

Отчёт по лабораторной работе 2

Настройка DHCP-сервера

Суннатилло Махмудов

Содержание

1	Цель работы	5
2	Теоретические сведения	6
2.1	Основные параметры DHCP	6
3	Выполнение лабораторной работы	8
3.1	Конфигурирование DHCP-сервера	8
3.2	Анализ работы DHCP-сервера	13
3.3	Настройка обновления DNS-зоны	15
3.4	Анализ работы DHCP-сервера после настройки обновления DNS-зоны	20
3.5	Внесение изменений в настройки внутреннего окружения виртуальной машины	20
4	Вывод	22
5	Контрольные вопросы	23
6	Список литературы	26

Список иллюстраций

3.1	Установка и резервное копирование конфигурации	8
3.2	Редактирование параметров domain-name и domain-search	9
3.3	Задание конфигурации подсети	10
3.4	Привязка DHCP-сервера к интерфейсу	10
3.5	Файл прямой зоны	11
3.6	Файл обратной зоны	11
3.7	Проверка доступности DHCP-сервера по имени	12
3.8	Разрешение DHCP в firewall	13
3.9	Список аренд DHCP	14
3.10	Результат работы ifconfig	14
3.11	Создание ключа TSIG	16
3.12	Разрешение обновлений зоны	17
3.13	Файл tsig-keys.json	18
3.14	Файл kea-dhcp-ddns.conf	18
3.15	Запуск kea-dhcp-ddns.service	19
3.16	Изменения в конфигурации DHCP	19
3.17	Перезапуск kea-dhcp4	20
3.18	Скрипт настройки внутреннего окружения	21

Список таблиц

1 Цель работы

Приобретение практических навыков по установке и конфигурированию DHCP-сервера.

2 Теоретические сведения

DHCP (Dynamic Host Configuration Protocol) — сетевой протокол, позволяющий автоматически выдавать клиентам IP-адреса и другие параметры для работы в сети. Он избавляет администратора от необходимости вручную настраивать каждый хост.

- **DHCP-сервер** — программа или служба, автоматически раздающая сетевые параметры клиентам.
- **DHCP-клиент** — устройство или программа, запрашивающая настройки у сервера.
- **DHCP-ретранслятор (Relay Agent)** — пересылает запросы от клиентов в другой подсети к серверу DHCP.
- **Пул адресов (scope)** — диапазон IP-адресов, из которых сервер выдает клиентам уникальные адреса.

2.1 Основные параметры DHCP

- **IP-адрес** клиента;
- **Маска подсети (subnet mask)**;

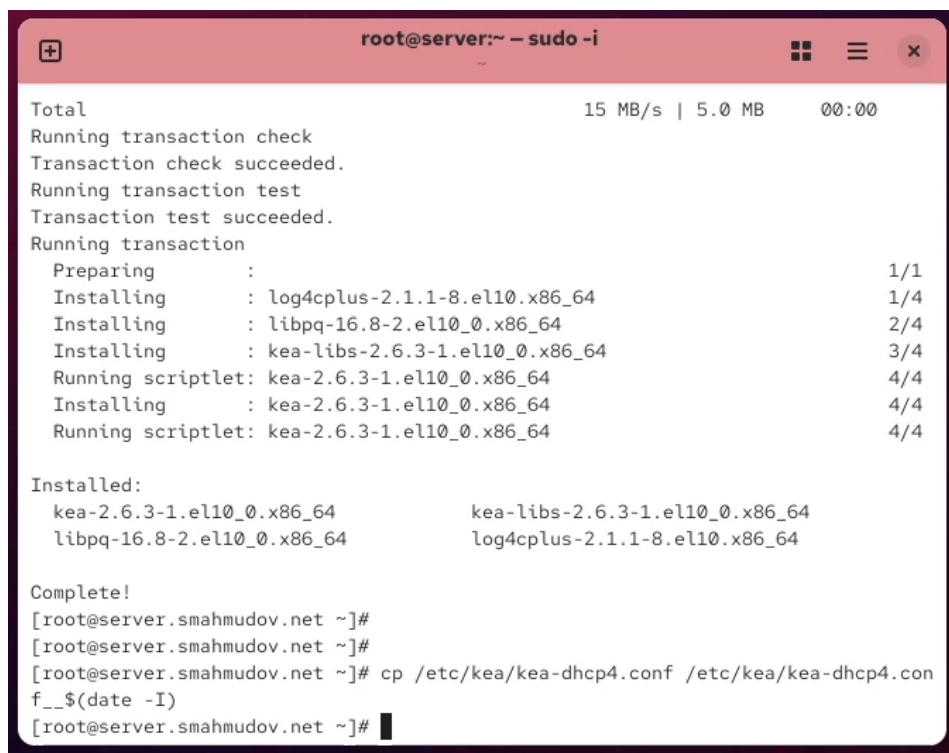
- **Основной шлюз (default gateway);**
 - **Адреса DNS-серверов;**
 - **Время аренды (lease time)** — срок действия выданного адреса;
 - **Дополнительные параметры (WINS, доменное имя, TFTP-сервер и др.).**
1. **Discover** — клиент отправляет широковещательный запрос о наличии DHCP-сервера.
 2. **Offer** — сервер предлагает свободный IP-адрес и параметры.
 3. **Request** — клиент подтверждает выбор предложенного адреса.
 4. **Acknowledge (ACK)** — сервер закрепляет адрес за клиентом и сообщает время аренды.

