

Отчёт по лабораторной работе 3

Настройка DHCP-сервера

Сейдалиев Тагиетдин Ровшенович

Содержание

1 Цель работы	5
2 Выполнение	6
2.1 Конфигурирование DHCP-сервера	6
2.2 Анализ работы DHCP-сервера	10
2.3 Настройка обновления DNS-зоны	13
2.4 Анализ работы DHCP-сервера после настройки обновления DNS-зоны	19
2.5 Внесение изменений в настройки внутреннего окружения виртуальной машины	21
3 Заключение	23
4 Контрольные вопросы	24
4.1 1. В каких файлах хранятся настройки сетевых подключений?	24
4.2 2. За что отвечает протокол DHCP?	24
4.3 3. Поясните принцип работы протокола DHCP. Какими сообщениями обмениваются клиент и сервер?	25
4.4 4. В каких файлах обычно находятся настройки DHCP-сервера? За что отвечает каждый из файлов?	26
4.5 5. Что такое DDNS? Для чего применяется DDNS?	26
4.6 6. Какую информацию можно получить, используя утилиту ifconfig? Приведите примеры с использованием различных опций.	27
4.6.1 Примеры:	27
4.7 7. Какую информацию можно получить, используя утилиту ping? Приведите примеры с использованием различных опций.	28
4.7.1 Примеры:	28

Список иллюстраций

2.1	Фрагмент kea-dhcp4.conf с параметрами домена и DNS	6
2.2	Конфигурация подсети DHCP	7
2.3	Проверка и включение службы kea-dhcp4	8
2.4	Файл прямой зоны после изменения	8
2.5	Файл обратной зоны	9
2.6	Проверка ping по имени dhcp..net	9
2.7	Изменения в firewall и SELinux	9
2.8	Скрипт настройки маршрутизации клиента	10
2.9	Вывод ifconfig на клиенте	11
2.10	Содержимое файла kea-leases4.csv	12
2.11	Создание ключа DHCP_UPDATER	13
2.12	Включение ключа в named.conf	14
2.13	Разрешение обновлений в прямой зоне	15
2.14	Разрешение обновлений в обратной зоне	15
2.15	Файл tsig-keys.json	16
2.16	Фрагмент kea-dhcp-ddns.conf	17
2.17	Запуск службы kea-dhcp-ddns	18
2.18	Проверка и запуск Kea DHCP4	19
2.19	Результат выполнения dig	20
2.20	Содержимое dhcp.sh	22

Список таблиц

1 Цель работы

Приобретение практических навыков по установке и конфигурированию DHCP-сервера.

2 Выполнение

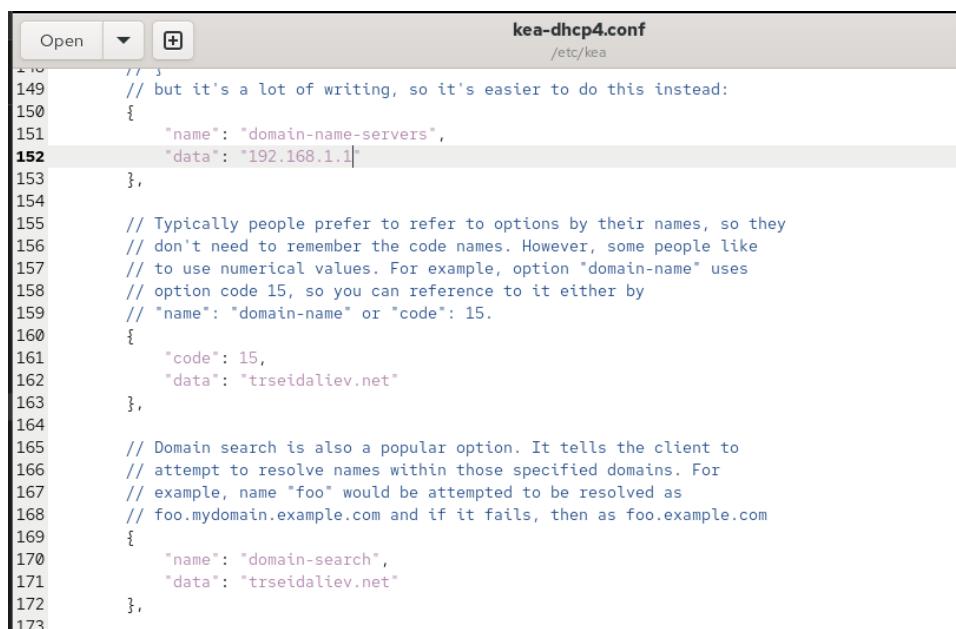
2.1 Конфигурирование DHCP-сервера

Перед началом настройки была выполнена резервная копия конфигурационного файла Kea DHCP.

После открытия файла `/etc/kea/kea-dhcp4.conf` внесены изменения в параметры домена.

В блоке `option-data` значения `domain-name` и `domain-search` заменены на домен вида `<логин>.net`,

а список DNS-серверов заменён на `192.168.1.1`.



