Отчёт о лабораторной работе

Лабораторная работа 3

Андрюшин Никита Сергеевич

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Приобретение практических навыков по установке и конфигурированию DHCP-сервера.

2 Выполнение лабораторной работы

Для начала запустим сервер через vagrant (рис. 2.1).

```
C:\work\nsandryushin\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.
==> server: Clearing any previously set forwarded ports...
==> server: Clearing any previously set network interfaces...
==> server: Preparing network interfaces based on configuration...
    server: Adapter 1: nat
    server: Adapter 2: intnet
==> server: Forwarding ports...
```

Рис. 2.1: Запуск сервера

Зайдём под рутом и установим пакет для настройки dhcp - kea (рис. 2.2).

```
[nsandryushin@server.nsandryushin.net ~]$ sudo -i
[sudo] password for nsandryushin:
[root@server.nsandryushin.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 - BaseOS
Rocky Linux 10 - AppStream
Rocky Linux 10 - AppStream
Rocky Linux 10 - CRB
Rocky Linux 10 - CRB
Rocky Linux 10 - Extras
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

Transaction Summary
================================================================================
Install  6 Packages

Total download size: 5.3 M
Installed size: 19 M
Downloading Packages:
(1/6): libpq-16.8-2.el10_0.x86_64.rpm
(2/6): kea-libs-2.6.3-1.el10_0.x86_64.rpm
(3/6): mariadb-connector-c-3.4.4-1.el10.x86_64.rpm
(4/6): mariadb-connector-c-config-3.4.4-1.el10.noarch.rpm
(5/6): log4cplus-2.1.1-8.el10.x86_64.rpm
(6/6): kea-2.6.3-1.el10_0.x86_64.rpm
```

Рис. 2.2: Установка kea

Перед изменением конфигурационного файла, на всякий случай сделаем бекап и отредактируем его (рис. 2.3).

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

Рис. 2.3: Бекап конфига

Мы поменяем изначальные данные на свои - изменим доменное имя на собственное, а также поставим ip на ip нашей магины - 192.168.1.1 (рис. 2.4).

```
GNU nano 8.1                                                    /etc/kea/kea-dhcp4.conf
        // domain-name-servers you could 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 names. 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": "nsandryushin.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": "nsandryushin.net"
        },
```

Рис. 2.4: Редактирование конфига

Спустимся ниже и настроим свою подсеть следующим образом (рис. 2.5).

```
"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 actual 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" } ],

        // These are options that are subnet specific. In most cases,
        // you need to define at least routers option, as without this
        // option your clients will not be able to reach their default
        // gateway and will not have Internet connectivity.
        "option-data": [
            {
                // For each IPv4 subnet you most likely need to specify at
                // least one router.
                "name": "routers",
                "data": "192.168.1.1"
            }
        ],
```

Рис. 2.5: Настройка подсети

И установим интерфейс для dhcp как eth1 (рис. 2.6).

```
"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"
    },
```

Рис. 2.6: Установка интерфейса

Загрузим конфиг и убедимся, что нигде нет критическиз ошибок (рис. 2.7).

```
[root@server.nsandryushin.net ~]# kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
2025-09-20 12:08:22.056 INFO  [kea-dhcp4.hosts/10508.139798805010624] HOSTS_BACKENDS_REGISTERED the following host backend types are available: mysql postgresql
2025-09-20 12:08:22.057 WARN  [kea-dhcp4.dhcpsrv/10508.139798805010624] DHCPSRV_MT_DISABLED_QUEUE_CONTROL disabling dhcp queue control when multi-threading is enabled.
2025-09-20 12:08:22.057 WARN  [kea-dhcp4.dhcp4/10508.139798805010624] DHCP4_RESERVATIONS_LOOKUP_FIRST_ENABLED Multi-threading is enabled and host reservations lookup is always perfo
rmed first.
2025-09-20 12:08:22.057 INFO  [kea-dhcp4.dhcpsrv/10508.139798805010624] 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
2025-09-20 12:08:22.058 INFO  [kea-dhcp4.dhcpsrv/10508.139798805010624] DHCPSRV_CFGMGR_SOCKET_TYPE_SELECT using socket type raw
2025-09-20 12:08:22.058 INFO  [kea-dhcp4.dhcpsrv/10508.139798805010624] DHCPSRV_CFGMGR_ADD_IFACE listening on interface eth1
2025-09-20 12:08:22.058 INFO  [kea-dhcp4.dhcpsrv/10508.139798805010624] DHCPSRV_CFGMGR_SOCKET_TYPE_DEFAULT "dhcp-socket-type" not specified , using default socket type raw
```

Рис. 2.7: Загрузка конфига

Перезагрузим системные даемоны (рис. 2.8).