3 Выполнение лабораторной работы

3.1 Конфигурирование DHCP-сервера

1. На виртуальной машине **server** был установлен пакет **kea-dhcp4**, а также необходимые зависимости. Для сохранности конфигурации был выполнен бэкап файла:

```
cp /etc/kea/kea-dhcp4.conf /etc/kea/kea-dhcp4.conf_$(date -I)
```



```
root@server:~ - sudo -i
Total 15 MB/s | 5.0 MB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : 1/1
  Installing     : log4cplus-2.1.1-8.el10.x86_64 1/4
  Installing     : libpq-16.8-2.el10_0.x86_64 2/4
  Installing     : kea-libs-2.6.3-1.el10_0.x86_64 3/4
  Running scriptlet: kea-2.6.3-1.el10_0.x86_64 4/4
  Installing     : kea-2.6.3-1.el10_0.x86_64 4/4
  Running scriptlet: kea-2.6.3-1.el10_0.x86_64 4/4

Installed:
  kea-2.6.3-1.el10_0.x86_64      kea-libs-2.6.3-1.el10_0.x86_64
  libpq-16.8-2.el10_0.x86_64    log4cplus-2.1.1-8.el10.x86_64

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

Рис. 3.1: Установка и резервное копирование конфигурации

2. Файл конфигурации **/etc/kea/kea-dhcp4.conf** был изменён в соответствии

с заданием:

– Установлены параметры **domain-name** и **domain-search** с использованием домена *smahmudov.net*;

– Определён DNS-сервер:

```
{ "name": "domain-name-servers", "data": "192.168.1.1" }
```

– Настроена подсеть с идентификатором **1**:

```
"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" } ]  
}]
```

```
// }  
// 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": "smahmudov.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": "smahmudov.net"  
},  
  
// String options that have a comma in their values need to have  
// it escaped (i.e. each comma is preceded by two backslashes).  
// That's because commas are reserved for separating fields in
```

Рис. 3.2: Редактирование параметров domain-name и domain-search

```
// 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 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. 3.3: Задание конфигурации подсети

3. Для привязки работы DHCP-сервера к интерфейсу **eth1** была внесена настройка:

“interfaces-config”: { “interfaces”: [“eth1”] }

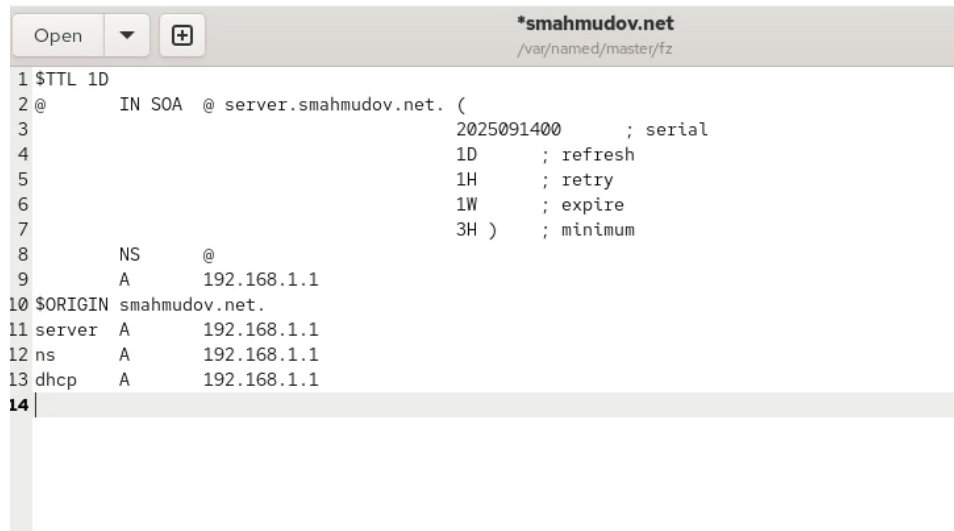
```
// DHCPv4 configuration starts here. This section will be read by DHCPv4 server
// and will be ignored by other components.
"Dhcp4": {
// Add names of your network interfaces to listen on.
"interfaces-config": {
// See section 8.2.4 for more details. You probably want to add just
// interface name (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"
},
}
```