The screenshot shows a text editor window with the title 'kea-dhcp4.conf' and the path '/etc/kea'. The code in the editor is as follows:

```
// but it's a lot of writing, so it's easier to do this instead:  
149 {  
150     "name": "domain-name-servers",  
151     "data": "192.168.1.1"  
152 },  
153  
154 // Typically people prefer to refer to options by their names, so they  
155 // don't need to remember the code names. However, some people like  
156 // to use numerical values. For example, option 'domain-name' uses  
157 // option code 15, so you can reference to it either by  
158 // "name": "domain-name" or "code": 15.  
159 {  
160     "code": 15,  
161     "data": "trseidaliev.net"  
162 },  
163  
164 // Domain search is also a popular option. It tells the client to  
165 // attempt to resolve names within those specified domains. For  
166 // example, name "foo" would be attempted to be resolved as  
167 // foo.mydomain.example.com and if it fails, then as foo.example.com  
168 {  
169     "name": "domain-search",  
170     "data": "trseidaliev.net"  
171 },  
172  
173
```

Рис. 2.1: Фрагмент `kea-dhcp4.conf` с параметрами домена и DNS

Создана собственная конфигурация подсети:

задана сеть 192.168.1.0/24, диапазон выдаваемых адресов 192.168.1.30–192.168.1.199, и указан адрес шлюза 192.168.1.1.

The screenshot shows a code editor window with the file 'kea-dhcp4.conf' open. The file is located at '/etc/kea'. The code defines a subnet with ID 1, a subnet of 192.168.1.0/24, and a pool of addresses from 192.168.1.30 to 192.168.1.199. It also specifies a router option with the value 192.168.1.1. The configuration is annotated with comments explaining the structure of the JSON objects.

```
// This is a list, denoted with [ ], or structures, each denoted with
// { }. Each structure describes a single subnet and may have several
// parameters. One of those parameters is "pools" that is also a list of
// 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"
            }
        ]
    }
], // You can add more subnets there.
```

Рис. 2.2: Конфигурация подсети DHCP

DHCP-сервер настроен на работу через интерфейс eth1, что зафиксировано в параметре `interfaces-config`.

Произведена проверка конфигурации и перезагрузка службы DHCP. Также включена автоматическая загрузка сервиса при старте системы.

```
[root@server.trseidaliev.net server]# 
[root@server.trseidaliev.net server]# kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
2025-11-19 19:00:06.379 INFO [kea-dhcp4.hosts/21246.139875138164928] HOSTS_BACKENDS_REGISTERED the following host backend types are available: mysql postgresql
2025-11-19 19:00:06.380 WARN [kea-dhcp4.dhcpsrv/21246.139875138164928] DHCP4_MT_DISABLED_QUEUE_CONTROL disabling dhcp queue control when multi-threading is enabled.
2025-11-19 19:00:06.380 WARN [kea-dhcp4.dhcpsrv/21246.139875138164928] DHCP4_RESERVATIONS_LOOKUP_FIRST_ENABLED Multi-threading is enabled and host reservations lookup is always performed first.
2025-11-19 19:00:06.381 INFO [kea-dhcp4.dhcpsrv/21246.139875138164928] DHCP4_CFMGR_NEW_SUBNET4 a new subnet has been added to configuration: 192.168.1.0/24 with params: t1=900, t2=1800, valid-lifetime=3600
2025-11-19 19:00:06.381 INFO [kea-dhcp4.dhcpsrv/21246.139875138164928] DHCP4_CFMGR_SOCKET_TYPE_SELECT using socket type raw
2025-11-19 19:00:06.381 INFO [kea-dhcp4.dhcpsrv/21246.139875138164928] DHCP4_CFMGR_ADD_IFACE listening on interface eth1
2025-11-19 19:00:06.381 INFO [kea-dhcp4.dhcpsrv/21246.139875138164928] DHCP4_CFMGR_SOCKET_TYPE_DEFAULT "dhcp-socket-type" not specified , using default socket type raw
[root@server.trseidaliev.net server]# systemctl --system daemon-reload
[root@server.trseidaliev.net server]# 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.trseidaliev.net server]# 
```

Рис. 2.3: Проверка и включение службы kea-dhcp4

В конце файла `/var/named/master/fz/<логин>.net` добавлена запись

Также обновлён серийный номер зоны.