Рис. 2.8: Перезагрузка даемонов

И слегка отредактируем наш файл с прошлой лабораторной работы в папке fz, добавив запись о dhcp (рис. 2.9).

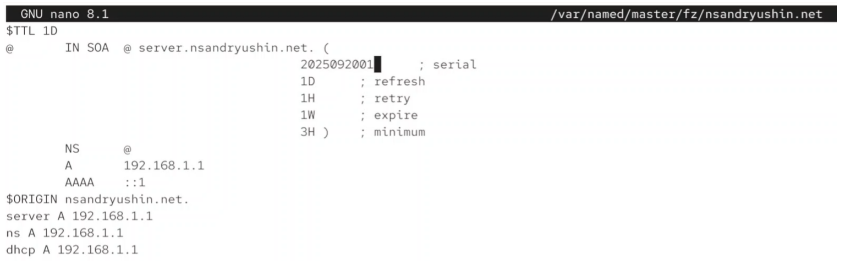

Рис. 2.9: Редактирование fz

То же самое сделаем с rz (рис. 2.10).

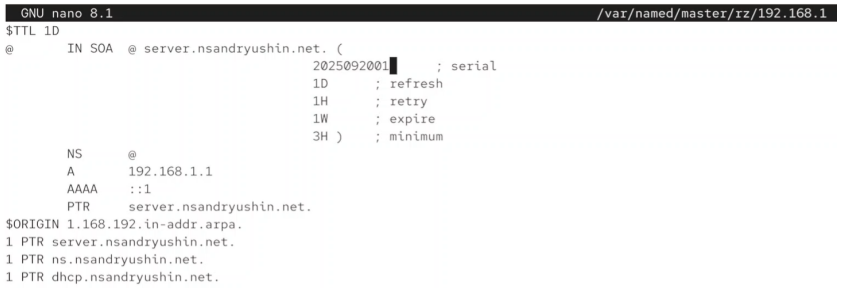

Рис. 2.10: rz

Перезагрузим сервер ДНС и убедимся, что мы можем пинганути dhcp сервер (рис. 2.11).

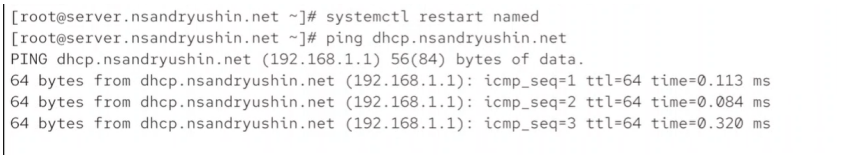

Рис. 2.11: Пинг dhcp

Теперь настроим firewall и обновим метки selinux (рис. 2.12).

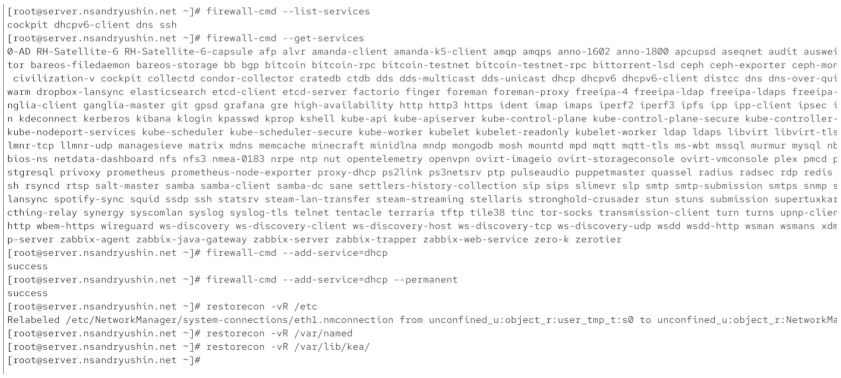

Рис. 2.12: firewall и selinux

Убедимся по логам, что сервер ДНС работает и не выдаёт ошибок (рис. 2.13).

