

**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: Бансимба Клодели Дъегра

Студ. билет № 1032215651

Группа: НПИбд-02-22

**МОСКВА**

2024 г.

### Цель работы:

Целью данной работы является приобретение практических навыков по установке и конфигурированию DNS-сервера, усвоение принципов работы системы доменных имён.

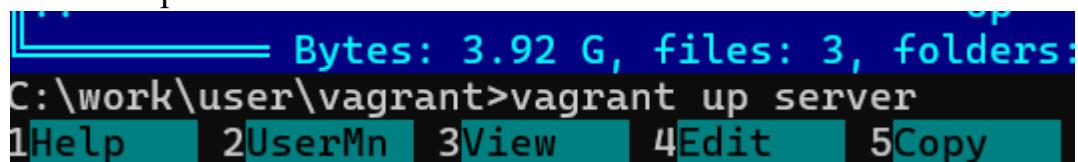
### Выполнение работы:

Загрузим нашу операционную систему и перейдем в рабочий каталог с проектом:

```
cd /var/tmp/claudely/vagrant
```

Далее запустим виртуальную машину server (Рис. 1.1):

```
make server-up
```



**Рис. 1.1.** Открытие рабочего каталога с проектом и запуск виртуальной

машины server.

На виртуальной машине server войдём под созданным нами в предыдущей работе пользователем и откройте терминал. Перейдём в режим суперпользователя: sudo -i

И установим bind и bind-utils (Рис. 1.2):

```
dnf -y install bind bind-utils
```

```
[user@server.user.net ~]$ sudo -i
[sudo] password for user:
[root@server.user.net ~]# dnf -y install bind bind-utils
Last metadata expiration check: 0:23:11 ago on Tue 06 Jan 2026 01:18:33 PM UTC.
Package bind-utils-32:9.16.23-18.el9_4.1.x86_64 is already installed.
Dependencies resolved.
=====
 Package          Architecture Version      Repository   Size
=====
Installing:
 bind             x86_64       32:9.16.23-34.el9_7.1 appstream    488 k
Upgrading:
 bind-libs        x86_64       32:9.16.23-34.el9_7.1 appstream    1.2 M
bind-license     noarch      32:9.16.23-34.el9_7.1 appstream    13 k
bind-utils       x86_64       32:9.16.23-34.el9_7.1 appstream    199 k
openssl          x86_64       1:3.5.1-4.el9_7      baseos       1.4 M
openssl-devel    x86_64       1:3.5.1-4.el9_7      appstream    3.4 M
openssl-libs     x86_64       1:3.5.1-4.el9_7      baseos       2.3 M
Installing dependencies:
 bind-dnssec-doc noarch      32:9.16.23-34.el9_7.1 appstream    45 k
openssl-fips-provider x86_64  1:3.5.1-4.el9_7      baseos       812 k
python3-bind     noarch      32:9.16.23-34.el9_7.1 appstream    61 k
python3-ply      noarch      3.11-14.el9.0.1     baseos       103 k
Installing weak dependencies:
 bind-dnssec-utils x86_64       32:9.16.23-34.el9_7.1 appstream    113 k
=====
Transaction Summary
=====
Install 6 Packages
Upgrade 6 Packages
```

**Рис. 1.2.** Переход в режим суперпользователя и установка bind,bind-utils.

С помощью утилиты dig сделаем запрос к DNSадресу www.yandex.ru  
(Рис. 1.3):

```
dig www.yandex.ru
=====
Complete!
[root@server.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: 58434
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

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

;; ANSWER SECTION:
www.yandex.ru.      3600    IN      A      5.255.255.77
www.yandex.ru.      3600    IN      A      77.88.44.55
www.yandex.ru.      3600    IN      A      77.88.55.88

;; Query time: 28 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Tue Jan 06 13:50:10 UTC 2026
;; MSG SIZE  rcvd: 79

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

**Рис. 1.3.** Запрос с помощью утилиты dig.

Просмотрим содержание файлов /etc/resolv.conf (Рис. 2.1),

/etc/named.conf (Рис. 2.2), /var/named/named.ca (Рис. 2.3),

/var/named/named.localhost (Рис. 2.4), /var/named/named.loopback (Рис. 2.5).

```
[root@server.user.net ~]# sudo -i
[root@server.user.net ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search user.net
nameserver 10.0.2.3
[root@server.user.net ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search user.net
nameserver 10.0.2.3
[root@server.user.net ~]# cat /etc/named.conf
//
// 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; };
    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.recurse";
    allow-query     { localhost; };

/*
 - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
```

**Рис. 2.2.** Просмотр содержания файла /etc/named.conf. и Просмотр содержания файла /etc/resolv.conf

```
root@server:~ [root@server.user.net ~]# cat /var/named/named.ca
; <>> DiG 9.18.20 <>> -4 +tcp +norec +nostats @d.root-servers.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47286
;; flags: qr aa; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 27
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1450
;; 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.
```

**Рис. 2.3.** Просмотр содержания файла /var/named/named.ca.

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

**Рис. 2.4.** Просмотр содержания файла /var/named/named.localhost.

Запустим DNS-сервер: systemctl

start named

Включим запуск DNS-сервера в автозапуск при загрузке системы:

systemctl enable named

Проанализируем отличие в выведенной на экран информации при выполнении команд:

dig www.yandex.ru (Рис. 2.6)

и

dig @127.0.0.1 www.yandex.ru (Рис. 2.7)

```
[root@server.user.net ~]# systemctl start named
[root@server.user.net ~]# systemctl enable named
Created symlink /etc/systemd/system/multi-user.target.wants/named.service → /usr/lib/systemd/system/named.service.
[root@server.user.net ~]#
```