Рис. 3.4: Привязка DHCP-сервера к интерфейсу

4. В прямой зоне DNS **/var/named/master/fz/smahmudov.net** добавлена запись для DHCP-сервера:

dhcp A 192.168.1.1

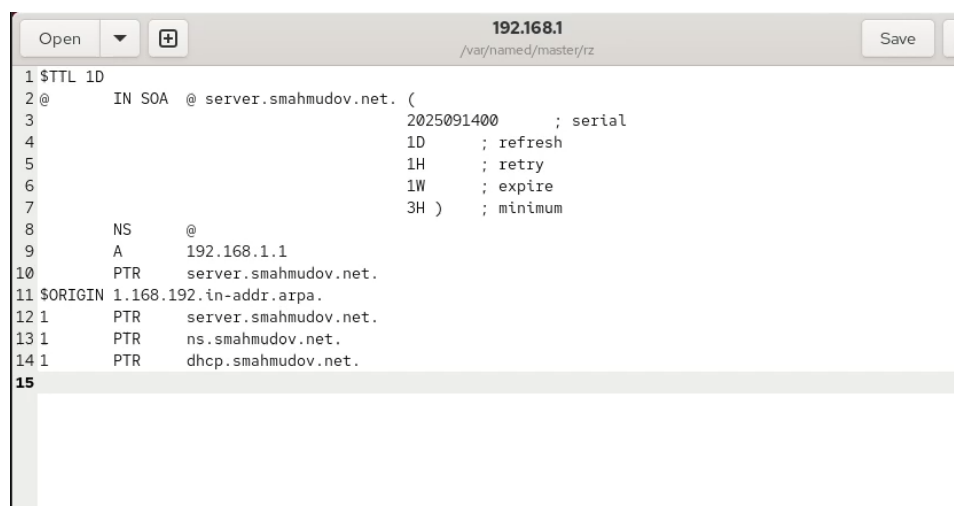


```
1 $TTL 1D
2 @      IN SOA  @ server.smahmudov.net. (
3                               2025091400      ; serial
4                               1D              ; refresh
5                               1H              ; retry
6                               1W              ; expire
7                               3H )            ; minimum
8      NS      @
9      A        192.168.1.1
10 $ORIGIN smahmudov.net.
11 server A      192.168.1.1
12 ns     A      192.168.1.1
13 dhcp   A      192.168.1.1
14
```

Рис. 3.5: Файл прямой зоны

5. В обратной зоне DNS **/var/named/master/rz/192.168.1** добавлена PTR-запись для DHCP-сервера:

1 PTR dhcp.smahmudov.net.



```
1 $TTL 1D
2 @      IN SOA  @ server.smahmudov.net. (
3                               2025091400      ; serial
4                               1D              ; refresh
5                               1H              ; retry
6                               1W              ; expire
7                               3H )            ; minimum
8      NS      @
9      A        192.168.1.1
10     PTR      server.smahmudov.net.
11 $ORIGIN 1.168.192.in-addr.arpa.
12 1        PTR      server.smahmudov.net.
13 1        PTR      ns.smahmudov.net.
14 1        PTR      dhcp.smahmudov.net.
15
```