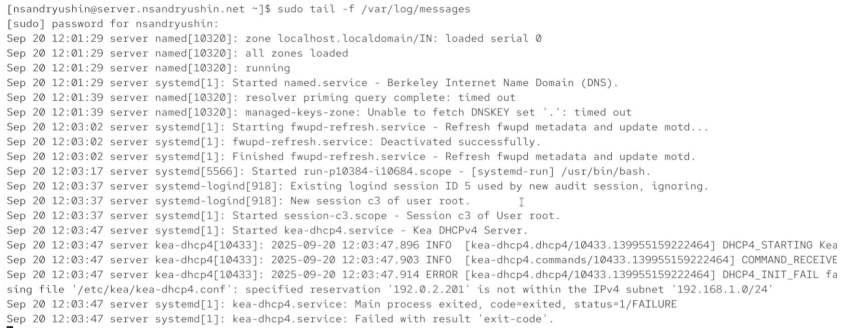

Рис. 2.13: Логи сервера

Теперь запускаем dhcp сервер (рис. 2.14).

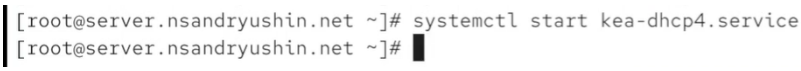

Рис. 2.14: Запуск dhcp

Посмотрим на лог и убедимся, что запуск был успешен (рис. 2.15).

Рис. 2.15: Сверка по логу

Далее убедимся в том, что в нашей папке клиента в vagrant представлен скрипт следующего содержания для настройки сети, берущей свой ip по dhcp (рис. 2.16).

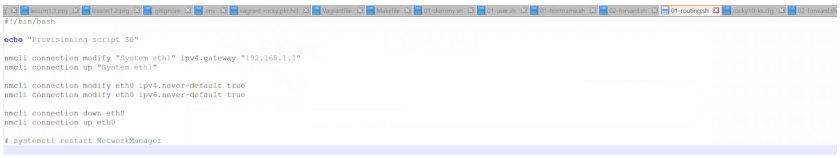

Рис. 2.16: Скрипт для клиента

В Vagrantfile мы убедимся, что этот скрипт прописан для запуска (рис. 2.17).

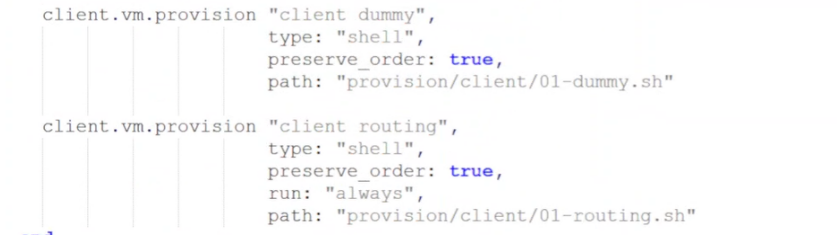

Рис. 2.17: Vagrantfile

Когда приготовления завершены, мы можем запустить клиент (рис. 2.18).

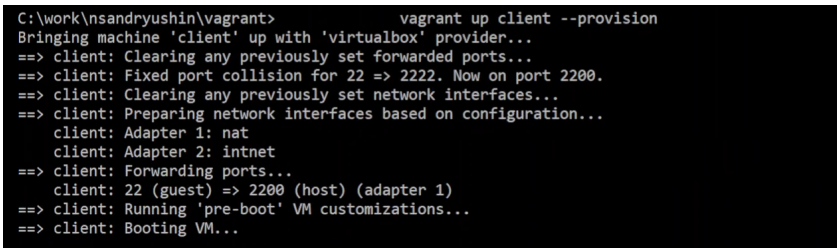

Рис. 2.18: Запуск клиента

Зайдя в клиент, через ifconfig убедимся, что айпи был получен с сервера. Это так, айпи назначился как 192.168.1.30 (рис. 2.19).

