

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

дисциплина: Администрирование сетевых подсистем

Студент: Кармацкий Н.С.

Группа: НФИбд-01-21

МОСКВА

2023 г.

Цель работы:

Приобретение практических навыков по установке и конфигурированию DHCP-сервера.

Постановка задачи

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

Выполнение работы

- 1. Установка на виртуальной машине server DHCP-сервера*
1. Запускаем виртуальную машину server. Переходим в режим суперпользователя и устанавливаем dhcp-server

```
root@server.
[root@server.nskarmatskiy.net ~]# dnf -y install dhcp-server
Rocky Linux 9 - BaseOS                2.5 kB/s | 4.1 kB    00:01
Rocky Linux 9 - AppStream              12 kB/s | 4.5 kB    00:00
Rocky Linux 9 - Extras                 6.3 kB/s | 2.9 kB    00:00
Package dhcp-server-12:4.4.2-18.b1.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@server.nskarmatskiy.net ~]#
```

Рис.1.1: Режим суперпользователя и установка dhcp-server

2. Конфигурирование DNCP-сервера

1. Скопируем файл примера конфигурации DHCP `dhcpd.conf.example` из каталога `/usr/share/doc/dhcp*` в каталог `/etc/dhcp` и переименуем его в файл с названием `dhcpd.conf`:

```
[root@server.nskarmatskiy.net ~]# cd /etc/dhcp
[root@server.nskarmatskiy.net dhcp]# cp /usr/share/doc/dhcp*/dhcpd.conf.example
/etc/dhcp
[root@server.nskarmatskiy.net dhcp]# ls
dhclient.d  dhcpd6.conf  dhcpd.conf  dhcpd.conf.example
[root@server.nskarmatskiy.net dhcp]# mv /etc/dhcp/dhcpd.conf.example /etc/dhcp/d
hcpd.conf
```

Рис.2.1: Копирование файла конфигурации

2. Откроем файл `/etc/dhcp/dhcpd.conf` на редактирование. Заменяем парус строк, раскомментируем строку `authoritative`, а так же зададим собственную конфигурацию dhcp-сети

```
# dhcpd.conf

option domain-name "nskarmatskiy.net";
option domain-name-servers ns.nskarmatskiy.net;

default-lease-time 600;
max-lease-time 7200;

# Use this to enable / disable dynamic dns updates globally.
#ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.30 192.168.1.199;
    option routers 192.168.1.1;
    option broadcast-address 192.168.1.255;
}
#This is a very basic subnet declaration.
```

Рис.2.2: Отредактированный файл dhcpd.conf

3. Настроим привязку dhcpd к интерфейсу eth1 виртуальной машины server. Для этого скопируем файл dhcpd.service из каталога /lib/systemd/system в каталог /etc/systemd/system. Откроем файл /etc/systemd/system/dhcpd.service на редактирование и заменим в нём строку. После этого перезагрузим конфигурацию dhcpd и разрешим загрузку DHCP-сервера при запуске нашей машины

```
[service]
Type=notify
EnvironmentFile=-/etc/sysconfig/dhcpd
ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid eth1
StandardError=null
```

Рис.2.3: Привязка dhcpd к интерфейсу eth1

```
[root@server.nskarmatskiy.net dhcp]# nano /etc/systemd/system/dhcpd.service
[root@server.nskarmatskiy.net dhcp]# systemctl --system daemon-reload
[root@server.nskarmatskiy.net dhcp]# systemctl enable dhcpd
[root@server.nskarmatskiy.net dhcp]#
```

Рис.2.4: Перезапуск конфигурации и разрешение на загрузку

4. Добавим запись для DHCP-сервера в конце файла прямой и обратной DNS-зоны

```
$ORIGIN .
$TTL 86400      ; 1 day
nskarmatskiy.net IN SOA  nskarmatskiy.net. server.nskarmatskiy.net. (
                        2023111406 ; serial
                        86400      ; refresh (1 day)
                        3600       ; retry (1 hour)
                        604800     ; expire (1 week)
                        10800      ; minimum (3 hours)
                        )
                        NS       nskarmatskiy.net.
                        A       192.168.1.1
$ORIGIN nskarmatskiy.net.
dhcp          A       192.168.1.1
```

