

Отчёт по лабораторной работе №3

Администрирование сетевых систем

Ищенко Ирина НПИбд-02-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	24
4	Ответы на контрольные вопросы	25

Список иллюстраций

2.1	Установка DHCP	6
2.2	Копирование файла примера конфигурации и переименование . .	7
2.3	/etc/dhcp/dhcpd.conf	7
2.4	Копирование файла dhcpd.service	7
2.5	Редактирование файла /etc/systemd/system/dhcpd.service	8
2.6	Перезагрузка конфигурации и автозагрузка DHCP-сервера	8
2.7	Редактирование файла прямой DNS-зоны	9
2.8	Редактирование файла обратной DNS-зоны	9
2.9	Перезагрузка DNS-сервера и пинг	10
2.10	Внесение изменений в настройки межсетевого экрана, восстано- вление контекста безопасности	10
2.11	Внесение изменений в настройки межсетевого экрана, восстано- вление контекста безопасности	10
2.12	Запуск DHCP-сервера	11
2.13	01-routing.sh	11
2.14	Vagrantfile	11
2.15	Запись о подключении к ВМ узла client и выдачи ему IP-адреса . .	12
2.16	Просмотр файла /var/lib/dhcpd/dhcpd.leases	12
2.17	ifconfig на ВМ client	14
2.18	Редактирование файла /etc/named/ioithenko.net	17
2.19	Редактирование файла /etc/dhcp/dhcpd.conf	18
2.20	Успешный перезапуск DHCP-сервера	18
2.21	Проверка DNS-записи о клиенте в прямой DNS-зоне	19
2.22	Заполнение DHCP	22
2.23	Создание скрипта dhcp.sh	22
2.24	Vagrantfile	23

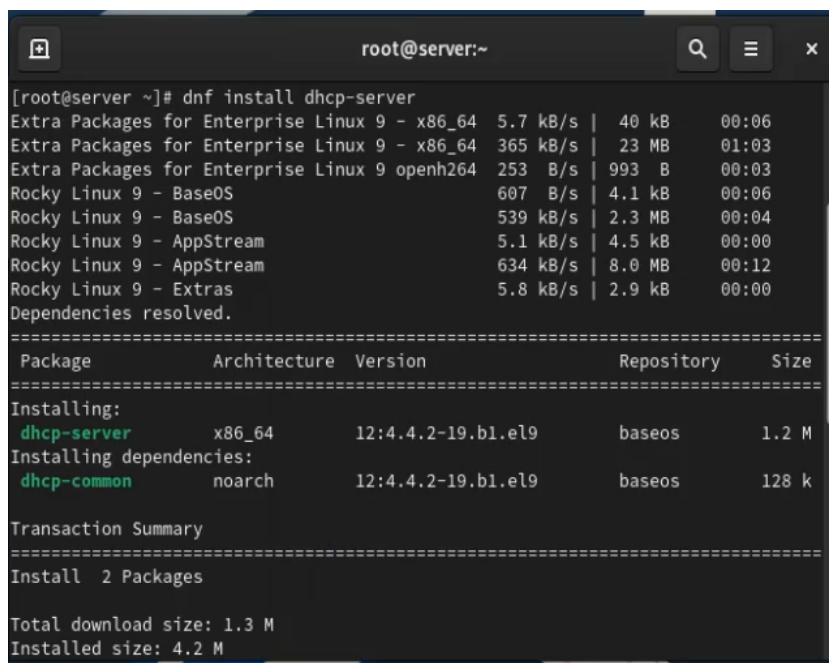
Список таблиц

1 Цель работы

Приобретение практических навыков по установке и конфигурированию DHCP-сервера.