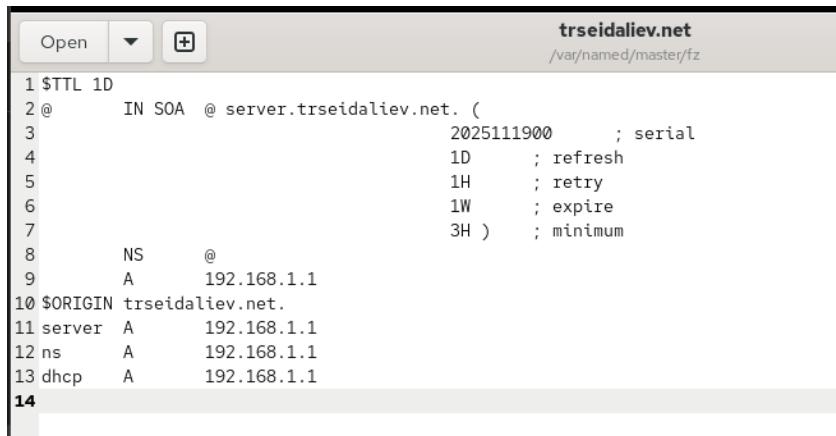


Рис. 2.4: Файл прямой зоны после изменения

В файл `/var/named/master/rz/192.168.1` добавлена PTR-запись

Серийный номер зоны обновлён.

```

1 $TTL 1D
2 @ IN SOA @ server.trseidaliev.net. (
3                               2025111900 ; serial
4                               1D      ; refresh
5                               1H      ; retry
6                               1W      ; expire
7                               3H )    ; minimum
8 NS      @
9 A       192.168.1.1
10 AAAA   ::1
11 PTR    server.trseidaliev.net.
12 $ORIGIN 1.168.192.in-addr.arpa.
13 1     PTR    server.trseidaliev.net.
14 1     PTR    ns.trseidaliev.net.
15 1     PTR    dhcp.trseidaliev.net.
16 |

```

Рис. 2.5: Файл обратной зоны

После перезапуска DNS-сервера выполнена проверка доступности DHCP-узла по имени.

```

[root@server.trseidaliev.net server]#
[root@server.trseidaliev.net server]# systemctl restart named
[root@server.trseidaliev.net server]#
[root@server.trseidaliev.net server]# ping dhcp.trseidaliev.net
PING dhcp.trseidaliev.net (192.168.1.1) 56(84) bytes of data.
64 bytes from server.trseidaliev.net (192.168.1.1): icmp_seq=1 ttl=64 time=0.012 ms
64 bytes from server.trseidaliev.net (192.168.1.1): icmp_seq=2 ttl=64 time=0.084 ms
64 bytes from server.trseidaliev.net (192.168.1.1): icmp_seq=3 ttl=64 time=0.037 ms
64 bytes from server.trseidaliev.net (192.168.1.1): icmp_seq=4 ttl=64 time=0.064 ms
64 bytes from server.trseidaliev.net (192.168.1.1): icmp_seq=5 ttl=64 time=0.061 ms
64 bytes from server.trseidaliev.net (192.168.1.1): icmp_seq=6 ttl=64 time=0.077 ms
^C
--- dhcp.trseidaliev.net ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5577ms
rtt min/avg/max/mdev = 0.012/0.055/0.084/0.024 ms
[root@server.trseidaliev.net server]#

```

Рис. 2.6: Проверка ping по имени dhcp..net

В межсетевом экране разрешена служба DHCP, а также восстановлены контексты безопасности SELinux для каталогов /etc, /var/named и /var/lib/kea.

```

[root@server.trseidaliev.net server]#
[root@server.trseidaliev.net server]# firewall-cmd --add-service=dhcp
success
[root@server.trseidaliev.net server]# firewall-cmd --add-service=dhcp --permanent
success
[root@server.trseidaliev.net server]# restorecon -vR /etc
[root@server.trseidaliev.net server]# restorecon -vR /var/named/
[root@server.trseidaliev.net server]# restorecon -vR /var/lib/kea/
[root@server.trseidaliev.net server]# systemctl start kea-dhcp4.service
[root@server.trseidaliev.net server]#

```

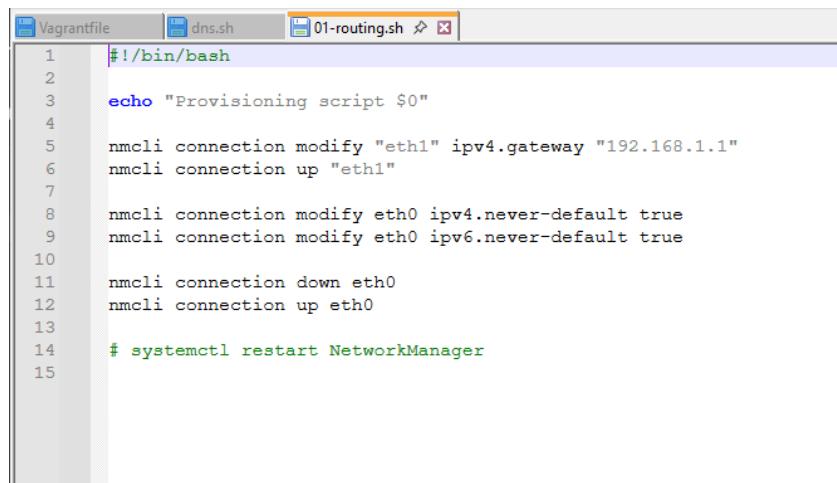
Рис. 2.7: Изменения в firewall и SELinux

2.2 Анализ работы DHCP-сервера

Для корректной работы клиентской машины был создан скрипт `01-routing.sh`, который изменяет параметры NetworkManager, направляя весь исходящий трафик через интерфейс `eth1`.