Рис.2.5: Добавление записи в конец прямой зоны

```
$ORIGIN .
$TTL 86400      ; 1 day
1.168.192.in-addr.arpa IN SOA  1.168.192.in-addr.arpa. server.nskarmatskiy.net. (
                        2023111404 ; serial
                        86400      ; refresh (1 day)
                        3600       ; retry (1 hour)
                        604800     ; expire (1 week)
                        10800      ; minimum (3 hours)
                        )
                        NS       1.168.192.in-addr.arpa.
                        A       192.168.1.1
                        PTR     server.nskarmatskiy.net.
$ORIGIN 1.168.192.in-addr.arpa.
$TTL 300        ; 5 minutes
1               PTR     server.nskarmatskiy.net.
                PTR     ns.nskarmatskiy.net.
                PTR     dhcp.nskarmatskiy.net.
```

Рис. 2.6: Добавление записи в конец обратной зоны

5. Перезапустим named. Так же проверим, что можно обратиться у DHCP-серверу по имени, потом так же перезапускаем named

```
[root@server.nskarmatskiy.net dhcp]# nano /var/named/master/rz/192.168.1
[root@server.nskarmatskiy.net dhcp]# systemctl restart named
[root@server.nskarmatskiy.net dhcp]# ping dhcp.nskarmatskiy.net
PING dhcp.nskarmatskiy.net (192.168.1.1) 56(84) bytes of data.
64 bytes from www.nskarmatskiy.net (192.168.1.1): icmp_seq=1 ttl=64 time=0.041 ms
64 bytes from ns.nskarmatskiy.net (192.168.1.1): icmp_seq=2 ttl=64 time=0.209 ms
64 bytes from www.nskarmatskiy.net (192.168.1.1): icmp_seq=3 ttl=64 time=0.115 ms
64 bytes from www.nskarmatskiy.net (192.168.1.1): icmp_seq=4 ttl=64 time=0.157 ms
64 bytes from server.nskarmatskiy.net (192.168.1.1): icmp_seq=5 ttl=64 time=0.160 ms
64 bytes from ns.nskarmatskiy.net (192.168.1.1): icmp_seq=6 ttl=64 time=0.147 ms
64 bytes from server.nskarmatskiy.net (192.168.1.1): icmp_seq=7 ttl=64 time=0.163 ms
^C
--- dhcp.nskarmatskiy.net ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6062ms
rtt min/avg/max/mdev = 0.041/0.141/0.209/0.048 ms
[root@server.nskarmatskiy.net dhcp]# systemctl restart named
[root@server.nskarmatskiy.net dhcp]#
```

Рис.2.7: Перезапуск named и обращение к нашему dhcp-серверу

6. Внесем изменения в настройки межсетевого экрана узла server, разрешив работу с DHCP, а так же восстановим контекст безопасности в SELinux

```
mtp-submission smtps snmp snmpd snmpd-trap snmptrap spideroak-lansync spotify-sync squid s
sdp ssh steam-streaming svdrp svn syncthing syncthing-guis synergy syslog syslog-tls telnet t
tacle tftp tile38 tinc tor-socks transmission-client upnp-client vdsms vnc-server wbe
m-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsman wsm
ans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier
[root@server.nskarmatskiy.net dhcp]# firewall-cmd --add-service=dhcp
Warning: ALREADY_ENABLED: 'dhcp' already in 'public'
success
[root@server.nskarmatskiy.net dhcp]# firewall-cmd --add-service=dhcp --permanent
Warning: ALREADY_ENABLED: dhcp
success
[root@server.nskarmatskiy.net dhcp]# restorecon -vR /etc
Relabeled /etc/sysconfig/network-scripts/ifcfg-eth1 from unconfined_u:object_r:user_tmp_t:s0 t
o unconfined_u:object_r:net_conf_t:s0
[root@server.nskarmatskiy.net dhcp]# restorecon -vR /var/named
[root@server.nskarmatskiy.net dhcp]# restorecon -vR /var/lib/dhcpd/
[root@server.nskarmatskiy.net dhcp]#
```