2 Выполнение лабораторной работы

Запускаем ВМ через рабочий каталог. На ВМ server входим под моим пользователем и переходим в режим суперпользователя. Устанавливаем dhcp(рис. 2.1).



```
root@server:~  
[root@server ~]# dnf install dhcp-server  
Extra Packages for Enterprise Linux 9 - x86_64 5.7 kB/s | 40 kB 00:06  
Extra Packages for Enterprise Linux 9 - x86_64 365 kB/s | 23 MB 01:03  
Extra Packages for Enterprise Linux 9 openh264 253 B/s | 993 B 00:03  
Rocky Linux 9 - BaseOS 607 B/s | 4.1 kB 00:06  
Rocky Linux 9 - BaseOS 539 kB/s | 2.3 MB 00:04  
Rocky Linux 9 - AppStream 5.1 kB/s | 4.5 kB 00:00  
Rocky Linux 9 - AppStream 634 kB/s | 8.0 MB 00:12  
Rocky Linux 9 - Extras 5.8 kB/s | 2.9 kB 00:00  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
Installing:				
dhcp-server	x86_64	12:4.4.2-19.b1.el9	baseos	1.2 M
Installing dependencies:				
dhcp-common	noarch	12:4.4.2-19.b1.el9	baseos	128 k

```
Transaction Summary  
=====
```

Transaction Summary	
Install	2 Packages

```
Total download size: 1.3 M  
Installed size: 4.2 M
```

Рис. 2.1: Установка DHCP

Копируем файл примера конфигурации DHCP `dhcpcd.conf.example` из каталога `/usr/share/doc/dhcp*` в каталог `/etc/dhcp` и переименовываем его в файл с названием `dhcpcd.conf` (рис. 2.2).

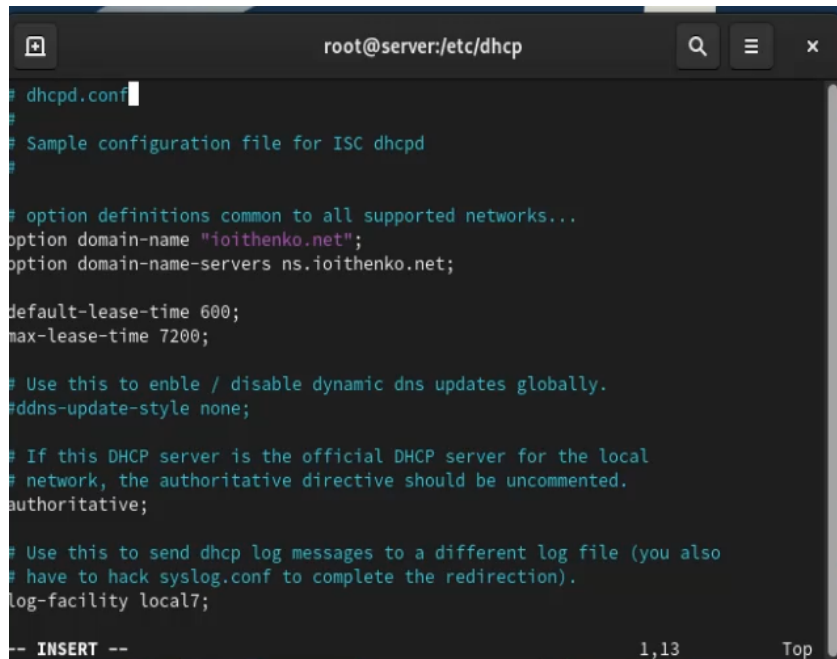
```

[root@server ~]# cd /etc/dhcp
[root@server dhcp]# cp /usr/share/doc/dhcp*/dhcp.conf.example /etc/dhcp
cp: cannot stat '/usr/share/doc/dhcp*/dhcp.conf.example': No such file or directory
[root@server dhcp]# cp /usr/share/doc/dhcp*/dhcpd.conf.example /etc/dhcp
[root@server dhcp]# mv /etc/dhcp/dhcpd.conf.example /etc/dhcp/dhcpd.conf

```

Рис. 2.2: Копирование файла примера конфигурации и переименование

Редактируем файл `/etc/dhcp/dhcpd.conf` (рис. 2.3)



```

# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# option definitions common to all supported networks...
option domain-name "ioithenko.net";
option domain-name-servers ns.ioithenko.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;

