

# Отчёт о лабораторной работе

Лабораторная работа 3

Мошаров Денис Максимович

## Содержание

## Цель работы

Приобретение практических навыков по установке и конфигурированию DHCP-сервера.

## Выполнение лабораторной работы

Для начала запустим сервер через vagrant (рис. [-@fig:001]).

```
C:\work\asp\dmosharov\vagrant>vagrant up server
Bringing machine 'server' up with 'virtualbox' provider...
  => server: You assigned a static IP ending in ".1" or ":1" to this machine.
  => server: This is very often used by the router and can cause the
  => server: network to not work properly. If the network doesn't work
  => server: properly, try changing this IP.
  => server: You assigned a static IP ending in ".1" or ":1" to this machine.
  => server: This is very often used by the router and can cause the
  => server: network to not work properly. If the network doesn't work
  => server: properly, try changing this IP.
```

Запуск сервера

Зайдём под рутом и установим пакет для настройки dhcp - kea (рис. [-@fig:002]).

```
[root@server dmosharov.net ~]# dnf -y install kea
Rocky Linux 10 - BaseOS          3.1 kB/s | 4.3 kB  00:01
Rocky Linux 10 - AppStream        5.3 kB/s | 4.3 kB  00:00
Rocky Linux 10 - CRH              4.1 kB/s | 4.3 kB  00:01
Rocky Linux 10 - Extras           3.0 kB/s | 3.1 kB  00:01
Dependencies resolved.
=====
 Package           Architecture Version       Repository   Size
=====
 Installing:
  kea               x86_64      3.0.1-2.el10_1   baseos      1.4 M
 Installing dependencies:
  kea-tbs            x86_64      3.0.1-2.el10_1   baseos      3.1 M
  libtbs             x86_64      16.8-2.el10_0    baseos     255 k
  log4plus            x86_64      2.1.1-8.el10    baseos     351 k
  mariadb-connector-c x86_64      3.4.4-1.el10    baseos     206 k
  mariadb-connector-c-config  noarch    3.4.4-1.el10    baseos     8.9 k
 Transaction Summary
 =====
 Install 6 Packages

 Total download size: 5.3 M
 Installed size: 19 M
 Downloading Packages:
 (1/6): libtbs-16.8-2.el10_0.x86_64.rpm           233 kB/s | 250 kB  00:01
 (2/6): log4plus-2.1.1-8.el10.x86_64.rpm          2.3 kB/s | 3.1 MB  00:01
 (3/6): kea-3.0.1-2.el10_1.x86_64.rpm            915 kB/s | 1.4 MB  00:01
 (4/6): mariadb-connector-c-config-3.4.4-1.el10.noarch.rpm 31 kB/s | 8.9 kB  00:00
 (5/6): log4plus-2.1.1-8.el10.x86_64.rpm          380 kB/s | 351 kB  00:00
 (6/6): mariadb-connector-c-3.4.4-1.el10.x86_64.rpm 276 kB/s | 206 kB  00:00
 Total                                         2.1 MB/s | 5.3 MB  00:02
```

Установка kea

Перед изменением конфигурационного файла, на всякий случай сделаем бекап и отредактируем его (рис. [-@fig:003]).

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

Бекап конфига

Мы поменяем изначальные данные на свои - изменим доменное имя на собственное, а также поставим ip на ip нашей машины - 192.168.1.1 (рис. [-@fig:004]).

```

GNU nano 8.1                               /etc/kea/kea-dhcp4.conf      Modified
// domain-name-servers $ can do this:
[{
  {
    "name": "domain-name-servers",
    "code": 6,
    "csv-format": true,
    "space": "dhcp4",
    "data": "192.0.2.1, 192.0.2.2"
  }
]
// but it's a lot of writing, so it's easier to do this instead:
{
  "name": "domain-name-servers",
  "data": "192.168.1.1"
}.

// Typically people prefer to refer to options by their names, so they
// don't need to remember the code name. However, some people like
// to use numerical values. For example, option 'domain-name' uses
// option code 15, so you can reference to it either by
// 'name': 'domain-name' or 'code': 15.
{
  "code": 15,
  "data": "dmmosharov.net"
}.
{
  "name": "domain-search",
  "data": "dmmosharov.net"
}.

// Domain search is also a popular option. It tells the client to
// attempt to resolve names within those specified domains. For
// example, name "foo" would be attempted to be resolved as
// foo.mydomain.example.com and if it fails, then as foo.example.com
{
  "name": "domain-search",
  "data": "mydomain.example.com, example.com"
}