Рис.2.8: Разрешаем работу с DHCP и восстанавливаем контекст безопасности

7. В дополнительном терминале запустим мониторинг происходящих в системе процессов в реальном времени, а в основном рабочем терминале запустим DHCP-сервер:

```
[nskarmatskiy@server.nskarmatskiy.net ~]$ tail -f /var/log/messages
Nov 16 10:29:17 server named[8886]: network unreachable resolving './DNSKEY/IN': 2001:500:12::
d0d#53
Nov 16 10:29:17 server named[8886]: network unreachable resolving './DNSKEY/IN': 2001:503:ba3e
::2:30#53
Nov 16 10:29:17 server named[8886]: timed out resolving './DNSKEY/IN': 127.0.0.1#53
Nov 16 10:29:17 server named[8886]: network unreachable resolving './DNSKEY/IN': 2001:7fe::53#
53
Nov 16 10:29:17 server named[8886]: network unreachable resolving './DNSKEY/IN': 2001:500:12::
d0d#53
Nov 16 10:29:17 server named[8886]: network unreachable resolving './DNSKEY/IN': 2001:503:ba3e
::2:30#53
Nov 16 10:29:17 server named[8886]: managed-keys-zone: Key 20326 for zone . is now trusted (ac
ceptance timer complete)
Nov 16 10:37:55 server firewalld[629]: WARNING: ALREADY_ENABLED: 'dhcp' already in 'public'
Nov 16 10:37:59 server firewalld[629]: WARNING: ALREADY_ENABLED: dhcp
Nov 16 10:40:55 server systemd[2485]: Started VTE child process 9037 launched by gnome-termina
l-server process 6707.
```

Рис.2.9: Мониторинг процессов

3. Анализ работы DHCP-сервера

1. Перед запуском виртуальной машины клиент создадим скрипт 01-routing.sh для настроек трафика.

Скрипт:

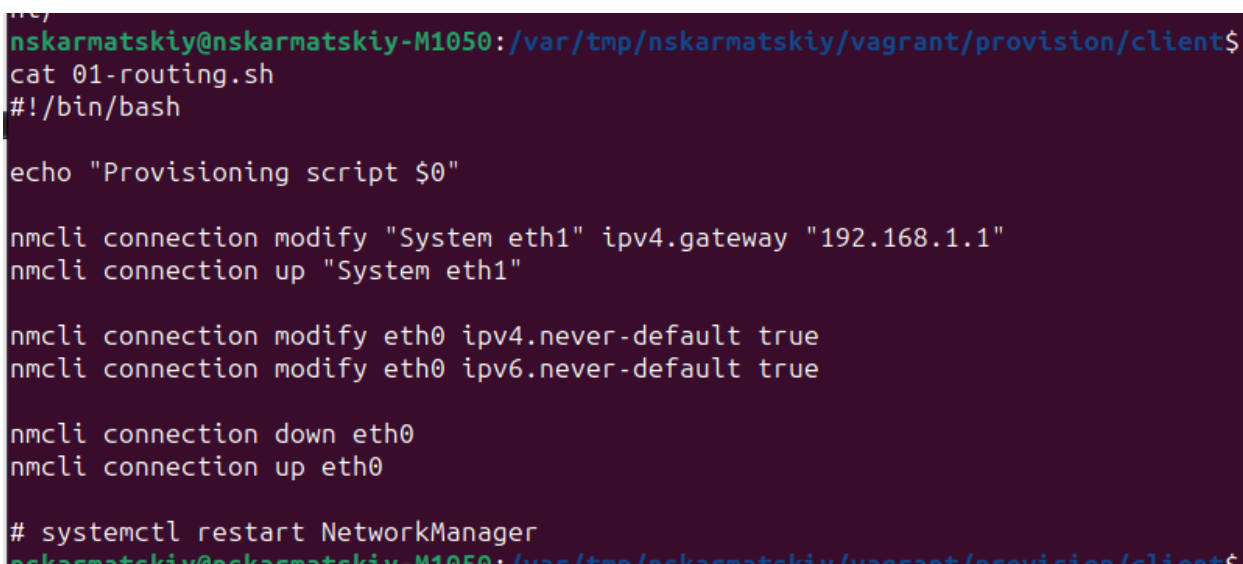
```
#!/bin/bash

echo "Provisioning script $0"

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

A screenshot of a terminal window with a dark background and light-colored text. The prompt is 'nskarmatskiy@nskarmatskiy-M1050:/var/tmp/nskarmatskiy/vagrant/provision/client\$'. The user has entered 'cat 01-routing.sh' and the terminal displays the script's content, which matches the text in the previous block. The prompt at the bottom is partially cut off.

