

Отчет по лабораторной работе №3

Дисциплина: Администрирование сетевых подсистем

Иванов Сергей Владимирович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Установка DHCP-сервера	7
3.2	Конфигурирование DHCP-сервера	8
3.3	Анализ работы DHCP-сервера	13
3.4	Настройка обновления DNS-зоны	17
3.5	Анализ работы DHCP-сервера после настройки обновления DNS-зоны	21
3.6	Внесение изменений в настройки внутреннего окружения виртуальной машины	22
4	Ответы на контрольные вопросы	24
5	Выводы	27

Список иллюстраций

3.1	Запуск server	7
3.2	Установка dhcp	7
3.3	Сохранение конф. файла	8
3.4	Файл /etc/kea/kea-dhcp4.conf	10
3.5	Файл /etc/kea/kea-dhcp4.conf	10
3.6	Проверка правильности файла	10
3.7	Перезагрузка конфигурации	11
3.8	Редактирование файлов	11
3.9	Редактирование файлов	11
3.10	Проверка обращения к серверу	12
3.11	Изменения в настройках firewall	12
3.12	Восстановление меток в SELinux	12
3.13	Запуск мониторинга	13
3.14	Запуск DHCP-сервера	13
3.15	Редактирование 01-routing.sh	14
3.16	Запуск client	14
3.17	Анализ подключения	15
3.18	Анализ ifconfig	16
3.19	Список выданных адресов	17
3.20	Создание ключа	17
3.21	Права доступа	17
3.22	Подключение ключа	17
3.23	Редактирование файла /etc/named/user.net	18
3.24	Проверка и перезапуск сервера	18
3.25	Формирование ключа	18
3.26	Смена владельца и прав	19
3.27	Настройка	19
3.28	Смена владельца и проверка ошибок	19
3.29	Запуск и проверка статуса	19
3.30	Добавление динамического обновления DNS-записей	20
3.31	Проверка ошибок	20
3.32	Запуск и проверка статуса	20
3.33	Переполучение адреса	21
3.34	Анализ запроса	21
3.35	Замена конф. файлов	22
3.36	Запуск и проверка статуса	23

3.37 Редактирование Vagrantfile	23
---	----

1 Цель работы

Целью данной работы является приобретение практических навыков по установке и конфигурированию DHCP сервера.

2 Задание

1. Установите на виртуальной машине server DHCP-сервер (см. раздел 3.4.1).
2. Настройте виртуальную машину server в качестве DHCP-сервера для виртуальной внутренней сети (см. раздел 3.4.2).
3. Проверьте корректность работы DHCP-сервера в виртуальной внутренней сети путём запуска виртуальной машины client и применения соответствующих утилит диагностики (см. раздел 3.4.3).
4. Настройте обновление DNS-зоны при появлении в виртуальной внутренней сети новых узлов (см. раздел 3.4.4).
5. Проверьте корректность работы DHCP-сервера и обновления DNS-зоны в виртуальной внутренней сети путём запуска виртуальной машины client и применения соответствующих утилит диагностики (см. раздел 3.4.5).
6. Напишите скрипт для Vagrant, фиксирующий действия по установке и настройке DHCP-сервера во внутреннем окружении виртуальной машины server. Соответствующим образом внести изменения в Vagrantfile (см. раздел 3.4.6).

3 Выполнение лабораторной работы

3.1 Установка DHCP-сервера

Загрузим операционную систему и перейдем в рабочий каталог с проектом:
cd /var/tmp/user_name/vagrant . Запустим виртуальную машину server: vagrant
up server . (рис. 1).

```
C:\Users\lserg>cd C:\work_asp\svivanov\vagrant
C:\work_asp\svivanov\vagrant>vagrant up server
```

Рис. 3.1: Запуск server

На виртуальной машине server войдем под нашим пользователем и откроем терминал. Перейдем в режим суперпользователя: sudo -i . Установим dhcp: dnf
-y install kea (рис. 2).

```
[svivanov@server svivanov.net ~]$ sudo -i
[svivanov@server svivanov.net ~]$ dnf -y install kea
[root@server svivanov.net ~]# dnf -y install kea
Extra Packages for Enterprise Linux 10 - x86_64
Errors during downloading metadata for repository 'epel':
 - Curl error (6): Could not resolve hostname for https://mirrors.fedoraproject.org/metalink?repo=epel-z-10&arch=x86_64 [Could not resolve host: mirrors.fedoraproject.org]
Error: Failed to download metadata for repo 'epel': Cannot prepare internal mirrorlist: Curl error (6): Could not resolve host: mirrors.fedoraproject.org
[root@server svivanov.net ~]# dnf -y install kea
Extra Packages for Enterprise Linux 10 - x86_64
Extra Packages for Enterprise Linux 10 - x86_64
Rocky Linux 10 - BaseOS
Rocky Linux 10 - AppStream
Rocky Linux 10 - Extras
Dependencies resolved.
=====
Package                                Architecture    Version
=====
Installing:
kea                                    x86_64          2.6.3-1.el10_0
Installing dependencies:
kea-libs                              x86_64          2.6.3-1.el10_0
libpq                                  x86_64          16.8-2.el10_0
log4cplus                              x86_64          2.1.1-8.el10
mariadb-connector-c                    x86_64          3.4.4-1.el10
mariadb-connector-c-config             noarch          3.4.4-1.el10
=====
```