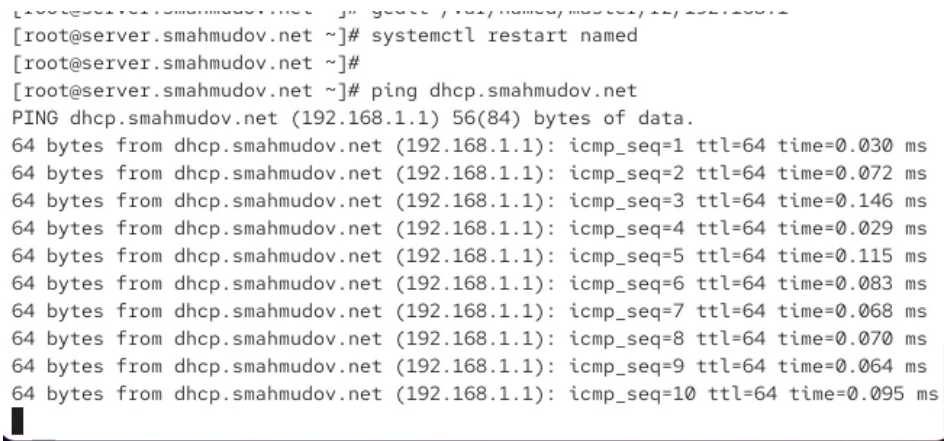
Рис. 3.6: Файл обратной зоны

6. После редактирования конфигурационных файлов был перезапущен сервис **named**:

```
systemctl restart named
```

7. Проверка доступности DHCP-сервера по имени подтвердила правильность настроек:

```
ping dhcp.smahmudov.net
```



```
[root@server.smahmudov.net ~]# systemctl restart named
[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]# ping dhcp.smahmudov.net
PING dhcp.smahmudov.net (192.168.1.1) 56(84) bytes of data.
64 bytes from dhcp.smahmudov.net (192.168.1.1): icmp_seq=1 ttl=64 time=0.030 ms
64 bytes from dhcp.smahmudov.net (192.168.1.1): icmp_seq=2 ttl=64 time=0.072 ms
64 bytes from dhcp.smahmudov.net (192.168.1.1): icmp_seq=3 ttl=64 time=0.146 ms
64 bytes from dhcp.smahmudov.net (192.168.1.1): icmp_seq=4 ttl=64 time=0.029 ms
64 bytes from dhcp.smahmudov.net (192.168.1.1): icmp_seq=5 ttl=64 time=0.115 ms
64 bytes from dhcp.smahmudov.net (192.168.1.1): icmp_seq=6 ttl=64 time=0.083 ms
64 bytes from dhcp.smahmudov.net (192.168.1.1): icmp_seq=7 ttl=64 time=0.068 ms
64 bytes from dhcp.smahmudov.net (192.168.1.1): icmp_seq=8 ttl=64 time=0.070 ms
64 bytes from dhcp.smahmudov.net (192.168.1.1): icmp_seq=9 ttl=64 time=0.064 ms
64 bytes from dhcp.smahmudov.net (192.168.1.1): icmp_seq=10 ttl=64 time=0.095 ms
```

Рис. 3.7: Проверка доступности DHCP-сервера по имени

8. Для разрешения работы DHCP-сервиса в межсетевом экране были внесены изменения:

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

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

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

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

```

[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]# firewall-cmd --add-service=dhcp
success
[root@server.smahmudov.net ~]# firewall-cmd --add-service=dhcp --permanent
success
[root@server.smahmudov.net ~]# restorecon -vR /etc
Relabeled /etc/NetworkManager/system-connections/eth1.nmconnection from unconfined_u:object_r:user_tmp_t:s0 to unconfined_u:object_r:NetworkManager_etc_rw_t:s0
[root@server.smahmudov.net ~]# restorecon -vR /var/named/
[root@server.smahmudov.net ~]# restorecon -vR /var/lib/kea/
[root@server.smahmudov.net ~]#

```

Рис. 3.8: Разрешение DHCP в firewall

9. Для восстановления контекста безопасности в **SELinux** были выполнены команды:

```

restorecon -vR /etc
restorecon -vR /var/named
restorecon -vR /var/lib/kea/

```

10. В дополнительном терминале был запущен мониторинг происходящих в системе процессов в реальном времени:

```
tail -f /var/log/messages
```

11. В основном рабочем терминале был запущен сервис **kea-dhcp4**:

```
systemctl start kea-dhcp4.service
```

12. После успешного запуска DHCP-сервера, не выключая виртуальной машины **server** и не прерывая мониторинга процессов, был выполнен переход к анализу работы DHCP-сервера на клиенте.

3.2 Анализ работы DHCP-сервера

1. Перед запуском виртуальной машины **client** был создан скрипт **01-routing.sh** в каталоге *vagrant/provision/client*. Он изменяет настройки **NetworkManager**, чтобы весь трафик клиента проходил через интерфейс **eth1**.