```
nskarmatskiy@nskarmatskiy-M1050:/var/tmp/nskarmatskiy/vagrant/provision/client$
cat 01-routing.sh
#!/bin/bash

echo "Provisioning script $0"

nmcli connection modify "System eth1" ipv4.gateway "192.168.1.1"
nmcli connection up "System 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
nskarmatskiy@nskarmatskiy-M1050:/var/tmp/nskarmatskiy/vagrant/provision/client$
```

Рис.3.1: Скриптовый файл

2. В Vagrantfile подключим этот скрипт в разделе конфигурации для клиента:

```
client.vm.provision "client routing",
  type: "shell",
  preserve_order: true,
  run: "always",
  path: "provision/client/01-routing.sh"
```

```
client.vm.provision "client routing",
  type: "shell",
  preserve_order: true,
  run: "always",
  path: "provision/client/01-routing.sh"

client.vm.provider :virtualbox do |v|
```

Рис.3.2: Добавленный скрипт

3. Фиксируем внесенные изменения и запускаем client. Так же после запуска выведем на экран информацию об имеющихся интерфейсах

```
[nskarmatskiy@client.nskarmatskiy.net ~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe2f:cca5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:2f:cc:a5 txqueuelen 1000 (Ethernet)
    RX packets 1366 bytes 153886 (150.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1320 bytes 199498 (194.8 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::a00:27ff:feld:6be1 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1d:6b:e1 txqueuelen 1000 (Ethernet)
    RX packets 12 bytes 2256 (2.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 203 bytes 23718 (23.1 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 17 bytes 2045 (1.9 KiB)
```

Рис.3.3: Имеющиеся интерфейсы

Давайте разберем каждую строчку:

1. eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

- eth0 - имя сетевого интерфейса.

- flags=4163 - флаги, указывающие текущие характеристики интерфейса. В данном случае, UP (вверх), BROADCAST (поддержка широковещательных сообщений), RUNNING (интерфейс в состоянии выполнения), MULTICAST (поддержка многозадачности).

- mtu 1500 - максимальный размер передаваемого пакета в байтах.

2. inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255

- inet 10.0.2.15 - IPv4-адрес интерфейса.
- netmask 255.255.255.0 - сетевая маска.
- broadcast 10.0.2.255 - широковещательный адрес.

3. inet6 fe80::a00:27ff:fe2f:cca5 prefixlen 64 scopeid 0x20<link>

- inet6 fe80::a00:27ff:fe2f:cca5 - IPv6-адрес интерфейса.
- prefixlen 64 - префиксная длина IPv6-адреса.
- scopeid 0x20<link> - область действия адреса, в данном случае, ссылка на локальный интерфейс.

4. ether 08:00:27:2f:cc:a5 txqueuelen 1000 (Ethernet)

- ether 08:00:27:2f:cc:a5 - MAC-адрес (физический адрес) интерфейса.
- txqueuelen 1000 - длина очереди передачи (TX queue length).
- (Ethernet) - тип интерфейса.

5. RX packets 1366 bytes 153886 (150.2 KiB)

- RX packets 1366 - количество принятых пакетов.
- bytes 153886 - количество принятых байт.
- (150.2 KiB) - эквивалент в килобайтах.

6. RX errors 0 dropped 0 overruns 0 frame 0

- RX errors 0 - количество ошибок при приеме данных.
- dropped 0 - количество отброшенных пакетов.