```

### Редактирование конфига

Спустимся ниже и настроим свою подсеть следующим образом (рис. [-@fig:005]).

```

GNU nano 8.1                               /etc/kea/kea-dhcp4.conf      Modified
// structures.
"subnet4": [
  {
    // This defines the whole subnet. Kea will use this information to
    // determine where the clients are connected. This is the whole
    // subnet in your network.

    // Subnet Identifier should be unique for each subnet.
    'id': 1,
    // This is mandatory parameter for each subnet.
    'subnet': '192.168.1.0/24',
    // Pools define the active part of your subnet that is governed
    // by Kea. Technically this is optional parameter, but it's
    // almost always needed for DHCP to do its job. If you omit it,
    // clients won't be able to get addresses, unless there are
    // host reservations defined for them.
    'pools': [ { 'pool': '192.168.1.30 - 192.168.1.199' } ],
    // This is one of the subnet selectors. Uncomment the 'interface'
    'option-data': [
      {
        // For each IPv4 subnet you most likely need to specify at
        // least one router.
        'name': 'routers',
        'data': '192.168.1.1'
      }
    ],
  }
].

```

### Настройка подсети

И установим интерфейс для dhcp как eth1 (рис. [-@fig:006]).

```

// DHCPv4 configuration starts here. This section will be read by DHCPv4 server
// and will be ignored by other components.
"DHCP4": {
  // Add names of your network interfaces to listen on.
  "interfaces-config": {
    // See section 8.2.4 for more details. You probably want to add just
    // interface name (e.g. "eth0" or specific IPv4 address on that
    // interface name (e.g. "eth0/192.0.2.1").
    "interfaces": [ "eth1" ]
  },
  // Kea DHCPv4 server by default listens using raw sockets. This ensures
  // all packets, including those sent by directly connected clients
  // that don't have IPv4 address yet, are received. However, if your
  // traffic is always relayed, it is often better to use regular
  // UDP sockets. If you want to do that, uncomment this line:
  // "dhcp-socket-type": "udp"
},

```

### Установка интерфейса

Загрузим конфиг и убедимся, что нигде нет критических ошибок (рис. [-@fig:007]).

```

2025-12-17 00:12:25.544 INFO [kea-dhcp4.dhcp4v4/32183.139784297804928] DHCP4SRV_CFMGR_SOCKET_TYPE
E_DEFAULT_ "dhcp-socket-type" not specified, using default socket type raw
2025-12-17 00:12:25.544 INFO [kea-dhcp4.dhcp4v4/32183.139784297804928] DHCP4SRVLEASE_MGR_BACKEND
S_REGISTERED the following lease backend types are available: memfile
2025-12-17 00:12:25.544 INFO [kea-dhcp4.dhcp4v4/32183.139784297804928] HOSTS_BACKENDS_REGISTERED t
he following hosts backend types are available:
2025-12-17 00:12:25.544 INFO [kea-dhcp4.dhcp4v4/32183.139784297804928] DHCP4SRV_FORENSIC_BACKENDS
_REGISTERED the following forensic backend types are available:
2025-12-17 00:12:25.544 INFO [kea-dhcp4.database/32183.139784297804928] CONFIO_BACKENDS_REGISTER
ED the following config backend types are available:

```

### Загрузка конфига

Перезагрузим системные дæмоны (рис. [-@fig:008]).

```

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