Скрипт прописан в каталоге `vagrant/provision/client` и подключён в секции конфигурации клиента в `Vagrantfile`.

После запуска провижининга клиентской машины настройки применились:



```
#!/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. 2.8: Скрипт настройки маршрутизации клиента

После запуска виртуальной машины `client` выполнена проверка сетевых интерфейсов.

Интерфейс `eth1` успешно получил IP-адрес из диапазона DHCP-сервера:

```
RX packets 1927 bytes 232458 (227.0 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1660 bytes 268730 (262.4 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::86cf:247e:f9d1:8ac9 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:36:98:16 txqueuelen 1000 (Ethernet)
RX packets 62 bytes 7039 (6.8 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 238 bytes 22681 (22.1 KiB)
TX errors 0 dropped 0 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 18 bytes 2112 (2.0 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 18 bytes 2112 (2.0 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 2.9: Вывод ifconfig на клиенте

Постстрочный анализ:

- **eth1** – активный интерфейс, через который осуществляется подключение к внутренней сети.
- **inet 192.168.1.30** – клиент получил адрес из диапазона 192.168.1.30–192.168.1.199, что подтверждает корректную работу DHCP.
- **netmask 255.255.255.0** – маска подсети соответствует сети 192.168.1.0/24.
- **broadcast 192.168.1.255** – широковещательный адрес установлен автоматически.
- **ether 08:00:27:36:98:16** – MAC-адрес интерфейса, используемый DHCP-сервером для идентификации клиента.

- Статистика RX/TX показывает обмен пакетами внутри сети — интерфейс функционирует.

Интерфейс `lo` — стандартный Loopback, работает корректно.

На сервере зафиксировано несколько записей в файле `/var/lib/kea/kea-leases4.csv`:

```
[root@server.trseidaliev.net server]#
[root@server.trseidaliev.net server]# cat /var/lib/kea/kea-leases4.csv
address,hwaddr,client_id,valid_lifetime,expire,subnet_id,fqdn_fwd,fqdn_rev,hostname,state,user_context,pool_id
192.168.1.30,08:00:27:36:98:16,01:08:00:27:36:98:16,3600,1763571977,1,0,0,client,,0
192.168.1.30,08:00:27:36:98:16,01:08:00:27:36:98:16,3600,1763571977,1,0,0,client,,0
192.168.1.30,08:00:27:36:98:16,01:08:00:27:36:98:16,3600,1763571982,1,0,0,client,,0
[root@server.trseidaliev.net server]#
```

Рис. 2.10: Содержимое файла `kea-leases4.csv`

Построчное объяснение (формат Kea):

- **192.168.1.30** — IP, выданный клиенту.
- **08:00:27:36:98:16** — MAC-адрес клиента.
- **01:08:00:27:36:98:16** — client-id, автоматически сгенерированный.
- **3600** — время аренды в секундах (1 час).
- **1763571977** — timestamp истечения аренды.
- **1** — ID подсети (subnet4 ID = 1).
- **0,0** — FQDN обновления (вперёд/назад) отключены.
- **client** — имя хоста.

- **0** – поле состояния (lease active).
- **0** – ID пула, из которого выдан адрес.

2.3 Настройка обновления DNS-зоны

Для обеспечения автоматического создания A- и PTR-записей при появлении новых клиентов в сети была выполнена настройка механизма **DDNS (Dynamic DNS Updates)** между Kea DHCP и Bind9.

На сервере была создана директория для ключей и сформирован TSIG-ключ для авторизации обновлений:

```
[root@server.trseidaliev.net server]#
[root@server.trseidaliev.net server]# mkdir -p /etc/named/keys
[root@server.trseidaliev.net server]# tsig-keygen -a HMAC-SHA512 DHCP_UPDATER > /etc/named/keys/dhcp_updater.key
[root@server.trseidaliev.net server]# cat /etc/named/keys/dhcp_updater.key
key "DHCP_UPDATER" {
    algorithm hmac-sha512;
    secret "vn63cmyZgm5UQRNJs0feoxh5aWl/QexHcWouly6Kn6IuLJBvEvQO/RIWTad/lkdBSHKiVa+OmwAa3AGyoSr1A==";
};
[root@server.trseidaliev.net server]# chown -R named:named /etc/named/keys/
[root@server.trseidaliev.net server]#
```

Рис. 2.11: Создание ключа DHCP_UPDATER

После генерации ключа были установлены корректные права доступа:

В файл `/etc/named.conf` добавлено подключение ключа:

```
named.conf
/etc
28     attacks. Implementing BCP38 within your network would greatly
29     reduce such attack surface
30 */
31 recursion yes;
32
33 dnssec-validation yes;
34
35 managed-keys-directory "/var/named/dynamic";
36 geoip-directory "/usr/share/GeoIP";
37
38 pid-file "/run/named/named.pid";
39 session-keyfile "/run/named/session.key";
40
41 /* https://fedoraproject.org/wiki/Changes/CryptoPolicy */
42 include "/etc/crypto-policies/back-ends/bind.config";
43 };
44
45 logging {
46     channel default_debug {
47         file "data/named.run";
48         severity dynamic;
49     };
50 };
51
52 zone "." IN {
53     type hint;
54     file "named.ca";
55 };
56
57 include "/etc/named.rfc1912.zones";
58 include "/etc/named.root.key";
59 include "/etc/named/trseidaliev.net";
60 include "/etc/named/keys/dhcp_updater.key";
```

Рис. 2.12: Включение ключа в named.conf

В файле зоны <логин>.net указано правило обновления:

```

1 // named.rfc1912.zones:
2 //
3 // Provided by Red Hat caching-nameserver package
4 //
5 // ISC BIND named zone configuration for zones recommended by
6 // RFC 1912 section 4.1 : localhost TLDs and address zones
7 // and https://tools.ietf.org/html/rfc6303
8 // (c)2007 R W Franks
9 //
10 // See /usr/share/doc/bind*/sample/ for example named configuration files.
11 //
12 // Note: empty-zones-enable yes; option is default.
13 // If private ranges should be forwarded, add
14 // disable-empty-zone "."; into options
15 //
16
17 zone "trseidaliev.net" IN {
18     type master;
19     file "master/fz/trseidaliev.net";
20     update-policy {
21         grant DHCP_UPDATER wildcard *.trseidaliev.net A DHCID;
22     };
23 };
24
25 zone "1.168.192.in-addr.arpa" IN {
26     type master;
27     file "master/rz/192.168.1";
28     update-policy {
29         grant DHCP_UPDATER wildcard *.1.168.192.in-addr.arpa PTR DHCID;
30     };
31 };

```

Рис. 2.13: Разрешение обновлений в прямой зоне

В обратной зоне `1.168.192.in-addr.arpa` аналогично разрешены обновления PTR-записей:

```

{
    "tsig-keys": [
        {
            "name": "DHCP_UPDATER",
            "algorithm": "hmac-sha512",
            "secret": "vn63cmYZgm5UQRNjs0feoxh5aWl//QexHcWouly6Kn6IuLJBvEvQO/RIWTad/lkBShKiVa+OmwaA3AGyoSr1A=="
        }
    ]
}

```

Рис. 2.14: Разрешение обновлений в обратной зоне

После внесения изменений конфигурация проверена

DNS-сервер перезапущен.

Создан файл `/etc/kea/tsig-keys.json` и в него перенесён ключ в формате JSON:

```
*kea-dhcp-ddns.conf
/etc/kea
19 // See Section 11 for examples and details description.
20 "DhcpDdns":
21 {
22     "ip-address": "127.0.0.1",
23     "port": 53001,
24     "control-socket": {
25         "socket-type": "unix",
26         "socket-name": "/run/kea/kea-ddns-ctrl-socket"
27     },
28     <?include "/etc/kea/tsig-keys.json" ?>
29
30     "forward-ddns" : {
31         "ddns-domains" : [
32             {
33                 "name": "trseidaliev.net.",
34                 "key-name": "DHCP_UPDATER",
35                 "dns-servers": [
36                     { "ip-address": "192.168.1.1" }
37                 ]
38             }
39         ]
40     },
41
42     "reverse-ddns" : {
43         "ddns-domains" : [
44             {
45                 "name": "1.168.192.in-addr.arpa.",
46                 "key-name": "DHCP_UPDATER",
47                 "dns-servers": [
48                     { "ip-address": "192.168.1.1" }
49                 ]
50             }
51         ]
52     }
53 }
```

Рис. 2.15: Файл tsig-keys.json

Установлены права и назначен владелец

В файл /etc/kea/kea-dhcp-ddns.conf добавлены параметры для forward и reverse обновлений:

```

[root@server.trseidaliev.net server]#
[root@server.trseidaliev.net server]# kea-dhcp-ddns -t /etc/kea/kea-dhcp-ddns.conf
2025-11-19 19:21:41.818 INFO [kea-dhcp-ddns.dctl/24626.140596514251072] 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.trseidaliev.net server]# 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.trseidaliev.net server]# 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 Wed 2025-11-19 19:22:08 MSK; 13s ago
        Invocation: 0fe927993944887969997b06645d8ef
          Docs: man:kea-dhcp-ddns(8)
     Main PID: 24872 (kea-dhcp-ddns)
       Tasks: 5 (limit: 10381)
      Memory: 1.7M (peak: 6.1M)
        CPU: 10ms
      CGroup: /system.slice/kea-dhcp-ddns.service
              └─24872 /usr/sbin/kea-dhcp-ddns -c /etc/kea/kea-dhcp-ddns.conf

Nov 19 19:22:08 server.trseidaliev.net systemd[1]: Started kea-dhcp-ddns.service - Kea DHCP-DDNS Server.
Nov 19 19:22:08 server.trseidaliev.net kea-dhcp-ddns[24872]: 2025-11-19 19:22:08.247 INFO [kea-dhcp-ddns.dctl/24872]>
Nov 19 19:22:08 server.trseidaliev.net kea-dhcp-ddns[24872]: INFO COMMAND_ACCEPTOR_START Starting to accept connecti>
Nov 19 19:22:08 server.trseidaliev.net kea-dhcp-ddns[24872]: INFO DCTL_CONFIG_COMPLETE server has completed configur>
Nov 19 19:22:08 server.trseidaliev.net kea-dhcp-ddns[24872]: INFO DHCP_DDNS_STARTED Kea DHCP-DDNS server version 2.6>
lines 1-17/17 (END)