-- INSERT --
1,13
Top

```

Рис. 2.3: `/etc/dhcp/dhcpd.conf`

Настраиваем привязку `dhcpd` к интерфейсу `eth1` виртуальной машины `server`. Вводим (рис. 2.4):

`cp /lib/systemd/system/dhcpd.service /etc/systemd/system/`

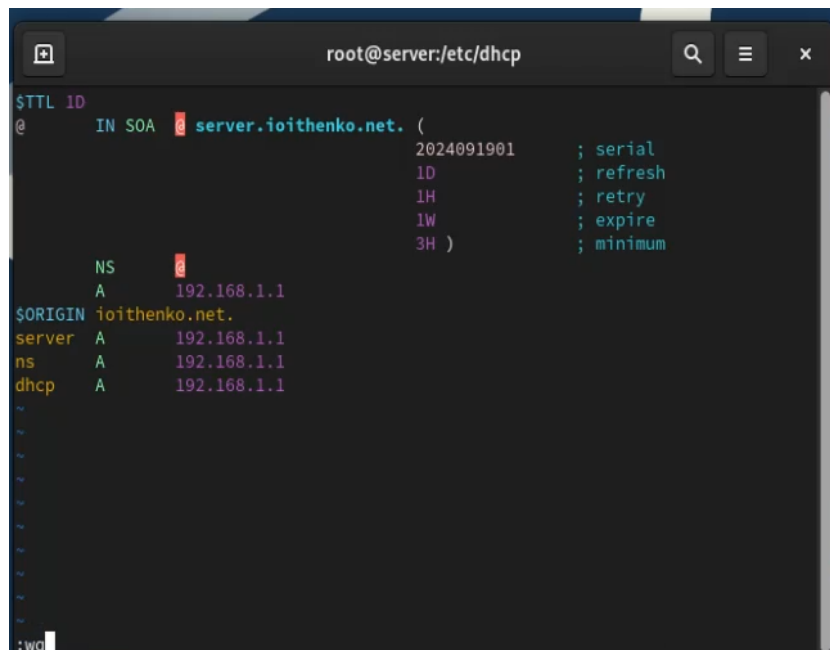
```

[root@server dhcp]# vim dhcpd.conf
[root@server dhcp]# cp /lib/systemd/system/dhcpd.service /etc/systemd/system/
[root@server dhcp]# vim /etc/systemd/system/dhcpd.service

```

Рис. 2.4: Копирование файла `dhcpd.service`

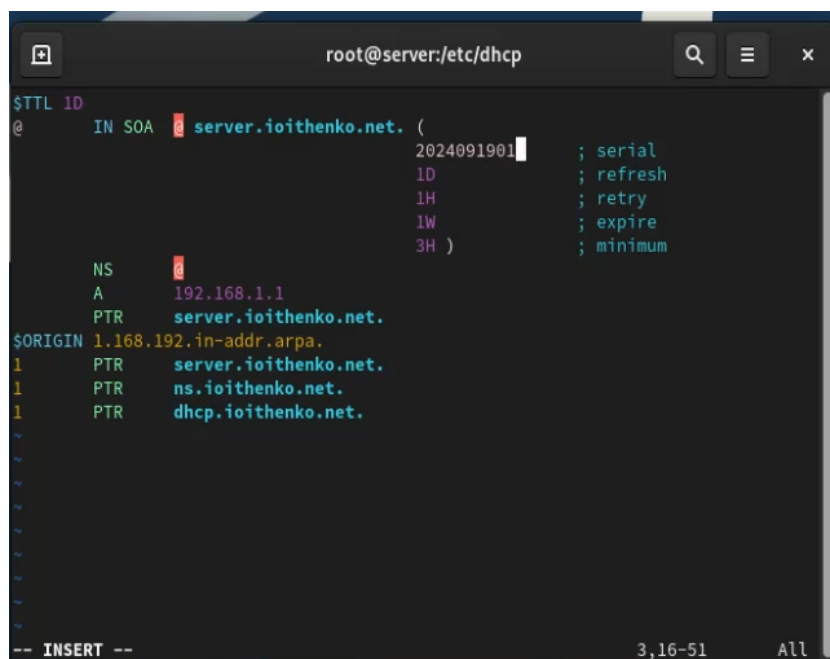
и редактируем файл `/etc/systemd/system/dhcpd.service` (рис. 2.5)