Рис. 3.2: Установка dhcp

3.2 Конфигурирование DHCP-сервера

Сохраним на всякий случай конфигурационный файл: `cp /etc/kea/kea-dhcp4.conf /etc/kea/kea-dhcp4.conf_$(date -I)` (рис. 3)

```
[root@server.svivanov.net ~]# cp /etc/kea/kea-dhcp4.conf /etc/kea/kea-dhcp4.conf_$(date -I)
[root@server.svivanov.net ~]#
```

Рис. 3.3: Сохранение конф. файла

Откроем файл `/etc/kea/kea-dhcp4.conf` на редактирование. В этом файле: – заменим шаблон для `domain-name`

```
{
  "code": 15,
  "data": "example.org"
},
```

```
{
  "name": "domain-search",
  "data": "srv.world"
},
```

на описание

```
{
  "code": 15,
  "data": "user.net"
},
```

```
{
  "name": "domain-search",
  "data": "user.net"
},
```

– заменим блок

```
{
  "name": "domain-name-servers",
```



```
“data”: “192.0.2.1, 192.0.2.2”
```

```
},
```

на блок

```
{
```

```
“name”: “domain-name-servers”,
```

```
“data”: “192.168.1.1”
```

```
},
```

– на базе одного из приведённых в файле примеров конфигурирования подсети зададим собственную конфигурацию dhcp-сети, задав адрес подсети, диапазон адресов для распределения клиентам, адрес маршрутизатора и broadcast-адрес:

```
“subnet4”: [
```

```
{
```

```
“id”: 1,
```

```
// specify subnet that DHCP is used
```

```
“subnet”: “192.168.1.0/24”,
```

```
// specify the range of IP addresses to be leased
```

```
“pools”: [ { “pool”: “192.168.1.30 - 192.168.1.199” } ],
```

```
“option-data”: [
```

```
{
```

```
// specify your gateway
```

```
“name”: “routers”,
```

```
“data”: “192.168.1.1”
```

```
}
```

```
]
```

```
}
```

```
],
```

Остальные примеры задания конфигураций подсетей удалим. (рис. 4)

```
// structures.
"subnet4": [
  {
    "id": 1,
    "subnet": "192.168.1.0/24",
    "pools": [ { "pool": "192.168.1.30 - 192.168.1.199" } ],

    "option-data": [
      {
        "name": "routers",
        "data": "192.168.1.1"
      }
    ]
  },
]
```

Рис. 3.4: Файл /etc/kea/kea-dhcp4.conf

Настроим привязку dhcpd к интерфейсу eth1 виртуальной машины server:

“interfaces-config”: {

"interfaces": ["eth1"]

}, (рис. 5)

```
"Dhcp4": {
  // Add names of your network interfaces to listen on.
  "interfaces-config": {
    "interfaces": [ "eth1" ]
  }
}
```

Рис. 3.5: Файл /etc/kea/kea-dhcp4.conf

Проверим правильность конфигурационного файла: kea-dhcp4 -t /etc/kea/kea-dhcp4.conf (рис. 6)

```
[root@server.svivanov.net kea]# kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
2025-09-15 09:35:04.859 INFO [kea-dhcp4.hosts/18931.140564534962304] HOSTS_BACKENDS_REGISTERED the following host backend types are available: mysql postgresql
2025-09-15 09:35:04.859 WARN [kea-dhcp4.dhcp4/18931.140564534962304] DHCP4_RESERVED_QUEUE_CONTROL disabled dhcp queue control when multi-threading is enabled.
2025-09-15 09:35:04.859 WARN [kea-dhcp4.dhcp4/18931.140564534962304] DHCP4_RESERVATIONS_LOOKUP_FIRST_ENABLED Multi-threading is enabled and host reservations lookup is always performed first.
2025-09-15 09:35:04.859 INFO [kea-dhcp4.dhcp4/18931.140564534962304] DHCP4_CFGMGR_NEW_SUBNET4 a new subnet has been added to configuration: 192.168.1.0/24 with params: t1=900, t2=1800, valid-lifetime=3600
2025-09-15 09:35:04.860 INFO [kea-dhcp4.dhcp4/18931.140564534962304] DHCP4_CFGMGR_SOCKET_TYPE_SELECT using socket type raw
2025-09-15 09:35:04.860 INFO [kea-dhcp4.dhcp4/18931.140564534962304] DHCP4_CFGMGR_ADD_IFACE listening on interface eth1
2025-09-15 09:35:04.860 INFO [kea-dhcp4.dhcp4/18931.140564534962304] DHCP4_CFGMGR_SOCKET_TYPE_DEFAULT "dhcp -socket-type" not specified, using default socket type raw
[root@server.svivanov.net kea]#
```