```

Рис. 2.16: Фрагмент kea-dhcp-ddns.conf

Особенно важно наличие точки в конце имён зон (`user.net.` и `1.168.192.in-addr.arpa.`),
иначе обновления будут завершаться ошибкой.

Файл проверен на корректность

Служба запущена и активирована:

The screenshot shows a terminal window with the configuration file 'kea-dhcp4.conf' open. The file is located at '/etc/kea'. The code in the file is as follows:

```
26 // Dhcpv4 configuration starts here. This section will be read by Dhcpv4 server
27 // and will be ignored by other components.
28 "Dhcp4": {
29     // Add names of your network interfaces to listen on.
30     "interfaces-config": {
31         // See section 8.2.4 for more details. You probably want to add just
32         // interface name (e.g. "eth0" or specific IPv4 address on that
33         // interface name (e.g. "eth0/192.0.2.1").
34         "interfaces": [ "eth1" ]
35
36         // Kea DHCPv4 server by default listens using raw sockets. This ensures
37         // all packets, including those sent by directly connected clients
38         // that don't have IPv4 address yet, are received. However, if your
39         // traffic is always relayed, it is often better to use regular
40         // UDP sockets. If you want to do that, uncomment this line:
41         // "dhcp-socket-type": "udp"
42     },
43
44     "dhcp-ddns" : {
45         "enable-updates": true
46     },
47     "ddns-qualifying-suffix": "trseidaliev.net",
48     "ddns-override-client-update": true,
49
50     // Kea supports control channel, which is a way to receive management
51     // commands while the server is running. This is a Unix domain socket that
52     // receives commands formatted in JSON, e.g. config-set (which sets new
```

Рис. 2.17: Запуск службы kea-dhcp-ddns

В основной конфигурации DHCP включено использование DDNS

После внесения изменений конфигурация проверена и DHCP-сервер перезапущен:

```
[root@server.trseidaliev.net server]# kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
2025-11-19 19:24:27.584 INFO [kea-dhcp4.hosts/25203.139864501876928] HOSTS_BACKENDS_REGISTERED the following host backends are available: mysql postgresql
2025-11-19 19:24:27.585 WARN [kea-dhcp4.dhcpsrv/25203.139864501876928] DHCPSRV_MT_DISABLED_QUEUE_CONTROL disabling dhcp queue control when multi-threading is enabled.
2025-11-19 19:24:27.585 WARN [kea-dhcp4.dhcp4/25203.139864501876928] DHCP4_RESERVATIONS_LOOKUP_FIRST_ENABLED Multi-threading is enabled and host reservations lookup is always performed first.
2025-11-19 19:24:27.585 INFO [kea-dhcp4.dhcpsrv/25203.139864501876928] DHCPSRV_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
2025-11-19 19:24:27.585 INFO [kea-dhcp4.dhcpsrv/25203.139864501876928] DHCPSRV_CFGMGR_SOCKET_TYPE_SELECT using socket type raw
2025-11-19 19:24:27.585 INFO [kea-dhcp4.dhcpsrv/25203.139864501876928] DHCPSRV_CFGMGR_ADD_IFACE listening on interface eth1
2025-11-19 19:24:27.585 INFO [kea-dhcp4.dhcpsrv/25203.139864501876928] DHCPSRV_CFGMGR_SOCKET_TYPE_DEFAULT "dhcp-socket-type" not specified, using default socket type raw
[root@server.trseidaliev.net server]#
[root@server.trseidaliev.net server]# systemctl restart kea-dhcp4.service
[root@server.trseidaliev.net server]# 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 Wed 2025-11-19 19:24:41 MSK; 5s ago
     Invocation: cb7e5da1c0e840fb954f99de7c5ce3c3
       Docs: man:kea-dhcp4(8)
     Main PID: 25273 (kea-dhcp4)
        Tasks: 7 (limit: 10381)
      Memory: 2.5M (peak: 6M)
        CPU: 13ms
      CGroup: /system.slice/kea-dhcp4.service
              └─25273 /usr/sbin/kea-dhcp4 -c /etc/kea/kea-dhcp4.conf

Nov 19 19:24:41 server.trseidaliev.net systemd[1]: Started kea-dhcp4.service - Kea DHCPv4 Server.
Nov 19 19:24:41 server.trseidaliev.net kea-dhcp4[25273]: 2025-11-19 19:24:41.161 INFO [kea-dhcp4.dhcp4/25273.1399095]
Nov 19 19:24:41 server.trseidaliev.net kea-dhcp4[25273]: 2025-11-19 19:24:41.161 INFO [kea-dhcp4.commands/25273.1399]
lines 1-15/15 (END)
```