- overruns 0 - количество переполнений приемного буфера.
- frame 0 - количество ошибок кадра.

7. TX packets 1320 bytes 199498 (194.8 KiB)

- TX packets 1320 - количество переданных пакетов.
- bytes 199498 - количество переданных байт.
- (194.8 KiB) - эквивалент в килобайтах.

8. TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

- TX errors 0 - количество ошибок при передаче данных.

- dropped 0 - количество отброшенных пакетов при передаче.
- overruns 0 - количество переполнений передающего буфера.
- carrier 0 - количество ошибок несущей.
- collisions 0 - количество коллизий в процессе передачи.

Аналогичные разборы применяются к блокам eth1 и lo, представляющим другие сетевые интерфейсы (вторичный Ethernet-интерфейс и интерфейс "loopback" соответственно).

4. Так же мы можем увидеть в мониторинге процессов на виртуальной машине server, как выдался ip-адрес VM client.

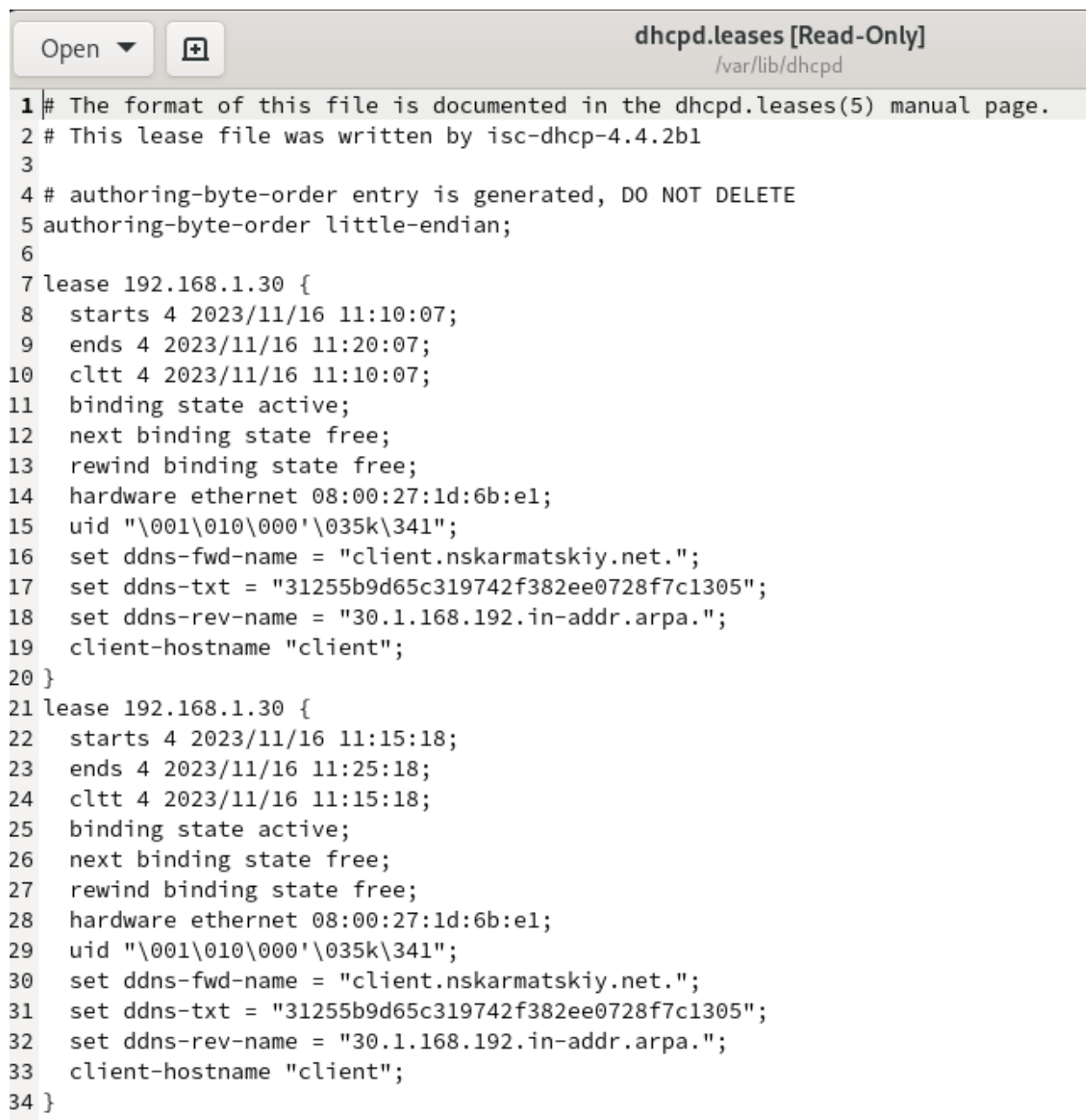
```
Nov 16 11:14:41 server named[8886]: network unreachable resolving 'anyns.pch.net/A/IN': 2001:418:3f4::5#53
Nov 16 11:14:41 server named[8886]: network unreachable resolving 'ns1.eu.bitnames.com/A/IN': 2001:500:14:6004:ad::1#53
Nov 16 11:14:41 server named[8886]: network unreachable resolving 'ns1.eu.bitnames.com/AAAA/IN': 2001:500:14:6004:ad::1#53
Nov 16 11:15:18 server dhcpd[1142]: DHCPREQUEST for 192.168.1.30 from 08:00:27:1d:6b:e1 (client) via eth1
Nov 16 11:15:18 server dhcpd[1142]: DHCPACK on 192.168.1.30 to 08:00:27:1d:6b:e1 (client) via eth1
Nov 16 11:16:03 server named[8886]: timed out resolving 'wildcard.fedoraproject.org/A/IN': 127.0.0.1#53
Nov 16 11:16:03 server named[8886]: timed out resolving 'wildcard.fedoraproject.org/AAAA/IN': 127.0.0.1#53
```

Рис.3.4: Выдача ip-адреса

Также информацию о работе DHCP-сервера можно наблюдать в файле /var/lib/dhcpd/dhcpd.leases. Проанализируем построчно весь файл.

1. authoring-byte-order little-endian;; Эта строка указывает порядок байт при записи данных. В данном случае, используется little-endian порядок.
2. lease 192.168.1.30 { ... }; Эта часть кода описывает параметры аренды IP-адреса для клиента с IP-адресом 192.168.1.30.
 - starts: Время начала аренды. - ends: Время окончания аренды.
 - cltt: Время, когда клиент в последний раз продлил аренду (client last transaction time).
 - binding state: Состояние аренды (в данном случае, "active" - активная).
 - next binding state: Следующее состояние аренды (в данном случае, "free" - свободная).
 - rewind binding state: Состояние аренды при откате (в данном случае, "free" - свободная).
 - hardware ethernet: MAC-адрес клиента. - uid: Уникальный идентификатор клиента.
 - set: Устанавливает опции для динамического обновления DNS (Dynamic DNS).
 - client-hostname: Имя хоста клиента.
3. После этого идентичный блок с новым временем начала и окончания аренды, но остальные параметры остаются теми же.

В общем, этот код определяет две аренды IP-адресов для устройства с определенным MAC-адресом и уникальным идентификатором, устанавливает опции для динамического обновления DNS и указывает имена хостов для прямого и обратного DNS.