Рис. 3.6: Проверка правильности файла

Перезагрузим конфигурацию dhcpd и разрешим загрузку DHCP-сервера при запуске виртуальной машины server:

systemctl --system daemon-reload

systemctl enable kea-dhcp4.service (рис. 7)

```
[root@server.svivanov.net kea]# systemctl --system daemon-reload
[root@server.svivanov.net kea]# systemctl enable kea-dhcp4.service
Created symlink '/etc/systemd/system/multi-user.target.wants/kea-dhcp4.service' ->
'/usr/lib/systemd/system/kea-dhcp4.service'.
[root@server.svivanov.net kea]#
```

Рис. 3.7: Перезагрузка конфигурации

Добавим запись для DHCP-сервера в конце файла прямой DNS-зоны /var/named/master/fz/user.net:

dhcp A 192.168.1.1

и в конце файла обратной зоны /var/named/master/rz/192.168.1:

1 PTR dhcp.user.net. (рис. 8, 9)

```
$TTL 1D
@       IN SOA  server.svivanov.net. (
                                2025090900    ; serial
                                1D             ; refresh
                                1H             ; retry
                                1W             ; expire
                                3H )           ; minimum

       NS      server.svivanov.net.
       A       192.168.1.1
$ORIGIN svivanov.net.
server  A       192.168.1.1
ns      A       192.168.1.1
dhcp    A       192.168.1.1
```

Рис. 3.8: Редактирование файлов

```
$TTL 1D
@       IN SOA  server.svivanov.net. (
                                2025090900    ; serial
                                1D             ; refresh
                                1H             ; retry
                                1W             ; expire
                                3H )           ; minimum

       NS      server.svivanov.net.
       A       192.168.1.1
       PTR     server.svivanov.net.
$ORIGIN 1.168.192.in-addr.arpa.
1      PTR     server.svivanov.net.
1      PTR     ns.svivanov.net.
1      PTR     dhcp.svivanov.net.
```

Рис. 3.9: Редактирование файлов

Перезапустим named: `systemctl restart named`. Проверим, что можно обратиться к DHCP-серверу по имени: `ping dhcp.user.net`. (рис. 10)

```
[root@server.svivanov.net rz]# ping dhcp.svivanov.net
PING dhcp.svivanov.net (192.168.1.1) 56(84) bytes of data.
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=1 ttl=64 time=0.029 ms
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=3 ttl=64 time=0.060 ms
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=4 ttl=64 time=0.038 ms
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=5 ttl=64 time=0.058 ms
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=6 ttl=64 time=0.080 ms
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=7 ttl=64 time=0.059 ms
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=8 ttl=64 time=0.059 ms
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=9 ttl=64 time=0.056 ms
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=10 ttl=64 time=0.074 ms
64 bytes from server.svivanov.net (192.168.1.1): icmp_seq=11 ttl=64 time=0.063 ms
```

Рис. 3.10: Проверка обращения к серверу

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

```
firewall-cmd --list-services
```

```
firewall-cmd --get-services
```

```
firewall-cmd --add-service=dhcp
```

```
firewall-cmd --add-service=dhcp --permanent (рис. 11)
```

```
[root@server.svivanov.net rz]# firewall-cmd --add-service=dhcp
success
[root@server.svivanov.net rz]# firewall-cmd --add-service=dhcp --permanent
success
[root@server.svivanov.net rz]# █
```

Рис. 3.11: Изменения в настройках firewall

Восстановим контекст безопасности в SELinux:

```
restorecon -vR /etc
```

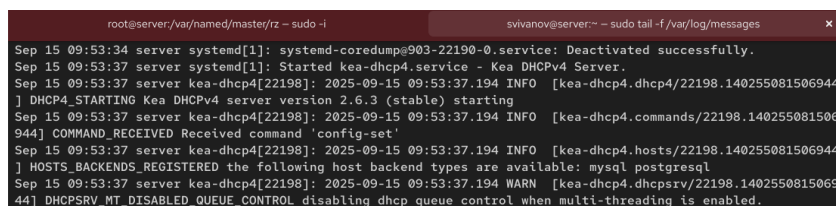
```
restorecon -vR /var/named
```

```
restorecon -vR /var/lib/kea/. (рис. 12)
```

```
[root@server.svivanov.net rz]# restorecon -vR /etc
Relabeled /etc/NetworkManager/system-connections/eth1.nmconnection
o unconfined_u:object_r:NetworkManager_etc_rw_t:s0
[root@server.svivanov.net rz]# restorecon -vR /var/named
[root@server.svivanov.net rz]# restorecon -vR /var/lib/kea/
[root@server.svivanov.net rz]# █
```