2. В файл **Vagrantfile** был добавлен вызов данного скрипта, после чего виртуальная машина **client** была запущена командой:

```
make client-provision
```

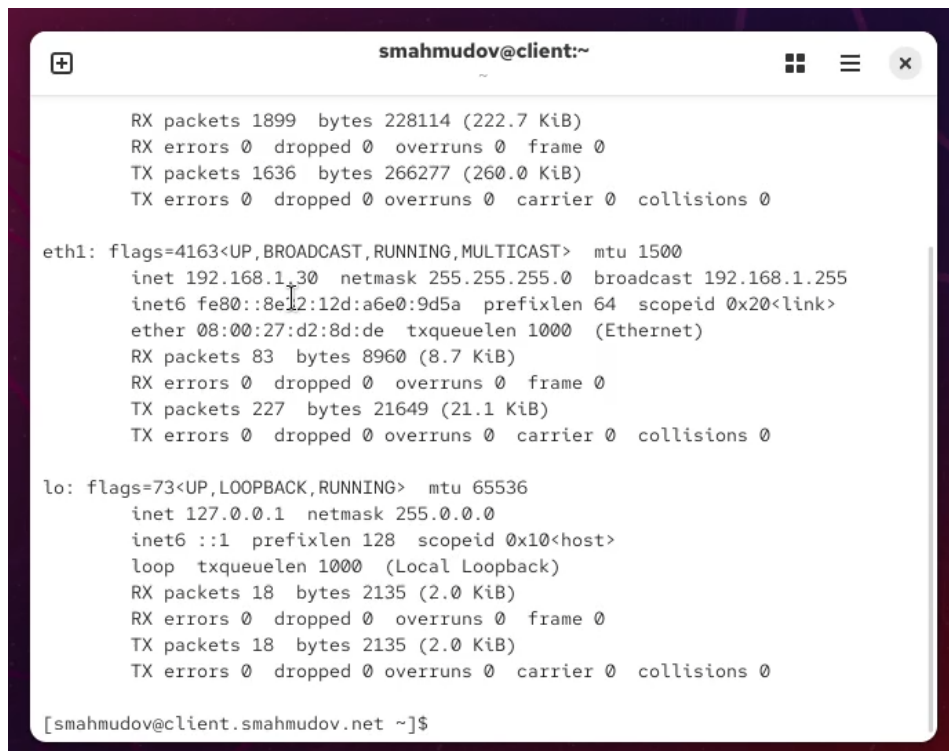
(для Windows — *vagrant up client --provision*).

3. После загрузки клиентской машины сервер **server** выдал клиенту IP-адрес из заданного диапазона. Это событие зафиксировано в логах и в файле **/var/lib/kea/kea-leases4.csv**.

```
[root@server.smahmudov.net ~]#  
[root@server.smahmudov.net ~]# systemctl start kea-dhcp4.service  
[root@server.smahmudov.net ~]#  
[root@server.smahmudov.net ~]# 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:d2:8d:de,01:08:00:27:d2:8d:de,3600,1757837446,1,0,0,client,0,,0  
192.168.1.30,08:00:27:d2:8d:de,01:08:00:27:d2:8d:de,3600,1757837446,1,0,0,client,0,,0  
192.168.1.30,08:00:27:d2:8d:de,01:08:00:27:d2:8d:de,3600,1757837452,1,0,0,client,0,,0  
[root@server.smahmudov.net ~]#
```

Рис. 3.9: Список аренд DHCP

4. На виртуальной машине **client** была выполнена команда **ifconfig**, что подтвердило получение IP-адреса **192.168.1.30** на интерфейсе **eth1**.



```
smahmudov@client:~  
RX packets 1899  bytes 228114 (222.7 KiB)  
RX errors 0  dropped 0  overruns 0  frame 0  
TX packets 1636  bytes 266277 (260.0 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::8e12:12d:a6e0:9d5a  prefixlen 64  scopeid 0x20<link>  
    ether 08:00:27:d2:8d:de  txqueuelen 1000  (Ethernet)  
    RX packets 83  bytes 8960 (8.7 KiB)  
    RX errors 0  dropped 0  overruns 0  frame 0  
    TX packets 227  bytes 21649 (21.1 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 2135 (2.0 KiB)  
    RX errors 0  dropped 0  overruns 0  frame 0  
    TX packets 18  bytes 2135 (2.0 KiB)  
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0  
  
[smahmudov@client.smahmudov.net ~]$
```

Рис. 3.10: Результат работы ifconfig

Комментарий к выводу:

- **eth1**: клиент получил адрес 192.168.1.30 с маской 255.255.255.0 и широковещательным адресом 192.168.1.255. Также отображён MAC-адрес сетевой карты клиента 08:00:27:d2:8d:de.
 - **lo**: интерфейс *loopback* имеет адрес 127.0.0.1, используется для локального взаимодействия внутри машины.
5. На сервере в файле **kea-leases4.csv** отображается информация о выданных адресах.