```

### Перезагрузка дæмонов

И слегка отредактируем наш файл с прошлой лабораторной работы в папке fz,

добавив запись о dhcp (рис. [-@fig:009]).

```
GNU nano 8.1          /var/named/master/fz/dmmosharov.net
$TTL 1D
@      IN SOA @ server.dmmosharov.net. (
                                2025121701 ; serial
                                1D       ; refresh
                                1H       ; retry
                                1W       ; expire
                                3H )    ; minimum

NS      @
A      192.168.1.1

$ORIGIN dmmosharov.net.
server A 192.168.1.1
ns A 192.168.1.1
dhcp A 192.168.1.1
```

## Редактирование fz

То же самое сделаем с `rz` (рис. [-@fig:010]).

```
GNU nano 8.1                               /var/named/master/rz/192.168.1
$TTL 1D
@       IN SOA @ server.dmmosharov.net. (
                                2025121701 ; serial
                                ID      ; refresh
                                IH      ; retry
                                IW      ; expire
                                3H )    ; minimum
NS      @
A       192.168.1.1
PTR    server.dmmosharov.net.
$ORIGIN 1.168.192.in-addr.arpa.
1 PTR server.dmmosharov.net.
1 PTR ns.dmmosharov.net.
1 PTR dhcp.dmmosharov.net
```

rz

Перезагрузим сервер DNS и убедимся, что мы можем пингануть dhcp сервер (рис. [-@fig:011]).

```
[root@server1 dmosharov.net ~]# systemctl restart named
[root@server1 dmosharov.net ~]# ping dhcp.dmosharov.net
PING dhcp.dmosharov.net (192.168.1.1) 56(84) bytes of data.
64 bytes from dhcp.dmosharov.net.1.168.192.in-addr.arpa (192.168.1.1): icmp_seq=1 ttl=64 time=0.042 ms
64 bytes from dhcp.dmosharov.net.1.168.192.in-addr.arpa (192.168.1.1): icmp_seq=2 ttl=64 time=0.062 ms
64 bytes from dhcp.dmosharov.net.1.168.192.in-addr.arpa (192.168.1.1): icmp_seq=3 ttl=64 time=0.059 ms
64 bytes from dhcp.dmosharov.net.1.168.192.in-addr.arpa (192.168.1.1): icmp_seq=4 ttl=64 time=0.083 ms
64 bytes from dhcp.dmosharov.net.1.168.192.in-addr.arpa (192.168.1.1): icmp_seq=5 ttl=64 time=0.059 ms
```

## Пинг dhcp

Теперь настроим firewall и обновим метки selinux (рис. [-@fig:012]).

## firewall и selinux

Убедимся по логам, что сервер DNS работает и не выдаёт ошибок (рис. [-@fig:013]).

## Логи сервера

Теперь запускаем dhcp сервер (рис. [-@fig:014]).

```
[root@server.dmmosharov.net ~]# systemctl start kea-dhcp4.service  
[root@server.dmmosharov.net ~]# █
```

## Запуск dhcp

Посмотрим на лог и убедимся, что запуск был успешен (рис. [-@fig:015]).

## Сверка по логу

Далее убедимся в том, что в нашей папке клиента в vagrant представлен скрипт следующего содержания для настройки сети, берущий свой ip по dhcp (рис. [-@fig016]).

```
$ 01-routing.sh >
C:> work_asp> dmmosharov vagrant provision > client > $ 01-routing.sh
1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5  nmcli connection modify "System eth1" ipv4.gateway "192.168.1.1"
6  nmcli connection up "System eth1"
7
8  nmcli connection modify eth0 ipv4.never-default true
9  nmcli connection modify eth0 ipv6.never-default true
10
11 nmcli connection down eth0
12 nmcli connection up eth0
13
14 # systemctl restart NetworkManager
15
```

## Скрипт для клиента

В Vagrantfile мы убедимся, что этот скрипт прописан для запуска (рис. [-@fig:017]).

```
110   client.vm.provision "client dummy",
111     type: "shell",
112     preserve_order: true,
113     path: "provision/client/01-dummy.sh"
114
115   client.vm.provision "client routing",
116     type: "shell",
117     preserve_order: true,
118     run: "always",
119     path: "provision/client/01-routing.sh"
```