```
root@server:/etc/dhcp
$TTL 1D
@      IN SOA  server.ioithenko.net. (
                                2024091901 ; serial
                                1D          ; refresh
                                1H          ; retry
                                1W          ; expire
                                3H )        ; minimum

      NS   server.ioithenko.net.
      A    192.168.1.1
$ORIGIN ioithenko.net.
server A    192.168.1.1
ns     A    192.168.1.1
dhcp   A    192.168.1.1
```

Рис. 2.7: Редактирование файла прямой DNS-зоны



```
root@server:/etc/dhcp
$TTL 1D
@      IN SOA  server.ioithenko.net. (
                                2024091901 ; serial
                                1D          ; refresh
                                1H          ; retry
                                1W          ; expire
                                3H )        ; minimum

      NS   server.ioithenko.net.
      A    192.168.1.1
      PTR  server.ioithenko.net.
$ORIGIN 1.168.192.in-addr.arpa.
1      PTR  server.ioithenko.net.
1      PTR  ns.ioithenko.net.
1      PTR  dhcp.ioithenko.net.
```

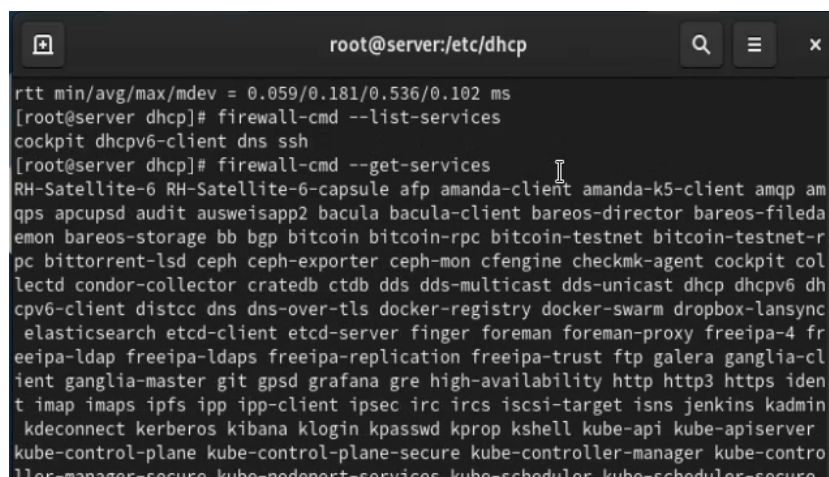
Рис. 2.8: Редактирование файла обратной DNS-зоны

Перезапускаем named и обращаемся к DHCP-серверу по имени (рис. 2.9).

```
[root@server dhcp]# systemctl restart named
[root@server dhcp]# ping dhcp.iothenko.net
ping: dhcp.iothenko.net: Name or service not known
[root@server dhcp]# ping dhcp.iothenko.net
PING dhcp.iothenko.net (192.168.1.1) 56(84) bytes of data.
64 bytes from ns.ioithenko.net (192.168.1.1): icmp_seq=1 ttl=64 time=0.059 ms
```

Рис. 2.9: Перезагрузка DNS-сервера и пинг

Вносим изменения в настройки межсетевого экрана узла server, разрешив работу с DHCP. Восстанавливаем контекст безопасности SELinux (рис. 2.10), (рис. 2.11)



```
rtt min/avg/max/mdev = 0.059/0.181/0.536/0.102 ms
[root@server dhcp]# firewall-cmd --list-services
cockpit dhcpv6-client dns ssh
[root@server dhcp]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 bacula bacula-client bareos-director bareos-filedemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gpsd grafana gre high-availability http http3 https idn imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-secure
```

Рис. 2.10: Внесение изменений в настройки межсетевого экрана, восстановление контекста безопасности