```
root@server:~ [root@server.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: 62703
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0
;;
;; QUESTION SECTION:
;www.yandex.ru.           IN      A
;;
;; ANSWER SECTION:
www.yandex.ru.        3600    IN      A      5.255.255.77
www.yandex.ru.        3600    IN      A      77.88.55.88
www.yandex.ru.        3600    IN      A      77.88.44.55
;;
;; Query time: 4 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Tue Jan 06 14:05:38 UTC 2026
;; MSG SIZE  rcvd: 79
[root@server.user.net ~]#
```

**Рис. 2.6.** Запуск DNS-сервера, включение запуска DNS-сервера в автозапуск при загрузке системы, анализ выведенной на экран информации при выполнении команды dig [www.yandex.ru](http://www.yandex.ru). Анализ выведенной на экран информации при выполнении команды dig @127.0.0.1 www.yandex.ru.

Сделаем DNS-сервер сервером по умолчанию для хоста server и внутренней виртуальной сети. Для этого изменим 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 (рис. 2.8):

```
[user@server 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
Unknown command: 'remove ipv4.dns'
nmcli> remove ipv4.dns
nmcli> set ipv4.ignore-auto-dns yes
nmcli> ipv4.dns 127.0.0.1
Unknown command: 'ipv4.dns 127.0.0.1'
nmcli> save
Connection 'eth0' (52b757e4-3833-4753-b0eb-de0af7f26f57) successfully updated.
nmcli> quit
[user@server user.net ~]$
```

**Рис. 2.9.** Повторяем действия для соединения System eth0.

Перезапустим NetworkManager: systemctl

restart NetworkManager

Проверим наличие изменений в файле /etc/resolv.conf (рис. 2.10):

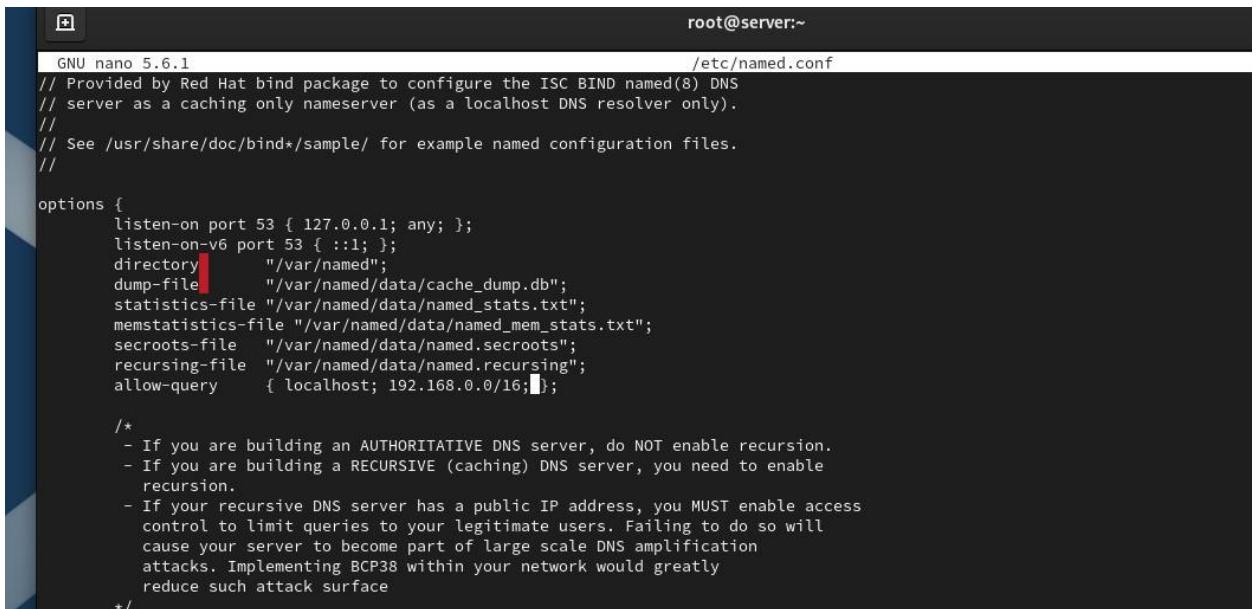
```
[root@server.user.net ~]# systemctl restart NetworkManager
[root@server.user.net ~]#
[root@server.user.net ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search user.net
nameserver 127.0.0.1
[root@server.user.net ~]#
```

**Рис. 2.10.** Перезапуск NetworkManager и проверка наличия изменений в файле /etc/resolv.conf.

Теперь нам требуется настроить направление DNS-запросов от всех узлов внутренней сети, включая запросы от узла server, через узел server (рис. 2.11). Для этого внесём изменения в файл /etc/named.conf, заменив строку

listen-on port 53 { 127.0.0.1; }; на listen-on port 53 { 127.0.0.1; any; }; и строку

allow-query { localhost; }; на allow-query { localhost; 192.168.0.0/16; };



```
root@server:~#
GNU nano 5.6.1                                     /etc/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; };