Рис. 3.12: Восстановление меток в SELinux

В дополнительном терминале запустим мониторинг происходящих в системе процессов в реальном времени: `tail -f /var/log/messages` (рис. 13)



```
root@server:/var/named/master/rz - sudo -i
svivanov@server:~ - sudo tail -f /var/log/messages
Sep 15 09:53:34 server systemd[1]: systemd-coredump@903-22190-0.service: Deactivated successfully.
Sep 15 09:53:37 server systemd[1]: Started kea-dhcp4.service - Kea DHCPv4 Server.
Sep 15 09:53:37 server kea-dhcp4[22198]: 2025-09-15 09:53:37.194 INFO [kea-dhcp4.dhcp4/22198.140255081506944
] DHCP4_STARTING Kea DHCPv4 server version 2.6.3 (stable) starting
Sep 15 09:53:37 server kea-dhcp4[22198]: 2025-09-15 09:53:37.194 INFO [kea-dhcp4.commands/22198.140255081506
944] COMMAND_RECEIVED Received command 'config-set'
Sep 15 09:53:37 server kea-dhcp4[22198]: 2025-09-15 09:53:37.194 INFO [kea-dhcp4.hosts/22198.140255081506944
] HOSTS_BACKENDS_REGISTERED the following host backend types are available: mysql postgresql
Sep 15 09:53:37 server kea-dhcp4[22198]: 2025-09-15 09:53:37.194 WARN [kea-dhcp4.dhcp4srv/22198.1402550815069
44] DHCP4_SRV_MT_DISABLED_QUEUE_CONTROL disabling dhcp queue control when multi-threading is enabled.
```

Рис. 3.13: Запуск мониторинга

В основном рабочем терминале запустим DHCP-сервер: `systemctl start kea-dhcp4.service` (рис. 14)



```
[root@server.svivanov.net rz]# systemctl start kea-dhcp4.service
[root@server.svivanov.net rz]#
```

Рис. 3.14: Запуск DHCP-сервера

3.3 Анализ работы DHCP-сервера

Перед запуском виртуальной машины `client` в каталоге с проектом в подкаталоге `vagrant/provision/client` отредактируем файл `01-routing.sh`. Этот скрипт изменяет настройки `NetworkManager` так, чтобы весь трафик на виртуальной машине `client` шёл по умолчанию через интерфейс `eth1`. (рис. 15)

```
#!/bin/bash

echo "Provisioning script $0"

nmcli connection modify "eth1" ipv4.gateway "192.168.1.1"
nmcli connection up "eth1"

nmcli connection modify eth0 ipv4.never-default true
nmcli connection modify eth0 ipv6.never-default true

nmcli connection down eth0
nmcli connection up eth0

# systemctl restart NetworkManager
```

Рис. 3.15: Редактирование 01-routing.sh

Изменения в файле Vagrantfile не требуются.

Запустим виртуальную машину client: `vagrant up client` (рис. 16)

```
C:\work_asp\svivanov\vagrant>vagrant up client
```

Рис. 3.16: Запуск client

После загрузки виртуальной машины client можем увидеть на виртуальной машине server на терминале с мониторингом происходящих в системе процессов записи о подключении к виртуальной внутренней сети узла client и выдачи ему IP-адреса из соответствующего диапазона адресов. Также информацию о работе DHCP-сервера можно наблюдать в файле `/var/lib/kea/kea-leases4.csv`.

Анализ файла `/var/lib/kea/kea-leases4.csv` :

- IP адрес: 192.168.1.30
- MAC адрес: 08:00:27:e7:37:d8
- Client ID: 01:08:00:27:e7:37:d8
- Время жизни: 3600 секунд (1 час)

- Время истечения: 1758021328
- Подсеть: 1
- FQDN: client.svivanov.net
- Состояние: 0 (активно) (рис. 17)

```

address,hwaddr,client_id,valid_lifetime,expire,subnet_id,fqdn_fwd,fqdn_rev,hostname,state,user_context,pool_id
192.168.1.30,08:00:27:e7:37:d8,01:08:00:27:e7:37:d8,3600,1758021328,1,1,1,client.svivanov.net,0,,0
192.168.1.30,08:00:27:e7:37:d8,01:08:00:27:e7:37:d8,3600,1758021336,1,1,1,client.svivanov.net,0,,0
~
~

```

Рис. 3.17: Анализ подключения

Войдем в систему виртуальной машины client под нашим пользователем и откроем терминал. В терминале введем ifconfig.

Анализ выведенной информации:

Интерфейс eth0:

- Состояние: UP, BROADCAST, RUNNING, MULTICAST - интерфейс активен и функционирует
- MTU: 1500 байт - стандартный размер для Ethernet
- IPv4 адрес: 10.0.2.15/24 - частный адрес в сети класса A
- IPv6 адреса:
- fe80::a00:27ff:fe83:f313 - link-local адрес
- fd17:625c:f037:2:a00:27ff:fe83:f313 - уникальный локальный IPv6 адрес
- MAC адрес: 08:00:27:83:f3:13 - виртуальный адрес
- Статистика: 1470 полученных, 1294 отправленных пакетов

Интерфейс eth1:

- Состояние: Аналогично eth0 - интерфейс активен
- IPv4 адрес: 192.168.1.30/24 - соответствует DHCP
- IPv6 адрес: fe80::a00:27ff:fe83:f313 - link-local адрес
- MAC адрес: 08:00:27:87:08
- Проблема: 12 dropped packets при передаче
- Статистика: 157 полученных, 403 отправленных пакетов