Пример записи:

```
192.168.1.30,08:00:27:d2:8d:de,01:08:00:27:d2:8d:de,3600,1757837446,1,0,0,client,
```

Пояснение к полям:

- 192.168.1.30 — IP-адрес, выданный клиенту.
- 08:00:27:d2:8d:de — MAC-адрес клиента.
- 01:08:00:27:d2:8d:de — client-id.
- 3600 — время аренды в секундах (1 час).
- 1757837446 — метка времени окончания аренды.
- 1 — идентификатор подсети, из которой был выдан адрес.
- client — hostname клиента.
- остальные поля зарезервированы и в текущем случае пустые.

3.3 Настройка обновления DNS-зоны

1. На сервере с **Bind9** был создан ключ для динамических обновлений:

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

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

```

16
17 zone "smahmudov.net" IN {
18     type master;
19     file "master/fz/smahmudov.net";
20     update-policy {
21         grant DHCP_UPDATER wildcard *.smahmudov.net A DHCID;
22     };
23
24 };
25
26 zone "1.168.192.in-addr.arpa" IN {
27     type master;
28     file "master/rz/192.168.1";
29     update-policy {
30         grant DHCP_UPDATER wildcard *.1.168.192.in-addr.arpa PTR DHCID;
31     };
32 };

```

Рис. 3.11: Создание ключа TSIG

2. Ключ был подключён в конфигурацию **named** через файл */etc/named.conf*.

В зоны были внесены правила обновления:

```

zone "smahmudov.net" IN {
type master;
file "master/fz/smahmudov.net";
update-policy {
grant DHCP_UPDATER wildcard *.smahmudov.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;
};
};

```



```

1 {
2   "ip-address": "127.0.0.1",
3   "port": 53001,
4   "control-socket": {
5     "socket-type": "unix",
6     "socket-name": "/run/kea/kea-ddns-ctrl-socket"
7   },
8   <?include "/etc/kea/tsig-keys.json"?>
9 }
10
11 "forward-ddns" : {
12   "ddns-domains" : [
13     {
14       "name": "smahmudov.net.",
15       "key-name": "DHCP_UPDATER",
16       "dns-servers": [
17         { "ip-address": "192.168.1.1" }
18       ]
19     }
20   ]
21 },
22
23 "reverse-ddns" : {
24   "ddns-domains" : [
25     {
26       "name": "1.168.192.in-addr.arpa.",
27       "key-name": "DHCP_UPDATER",
28       "dns-servers": [
29         { "ip-address": "192.168.1.1" }
30       ]
31     }
32   ]
33 },
34 ]

```

Рис. 3.12: Разрешение обновлений зоны

3. После проверки синтаксиса командой **named-checkconf** был перезапущен сервис **named**:

```
systemctl restart named
```

4. Для взаимодействия с Кеа DHCP был создан файл ключа **/etc/kea/tsig-keys.json**, в который был перенесён ключ в формате JSON:

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

```

[root@server.smahmudov.net ~]# gedit /etc/named/smahmudov.net
[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]# named-checkconf
[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]# systemctl restart named
[root@server.smahmudov.net ~]# touch /etc/kea/tsign-keys.json
[root@server.smahmudov.net ~]# cat /etc/named/keys/dhcp_updater.key
key "DHCP_UPDATER" {
    algorithm hmac-sha512;
    secret "9RDzHBpFC3HU7NmpqS2NUVvPBkJgRA7jKLfLkO3MgrEngLH7wLugm4X+PxVPi62kb6KzpCTdvoVWbvsgmZn9Rg==";
};
[root@server.smahmudov.net ~]# gedit /etc/kea/tsign-keys.json
[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]# mv /etc/kea/tsign-keys.json /etc/kea/tsig-keys.json
[root@server.smahmudov.net ~]# chown kea:kea /etc/kea/tsig-keys.json
[root@server.smahmudov.net ~]# chmod 640 /etc/kea/tsig-keys.json
[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]# gedit /etc/kea/kea-dhcp-ddns.conf
[root@server.smahmudov.net ~]# chown kea:kea /etc/kea/kea-dhcp-ddns.conf
[root@server.smahmudov.net ~]# 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.smahmudov.net ~]# systemctl s[REDACTED]kea-dhcp-ddns.service

```

Рис. 3.13: Файл tsig-keys.json

5. Настройка динамического обновления была выполнена в файле **/etc/kea/kea-dhcp-ddns.conf**. В нём определены forward- и reverse-зоны с указанием сервера и ключа:



Рис. 3.14: Файл kea-dhcp-ddns.conf

6. После проверки конфигурации сервис **kea-dhcp-ddns** был включён и запущен:

```
systemctl enable --now kea-dhcp-ddns.service
```

```
systemctl status kea-dhcp-ddns.service
```

```

[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]# systemctl enable --now kea-dhcp-ddns.service
[root@server.smahmudov.net ~]# 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 Sun 2025-09-14 07:33:49 UTC; 2s ago
  Invocation: 280576a8afd341cda3634a4672adc4fb
     Docs: man:kea-dhcp-ddns(8)
    Main PID: 18928 (kea-dhcp-ddns)
      Tasks: 5 (Limit: 10381)
    Memory: 1.7M (peak: 6.1M)
       CPU: 11ms
    CGroup: /system.slice/kea-dhcp-ddns.service
            └─18928 /usr/sbin/kea-dhcp-ddns -c /etc/kea/kea-dhcp-ddns.conf

Sep 14 07:33:49 server.smahmudov.net systemd[1]: Started kea-dhcp-ddns.service - Kea DHCP-DDNS Server.
Sep 14 07:33:49 server.smahmudov.net kea-dhcp-ddns[18928]: 2025-09-14 07:33:49.223 INFO [kea-dhcp-ddns.dctl/18928.140471451685184] DCTL_STARTING DhcpDdns s
Sep 14 07:33:49 server.smahmudov.net kea-dhcp-ddns[18928]: INFO COMMAND_ACCEPTOR_START Starting to accept connections via unix domain socket bound to /run/s
Sep 14 07:33:49 server.smahmudov.net kea-dhcp-ddns[18928]: INFO DCTL_CONFIG_COMPLETE server has completed configuration: listening on 127.0.0.1, port 53001
Sep 14 07:33:49 server.smahmudov.net kea-dhcp-ddns[18928]: INFO DHCP_DDNS_STARTED Kea DHCP-DDNS server version 2.6.3 started
[root@server.smahmudov.net ~]#

```

Рис. 3.15: Запуск kea-dhcp-ddns.service

7. В основной конфигурации DHCP (**/etc/kea/kea-dhcp4.conf**) была включена поддержка обновлений DNS:

```

“dhcp-ddns”: {
“enable-updates”: true
},
“ddns-qualifying-suffix”: “smahmudov.net”,
“ddns-override-client-update”: true

```

```

// and will be ignored by other components.
"Dhcp4": {
  "interfaces-config": {
    "interfaces": [ "eth1" ]
  },

  "control-socket": {
    "socket-type": "unix",
    "socket-name": "kea4-ctrl-socket"
  },

  "dhcp-ddns": {
    "enable-updates": true
  },

  "ddns-qualifying-suffix": "smahmudov.net",
  "ddns-override-client-update": true,

```

Рис. 3.16: Изменения в конфигурации DHCP

8. Сервис **kea-dhcp4** был перезапущен и проверен:

```
systemctl restart kea-dhcp4.service
```

```
systemctl status kea-dhcp4.service
```

```
socket type raw
[root@server.smahmudov.net ~]# systemctl restart kea-dhcp4.service
[root@server.smahmudov.net ~]# 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 Sun 2025-09-14 07:39:58 UTC; 6s ago
  Invocation: 23fc28443fc4687b2a6ee95c6226d14
     Docs: man:kea-dhcp4(8)
    Main PID: 19626 (kea-dhcp4)
      Tasks: 7 (limit: 10381)
     Memory: 2.5M (peak: 6.1M)
        CPU: 14ms
    CGroup: /system.slice/kea-dhcp4.service
            └─19626 /usr/sbin/kea-dhcp4 -c /etc/kea/kea-dhcp4.conf

Sep 14 07:39:58 server.smahmudov.net systemd[1]: Started kea-dhcp4.service - Kea DHCPv4 Server.
Sep 14 07:39:58 server.smahmudov.net kea-dhcp4[19626]: 2025-09-14 07:39:58.056 INFO [kea-dhcp4.dhcp4/19626.140186563156160] DHCP4_STARTING Kea DHCPv4 server
Sep 14 07:39:58 server.smahmudov.net kea-dhcp4[19626]: 2025-09-14 07:39:58.056 INFO [kea-dhcp4.commands/19626.140186563156160] COMMAND_RECEIVED Received command
lines 1-15/15 (END)
```