```
[root@server dhcp]# firewall-cmd --add-service=dhcp
success
[root@server dhcp]# firewall-cmd --add-service=dhcp --permanent
success
[root@server dhcp]# restorecon -vR /etc
Relabeled /etc/systemd/system/dhcpd.service from unconfined_u:object_r:systemd_unit_file_t:s0 to unconfined_u:object_r:dhcpd_unit_file_t:s0
Relabeled /etc/sysconfig/network-scripts/ifcfg-eth1 from unconfined_u:object_r:unconfined_u:object_r:net_conf_t:s0 to unconfined_u:object_r:net_conf_t:s0
[root@server dhcp]# restorecon -vR /var/named
[root@server dhcp]# restorecon -vR /var/lib/dhcpd
restorecon: lstat(/var/lib/dhcpd) failed: No such file or directory
[root@server dhcp]# restorecon -vR /var/lib/dhcpd
```

Рис. 2.11: Внесение изменений в настройки межсетевого экрана, восстановление контекста безопасности

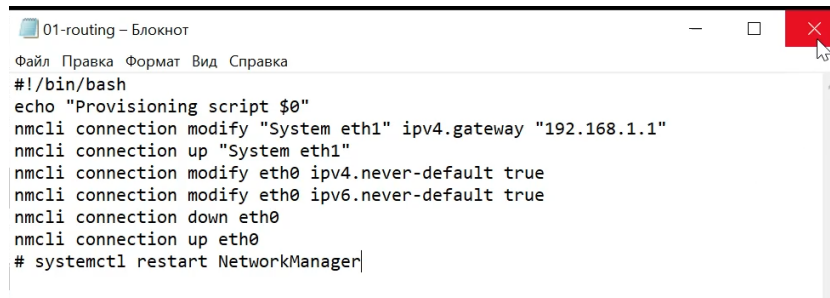
В дополнительном терминале запускаем мониторинг происходящих в системе процессов в реальном времени.

В основном терминале запускаем DHCP-сервер (рис. 2.12)

```
[root@server dhcp]# systemctl start dhcpd
[root@server dhcp]#
```

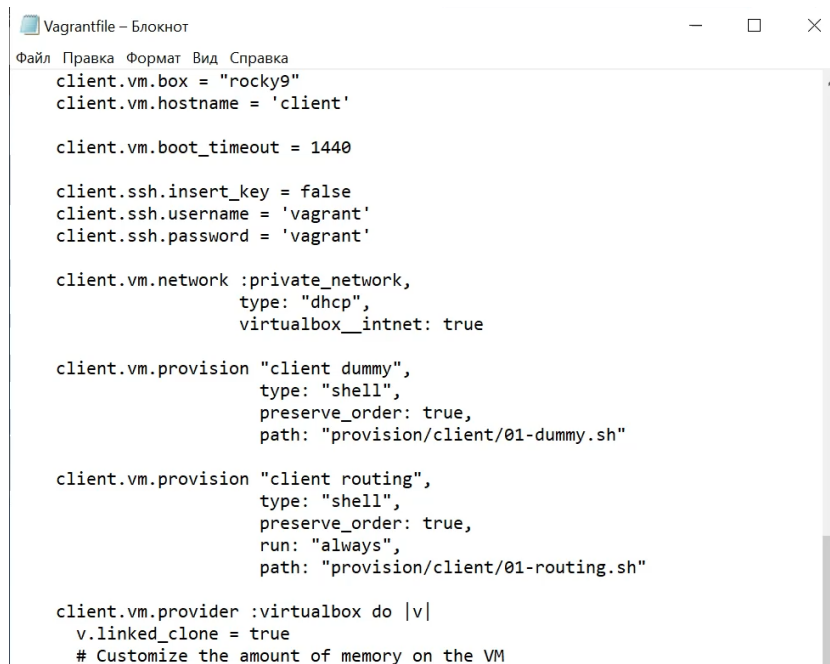
Рис. 2.12: Запуск DHCP-сервера

Проверяем файл 01-routing.sh в подкаталоге vagrant/provision/client (рис. 2.13). В Vagrantfile проверяем, что скрипт подключен (рис. 2.14).