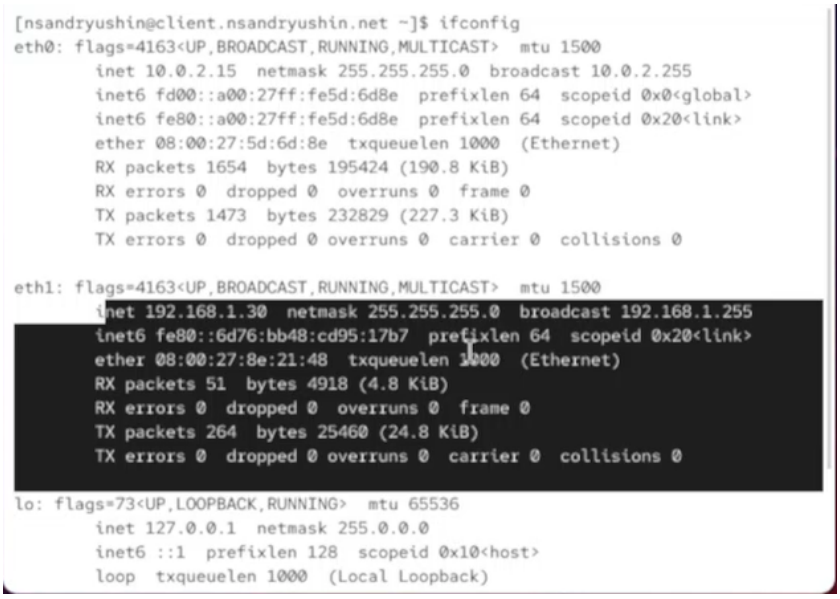

Рис. 2.19: ifconfig

Информация о назначении айпи также хранится в файле /var/lib/kea/kea-leases4.csv на сервере (рис. 2.20).

Рис. 2.20: Таблица с назначениями

Теперь создадим ключ sha512 и убедимся в том, что он создался (рис. 2.21).

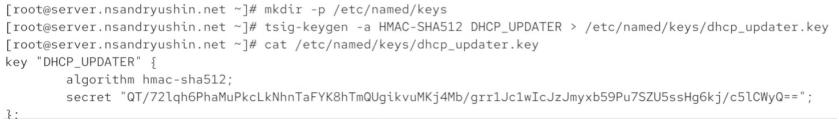

Рис. 2.21: Ключ sha512

Этот ключ добавим в /etc/named.conf (рис. 2.22).

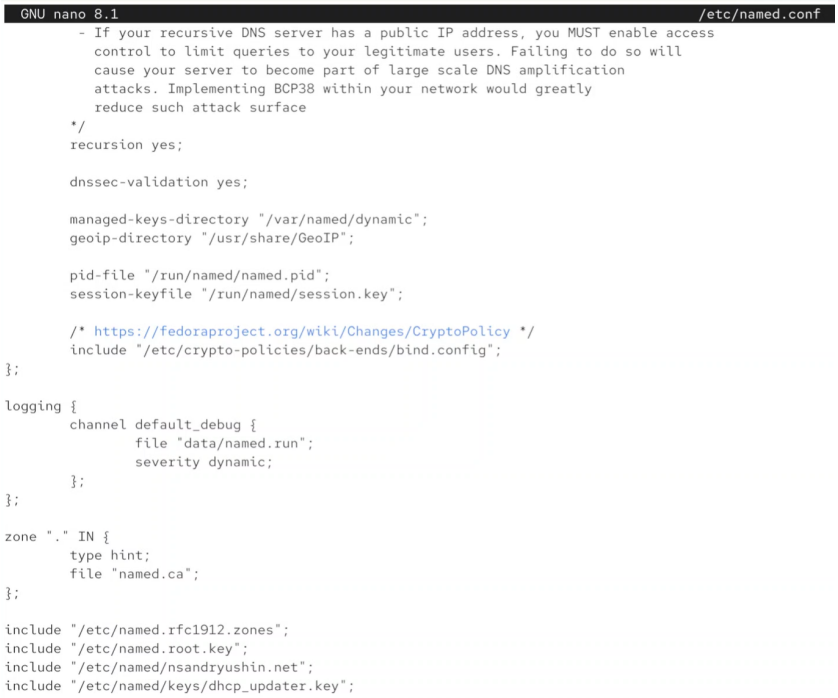

Рис. 2.22: Добавление ключа

Обновим файл /etc/named/nsandryushin.net, добавив туда dhcp (рис. 2.23).

