cd /vagrant
[root@server vagrant]# ls
Makefile  plugins  provision  Vagrantfile  vagrant-virtualbox-rocky-9-x86_64.box  vm
[root@server vagrant]# mkdir -p provision/server/dns/etc/named
[root@server vagrant]# mkdir -p provision/server/dns/var/named/master
[root@server vagrant]# cp -R /etc/named.conf ./provision/server/dns/etc/
[root@server vagrant]# cp -R /etc/named/* ./provision/server/dns/etc/named/
[root@server vagrant]# cp -R /var/named/master/* ./provision/server/dns/var/named/master/
[root@server vagrant]#
```

Рис. 3.23: vagrant

Создали скрипт `dns.sh`, который выполняет эту настройку автоматически.




```
work > dmgeneralov > vagrant > provision > server >  dns.sh
1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5  echo "Install needed packages"
6  dnf -y install bind bind-utils
7
8  echo "Copy configuration files"
9  cp -R /vagrant/provision/server/dns/etc/* /etc
10 cp -R /vagrant/provision/server/dns/var/named/* /var/named
11 chown -R named:named /etc/named
12 chown -R named:named /var/named
13 restorecon -vR /etc
14 restorecon -vR /var/named
15
16 echo "Configure firewall"
17 firewall-cmd --add-service=dns
18 firewall-cmd --add-service=dns --permanent
19
20 echo "Tuning SELinux"
21 setsebool named_write_master_zones 1
22 setsebool -P named_write_master_zones 1
23
24 echo "Change dns server address"
25 nmcli connection edit "eth0" <<EOF
26 remove ipv4.dns
27 set ipv4.ignore-auto-dns yes
28 set ipv4.dns 127.0.0.1
29 save
30 quit
31 EOF
32
33 systemctl restart NetworkManager
34
35 echo "Start named service"
36 systemctl enable named
37 systemctl start named
```

Рис. 3.24: vagrant

И добавили его как шаг настройки в Vagrantfile.

```
37
38     server.vm.provision "server dummy",
39         type: "shell",
40         preserve_order: true,
41         path: "provision/server/01-dummy.sh"
42
43     server.vm.provision "server dns",
44         type: "shell",
45         preserve_order: true,
46         path: "provision/server/dns.sh"
47
48     server.vm.provider :virtualbox do |v|
49         v.linked_clone = true
50         # Customize the amount of memory on the VM
```

Рис. 3.25: vagrant

## 4 Выводы

Я получил опыт настройки DNS-сервера BIND и сохранения сделанных настроек в систему Vagrant.

## 5 Контрольные вопросы

1. Что такое DNS? Протокол и система серверов, которые предоставляют доступ к базе данных, соотносящей доменные имена с IP-адресами и другой информацией.
2. Каково назначение кэширующего DNS-сервера? Временно сохранять ответы от авторитетных DNS-серверов, чтобы сэкономить трафик и нагрузку на них.
3. Чем отличается прямая DNS-зона от обратной? Прямая зона – от имени к IP-адресу, а обратная – от IP-адреса к имени.
4. В каких каталогах и файлах располагаются настройки DNS-сервера? Кратко охарактеризуйте, за что они отвечают. `/etc/named` – общие настройки сервера: какие интерфейсы слушать, какие протоколы, следует ли делать zone transfer... `/var/named` – файлы зоны, для которых сервер является авторитетным, и которые он отвечает сам.
5. Что указывается в файле `resolv.conf`? IP-адрес DNS-серверов, которые стоит использовать для разрешения имен на этом компьютере, а также search-домен (потенциальный постфикс имен) и настройки механизма разрешения имен.
6. Какие типы записи описания ресурсов есть в DNS и для чего они используются? Есть много различных, среди них:

- SOA: для этой зоны является авторитетом этот DNS-сервер и эта организация
  - NS: для этой зоны авторитетным DNS-сервером является это имя
  - A/AAAA: этот домен относится к этому IPv4/IPv6-адресу.
  - PTR: для этого IP-адреса (заданого в in-addr.arpa зоне) существует A-запись от этого домена
  - CNAME: обращения к данному домену должны быть переадресованы к другому домену
  - MX: для почты, направленной на данный домен, следует связываться с сервером, находящимся по этому имени.
7. Для чего используется домен in-addr.arpa? Поддомены этой зоны используются для обратных запросов (IP -> имя). По запросу PTR 4.3.2.1.in-addr.arpa можно узнать, какой домен указывает на IP-адрес 1.2.3.4.
  8. Для чего нужен демон named? Он слушает и отвечает на DNS-запросы, либо как авторитет для определенной зоны, либо рекурсивно.
  9. В чём заключаются основные функции slave-сервера и master-сервера? Master-сервер хранит авторитетную копию записей для данной зоны. Помимо обслуживания внешних запросов, он предоставляет zone-transfer для своих slave-серверов, которые могут быть использованы как резервные в случае недоступности master-сервера.
  10. Какие параметры отвечают за время обновления зоны? Этот параметр – refresh в SOA-записи. Он указывает, после какого времени slave-сервер должен скачать новое состояние зоны.
  11. Как обеспечить защиту зоны от скачивания и просмотра? Для этого нужно запретить AXFR-операцию на DNS-сервере.
  12. Какая запись RR применяется при создании почтовых серверов?

МХ.

13. Как протестировать работу сервера доменных имён? Использовать `dig example.com @192.168.1.1` или `host example.com 127.0.0.1`
14. Как запустить, перезапустить или остановить какую-либо службу в системе? `systemctl start <service>`, `systemctl restart`, `systemctl stop`.
15. Как посмотреть отладочную информацию при запуске какого-либо сервиса или службы? `systemctl status`
16. Где храниться отладочная информация по работе системы и служб? Как её посмотреть? Она хранится в системном журнале и её можно посмотреть с помощью `journalctl`.
17. Как посмотреть, какие файлы использует в своей работе тот или иной процесс? Приведите несколько примеров. `lsof -p <pid>`, `ls -l /proc/<pid>/fd`.
18. Приведите несколько примеров по изменению сетевого соединения при помощи командного интерфейса `nmcli`. Подключиться к Ethernet: `nmcli device connect enp3s0` Отключиться от Ethernet: `nmcli device disconnect enp3s0` Поменять IP-адрес подключения: `nmcli device modify enp3s0 ipv4.addresses "10.0.0.129"`
19. Что такое SELinux? Система мандатного контроля доступа, которая используется для ограничения возможностей программ и пользователей в системе.
20. Что такое контекст (метка) SELinux? Информация о том, кто имеет право выполнять какие операции с файлом.
21. Как восстановить контекст SELinux после внесения изменений в конфигурационные файлы? `restorecon /etc/conf.d`

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

```
audit2allow -M local << EOF
```

```
audit(...): avc: denied { write } for pid=... comm="..." name="..." dev=... in
```

```
EOF
```

```
semodule -i local.pp
```

23. Что такое булевый переключатель в SELinux? Это значение, которое может быть true или false, и используется для настройки готовой SELinux-политики.
24. Как посмотреть список переключателей SELinux и их состояние? semanage boolean -l
25. Как изменить значение переключателя SELinux? setsebool <boolean> on/off