/*
 - 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
*/
```

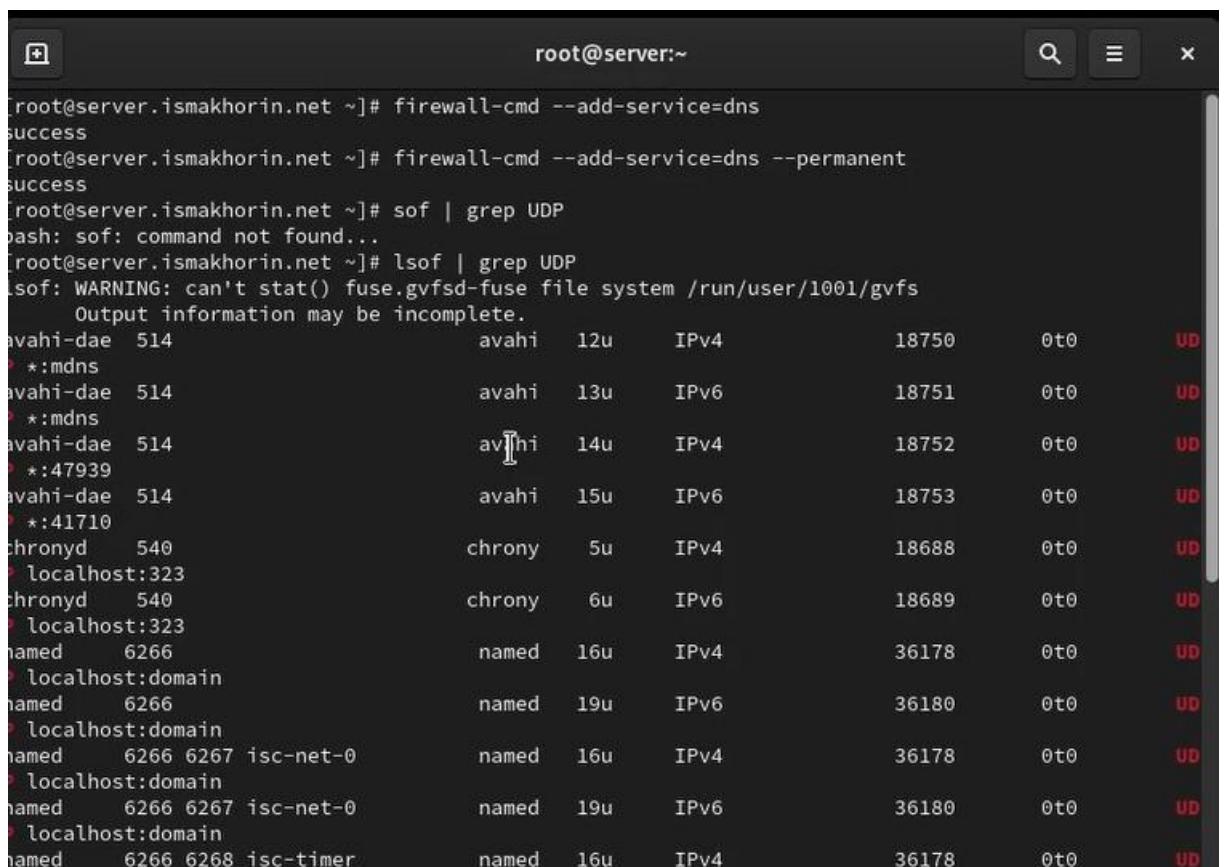
**Рис. 2.11.** Настройка направление DNS-запросов от всех узлов внутренней сети, включая запросы от узла server, через узел server.

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

```
firewall-cmd --add-service=dns  
firewall-cmd  
--add-service=dns --permanent
```

Убедимся, что DNS-запросы идут через узел server, который прослушивает порт 53. Для этого на данном этапе используем команду lsof (рис. 2.12):

```
lsof | grep UDP
```



```
[root@server.ismakhorin.net ~]# firewall-cmd --add-service=dns  
success  
[root@server.ismakhorin.net ~]# firewall-cmd --add-service=dns --permanent  
success  
[root@server.ismakhorin.net ~]# lsof | grep UDP  
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs  
      Output information may be incomplete.  
avahi-dae 514 avahi 12u IPv4 18750 0t0 UD  
avahi-dae 514 avahi 13u IPv6 18751 0t0 UD  
avahi-dae 514 avahi 14u IPv4 18752 0t0 UD  
*:47939 avahi-dae 514 avahi 15u IPv6 18753 0t0 UD  
*:41710 chronyd 540 chrony 5u IPv4 18688 0t0 UD  
localhost:323 chronyd 540 chrony 6u IPv6 18689 0t0 UD  
named 6266 named 16u IPv4 36178 0t0 UD  
localhost:domain named 6266 named 19u IPv6 36180 0t0 UD  
named 6266 6267 isc-net-0 localhost:domain named 16u IPv4 36178 0t0 UD  
named 6266 6267 isc-net-0 localhost:domain named 19u IPv6 36180 0t0 UD  
named 6266 6268 isc-timer named 16u IPv4 36178 0t0 UD
```

**Рис. 2.12.** Внос изменений в настройки межсетевого экрана узла server, разрешив работу с DNS. Проверка, что DNS-запросы идут через узел server, который прослушивает порт 53.

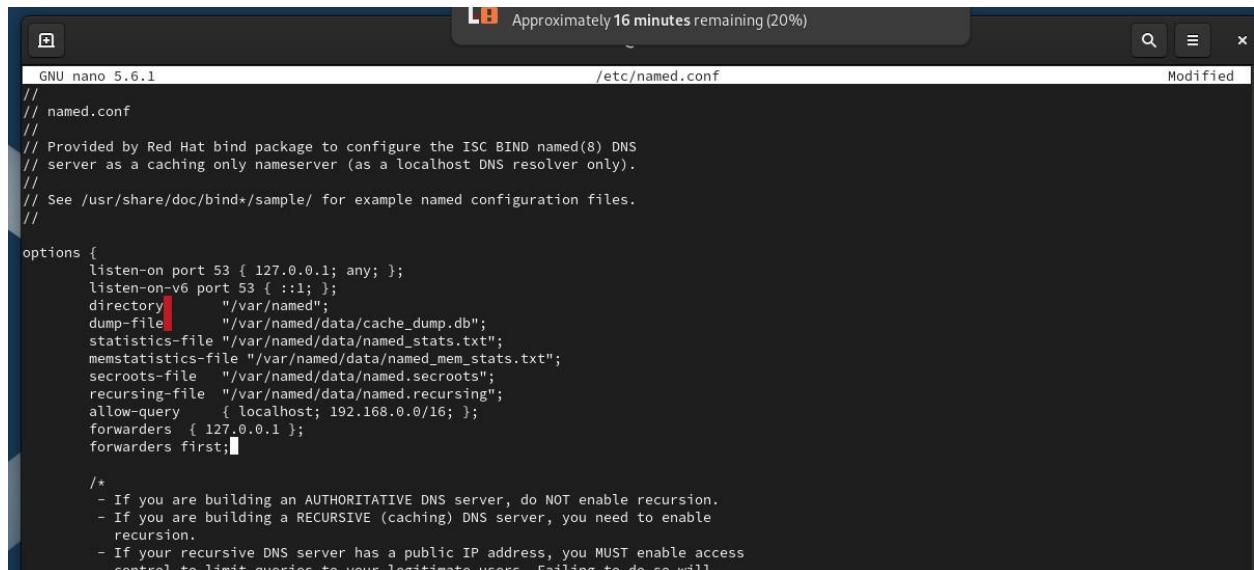
В случае возникновения в сети ситуации, когда DNS-запросы от сервера фильтруются сетевым оборудованием, следует добавить перенаправление DNS-запросов на конкретный вышестоящий DNS-сервер. Для этого в конфигурационный файл named.conf в секцию options добавим:

```
forwarders { список DNS-серверов }; forward  
first;
```

Кроме того, возможно вышестоящий DNS-сервер может не поддерживать технологию DNSSEC, тогда в конфигурационном файле named.conf укажем следующие настройки (рис. 3):

```
dnssec-enable no;
```

```
dnssec-validation no;
```



```
GNU nano 5.6.1          /etc/named.conf      Modified  
//  
// 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.recurse";  
    allow-query   { localhost; 192.168.0.0/16; };  
    forwarders   { 127.0.0.1 };  
    forwarders first;  
  
/*  
 - 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  
     break name resolution for other users.  
*/
```

**Рис. 3.** Добавление перенаправлений DNS-запросов на конкретный вышестоящий DNS-сервер и дополнительных настроек.

Откроем файл /etc/named/user.net на редактирование и вместо зоны пропишем свою прямую зону. Далее, вместо зоны пропишем свою обратную зону. Остальные записи в файле /etc/named/claudely.net удалим (рис. 4.3):

```
root@server:/etc/named
GNU nano 5.6.1
// 
zone "localhost.localdomain" IN {
    type master;
    file "named.localhost";
    allow-update { none; };
};

zone "localhost" IN {
    type master;
    file "named.localhost";
    allow-update { none; };
};

zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" IN {
    type master;
    file "named.loopback";
    allow-update { none; };
};

zone "1.0.0.127.in-addr.arpa" IN {
    type master;
    file "named.loopback";
    allow-update { none; };
};

zone "0.in-addr.arpa" IN {
    type master;
    file "named.empty";
    allow-update { none; };
};

include "/etc/named/claudely.net"
```

Рис. 4.3. Открытие файла /etc/named/user.net на редактирование.

Прописывание своей прямой зоны, обратной зоны и удаление остальных записей в файле.

В каталоге /var/named создадим подкаталоги master/fz и master/rz, в которых будут располагаться файлы прямой и обратной зоны соответственно (рис. 4.4):

```
cd /var/named mkdir -p
```

```
/var/named/master/fz
```

```
mkdir -p /var/named/master/rz
```

```
[user@server.user.net ~]$ cd /var/named
bash: cd: /var/named: Permission denied
[user@server.user.net ~]$
[user@server.user.net ~]$
[user@server.user.net ~]$
[sudo] password for user:
[root@server.user.net ~]# cd /var/named
[root@server.user.net named]# mkdir -p /var/named/master/fz
[root@server.user.net named]# mkdir -p /var/named/master/rz
[root@server.user.net named]# ls
data dynamic master named.ca named.empty named.localhost named.loopback slaves
[root@server.user.net named]#
```

Рис. 4.4. В каталоге /var/named создание подкаталогов master/fz и master/rz.

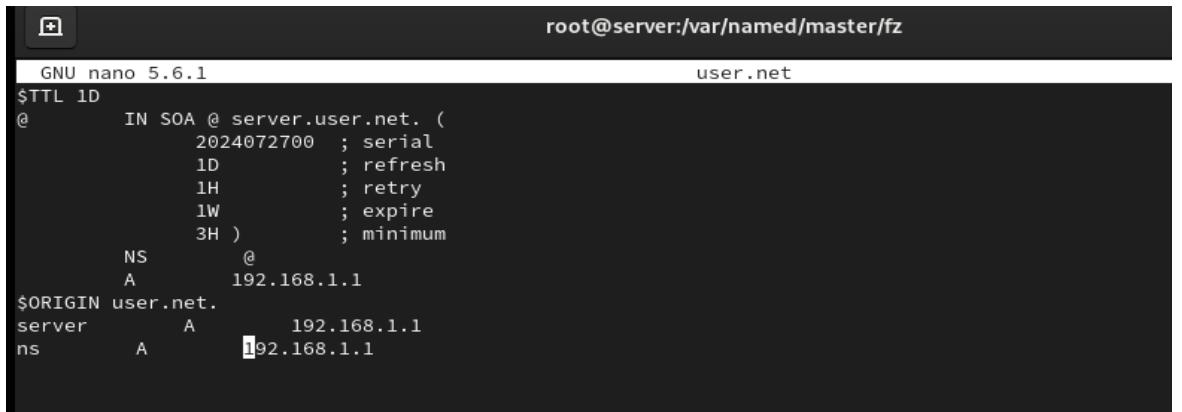
Скопируем шаблон прямой DNS-зоны named.localhost из каталога /var/named в каталог /var/named/master/fz и переименуем его в claudely.net (рис. 4.5):

```
cp /var/named/named.localhost /var/named/master/fz/  
cd /var/named/master/fz/ mv named.localhost  
user.net
```

```
[root@server.user.net named]# cp /var/named/named.localhost /var/named/master/fz/  
[root@server.user.net named]# cd /var/named/master/fz/  
[root@server.user.net fz]# mv named.localhost user.net  
[root@server.user.net fz]# ls  
user.net  
[root@server.user.net fz]#
```

**Рис. 4.5.** Копирование шаблона прямой DNS-зоны named.localhost из каталога /var/named в каталог /var/named/master/fz и изменение его названия.

Изменим файл /var/named/master/fz/user.net, указав необходимые DNS записи для прямой зоны. В этом файле DNS-имя сервера @ rname.invalid. заменим на @ server.user.net. Формат серийного номера ГГГГММДДВВ (ГГГГ — год, ММ — месяц, ДД — день, ВВ — номер ревизии) [1]; адрес в A- заменим с 127.0.0.1 на 192.168.1.1; в директиве \$ORIGIN зададим текущее имя домена user.net, а затем укажем имена и адреса серверов в этом домене в виде A- записей DNS (на данном этапе пропишем сервер с именем ns и адресом 192.168.1.1) (рис. 4.6):