Рис. 3.17: Перезапуск kea-dhcp4

9. На клиентской машине было выполнено переподключение интерфейса **eth1** для обновления IP-адреса:

```
nmcli connection down eth1
```

```
nmcli connection up eth1
```

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

В результате в DNS появилась динамически добавленная запись. Проверка с помощью утилиты **dig** подтвердила корректность работы DDNS:

```
dig @192.168.1.1 client.smahmudov.net
```

![Проверка DNS-записи клиента](18.png){ #fig:018 width=80% }

3.5 Внесение изменений в настройки внутреннего окружения виртуальной машины

Для автоматизации процесса настройки DHCP и DDNS был подготовлен скрипт, который выполняет следующие действия:

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

Рис. 3.18: Скрипт настройки внутреннего окружения

4 Вывод

В ходе лабораторной работы был развернут и настроен DHCP-сервер **Kea** с привязкой к сетевому интерфейсу и интеграцией с системой имен **DNS**. Реализовано автоматическое назначение IP-адресов клиентам, настройка прямой и обратной зон, а также динамическое обновление DNS-записей через DDNS. Проверка с помощью утилит **ifconfig**, **ping** и **dig** подтвердила корректную работу DHCP и DNS-сервисов.

5 Контрольные вопросы

1. В каких файлах хранятся настройки сетевых подключений?

Настройки сетевых подключений в Linux обычно хранятся в каталоге **/etc/sysconfig/network-scripts/** (например, файлы *ifcfg-eth0*, *ifcfg-eth1*).

В современных системах с **NetworkManager** конфигурация хранится в каталоге **/etc/NetworkManager/system-connections/** в виде файлов формата *.nmconnection*.

2. За что отвечает протокол DHCP?

DHCP (Dynamic Host Configuration Protocol) — это протокол динамической конфигурации хостов, который автоматизирует назначение IP-адресов, маски подсети, шлюза и DNS-серверов клиентским устройствам в сети.

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

Принцип работы DHCP основан на механизме аренды IP-адресов. Клиент и сервер обмениваются следующими сообщениями:

- **DHCPDISCOVER** — клиент ищет сервер;
- **DHCPOFFER** — сервер предлагает адрес и параметры;
- **DHCPREQUEST** — клиент подтверждает выбор;
- **DHCPACK** — сервер окончательно закрепляет за клиентом адрес.

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

Для DHCP-сервера **Kea** используются следующие файлы:

- */etc/kea/kea-dhcp4.conf* — основная конфигурация DHCPv4 (подсети, параметры аренды, опции);
- */etc/kea/kea-dhcp-ddns.conf* — настройки динамических обновлений DNS (DDNS);
- */etc/kea/tsig-keys.json* — ключи для аутентификации при обмене с DNS-сервером;
- */var/lib/kea/kea-leases4.csv* — база данных с информацией о выданных адресах.

5. Что такое DDNS? Для чего применяется DDNS?

DDNS (Dynamic DNS) — это механизм динамического обновления DNS-записей. Применяется для того, чтобы при выдаче клиенту нового IP-адреса DHCP-сервер автоматически добавлял (или обновлял) соответствующую A- и PTR-записи в DNS-зоне. Это обеспечивает корректное разрешение имён в сети.

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

Утилита **ifconfig** позволяет просмотреть и настроить сетевые интерфейсы.

Примеры:

- **ifconfig** — отображение всех активных интерфейсов;
- **ifconfig eth1** — подробная информация об интерфейсе eth1 (IP, MAC, статистика пакетов);
- **ifconfig eth1 down / ifconfig eth1 up** — отключение и включение интерфейса.

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

Утилита **ping** проверяет доступность узлов и измеряет время отклика.

Примеры:

- **ping 192.168.1.1** — проверка доступности маршрутизатора по IP;

- `ping dhcr.smahmudov.net` — проверка доступности узла по имени (через DNS);
- `ping -c 5 8.8.8.8` — отправка фиксированного количества пакетов (5);
- `ping -s 1024 8.8.8.8` — отправка пакетов увеличенного размера (1024 байта).

6 Список литературы

1. Barr D. Common DNS Operational and Configuration Errors: RFC / RFC Editor. — 02/1996. — DOI: 10.17487/rfc1912.
2. Droms R. Dynamic Host Configuration Protocol: RFC / RFC Editor. — 03/1997. — P. 1–45. — DOI: 10.17487/rfc2131.
3. Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136: RFC / P. Vixie, S. Thomson, Y. Rekhter, J. Bound; RFC Editor. — 04/1997. — DOI: 10.17487/RFC2136.