Рис. 2.18: Проверка и запуск Kea DHCP4

На клиентской машине адрес был переполучен

После получения IP-адреса DHCP-сервер инициирует обновление DNS-записей.

В каталоге прямой зоны появился файл-журнал динамических обновлений

2.4 Анализ работы DHCP-сервера после настройки

обновления DNS-зоны

После завершения настройки DDNS был выполнен тест с клиентской машины.

С помощью утилиты **dig** отправлен запрос к DNS-серверу (192.168.1.1) для проверки присутствия динамически созданной записи:

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

; <>> DiG 9.18.33 <>> @192.168.1.1 client.trseidaliev.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58475
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 129fa27f46fdde9e01000000691def862038615073907298 (good)
;; QUESTION SECTION:
;client.trseidaliev.net.          IN      A

;; ANSWER SECTION:
client.trseidaliev.net. 1200    IN      A      192.168.1.30

;; Query time: 1 msec
;; SERVER: 192.168.1.1#53(192.168.1.1) (UDP)
;; WHEN: Wed Nov 19 16:30:50 UTC 2025
;; MSG SIZE rcvd: 95

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

Рис. 2.19: Результат выполнения dig

Построчный разбор:

- ; <>> DiG 9.18.33 <>> @192.168.1.1 client.trseidAliev.net

Утилита dig отправляет запрос к DNS-серверу по адресу 192.168.1.1 на получение А-записи клиентского хоста.

- ;; Got answer:

Получен корректный ответ от DNS-сервера.

- ;; ->>HEADER<<- opcode: QUERY, status: NOERROR

Запрос выполнен успешно, ошибок нет. Статус NOERROR означает, что доменное имя найдено.

- qr aa rd ra

– qr – ответ является ответом сервера

– aa – authoritative answer (сервер является авторитетным для зоны)

- rd/ra – recursion desired/available (рекурсия доступна)

- QUESTION SECTION:

```
client.trseidAliев.net. IN A
```

Указано имя хоста, тип записи A – запрос IPv4-адреса.

- ANSWER SECTION:

```
client.trseidAliев.net. 1200 IN A 192.168.1.30
```

DNS-сервер сообщает, что имя client.<логин>.net связано с адресом **192.168.1.30**,

что подтверждает успешное динамическое создание записи DHCP-DDNS.

- SERVER: 192.168.1.1#53

Ответ получен от локального DNS-сервера Bind9.

- MSG SIZE rcvd: 95

Размер полученного сообщения – 95 байт.

2.5 Внесение изменений в настройки внутреннего окружения виртуальной машины

Для автоматизации развёртывания DHCP и DNS был подготовлен перенос текущих конфигураций серверных служб в директорию Vagrant.

В каталоге provisioning создана структура директорий для DHCP. В неё скопированы все актуальные файлы конфигурации DHCP-сервер.

В каталоге /vagrant/provision/server создан сценарий автоматической установки и запуска Kea DHCP/DDNS:

Фрагмент файла:

```
1  #!/bin/bash
2  echo "Provisioning script $0"
3  echo "Install needed packages"
4  dnf -y install kea
5  echo "Copy configuration files"
6  cp -R /vagrant/provision/server/dhcp/etc/kea/* /etc/kea/
7  echo "Fix permissions"
8  chown -R kea:kea /etc/kea
9  chmod 640 /etc/kea/tsig-keys.json
10 restorecon -vR /etc
11 restorecon -vR /var/lib/kea
12 echo "Configure firewall"
13 firewall-cmd --add-service dhcp
14 firewall-cmd --add-service dhcp --permanent
15 echo "Start dhcpcd service"
16 systemctl --system daemon-reload
17 systemctl enable --now kea-dhcp4.service
18 systemctl enable --now kea-dhcp-ddns.service
19 |
```

Рис. 2.20: Содержимое dhcp.sh

3 Заключение

В ходе выполнения работы:

- настроено динамическое обновление DNS-зон на основе взаимодействия Kea DHCP и Bind9 с использованием TSIG-ключей;
- сконфигурированы прямые и обратные зоны DNS с разрешением на автоматическое внесение A- и PTR-записей;
- подготовлены и подключены конфигурации Kea DHCP для работы механизма DDNS, включая файлы `tsig-keys.json` и `kea-dhcp-ddns.conf`;
- выполнена проверка работы службы DDNS и зафиксировано успешное создание динамических DNS-записей для клиента;
- подтверждена корректность функционирования DNS посредством запроса `dig`, показавшего автоматическое появление записи клиента в зоне;
- экспортированы текущие конфигурации DHCP и DNS в структуру каталога Vagrant для дальнейшей автоматизации;
- создан и настроен сценарий `dhcp.sh`, обеспечивающий автоматическое развёртывание Kea DHCP и Kea DDNS при последующих запусках проекта.