Интерфейс lo:

- Внутренний интерфейс для локального трафика
- IPv4: 127.0.0.1/8 - стандартный loopback адрес
- IPv6: ::1/128 - IPv6 loopback адрес
- MTU: 65536 байт - размер для локального трафика (рис. 18)

```
[svivanov@client.svivanov.net ~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe83:f313 prefixlen 64 scopeid 0x20<link>
    inet6 fd17:625c:f037:2:a00:27ff:fe83:f313 prefixlen 64 scopeid 0x0<global>
    ether 08:00:27:83:f3:13 txqueuelen 1000 (Ethernet)
    RX packets 1470 bytes 182960 (178.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1294 bytes 208451 (203.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.30 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::4d91:2872:e340:6fb9 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e7:37:d8 txqueuelen 1000 (Ethernet)
    RX packets 157 bytes 19856 (19.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 403 bytes 43773 (42.7 KiB)
    TX errors 0 dropped 12 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 23 bytes 2769 (2.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 23 bytes 2769 (2.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[svivanov@client.svivanov.net ~]$
```

Рис. 3.18: Анализ ifconfig

На машине server посмотрим список выданных адресов: `cat /var/lib/kea/kea-leases4.csv`. Этот файл уже проанализирован. (рис. 19)

```
[root@server.svivanov.net kea]# cat kea-leases4.csv
address,hwaddr,client_id,valid_lifetime,expire,subnet_id,fqdn_fwd,fqdn_rev,hostname,state,user_context,pool_id
192.168.1.30,08:00:27:e7:37:d8,01:08:00:27:e7:37:d8,3600,1758021328,1,1,1,client.svivanov.net,0,,0
192.168.1.30,08:00:27:e7:37:d8,01:08:00:27:e7:37:d8,3600,1758021336,1,1,1,client.svivanov.net,0,,0
[root@server.svivanov.net kea]#
```

Рис. 3.19: Список выданных адресов

3.4 Настройка обновления DNS-зоны

Создадим ключ на сервере с Bind9 (на виртуальной машине server):

```
mkdir -p /etc/named/keys
```

```
tsig-keygen -a HMAC-SHA512 DHCP_UPDATER > /etc/named/keys/dhcp_updater.key
```

Файл `/etc/named/keys/dhcp_updater.key` будет иметь следующий вид (рис. 20)

```
[root@server.svivanov.net kea]# mkdir -p /etc/named/keys
[root@server.svivanov.net kea]# tsig-keygen -a HMAC-SHA512 DHCP_UPDATER > /etc/named/keys/dhcp_updater.key
[root@server.svivanov.net kea]# cd
[root@server.svivanov.net ~]# cd /etc/named/keys/dhcp_updater.key
-bash: cd: /etc/named/keys/dhcp_updater.key: Это не каталог
[root@server.svivanov.net ~]# cd /etc/named/keys/
[root@server.svivanov.net keys]# cat dhcp_updater.key
key "DHCP_UPDATER" {
    algorithm hmac-sha512;
    secret "zssId9+A8D7*RTpeoxyPG4buhKMdp+QjFmhSidr3Zs272h0i3noG5rixrvR0cVyu7R0UUbPWeISivr2TIClQkw=";
};
[root@server.svivanov.net keys]#
```

Рис. 3.20: Создание ключа

Поправим права доступа: `chown -R named:named /etc/named/keys` (рис. 21)

```
[root@server.svivanov.net keys]# chown -R named:named /etc/named/keys
```

Рис. 3.21: Права доступа

Подключим ключ в файле `/etc/named.conf`: `include "/etc/named/keys/dhcp_updater.key"`; (рис. 22)

```
include "/etc/named/svivanov.net";
include "/etc/named/keys/dhcp_updater.key";
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
-- РЕЖИМ ВСТАВКИ --
```

Рис. 3.22: Подключение ключа

На виртуальной машине server под пользователем с правами суперпользователя отредактируем файл /etc/named/user.net, разрешив обновление зоны: (рис. 23)

```
zone "svivanov.net" IN {
    type primary;
    file "master/fz/svivanov.net";
    update-policy {
        grant DHCP_UPDATER wildcard *.user.net A DHCID;
    };
};

zone "1.168.192.in-addr.arpa" IN {
    type primary;
    file "master/rz/192.168.1";
    update-policy {
        grant DHCP_UPDATER wildcard *.1.168.192.in-addr.arpa PTR DHCID;
    };
};
```

Рис. 3.23: Редактирование файла /etc/named/user.net

Сделаем проверку конфигурационного файла: named-checkconf. Перезапустим DNS-сервер: systemctl restart named (рис. 24)

```
[root@server.svivanov.net named]# vim svivanov.net
[root@server.svivanov.net named]# named-checkconf
[root@server.svivanov.net named]# systemctl restart named
[root@server.svivanov.net named]#
```