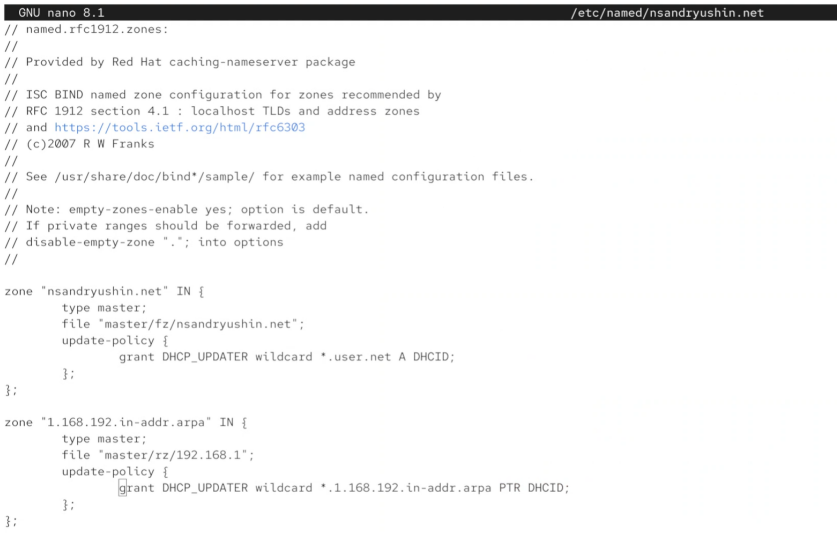

Рис. 2.23: Обновление файла

Проверим корректность конфига на синтаксис и перезапустим DNS службу, а также создадим файл с ключом (рис. 2.24).

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

Рис. 2.24: Применение изменений

В созданный файл вставим ключ, который мы сгенерировали ранее (рис. 2.25).

```
 GNU nano 8.1                                                    /etc/kea/tsig-keys.json
"tsig-keys": [
{
"name": "DHCP_UPDATER",
"algorithm": "hmac-sha512",
"secret": "QT/72lqhGPhaMuPkcLkNhnTaFYK8hTmQUgikvuMKj4Mb/grr1Jc1wIcJzJmyxb59Pu7SZU5ssHgGkj/c5lCWyQ=="
}
],
```

Рис. 2.25: Порядок монтирования

Поменяем права и владельца созданного файла, предоставив его системному пользователю службы (рис. 2.26).

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

Рис. 2.26: Смена прав файла

Теперь заполним файл конфигурации ddns, который перепишм с нуля согласно данному шаблону, поменяв имя на свой домен (рис. 2.27).

```
 GNU nano 8.1                                                    /etc/kea/kea-dhcp-ddns.conf
"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": "nsandryushin.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" }
]
}
]
},
"loggers": [
{
"name": "kea-dhcp-ddns",
"output_options": [
{
"output": "stdout",
"pattern": "%-5p %m\n"
```

Рис. 2.27: 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. 2.28).

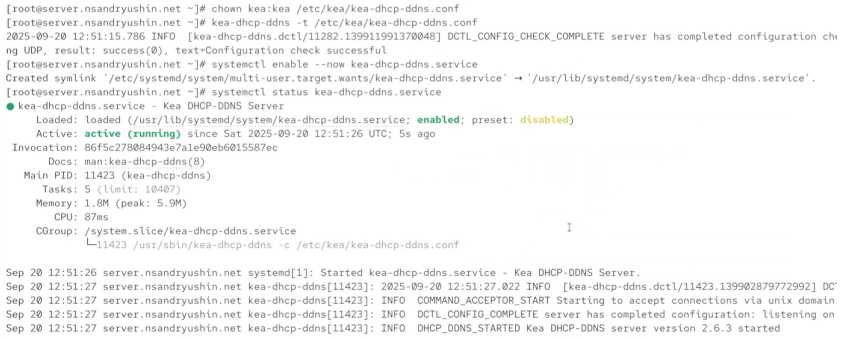

Рис. 2.28: Перезапуск ddns

Теперь добавим информацию о ddns в наш файл с конфигурацией dhcp (рис. 2.29).