```
1 # The format of this file is documented in the dhcpd.leases(5) manual page.
2 # This lease file was written by isc-dhcp-4.4.2b1
3
4 # authoring-byte-order entry is generated, DO NOT DELETE
5 authoring-byte-order little-endian;
6
7 lease 192.168.1.30 {
8     starts 4 2023/11/16 11:10:07;
9     ends 4 2023/11/16 11:20:07;
10    cltt 4 2023/11/16 11:10:07;
11    binding state active;
12    next binding state free;
13    rewind binding state free;
14    hardware ethernet 08:00:27:1d:6b:e1;
15    uid "\001\010\000'\035k\341";
16    set ddns-fwd-name = "client.nskarmatskiy.net.";
17    set ddns-txt = "31255b9d65c319742f382ee0728f7c1305";
18    set ddns-rev-name = "30.1.168.192.in-addr.arpa.";
19    client-hostname "client";
20 }
21 lease 192.168.1.30 {
22     starts 4 2023/11/16 11:15:18;
23     ends 4 2023/11/16 11:25:18;
24     cltt 4 2023/11/16 11:15:18;
25     binding state active;
26     next binding state free;
27     rewind binding state free;
28     hardware ethernet 08:00:27:1d:6b:e1;
29     uid "\001\010\000'\035k\341";
30     set ddns-fwd-name = "client.nskarmatskiy.net.";
31     set ddns-txt = "31255b9d65c319742f382ee0728f7c1305";
32     set ddns-rev-name = "30.1.168.192.in-addr.arpa.";
33     client-hostname "client";
34 }
```

Рис.3.5: Файл dhcpd.leases

4. Настройка обновления DNS-зоны

1. На виртуальной машине server под пользователем с правами суперпользователя отредактируем файл /etc/named/nskarmatskiy.net