Рис. 3.24: Проверка и перезапуск сервера

Сформируем ключ для Kea. Файл ключа назовём /etc/kea/tsig-keys.json: touch /etc/kea/tsig-keys.json. Перенесём ключ на сервер Kea DHCP и перепишем его в формате json (рис. 25)

```
{
  "tsig-keys": [
    {
      "name": "DHCP_UPDATER",
      "algorithm": "hmac-sha512",
      "secret": "XE54La0EnsVCUGWQU4twR9tRi10ndSkiGZRD0AgH5DM3pGJGVtwei4eb5o6LqXgM5WvYDjubwTXTofbDdo
xYDw=="
    }
  ]
}
```

Рис. 3.25: Формирование ключа

Сменим владельца: chown kea:kea /etc/kea/tsig-keys.json. Поправим права доступа: chmod 640 /etc/kea/tsig-keys.json (рис. 26)

```
[root@server.svivanov.net kea]# chown kea:kea /etc/kea/tsig-keys.json
[root@server.svivanov.net kea]# chmod 640 /etc/kea/tsig-keys.json
[root@server.svivanov.net kea]#
```

Рис. 3.26: Смена владельца и прав

Настройка происходит в файле /etc/kea/kea-dhcp-ddns.conf: (рис. 27)

```
"forward-ddns" : {
    "ddns-domains" : [
        {
            "name": "svivanov.net.",
            "key-name": "DHCP_UPDATER",
            "dns-servers": [
                {"ip-address": "192.168.1.1"}
            ]
        }
    ]
},
```

Рис. 3.27: Настройка

Изменим владельца файла: `chown kea:kea /etc/kea/kea-dhcp-ddns.conf`. Проверим файл на наличие возможных синтаксических ошибок: `kea-dhcp-ddns -t /etc/kea/kea-dhcp-ddns.conf` (рис. 28)

```
[root@server.svivanov.net kea]# chown kea:kea /etc/kea/kea-dhcp-ddns.conf
[root@server.svivanov.net kea]# kea-dhcp-ddns -t /etc/kea/kea-dhcp-ddns.conf
2025-09-15 11:15:38.503 INFO [kea-dhcp-ddns.dctl/34228.139883205669184] DCTL_CONFIG_CHECK_COMPLETE server has completed configuration check: listening on 127.0.0.1, port 53001, using UDP, result: success(0), text=Configuration check successful
[root@server.svivanov.net kea]#
```

Рис. 3.28: Смена владельца и проверка ошибок

Запустим службу ddns: `systemctl enable --now kea-dhcp-ddns.service`. Проверим статус работы службы: `systemctl status kea-dhcp-ddns.service` (рис. 29)

```
[root@server.svivanov.net kea]# systemctl enable --now kea-dhcp-ddns.service
Created symlink '/etc/systemd/system/multi-user.target.wants/kea-dhcp-ddns.service' → '/usr/lib/systemd/system/kea-dhcp-ddns.service'.
[root@server.svivanov.net kea]# systemctl status kea-dhcp-ddns.service
● kea-dhcp-ddns.service - Kea DHCP-DDNS Server
   Loaded: loaded (/usr/lib/systemd/system/kea-dhcp-ddns.service; enabled; preset: disabled)
   Active: active (running) since Mon 2025-09-15 11:16:00 UTC; 7s ago
     Invocation: 4c91603f305245ecbba3e926feb05c42
       Docs: man:kea-dhcp-ddns(8)
    Main PID: 34413 (kea-dhcp-ddns)
      Tasks: 5 (limit: 23144)
     Memory: 1.7M (peak: 5.9M)
        CPU: 21ms
    CGroup: /system.slice/kea-dhcp-ddns.service
            └─34413 /usr/sbin/kea-dhcp-ddns -c /etc/kea/kea-dhcp-ddns.conf
```

Рис. 3.29: Запуск и проверка статуса

Внесем изменения в конфигурационный файл /etc/kea/kea-dhcp4.conf, добавив в него разрешение на динамическое обновление DNS-записей с локального узла прямой и обратной зон: (рис. 30)

```
"dhcp-ddns": {
    "enable-updates": true
},
"ddns-qualifying-suffix": "svivanov.net",
"ddns-override-client-update": true,
}
}
-- РЕЖИМ ВСТАВКИ --
```

Рис. 3.30: Добавление динамического обновления DNS-записей

Проверим файл на наличие возможных синтаксических ошибок: kea-dhcp4 -t /etc/kea/kea-dhcp4.conf (рис. 31)

```
[root@server.svivanov.net kea]# kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
2025-09-15 11:19:41.630 WARN [kea-dhcp4.dhcp4/35101.140069864990848] DHCP4_CONFIG_SYNTAX_WARNING configuration syntax warning: /etc/kea/kea-dhcp4.conf:253.36: Extraneous comma. A piece of configuration may have been omitted.
2025-09-15 11:19:41.630 INFO [kea-dhcp4.hosts/35101.140069864990848] HOSTS_BACKENDS_REGISTERED the following host backend types are available: mysql postgresql
2025-09-15 11:19:41.631 WARN [kea-dhcp4.dhcp4/35101.140069864990848] DHCP4_MT_DISABLED_QUEUE_CONTROL disabling dhcp queue control when multi-threading is enabled.
2025-09-15 11:19:41.631 WARN [kea-dhcp4.dhcp4/35101.140069864990848] DHCP4_RESERVATIONS_LOOKUP_FIRST_ENABLED Multi-threading is enabled and host reservations lookup is always performed first.
2025-09-15 11:19:41.631 INFO [kea-dhcp4.dhcp4/35101.140069864990848] DHCP4_CFGMGR_NEW_SUBNET4 a new subnet has been added to configuration: 192.168.1.0/24 with parameters: t1=000, t2=1800, valid lifetime=2560
```