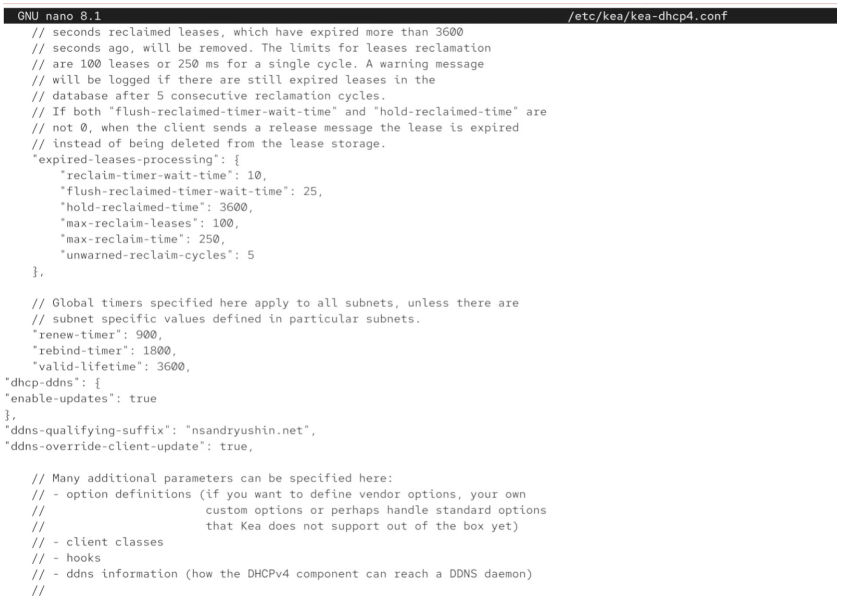

Рис. 2.29: Добавление информации о ddns

Вновь загрузим конфигурацию и перезапустим службу dhcp (рис. 2.30).

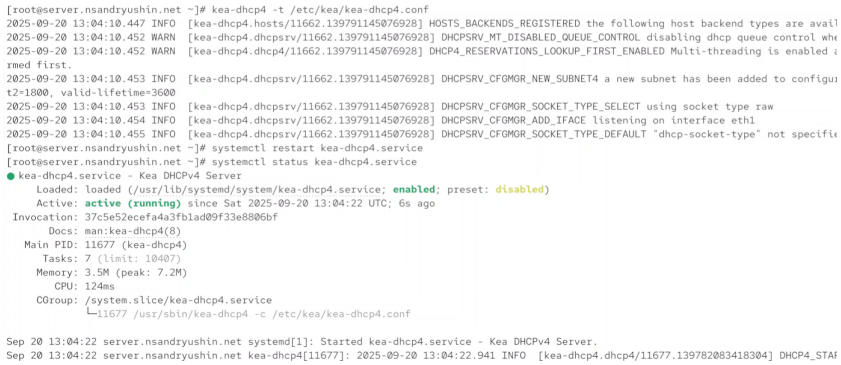

Рис. 2.30: Перезапуск службы с приминением изменений

Теперь на клиенте перезапустим интернет, чтобы обновить данные (рис. 2.31).

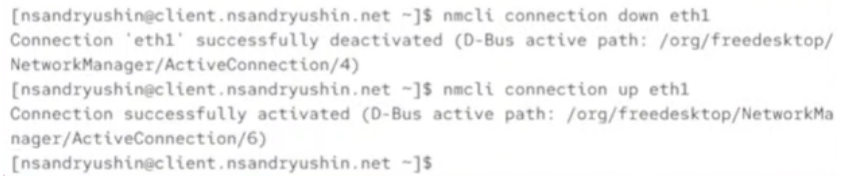

Рис. 2.31: Обновление данных

Теперь через dig получим информацию о нашем сервере (рис. 2.32).