```
// If private ranges should be forwarded, add
// disable-empty-zone "."; into options
//

zone "nskarmatskiy.net" IN {
    type master;
    file "master/fz/nskarmatskiy.net";
    allow-update { 127.0.0.1; };
};

zone "1.168.192.in-addr.arpa" IN {
    type master;
    file "master/rz/192.168.1";
    allow-update { 127.0.0.1; };
};
```

Рис.4.1: Обновление зоны с локального адреса

2. Перезапустим DNS-сервер. Внесем изменения в конфигурационный файл /etc/dhcp/dhcpd.conf, добавив в него разрешение на динамическое обновление DNS-записей с локального узла прямой и обратной зон. После этого снова перезапустим DHCP-сервер.

```
# Use this to enable / disable dynamic dns updates globally.
ddns-updates on;
ddns-update-style interim;
ddns-domainname "nskarmatskiy.net.";
ddns-rev-domainname "in-addr.arpa.";
zone nskarmatskiy.net. {
    primary 127.0.0.1;
}
zone 1.168.192.in-addr.arpa. {
    primary 127.0.0.1;
}
```

Рис.4.2: Изменения в конфигурационном файле DHCP

3. После успешного перезапуска в каталоге прямой DNS-зоны /var/named/master/fz должен появиться файл user.net.jnl, в котором в бинарном файле автоматически вносятся изменения записей зон

```
primary 127.0.0.1;
}
[root@server.nskarmatskiy.net ~]# cd /var/named/master/fz
[root@server.nskarmatskiy.net fz]# ls
nskarmatskiy.net  nskarmatskiy.net.jnl
[root@server.nskarmatskiy.net fz]# cat nskarmatskiy.net.jnl
;BIND LOG V9.2
x[nskarmatskiynetQ
nskarmatskiynetserver
nskarma
skiyonetx :0'client
nskarmatskiynet,
nskarmatskiynetQ
```

Рис.4.3 Бинарный файл

5. Анализ работы DHCP-сервера после настройки обновления DNS-зоны

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

```
et
; <<>> DiG 9.16.23-RH <<>> @192.168.1.1 client.nskarmatskiy.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 44120
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 1228f32bd1c72e480100000065570a0b2ee10699f7cbeaf2 (good)
;; QUESTION SECTION:
;client.nskarmatskiy.net.      IN      A

;; ANSWER SECTION:
client.nskarmatskiy.net. 300     IN      A      192.168.1.30

;; Query time: 7 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Fri Nov 17 06:37:01 UTC 2023
;; MSG SIZE rcvd: 96

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

Рис.5.1 Наличие DNS-записи

Давайте разберем каждую часть:

1. ; <<>> DiG 9.16.23-RH <<>> @192.168.1.1 client.nskarmatskiy.net: Эта строка указывает версию DiG, IP-адрес DNS-сервера (192.168.1.1) и доменное имя для запроса

(client.nskarmatskiy.net).

2. ;; global options: +cmd: Глобальные опции DiG, в данном случае, +cmd, что означает, что были использованы командные строки.

3. ;; Got answer:: Уведомление о полученном ответе от DNS-сервера.

4. ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44120: Заголовок ответа. Операция - запрос (QUERY), статус - NOERROR (запрос успешен), идентификатор запроса - 44120.

5. ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1: Флаги ответа. qr - ответ, aa - авторитетный ответ, rd - запрос с установленным битом Recursion Desired, ra - сервер поддерживает рекурсию. Далее следует статистика запроса: 1 запрос, 1 ответ, 0 авторитетных ответов, 1 дополнительная информация.

6. ;; OPT PSEUDOSECTION: ...: Дополнительная информация об опциях запроса, таких как EDNS (Extension mechanisms for DNS). Здесь указана версия EDNS, размер UDP-пакета и информация о COOKIE.

7. ;; QUESTION SECTION: ...: Секция вопроса, содержащая запрос на разрешение имени client.nskarmatskiy.net типа A (IPv4-адрес).