```
GNU nano 5.6.1                                     user.net
$TTL 1D
@       IN SOA @ server.user.net. (
                2024072700 ; serial
                1D          ; refresh
                1H          ; retry
                1W          ; expire
                3H )        ; minimum
                NS          @
                A           192.168.1.1
$ORIGIN user.net.
server      A           192.168.1.1
ns          A           192.168.1.1
```

**Рис. 4.6.** Изменение файла /var/named/master/fz/user.net, указав необходимые DNS записи для прямой зоны.

Скопируем шаблон обратной DNS-зоны named.loopback из каталога /var/named в каталог /var/named/master/rz и переименуем его в 192.168.1 (рис. 4.7):

```
cp /var/named/named.loopback /var/named/master/rz/ cd
/var/named/master/rz/
mv named.loopback 192.168.1
```

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

**Рис. 4.7.** Копирование шаблона обратной DNS-зоны named.loopback из каталога /var/named в каталог /var/named/master/rz и изменение его названия.

Изменим файл /var/named/master/rz/192.168.1, указав необходимые DNS записи для обратной зоны. В этом файле DNS-имя сервера @ rname.invalid заменим на @ server.user.net. формат серийного номера ГГГГММДДВВ (ГГГГ — год, ММ — месяц, ДД — день, ВВ — номер ревизии); адрес в Азаписи заменим с 127.0.0.1 на 192.168.1.1; в директиве \$ORIGIN зададим название обратной зоны в виде 1.168.192.in-addr.arpa., затем зададим PTRзаписи (на

данном этапе зададим PTR запись, ставящая в соответствие адресу 192.168.1.1 DNS-адрес ns.user.net)  
(рис. 4.8):



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

**Рис. 4.8.** Изменение файла /var/named/master/rz/192.168.1, указав необходимые DNS записи для обратной зоны.

Далее исправим права доступа к файлам в каталогах /etc/named и /var/named, чтобы демон named мог с ними работать:

```
chown -R named:named /etc/named chown  
-R named:named /var/named
```

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:

```
restorecon -vR /etc restorecon  
-vR /var/named
```

Для проверки состояния переключателей SELinux, относящихся к named, введём:

```
getsebool -a | grep named
```

Теперь дадим named разрешение на запись в файлы DNS-зоны:

```
setsebool named_write_master_zones 1 setsebool
```

```
-P named_write_master_zones 1
```

В дополнительном терминале запустим в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы (рис. 4.10):

journalctl -x -f и в первом терминале перезапустим DNS-

сервер (рис. 4.9): systemctl restart named

```
[root@server.user.net rz]# chown -R named:named /etc/named
[root@server.user.net rz]# chown -R named:named /var/named
[root@server.user.net rz]# 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.user.net rz]# restorecon -vR /var/named
[root@server.user.net rz]# getsebool -a | grep named
named_tcp_bind_http_port --> off
named_write_master_zones --> on
[root@server.user.net rz]# setsebool named_write_master_zones 1
[root@server.user.net rz]# setsebool -P named_write_master_zones 1
```

**Рис. 4.9.** Исправление прав доступа к файлам в каталогах /etc/named и /var/named, корректное восстановление их меток в SELinux, проверка состояния переключателей SELinux и перезапуск DNS-сервера.

```
[root@server.user.net rz]# journalctl -x -f
Feb 07 18:49:27 server.user.net kernel: SELinux: Converting 624 SID table entries...
Feb 07 18:49:27 server.user.net kernel: SELinux: policy capability network_peer_controls=1
Feb 07 18:49:27 server.user.net kernel: SELinux: policy capability open_perms=1
Feb 07 18:49:27 server.user.net kernel: SELinux: policy capability extended_socket_class=1
Feb 07 18:49:27 server.user.net kernel: SELinux: policy capability always_check_network=0
Feb 07 18:49:27 server.user.net kernel: SELinux: policy capability cgroup_seclabel=1
Feb 07 18:49:27 server.user.net kernel: SELinux: policy capability nnp_nosuid_transition=1
Feb 07 18:49:27 server.user.net kernel: SELinux: policy capability genfs_seclabel_symlinks=1
Feb 07 18:49:27 server.user.net setsebool[8795]: The named_write_master_zones policy boolean was changed to 1 by root
Feb 07 18:49:33 server.user.net dbus-broker-launch[5973]: avc: op=load_policy lsm=selinux seqno=4 res=1
Feb 07 18:50:39 server.user.net dbus-broker-launch[665]: avc: op=load_policy lsm=selinux seqno=4 res=1

Feb 07 18:51:52 server.user.net systemd[5943]: selinux: avc: op=load_policy lsm=selinux seqno=4 res=1
Feb 07 18:51:52 server.user.net systemd[5943]: Started VTE child process 8803 launched by gnome-terminal-server process 6885.
Subject: A start job for unit UNIT has finished successfully
Defined-By: systemd
Support: https://wiki.rockylinux.org/rocky/support

A start job for unit UNIT has finished successfully.

The job identifier is 616.
```

**Рис. 4.10.** Проверка корректности работы системы.

При помощи утилиты dig получим описание DNS-зоны с сервера ns.claudely.net (рис. 5.1):

```
dig ns.user.net
```

```
[root@server.user.net fz]# dig ns.user.net

; <>> DiG 9.16.23-RH <>> ns.user.net
;; global options: +cmd
;; connection timed out; no servers could be reached

[root@server.user.net fz]# █
```

**Рис. 5.1.** Получение описания DNS-зоны с сервера ns.user.net.

При помощи утилиты host проанализируем корректность работы DNSсервера (рис. 5.2):

```
host -l user.net host -a
```

```
user.net host -t A
```

```
user.net host -t PTR
```

```
192.168.1.1
```

```

[root@server.claudely.net rz]#
[root@server.claudely.net rz]# host -l claudely.net
claudely.net name server claudely.net.
claudely.net has address 192.168.1.1
ns.claudely.net has address 192.168.1.1
server.claudely.net has address 192.168.1.1
[root@server.claudely.net rz]# host -a claudely.net
Trying "claudely.net"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65326
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;claudely.net.           IN      ANY