```
#!/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
```

Рис. 2.13: 01-routing.sh



```
client.vm.box = "rocky9"
client.vm.hostname = 'client'

client.vm.boot_timeout = 1440

client.ssh.insert_key = false
client.ssh.username = 'vagrant'
client.ssh.password = 'vagrant'

client.vm.network :private_network,
                  type: "dhcp",
                  virtualbox____intnet: true

client.vm.provision "client dummy",
                    type: "shell",
                    preserve_order: true,
                    path: "provision/client/01-dummy.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|
  v.linked_clone = true
  # Customize the amount of memory on the VM
```

Рис. 2.14: Vagrantfile

Включаем VM client. На server вижу запись о подключении к VM узла client и выдачи ему IP-адреса из соответствующего диапазона адресов (рис. 2.15).

```

nyp.org/AAAA/IN : 2407:b9c0:0002:302:5054:f1:f6d:8c01#53
Sep 19 16:43:18 server dhcpd[7396]: DHCPREQUEST for 192.168.1.30 from 08:00:27:8d:1f:f7 (client) via eth1
Sep 19 16:43:18 server dhcpd[7396]: DHCPACK on 192.168.1.30 to 08:00:27:8d:1f:f7 (client) via eth1

```

Рис. 2.15: Запись о подключении к ВМ узла client и выдачи ему IP-адреса

Также просматриваем файл /var/lib/dhcpd/dhcpd.leases (рис. 2.16)

```

[root@server dhcp]# cat /var/lib/dhcpd/dhcpd.leases
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.4.2b1

# authoring-byte-order entry is generated, DO NOT DELETE
authoring-byte-order little-endian;

server-uid "\000\001\000\001.\177\017\271\010\000'\341\222\255";

lease 192.168.1.30 {
    starts 4 2024/09/19 16:42:22;
    ends 4 2024/09/19 16:52:22;
    cltt 4 2024/09/19 16:42:22;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 08:00:27:8d:1f:f7;
    uid "\001\010\000'\215\037\367";
    client-hostname "client";
}
[root@server dhcp]#

```

Рис. 2.16: Просмотр файла /var/lib/dhcpd/dhcpd.leases

Содержание файла:

1 author ing-byte-order Little-endian; - Указывает порядок байтов для записи данных. “Little-endian” означает, что младший байт хранится первым.

2. server-uid “\000\001\000\001.\177\017\271\010\000'\341\222\255”;

- Это уникальный идентификатор сервера (DUID), используемый для идентификации DHCP-сервера. Он представлен в виде строки байтов.

3. lease 192.168.1.30 {

- Начало блока аренды IP-адреса 192.168.1.30, который будет предоставлен клиенту.

4. starts 4 2024/09/19 16:42:22;

- Указывает время начала аренды (дата и время).

5. ends 4 2024/09/19 16:52:22;

- Указывает время окончания аренды.

6. cltt 4 2024/09/19 16:42:

- Время последнего запроса аренды (Client Last Transaction Time).

7. binding state active;

- Указывает, что текущее состояние привязки IP-адреса активно.

8. next binding state free;

- Указывает, что следующее состояние привязки будет свободным, когда аренда истечет.

9. rewind binding state free;

- Указывает на состояние возврата привязки, которое также свободно.

10. hardware ethernet 08:00:27:8d:1f:f7;

- Указывает аппаратный адрес (MAC-адрес) клиента, которому предоставлен этот IP-адрес.

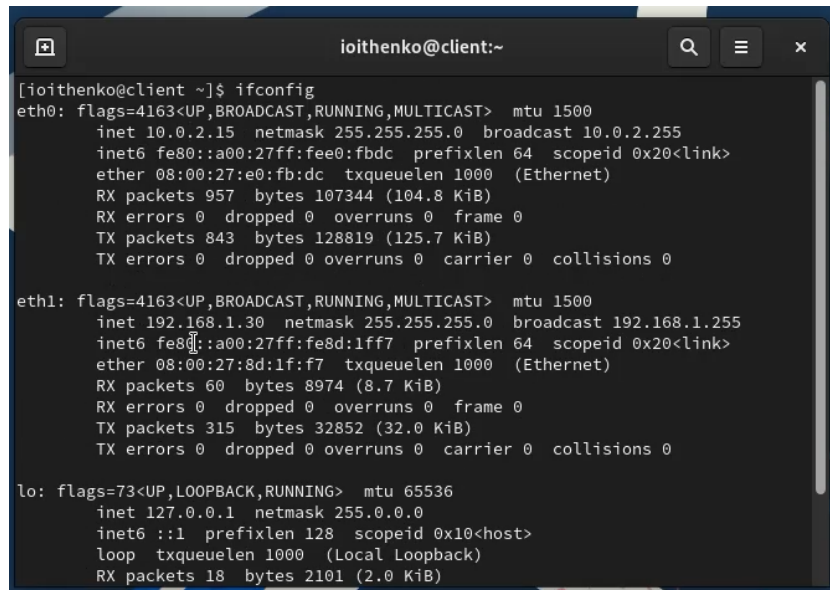
11. uid "\001\010\000'\215\637\367";

- Уникальный идентификатор клиента (UID), который может использоваться для идентификации устройства.

12. client-hostname "client";

- Имя хоста клиента, который запрашивает аренду IP-адреса.

На VM client вводим ifconfig и просматриваем имеющиеся интерфейсы (рис. 2.17)