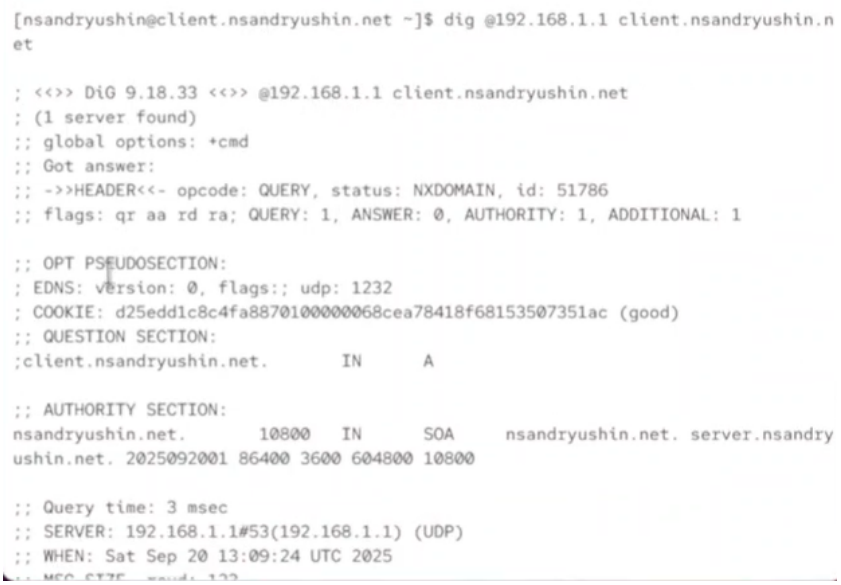

Рис. 2.32: dig

Теперь переместим данные созданных ранее конфигураций в вагрант, после

чего создадим скрипт (рис. 2.33).

```
[root@server.nsandryushin.net ~]# cd /vagrant/provision/server/
[root@server.nsandryushin.net server]# mkdir -p /vagrant/provision/server/dhcp/etc/kea
[root@server.nsandryushin.net server]# cp -R /etc/kea/* /vagrant/provision/server/dhcp/etc/kea/
[root@server.nsandryushin.net server]# cd /vagrant/provision/server/dns/
[root@server.nsandryushin.net dns]# cp -R /var/named/* /vagrant/provision/server/dns/var/named/
cp: overwrite '/vagrant/provision/server/dns/var/named/master/fz/nsandryushin.net'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/master/rz/192.168.1'? y
[root@server.nsandryushin.net dns]# cp -R /etc/named/* /vagrant/provision/server/dns/etc/named/
cp: overwrite '/vagrant/provision/server/dns/etc/named/nsandryushin.net'? y
[root@server.nsandryushin.net dns]# cd /vagrant/provision/server
[root@server.nsandryushin.net server]# touch dhcp.sh
[root@server.nsandryushin.net server]# chmod +x dhcp.sh
[root@server.nsandryushin.net server]#
```

Рис. 2.33: Перенос конфигурации

В скрипте напишем алгоритм настройки dhcp (рис. 2.34).

```
 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 dhcpd service"
systemctl --system daemon-reload
systemctl enable --now kea-dhcp4.service
systemctl enable --now kea-dhcp-ddns.service
```

Рис. 2.34: Скрипт vagrant

И добавим запуск скрипта в Vagrantfile (рис. 2.35).

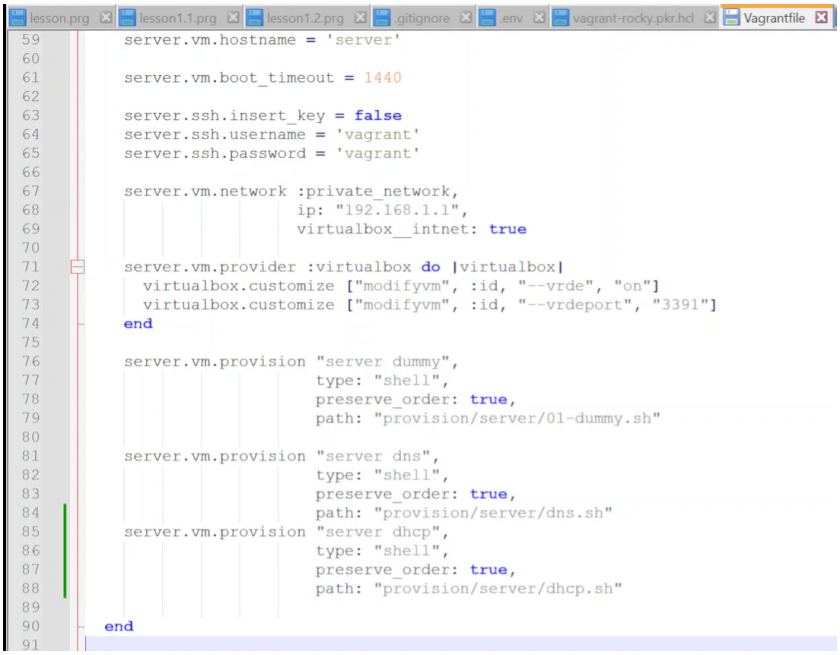

Рис. 2.35: Vagtantfile

3 Выводы

в результате выполнения работы были получены навыки настройки dhcp