;; ANSWER SECTION:
claudely.net.        86400   IN      SOA     claudely.net. server.claudely.net. 2024072700
                      86400  3600  604800 10800
claudely.net.        86400   IN      NS      claudely.net.
claudely.net.        86400   IN      A       192.168.1.1

;; ADDITIONAL SECTION:
claudely.net.        86400   IN      A       192.168.1.1

Received 119 bytes from 127.0.0.1#53 in 20 ms
[root@server.claudely.net rz]#
[root@server.claudely.net rz]# host -t A claudely.net
claudely.net has address 192.168.1.1
[root@server.claudely.net rz]# host -t PTR 192.168.1.1
1.1.168.192.in-addr.arpa domain name pointer server.claudely.net.
1.1.168.192.in-addr.arpa domain name pointer ns.claudely.net.
[root@server.claudely.net rz]# █

```

**Рис. 5.2.** Анализ корректности работы DNS-сервера.

На виртуальной машине server перейдём в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог dns, в который поместим в соответствующие каталоги конфигурационные файлы DNS (рис. 6.1):

```

cd /vagrant mkdir -p /vagrant/provision/server/dns/etc/named mkdir -p
/vagrant/provision/server/dns/var/named/master/ cp -R /etc/named.conf
/vagrant/provision/server/dns/etc/ cp -R /etc/named/*
/vagrant/provision/server/dns/etc/named/ cp -R /var/named/master/*
/vagrant/provision/server/dns/var/named/master/

```

```
[root@server.claudely.net vagrant]# mkdir -p /vagrant/provision/server/dns/etc/named
[root@server.claudely.net vagrant]# mkdir -p /vagrant/provision/server/dns/var/named/master/
[root@server.claudely.net vagrant]# cp -R /etc/named.conf /vagrant/provision/server/dns/etc/
cp: overwrite '/vagrant/provision/server/dns/etc/named.conf'? yes
[root@server.claudely.net vagrant]# cp -R /etc/named/* /vagrant/provision/server/dns/etc/named/
cp: overwrite '/vagrant/provision/server/dns/etc/named/claudely.net'? yes
[root@server.claudely.net vagrant]# cp -R /etc/master/* /vagrant/provision/server/dns/var/named/master/
cp: cannot stat '/etc/master/*': No such file or directory
[root@server.claudely.net vagrant]# cp -R /var/named/master/* /vagrant/provision/server/dns/var/named/master/
[root@server.claudely.net vagrant]#
[root@server.claudely.net vagrant]#
[root@server.claudely.net vagrant]#
```

**Рис. 6.1.** Переход в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создание в нём каталога dns, в который помещаем в соответствующие каталоги конфигурационные файлы DNS.

В каталоге /vagrant/provision/server создадим исполняемый файл dns.sh (рис. 6.2):

```
touch dns.sh
```

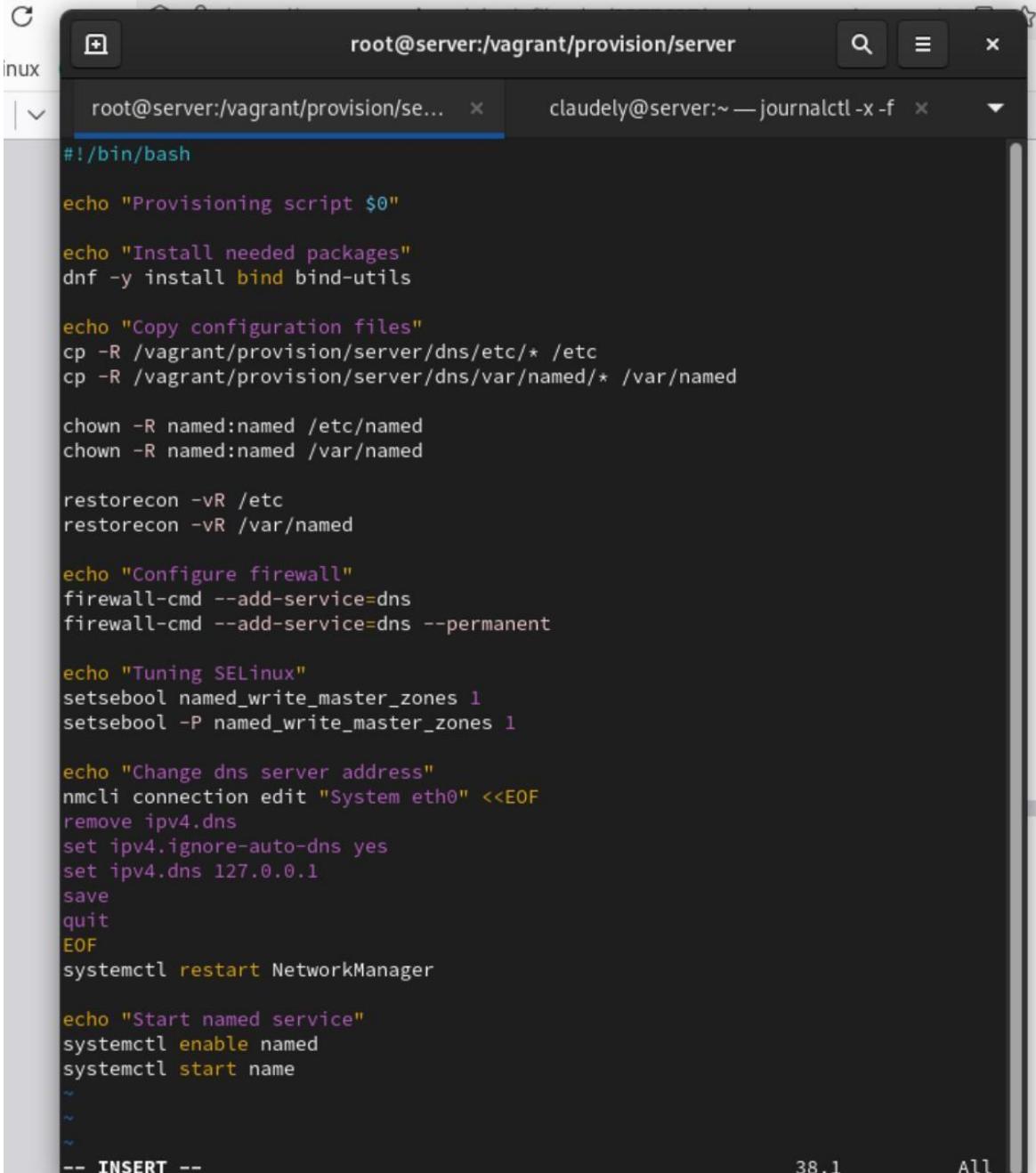
```
chmod +x dns.sh
```