Рис. 3.31: Проверка ошибок

Перезапустим DHCP-сервер: systemctl restart kea-dhcp4.service. Проверим статус: systemctl status kea-dhcp4.service: (рис. 32)

```
[root@server.svivanov.net kea]# systemctl restart kea-dhcp4.service
[root@server.svivanov.net kea]# systemctl status kea-dhcp4.service
● kea-dhcp4.service - Kea DHCPv4 Server
   Loaded: loaded (/usr/lib/systemd/system/kea-dhcp4.service; enabled; preset: disabled)
   Active: active (running) since Mon 2025-09-15 11:20:38 UTC; 5s ago
   Invocation: 7ffd0a96e40e4a04b504e92fc9610a1f
     Docs: man:kea-dhcp4(8)
    Main PID: 35235 (kea-dhcp4)
      Tasks: 9 (limit: 23144)
     Memory: 2.6M (peak: 5.8M)
        CPU: 26ms
    CGroup: /system.slice/kea-dhcp4.service
            └─35235 /usr/sbin/kea-dhcp4 -c /etc/kea/kea-dhcp4.conf
```

Рис. 3.32: Запуск и проверка статуса

На машине client переполучим адрес:

nmcli connection down eth1

nmcli connection up eth1 (рис. 33)

```
[svivanov@client.svivanov.net ~]$ nmcli connection down eth1
Подключение «eth1» успешно деактивировано (активный путь D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/4)
[svivanov@client.svivanov.net ~]$ nmcli connection up eth1
Подключение успешно активировано (активный путь D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/6)
[svivanov@client.svivanov.net ~]$
```

Рис. 3.33: Переполучение адреса

3.5 Анализ работы DHCP-сервера после настройки обновления DNS-зоны

На виртуальной машине client откроем терминал и с помощью утилиты dig убедимся в наличии DNS-записи о клиенте в прямой DNS-зоне:

Анализ выведенной информации: запрос выполнен без ошибок

- запрос: client.svivanov.net
- Status: NOERROR - домен найден
- ANSWER: 1 - есть ответная запись
- IP адрес: 192.168.1.30 - именно тот, который нужен (рис. 34)

```
[svivanov@client.svivanov.net ~]$ dig @192.168.1.1 client.svivanov.net

; <<>> DiG 9.18.33 <<>> @192.168.1.1 client.svivanov.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62266
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 66cf3b60ad1f4f350100000068c976bd63e1f1f36835821e (good)
;; QUESTION SECTION:
;client.svivanov.net.      IN      A

;; ANSWER SECTION:
client.svivanov.net.      86400   IN      A      192.168.1.30

;; Query time: 1 msec
;; SERVER: 192.168.1.1#53(192.168.1.1) (UDP)
;; WHEN: Tue Sep 16 14:39:56 UTC 2025
;; MSG SIZE  rcvd: 92
```

Рис. 3.34: Анализ запроса

3.6 Внесение изменений в настройки внутреннего окружения виртуальной машины

На виртуальной машине `server` перейдем в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создадим в нём каталог `dhcp`, в который поместим в соответствующие подкаталоги конфигурационные файлы DHCP:

```
cd /vagrant/provision/server
```

```
mkdir -p /vagrant/provision/server/dhcp/etc/kea
```

```
cp -R /etc/kea/* /vagrant/provision/server/dhcp/etc/kea/
```

Заменим конфигурационные файлы DNS-сервера:

```
cd /vagrant/provision/server/dns/
```

```
cp -R /var/named/* /vagrant/provision/server/dns/var/named/
```

```
cp -R /etc/named/* /vagrant/provision/server/dns/etc/named/ (рис. 35)
```



```
[root@server.svivanov.net fz]# cd /vagrant/provision/server
[root@server.svivanov.net server]# mkdir -p /vagrant/provision/server/dhcp/etc/kea
[root@server.svivanov.net server]# cp -R /etc/kea/* /vagrant/provision/server/dhcp/etc/kea/
[root@server.svivanov.net server]# cd /vagrant/provision/server/dns/
[root@server.svivanov.net dns]# cp -R /var/named/* /vagrant/provision/server/dns/var/named/
cp: overwrite '/vagrant/provision/server/dns/var/named/master/rz/192.168.1'? yes
cp: overwrite '/vagrant/provision/server/dns/var/named/master/fz/svivanov.net'? yes
[root@server.svivanov.net dns]# cp -R /etc/named/* /vagrant/provision/server/dns/etc/named/
cp: overwrite '/vagrant/provision/server/dns/etc/named/svivanov.net'? yes
[root@server.svivanov.net dns]#
```