```
[ioithenko@client ~]$ 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:fe0:fbdc prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e0:fb:dc txqueuelen 1000 (Ethernet)
    RX packets 957 bytes 107344 (104.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 843 bytes 128819 (125.7 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:fe8d:1ff7 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:8d:1f:f7 txqueuelen 1000 (Ethernet)
    RX packets 60 bytes 8974 (8.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 315 bytes 32852 (32.0 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 2101 (2.0 KiB)
```

Рис. 2.17: ifconfig на VM client

Содержимое ifconfig:

eth0:

1. eth0: flags=4163<UP BROADCAST, RUNNING, MULTICAST> mtu 1500
 - Интерфейс eth0 активен (UP), поддерживает широковещательную рассылку (BROADCAST), работает (RUNNING) и поддерживает многокаст (MULTICAST). MTU (Maximum Transmission Unit) равен 1500 байтам.
2. inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
 - IPv4-адрес интерфейса: 10.0.2.15. Маска подсети: 255.255.255.0, что означает, что сеть имеет 256 адресов. Широковещательный адрес сети: 10.0.2.255.
3. inet6 fe80::a00:27ff:feed:fbde prefixlen 64 scopeid 0x20
 - IPv6-адрес: fe80::a00:27ff:feed:fbde. Префикс длиной 64 бита, область действия — локальная сеть.
4. ether 68:00:27:e0:fb:de txqueuelen 1000 (Ethernet)

- MAC-адрес интерфейса: 68:00:27:e0:fb:de. Длина очереди передачи (txqueuelen) составляет 1000 пакетов.
5. RX packets 957 bytes 107344 (104.8 KiB)
 - Получено 957 пакетов, общий объем данных — 107344 байта (104.8 KiB).
 6. RX errors 0 dropped 0 overruns 6 frame 6
 - Ошибки при получении: 0 потерянных пакетов, 0 переполнений, 6 ошибок фрейма.
 7. TX packets 843 bytes 128819 (125.7 KiB)
 - Отправлено 843 пакета, общий объем данных — 128819 байт (125.7 KiB).