## Vagrantfile

Когда приготовления завершены, мы можем запустить клиент (рис. [-@fig:018]).

```
C:\work_asp\dmmosharov\vagrant>vagrant up client --provision
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Clearing any previously set forwarded ports...
==> client: Fixed port collision for 22 => 2222. Now on port 2200.
==> client: Clearing any previously set network interfaces...
==> client: Preparing network interfaces based on configuration...
    client: Adapter 1: nat
    client: Adapter 2: intnet
==> client: Forwarding ports...
    client: 22 (guest) => 2200 (host) (adapter 1)
==> client: Running 'pre-boot' VM customizations...
==> client: Booting VM...
```

## Запуск клиента

Зайдя в клиент, через ifconfig убедимся, что айпи был получен с сервера. Это так, айпи назначился как 192.168.1.30 (рис. [-@fig:019']).

```
eth1: flags=4313UP,BROADCAST,RUNNING,MULTICAST mtu 1500  
        broadcast 192.168.1.255  
        netmask 0.0.0.0 brd 192.168.1.255  
        mac 00:0c:29:7a:0e:01 linklayer  
inet0 192.168.1.254 brd 192.168.1.255 scopeid 0x20<link>  
    ether 00:0c:29:7a:0e:01 txqueuelen 1000 (Ethernet)  
    RX packets 328 bytes 42793 (41.7 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 889 bytes 72631 (70.9 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## ifconfig

Информация о назначении айпи также хранится в файле /var/lib/kea/kea-leases4.csv на сервере (рис. [-@fig:020]).

```
[root@centos ~]# nano 0.1 /var/lib/keystone-ten-leases4.csv  
address,header,client_id,valid_lifetime,expire,subnet_id,fqdn_fed,fqdn_rev,hostname,state,user_context,pool_id  
192.168.1.20,08:00:27:9a:9e:01,01:08:00:27:9a:9e:01,3600,1765998337,1.0.0.client.,0..0
```

### Таблица с назначениями

Теперь создадим ключ sha512 и убедимся в том, что он создался (рис. [-@fig:021]).

```
[root@server.dmmosharov.net ~]# mkdir -p /etc/named/keys
[root@server.dmmosharov.net ~]# tsig-keygen -a HMAC-SHA512 DHCP_UPDATER > /
etc/named/keys/dhcp_updater.key
[root@server.dmmosharov.net ~]# cat etc/named/keys/dhcp_updater.key
cat: etc/named/keys/dhcp_updater.key: No such file or directory
[root@server.dmmosharov.net ~]# cat /etc/named/keys/dhcp_updater.key
key "DHCP_UPDATER" {
    algorithm hmac-sha512;
    secret "xPISaX6DeMwhr3j#rdGArdhLCwYynXXvyV8pAud8svkz55ld+h5eNrZQz
pmwZEvV3shfty2rKGauYNQDUoA=";
};

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

### Ключ sha512

Этот ключ добавим в /etc/named.conf (рис. [-@fig:022]).



```
GNU nano 8.1          /etc/named.conf          Modified
/*
recursion yes;

dnssec-validation no;

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

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

/* https://fedoraproject.org/wiki/Changes/CryptoPolicy */
include '/etc/crypto-policies/back-ends/bind.config';
};

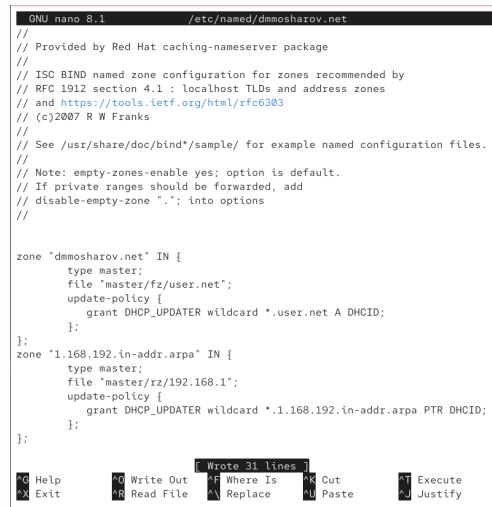
logging {
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };
};

zone "." IN {
    type hint;
    file "named.ca";
};

include '/etc/named.rfc1912.zones';
include '/etc/named.root.key';
include '/etc/named/dmmosharov.net';
include '/etc/named/keys/dhcp_updater.key';
```