```
[root@server.user.net vagrant]#
[root@server.user.net vagrant]# cd /vagrant//provision/server
[root@server.user.net server]# touch dns.sh
[root@server.user.net server]# chmod +x dns.sh
[root@server.user.net server]#
```

**Рис. 6.2.** Создание в каталоге /vagrant/provision/server исполняемого файла dns.sh.

Откроем его на редактирование и пропишем в нём следующий скрипт (приведён в лабораторной работе). Этот скрипт, по сути, повторяет произведённые нами действия по установке и настройке DNS-сервера (рис. 6.3):

1. подставляет в нужные каталоги подготовленные вами конфигурационные файлы;
2. меняет соответствующим образом права доступа, метки безопасности SELinux и правила межсетевого экрана;
3. настраивает сетевое соединение так, чтобы сервер выступал DNSсервером по умолчанию для узлов внутренней виртуальной сети;
4. запускает DNS-сервер;



The screenshot shows a terminal window titled "root@server:/vagrant/provision/server". The window has two tabs: "root@server:/vagrant/provision/se..." and "claudely@server:~ — journalctl -x -f". The main pane displays a shell script with the following content:

```

#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
dnf -y install bind bind-utils

echo "Copy configuration files"
cp -R /vagrant/provision/server/dns/etc/* /etc
cp -R /vagrant/provision/server/dns/var/named/* /var/named

chown -R named:named /etc/named
chown -R named:named /var/named

restorecon -vR /etc
restorecon -vR /var/named

echo "Configure firewall"
firewall-cmd --add-service=dns
firewall-cmd --add-service=dns --permanent

echo "Tuning SELinux"
setsebool named_write_master_zones 1
setsebool -P named_write_master_zones 1

echo "Change dns server address"
nmcli connection edit "System eth0" <<EOF
remove ipv4.dns
set ipv4.ignore-auto-dns yes
set ipv4.dns 127.0.0.1
save
quit
EOF
systemctl restart NetworkManager

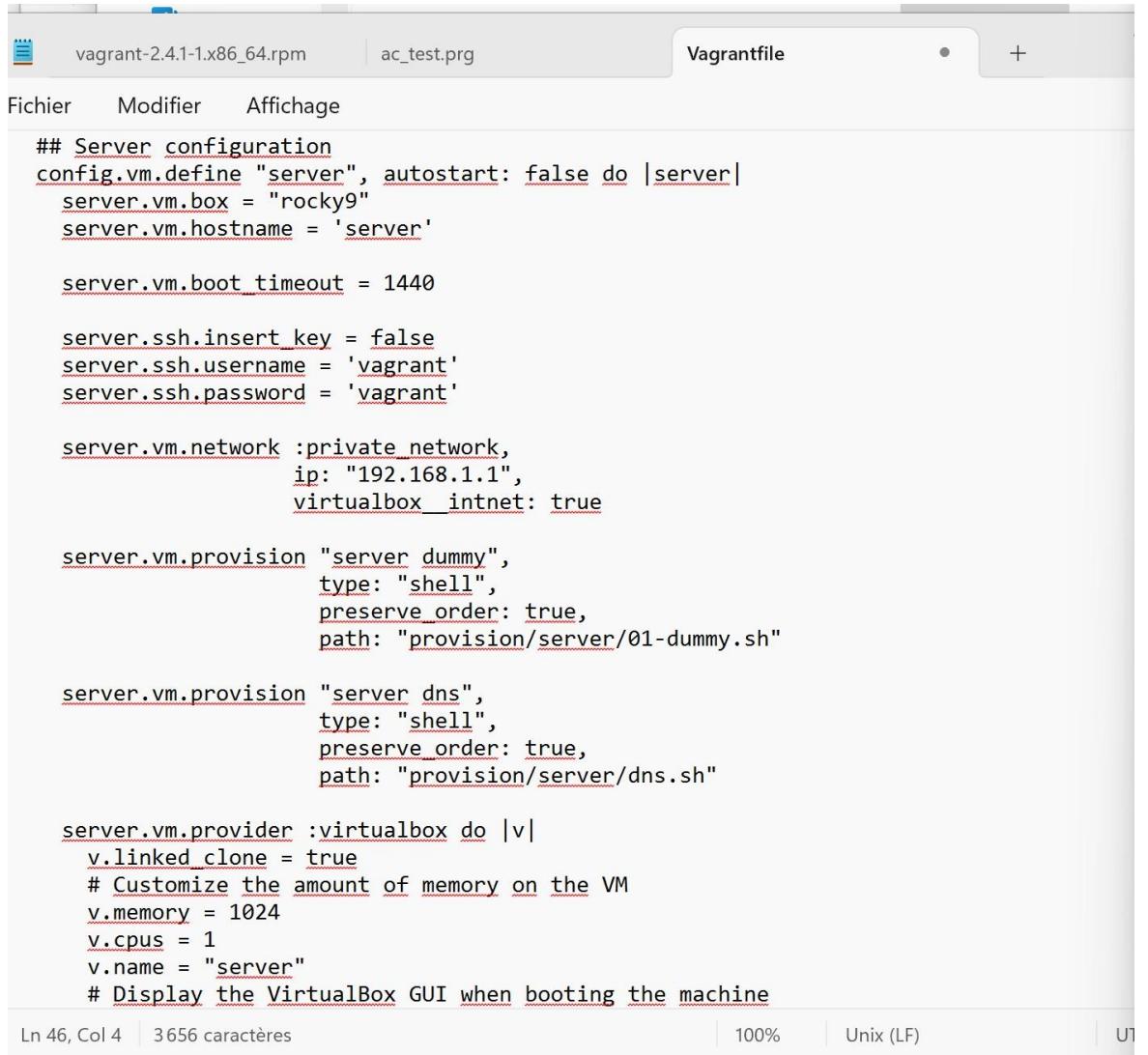
echo "Start named service"
systemctl enable named
systemctl start named
~^
~^
~^