4 Контрольные вопросы

4.1 1. В каких файлах хранятся настройки сетевых подключений?

Настройки сетевых подключений хранятся в следующих файлах:

- в системах семейства RHEL/CentOS/Rocky Linux:
`/etc/sysconfig/network-scripts/ifcfg-<интерфейс>` — основной файл конфигурации каждого сетевого интерфейса;
- в NetworkManager:
`/etc/NetworkManager/system-connections/<профиль>.nmconnection` — профили сетевых подключений;
- общесистемные параметры сети:
`/etc/sysconfig/network` — базовые сетевые настройки (шлюз, hostname, маршрутизация).

Эти файлы определяют IP-адреса, маску сети, шлюз, DNS-серверы, режим работы интерфейсов и прочие параметры сети.

4.2 2. За что отвечает протокол DHCP?

DHCP (Dynamic Host Configuration Protocol) отвечает за автоматическое назначение сетевых параметров клиентам в сети.

Основные функции:

- выдача IP-адресов из пула;
- передача маски сети, шлюза по умолчанию, DNS-серверов;
- управление временем аренды IP-адреса;
- автоматизация настройки сетевых устройств без ручной конфигурации.

4.3 3. Поясните принцип работы протокола DHCP.

Какими сообщениями обмениваются клиент и сервер?

Работа DHCP основана на механизме аренды IP-адреса и проходит в четыре этапа (DORA):

1. **DHCPDISCOVER** – клиент ищет DHCP-сервер, отправляя широковещательный запрос.
2. **DHCPOFFER** – сервер предлагает клиенту свободный IP-адрес.
3. **DHCPREQUEST** – клиент выбирает один из предложенных адресов и запрашивает его назначение.
4. **DHCPCACK** – сервер подтверждает аренду и передаёт клиенту полный набор сетевых параметров.

При обновлении аренды используются сообщения:

- **DHCPREQUEST** – запрос продления,
- **DHCPCACK** – подтверждение,
- **DCHPNAK** – отказ при недоступности адреса.

4.4 4. В каких файлах обычно находятся настройки

DHCP-сервера? За что отвечает каждый из файлов?

Типичные файлы конфигурации Kea DHCP находятся в каталоге:

/etc/kea/

Основные файлы:

- **kea-dhcp4.conf** – конфигурация IPv4 DHCP-сервера (пулы адресов, опции, подсети, интерфейсы).
- **kea-dhcp-ddns.conf** – настройки динамического обновления DNS.
- **tsig-keys.json** – ключи авторизации для взаимодействия Kea с Bind9.
- **kea.conf** (если есть) – глобальная конфигурация Kea.

Файлы зоны DNS для работы DDNS расположены в:

/var/named/ – хранят A- и PTR-записи, которые обновляются DHCP-сервером.

4.5 5. Что такое DDNS? Для чего применяется DDNS?

DDNS (Dynamic DNS) – механизм автоматического обновления DNS-записей о хостах.

Используется для:

- автоматического добавления A- и PTR-записей в DNS при появлении новых клиентов DHCP;
- упрощения администрирования сети – DNS всегда содержит актуальные имена и IP-адреса;

- повышения автоматизации и уменьшения ручных действий администратора.

Kea DHCP обновляет Bind9 через защищённое TSIG-взаимодействие.

4.6 6. Какую информацию можно получить, используя утилиту ifconfig? Приведите примеры с использованием различных опций.

ifconfig позволяет получить сведения о сетевых интерфейсах:

- IP-адреса (IPv4/IPv6);
- MAC-адрес;
- маску сети;
- состояние интерфейсов (UP/DOWN);
- статистику пакетов (RX/TX);
- параметры MTU.

4.6.1 Примеры:

- ifconfig – отображает все активные интерфейсы.
- ifconfig eth0 – информация только по указанному интерфейсу.

- `ifconfig eth1 down` – выключение интерфейса.
- `ifconfig eth1 up` – включение интерфейса.

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

Утилита `ping` используется для проверки доступности узлов в сети. Она показывает:

- факт доступности хоста;
- время отклика (RTT);
- процент потерь пакетов;
- работу маршрутизации;
- качество соединения.

4.7.1 Примеры:

- `ping 192.168.1.1` – проверка доступности маршрутизатора.
- `ping -c 4 google.com` – отправка 4 пакетов и завершение.
- `ping -i 0.5 8.8.8.8` – интервал 0.5 секунды между пакетами.

- `ping -s 1500 host` – отправка пакетов увеличенного размера.