### Добавление ключа

Обновим файл /etc/named/dmmosharov.net, добавив туда dhcp (рис. [-@fig:023]).



```
GNU nano 8.1          /etc/named/dmmosharov.net
// Provided by Red Hat caching-namenserver package
//
// ISC BIND named zone configuration for zones recommended by
// RFC 1912 section 4.1 : localhost TLDs and address zones
// and https://tools.ietf.org/html/rfc6303
// (c)2007 R W Franks
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
// Note: empty-zones-enable yes; option is default.
// If private ranges should be forwarded, add
// disable-empty-zone "."; into options
//

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

### Обновление файла

Проверим корректность конфига на синтаксис и перезапустим DNS службу, а также создадим файл с ключом (рис. [-@fig:024]).

```
[root@server.dmmosharov.net ~]# nano /etc/named/dmmosharov.net
[root@server.dmmosharov.net ~]# named-checkconf
[root@server.dmmosharov.net ~]# systemctl restart named
[root@server.dmmosharov.net ~]# touch /etc/kea/tsig-keys.json
[root@server.dmmosharov.net ~]#
```

## Применение изменений

В созданный файл вставим ключ, который мы сгенерировали ранее (рис. [-@fig:025]).



```
root@server:~# sudo -i
[GNU nano 8.1          /etc/kea/tsig-keys.json      Modified]
{"tsig-keys": [
    {
        "name": "DHCP_UPDATER",
        "algorithm": "hmac-sha512",
        "secret": "P1SaX6XDeWhr3jErdGArhLcYynXXvyV8pAud8svkz5Sld+hSeNrzQZqmwZEvyV3shfty2rKG6auYNs"
    }
]},
```

## Порядок монтирования

Поменяем права и владельца созданного файла, предоставив его системному пользователю службы (рис. [-@fig:026]).

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

## Смена прав файла

Теперь заполним файл конфигурации ddns, который перепишем с нуля согласно данному шаблону, поменяв имя на свой домен (рис. [-@fig:027]).



```
GNU nano 8.1          /etc/kea/kea-dhcp-ddns.conf
{
    "DhcpDdns": [
        {
            "ip-address": "127.0.0.1",
            "port": 53001,
            "control-socket": {
                "socket-type": "unix",
                "socket-name": "/run/kea/kea-ddns-ctrl-socket"
            },
            "<?include /etc/kea/tsig-keys.json?>",
            "forward-ddns": {
                "ddns-domains": [
                    {
                        "name": "dmmosharov.net.",
                        "key-name": "DHCP_UPDATER",
                        "dns-servers": [
                            { "ip-address": "192.168.1.1" }
                        ]
                    }
                ],
                "reverse-ddns": {
                    "ddns-domains": [
                        {
                            "name": "1.168.192.in-addr.arpa.",
                            "key-name": "DHCP_UPDATER",
                            "dns-servers": [
                                { "ip-address": "192.168.1.1" }
                            ]
                        }
                    ]
                }
            }
        }
    ],
    "reverse-ddns": {
        "ddns-domains": [
            {
                "name": "1.168.192.in-addr.arpa.",
                "key-name": "DHCP_UPDATER",
                "dns-servers": [
                    { "ip-address": "192.168.1.1" }
                ]
            }
        ]
    }
},
```

## Изменение конфигурации

Предоставим этот файл во владение системному пользователю, а так же загрузим эту конфигурацию, и убедимся, что она загружена успешно. После этого перезапустим ddns службу (рис. [-@fig:028]).