-- INSERT --

```

**Рис. 6.3.** Открытие файла на редактирование и прописывание в нём скрипта.

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile добавим определённые параметры в разделе конфигурации для сервера (рис. 6.4):



The screenshot shows a code editor window with the file 'Vagrantfile' open. The code defines a server configuration with various parameters like IP, SSH keys, and provisioning scripts. The code is as follows:

```
## Server configuration
config.vm.define "server", autostart: false do |server|
    server.vm.box = "rocky9"
    server.vm.hostname = 'server'

    server.vm.boot_timeout = 1440

    server.ssh.insert_key = false
    server.ssh.username = 'vagrant'
    server.ssh.password = 'vagrant'

    server.vm.network :private_network,
        ip: "192.168.1.1",
        virtualbox_intnet: true

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

    server.vm.provision "server dns",
        type: "shell",
        preserve_order: true,
        path: "provision/server/dns.sh"

    server.vm.provider :virtualbox do |v|
        v.linked_clone = true
        # Customize the amount of memory on the VM
        v.memory = 1024
        v.cpus = 1
        v.name = "server"
        # Display the VirtualBox GUI when booting the machine
    end

```

Ln 46, Col 4 | 3656 caractères | 100% | Unix (LF) | U1

**Рис. 6.4.** Добавление параметров в конфигурационном файле Vagrantfile в разделе конфигурации для сервера.

**Вывод:**

В ходе выполнения лабораторной работы были приобретены практические навыки по установке и конфигурированию DNS-сервера, а также усвоили принципы работы системы доменных имён.

### **Ответы на контрольные вопросы:**

1. Что такое DNS? - Это система, предназначенная для преобразования человекочитаемых доменных имен в IP-адреса, используемые компьютерами для идентификации друг друга в сети.
2. Каково назначение кэширующего DNS-сервера? - Его задача - хранить результаты предыдущих DNS-запросов в памяти. Когда клиент делает запрос, кэширующий DNS проверяет свой кэш, и если он содержит соответствующую информацию, сервер возвращает ее без необходимости обращаться к другим DNS-серверам. Это ускоряет процесс запроса.
3. Чем отличается прямая DNS-зона от обратной? - Прямая зона преобразует доменные имена в IP-адреса, обратная зона выполняет обратное: преобразует IP-адреса в доменные имена.
4. В каких каталогах и файлах располагаются настройки DNS-сервера? Кратко охарактеризуйте, за что они отвечают. - В Linux-системах обычно используется файл /etc/named.conf для общих настроек. Зоны хранятся в файлах в каталоге /var/named/, например, /var/named/example.com.zone.
5. Что указывается в файле resolv.conf? - Содержит информацию о DNS-серверах, используемых системой, а также о параметрах конфигурации.
6. Какие типы записи описания ресурсов есть в DNS и для чего они используются? - A (IPv4-адрес), AAAA (IPv6-адрес), CNAME

(каноническое имя), MX (почтовый сервер), NS (имя сервера), PTR (обратная запись), SOA (начальная запись зоны), TXT (текстовая информация).

7. Для чего используется домен in-addr.arpa? - Используется для обратного маппинга IP-адресов в доменные имена.
8. Для чего нужен демон named? - Это DNS-сервер, реализация BIND (Berkeley Internet Name Domain).
9. В чём заключаются основные функции slave-сервера и master-сервера?
  - Master-сервер хранит оригинальные записи зоны, slave-серверы получают копии данных от master-сервера.
- 10.Какие параметры отвечают за время обновления зоны? - refresh, retry, expire, и minimum.
- 11.Как обеспечить защиту зоны от скачивания и просмотра? - Это может включать в себя использование TSIG (Transaction SIGnatures) для аутентификации между серверами.
- 12.Какая запись RR применяется при создании почтовых серверов? - MX (Mail Exchange).
- 13.Как протестировать работу сервера доменных имён? - Используйте команды nslookup, dig, или host.
- 14.Как запустить, перезапустить или остановить какую-либо службу в системе? - systemctl start|stop|restart <service>.
- 15.Как посмотреть отладочную информацию при запуске какого-либо сервиса или службы? - Используйте опции, такие как -d или -v при запуске службы.
- 16.Где храниться отладочная информация по работе системы и служб? Как её посмотреть? - В системных журналах, доступных через journalctl.

17. Как посмотреть, какие файлы использует в своей работе тот или иной процесс? Приведите несколько примеров. - lsof -p <pid> или fuser -v <file>.
18. Приведите несколько примеров по изменению сетевого соединения при помощи командного интерфейса nmcli. - Примеры включают nmcli connection up|down <connection\_name>.
19. Что такое SELinux? - Это мандатный контроль доступа для ядра Linux.
20. Что такое контекст (метка) SELinux? - Метка, определяющая, какие ресурсы могут быть доступны процессу или объекту.
21. Как восстановить контекст SELinux после внесения изменений в конфигурационные файлы? - restorecon -Rv <directory>.
22. Как создать разрешающие правила политики SELinux из файлов журналов, содержащих сообщения о запрете операций? - Используйте audit2allow.
23. Что такое булевый переключатель в SELinux? - Это параметр, который включает или отключает определенные аспекты защиты SELinux.
24. Как посмотреть список переключателей SELinux и их состояние? - getsebool -a.
25. Как изменить значение переключателя SELinux? - setsebool -P <boolean\_name> <on|off>.