Рис. 3.35: Замена конф. файлов

В каталоге `/vagrant/provision/server` создадим исполняемый файл `dhcp.sh`:

```
cd /vagrant/provision/server
```

```
touch dhcp.sh
```

```
chmod +x dhcp.sh
```

Открыв его на редактирование, пропишем в нём следующий скрипт: (рис. 36)

```
#!/bin/bash
echo "Provisioning script $0"
echo "Install needed packages"
dnf -y install kea
echo "Copy configuration files"
cp -R /vagrant/provision/server/dhcp/etc/kea/* /etc/kea/
echo "Fix permissions"
chown -R kea:kea /etc/kea
chmod 640 /etc/kea/tsig-keys.json
restorecon -vR /etc
restorecon -vR /var/lib/kea
echo "Configure firewall"
firewall-cmd --add-service dhcp
firewall-cmd --add-service dhcp --permanent
echo "Start dhcpd service"
systemctl --system daemon-reload
systemctl enable --now kea-dhcp4.service
systemctl enable --now kea-dhcp-ddns.service
```

Рис. 3.36: Запуск и проверка статуса

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile необходимо добавить в разделе конфигурации для сервера: (рис. 37)

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

Рис. 3.37: Редактирование Vagrantfile

4 Ответы на контрольные вопросы

1. В каких файлах хранятся настройки сетевых подключений?

Настройки сетевых подключений в Linux, использующих NetworkManager, хранятся в файлах по пути:

`/etc/sysconfig/network-scripts/ifcfg-`

Например: `/etc/sysconfig/network-scripts/ifcfg-eth0`. Также NetworkManager может хранить настройки в собственной бинарной базе данных. Для управления подключениями через NetworkManager используются утилиты `nmcli` или `nmtui`.

2. За что отвечает протокол DHCP?

DHCP (Dynamic Host Configuration Protocol) отвечает за автоматическую настройку сетевых параметров на клиентских устройствах. Он предоставляет:

Динамическое назначение IP-адресов из заданного пула, указание маски подсети (`netmask`), назначение шлюза по умолчанию (`default gateway`), указание адресов DNS-серверов, другие параметры сети

3. Поясните принцип работы протокола DHCP. Какими сообщениями обмениваются клиент и сервер, используя протокол DHCP?

Принцип работы DHCP:

DISCOVER - клиент широковещательно ищет доступные DHCP-серверы

OFFER - сервер предлагает клиенту IP-адрес и параметры конфигурации

REQUEST - клиент подтверждает принятие предложения

ACK - сервер подтверждает выделение адреса и параметров

Дополнительные сообщения:

NAK - отказ сервера в предоставлении адреса

RELEASE - клиент освобождает IP-адрес

DECLINE - клиент отказывается от предложенного адреса

4. В каких файлах обычно находятся настройки DHCP-сервера? За что отвечает каждый из файлов?

Для DHCP-сервера Kea:

/etc/kea/kea-dhcp4.conf - основной конфигурационный файл DHCPv4-сервера

/etc/kea/kea-dhcp-ddns.conf - конфигурация динамического DNS

/etc/kea/tsig-keys.json - файл ключей TSIG для безопасного обновления DNS

/var/lib/kea/kea-leases4.csv - база данных выданных аренд (lease database)

5. Что такое DDNS? Для чего применяется DDNS?

DDNS (Dynamic DNS) - технология, позволяющая автоматически обновлять DNS-записи при изменении IP-адресов или появлении новых узлов в сети.

Применение DDNS: Автоматическое создание DNS-записей для клиентов, получающих адреса по DHCP, обновление записей при смене IP-адресов, синхронизация прямой и обратной DNS-зон, упрощение администрирования сетей с динамическими адресами.

6. Какую информацию можно получить, используя утилиту ifconfig?

Приведите примеры с использованием различных опций.

ifconfig показывает информацию о сетевых интерфейсах:

Примеры использования:

Показать все активные интерфейсы

- ifconfig

Показать конкретный интерфейс

- ifconfig eth0

Показать все интерфейсы

- ifconfig -a

Включить интерфейс

- `ifconfig eth0 up`

Выключить интерфейс

- `ifconfig eth0 down`

Выводимая информация: IP-адрес, маска подсети, MAC-адрес, статистика приема/передачи пакетов, флаги состояния интерфейса.

7. Какую информацию можно получить, используя утилиту ping? Приведите примеры с использованием различных опций.

ping используется для проверки доступности сетевых узлов и измерения задержки:

Базовая проверка доступности

- `ping example.com`

Ограничить количество пакетов - `ping -c 5 example.com`

Указать размер пакета - `ping -s 1000 example.com`

Указать интервал между пакетами (в секундах) - `ping -i 2 example.com`

Непрерывный ping - `ping -t example.com`

Ping с временными метками - `ping -D example.com`

5 Выводы

В ходе выполнения лабораторной работы мы приобрели практические навыки по установке и конфигурированию DHCP сервера.