```
Dec 17 20:22:33 server.dmmosharov.net systemd[1]: kea-dhcp-ddns.service: Main process exited, code=exited, status=1/FAILURE
Dec 17 20:22:33 server.dmmosharov.net systemd[1]: kea-dhcp-ddns.service: Failed with result 'exit-code'.
[root@server.dmmosharov.net ~]# sed -i 's|/var/run/kea||' /etc/kea/kea-dhcp-ddns.conf
[root@server.dmmosharov.net ~]# grep socket-name /etc/kea/kea-dhcp-ddns.conf
"socket-name": "/run/kea/kea-ddns-ctrl-socket"
[root@server.dmmosharov.net ~]# systemctl restart kea-dhcp-ddns
systemctl status kea-dhcp-ddns
● kea-dhcp-ddns.service - Kea DHCP-DDNS Server
   Loaded: loaded (/usr/lib/systemd/system/kea-dhcp-ddns.service; enabled; )
   Active: active (running) since Wed 2025-12-17 20:25:02 UTC; 16ms ago
     Invocation: 8056c7f7-4244-7988-f99f-74e39962
       Docs: man/kea-dhcp-ddns(8)
      Main PID: 35308 ((hcp-ddns))
         Tasks: 1 (limit: 28334)
        Memory: 1.7M (peak: 1.7M)
         CPU: 4ms
        CGroup: /system.slice/kea-dhcp-ddns.service
                  └─35308 ((hcp-ddns))

Dec 17 20:25:02 server.dmmosharov.net systemd[1]: Started kea-dhcp-ddns.service
lines 1-13/13 (END)
```

## Перезапуск ddns

Теперь добавим информацию о ddns в наш файл с конфигурацией dhcp (рис. [-@fig:029]).

```
"renew-timer": 900,
"rebind-timer": 1800,
"valid-lifetime": 3600,
"dhcp-ddns": {
    "enable-updates": true
},
"ddns-qualifying-suffix": "dmmosharov.net",
"ddns-override-client-update": true,
"option-data": [
{
    "name": "domain-name-servers",
    "data": "192.168.1.1"
}.
```

Добавление информации о ddns

Вновь загрузим конфигурацию и перезапустим службу dhcp (рис. [-@fig:030]).

```
ns lookup is always performed first.
2025-12-17 20:29:19.996 INFO [kea-dhcp4.dhcpsrv/35874.139723378295936] DHCPSR
V_CFMOR_NEW_SUBNET a new subnet has been added to configuration: 192.168.1.0
/24 with params: tl=9000 t2=1800 valid-lifetime=3600
2025-12-17 20:29:19.996 INFO [kea-dhcp4.dhcpsrv/35874.139723378295936] DHCPSR
V_CFMOR_SOCKET_TYPE selected using socket type raw
2025-12-17 20:29:19.996 INFO [kea-dhcp4.dhcpsrv/35874.139723378295936] DHCPSR
V_CFMOR_ADD_INTERFACE listening on interface eth1
2025-12-17 20:29:19.996 INFO [kea-dhcp4.dhcpsrv/35874.139723378295936] DHCPSR
V_CFMOR_SOCKET_TYPE_DEFAULT "dhcp-socket-type" not specified , using default
socket-type raw
2025-12-17 20:29:19.996 INFO [kea-dhcp4.dhcpsrv/35874.139723378295936] DHCPSR
VLEASE_MGR_BACKENDS_REGISTERED the following lease backend types are available:
memfile
2025-12-17 20:29:19.996 INFO [kea-dhcp4.hosts/35874.139723378295936] HOSTS_BACKENDS_REGISTERED the following host backend types are available:
2025-12-17 20:29:19.996 INFO [kea-dhcp4.dhcpsrv/35874.139723378295936] DHCPSR
V_FORENSIC_BACKENDS_REGISTERED the following forensic backend types are available:
2025-12-17 20:29:19.996 INFO [kea-dhcp4.database/35874.139723378295936] CONF
O_BACKENDS_REGISTERED the following config backend types are available:
[root@server dmmosharov.net ~]# systemctl restart kea-dhcp4.service
[root@server dmmosharov.net ~]# systemctl status kea-dhcp4.service
● kea-dhcp4.service - Kea DHCPv4 Server
   Loaded: loaded (/usr/lib/systemd/system/kea-dhcp4.service; enabled; pres
   Active: active (running) since Wed 2025-12-17 20:29:31 UTC; 8s ago
  Invocation: ef17d150e87f474085a7a7ad327e0d06
     Docs: man:kea-dhcp(8)
 Main PID: 35912 (kea-dhcp4)
   Status: "Dispatching packets..."
    Tasks: 9 (limit: 22834)
   Memory: 2.5M (peak: 6.1M)
    CPU: 28ms
   CGroup: /system.slice/kea-dhcp4.service
          └─ 35912 /usr/sbin/kea-dhcp4 -c /etc/kea/kea-dhcp4.conf
```

Перезапуск службы с применением изменений

Теперь на клиенте перезапустим интернет, чтобы обновить данные (рис. [-@fig:031]).