8. ;; ANSWER SECTION: ...: Секция ответа, содержащая полученный ответ на запрос. В данном случае, получен IP-адрес для client.nskarmatskiy.net: 192.168.1.30, с TTL (временем жизни записи) равным 300 секундам.

9. ;; Query time: 7 msec: Время выполнения запроса - 7 миллисекунд.

10. ;; SERVER: 192.168.1.1#53(192.168.1.1): Информация о DNS-сервере, который обработал запрос. IP-адрес DNS-сервера и номер порта.

11. ;; WHEN: Fri Nov 17 06:37:01 UTC 2023: Время выполнения запроса.

12. ;; MSG SIZE rcvd: 96: Размер полученного сообщения - 96 байт.

6. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, и создав каталоги dhcp/etc/dhcp и /hcp/etc/systemd/system скопируем файлы конфигурации DHCP-сервера. Так же в папке vagrant/provision/server создадим исполняемый скрипт, который будет автоматически запускать DHCP-сервер и запускать настройки конфигурации его.

```
[root@server.nskarmatskiy.net server]# cat dhcp.sh
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
dnf -y install dhcp-server

echo "Copy configuration files"
cp -R /vagrant/provision/server/dhcp/etc/* /etc

chown -R dhcpd:dhcpd /etc/dhcp

restorecon -vR /etc
restorecon -vR /var/lib/dhcpd

echo "Configure firewall"
firewall-cmd --add-service=dhcp
firewall-cmd --add-service=dhcp --permanent

echo "Start dhcpd service"
systemctl --system daemon-reload
systemctl enable dhcpd
systemctl start dhcpd
[root@server.nskarmatskiy.net server]#
```

Рис.6.1: Исполняемый скрипт dhcp.sh

2. Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile добавим в разделе конфигурации для сервера: server.vm.provision "server dhcp",

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

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

Рис.6.2. Дополнение конфигурации

Вывод: Мы приобрели практические навыки по установке и конфигурированию DHCP-сервера.

Контрольные вопросы

1. Настройки сетевых подключений:

- В большинстве Linux-систем информация о сетевых подключениях хранится в текстовых файлах в директории /etc/network/ или /etc/sysconfig/network-scripts/. Например, файлы interfaces в Debian-подобных системах или ifcfg-<interface> в Red Hat-подобных системах содержат конфигурацию сетевых интерфейсов.

- В Windows файлы реестра, такие как HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters, хранят информацию о сетевых настройках.

2. Протокол DHCP (Dynamic Host Configuration Protocol):

- Отвечает за динамическое предоставление IP-адресов и других сетевых настроек клиентам в локальной сети.

3. Принцип работы DHCP:

- Когда клиент включается в сеть, он отправляет DHCP Discover (обнаружение) на широковещательный адрес.

- DHCP-сервер в сети получает запрос и отправляет DHCP Offer (предложение) с доступными параметрами.

- Клиент принимает предложение и отправляет DHCP Request (запрос) для подтверждения.

- DHCP-сервер отправляет DHCP Acknowledgment (подтверждение), предоставляя клиенту запрошенные параметры.

4. Настройки DHCP-сервера:

- В Linux-системах, файлы конфигурации DHCP-сервера обычно находятся в /etc/dhcp/. Например, файл dhcpd.conf содержит основные настройки, а файлы типа dhcpd.leases хранят информацию о выданных адресах.

5. DDNS (Dynamic Domain Name System):

- DDNS представляет собой расширение DNS, позволяющее автоматически обновлять записи DNS при изменении IP-адресов устройств в сети.

6. Утилита ifconfig:

- ifconfig используется для отображения и настройки сетевых интерфейсов. Примеры:
- ifconfig: Показывает все активные интерфейсы и их настройки.
- ifconfig eth0: Показывает информацию о конкретном интерфейсе eth0.
- ifconfig -a: Показывает все интерфейсы, включая выключенные.

7. Утилита ping:

- ping используется для проверки доступности узла в сети. Примеры:
- ping google.com: Пингует домен google.com.
- ping -c 4 8.8.8.8: Пингует IP-адрес 8.8.8.8 с отправкой 4 пакетов.
- ping -t 10 google.com: Пингует google.com, отправляя пакеты с таймаутом 10 секунд.