```
[dmmosharov@client.dmmosharov.net ~]$ nmcli connection down eth1
Connection 'eth1' successfully deactivated (D-Bus active path: /org/freedesktop
NetworkManager/ActiveConnection/4)
[dmmosharov@client.dmmosharov.net ~]$ nmcli connection up eth1
Connection successfully activated (D-Bus active path: /org/freedesktop/Networ
kManager/ActiveConnection/6)
[dmmosharov@client.dmmosharov.net ~]$ █
```

Обновление данных

Теперь через dig получим информацию о нашем сервере (рис. [-@fig:032]).

```
[dmmosharov@client.dmmosharov.net ~]$ dig @192.168.1.1 client.dmmosharov.net
; <>> DIG 9.18.33 <>> @192.168.1.1 client.dmmosharov.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 59287
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 6cdae30253f866d10180000006943133846cfffe0da90db136 (good)
;; QUESTION SECTION:
client.dmmosharov.net. IN A
;; ANSWER SECTION:
client.dmmosharov.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 Dec 17 20:31:48 UTC 2025
;; MSG SIZE rcvd: 94
[dmmosharov@client.dmmosharov.net ~]$ █
```

dig

Теперь переместим данные созданных ранее конфигураций в вагрант, после чего создадим скрипт (рис. [-@fig:033]).

```
[root@server dmmosharov.net ~]# cd /vagrant/provision/server
[root@server dmmosharov.net server]# mkdir -p /vagrant/provision/server/dhcp/etc/kea
[root@server dmmosharov.net server]# cp -R /etc/kea/* /vagrant/provision/server/dhcp/etc/kea/
[root@server dmmosharov.net server]# cd /vagrant/provision/server/dns/
[root@server dmmosharov.net dns]# cp -R /var/named/* /vagrant/provision/server/dns/var/named/
cp: overwrite '/vagrant/provision/server/dns/var/named/master/fz/dmmosharov.net'?
y
cp: overwrite '/vagrant/provision/server/dns/var/named/master/rz/192.168.1.7'?
y
[root@server dmmosharov.net dns]# cp -R /etc/named/* /vagrant/provision/server/dns/etc/named/
cp: overwrite '/vagrant/provision/server/dns/etc/named/dmmosharov.net'?
y
[root@server dmmosharov.net dns]# cd /vagrant/provision/server
[root@server dmmosharov.net server]# touch dhcp.sh
[root@server dmmosharov.net server]# chmod +x dhcp.sh
[root@server dmmosharov.net server]#
```

#### Перенос конфигурации

В скрипте напишем алгоритм настройки dhcp (рис. [-@fig:034]).

```
GNU nano 8.1          dhcp.sh
#!/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 dhcpcd service"
systemctl --system daemon-reload
systemctl enable --now kea-dhcp4.service
systemctl enable --now kea-dhcp-ddns.service
```

#### Скрипт vagrant

И добавим запуск скрипта в Vagrantfile (рис. [-@fig:035]).

```
81   server.vm.provision "shell",
82     type: "shell",
83     preserve_order: true,
84     path: "provision/server/dns.sh"
85
86
87   server.vm.provision "shell",
88     type: "shell",
89     preserve_order: true,
90     path: "provision/server/dhcp.sh"
91 end
92
93
```

#### Vagrantfile

## Выводы

в результате выполнения работы были получены навыки настройки dhcp