 8. TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
 - Ошибки при передаче: 0 потерянных пакетов, 0 переполнений, 0 ошибок связи, 0 коллизий.

eth1:

1. eth1: flags=4163<UP BROADCAST, RUNNING, MULTICAST> mtu 1500
 - Интерфейс eth1 также активен и имеет те же характеристики, что и eth0.
2. inet 192.168.1.30 netmask 255.255.255.0 broadcast 192.168.1.255
 - IPv4-адрес интерфейса: 192.168.1.30. Маска подсети и широковещательный адрес аналогичны.
3. inet6 fe80::a00:27ff:fe8d:1ff7 prefixlen 64 scopeid 0x20
 - IPv6-адрес: fe80::a00:27ff:fe8d:1ff7, префикс длиной 64 бита, область действия — локальная сеть.

4. ether 08:00:27:8d:1f:f7 txqueuelen 1900 (Ethernet)

- MAC-адрес интерфейса: 08:00:27:8d:1f:f7. Длина очереди передачи составляет 1900 пакетов.

5. RX packets 69 bytes 8974 (8.7 KiB)

- Получено 69 пакетов, общий объем данных — 8974 байта (8.7 KiB).

6. RX errors 0 dropped 0 overruns 0 frame 0

- Ошибки при получении: нет потерянных пакетов, переполнений или ошибок фрейма.

7. TX packets 315 bytes 32852 (32.0 KiB)

- Отправлено 315 пакетов, общий объем данных — 32852 байта (32 KiB).

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

- Ошибки при передаче: нет потерянных пакетов, переполнений, ошибок связи или коллизий.

lo (Loopback):

1. lo: flags=73<UP, LOOPBACK, RUNNING> mtu 65536

- Интерфейс lo (loopback) активен и работает с MTU равным 65536 байтам.

2. inet 127.0.0.1 netmask 255.0.0.0

- IPv4-адрес интерфейса loopback: 127.0.0.1, маска подсети — 255.0.0.0.

3. inet6 ::1 prefixlen 128 scopeid 0x10

- IPv6-адрес loopback: ::1, префикс длиной 128 бит, область действия — локальный хост.

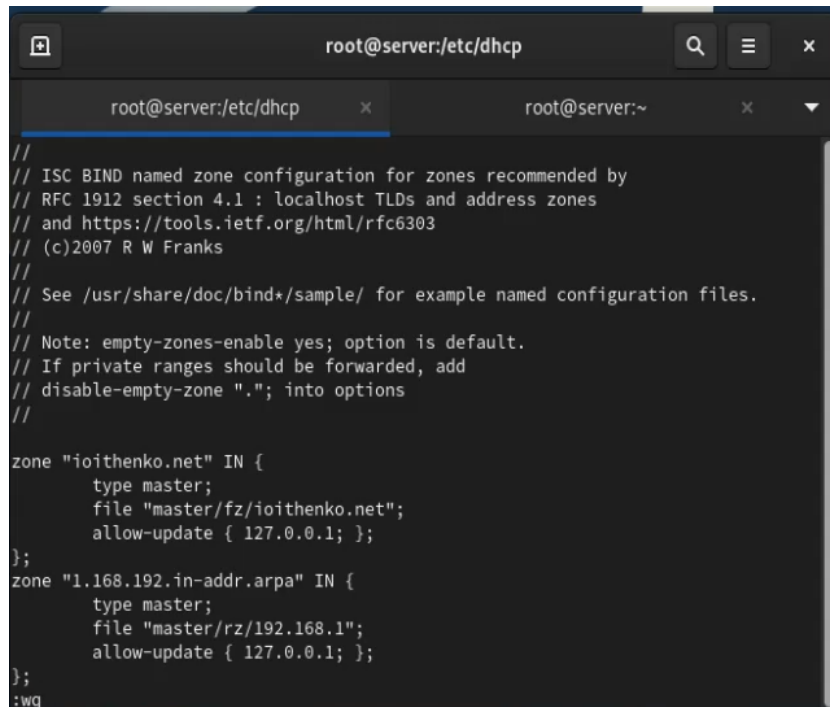
4. loop txqueuelen 1009 (Local Loopback)

- Длина очереди передачи для интерфейса loopback составляет 1009 пакетов.

5. RX packets 18 bytes 2101 (2.0 KiB)

- Получено 18 пакетов, общий объем данных — 2101 байт (2 KiB).

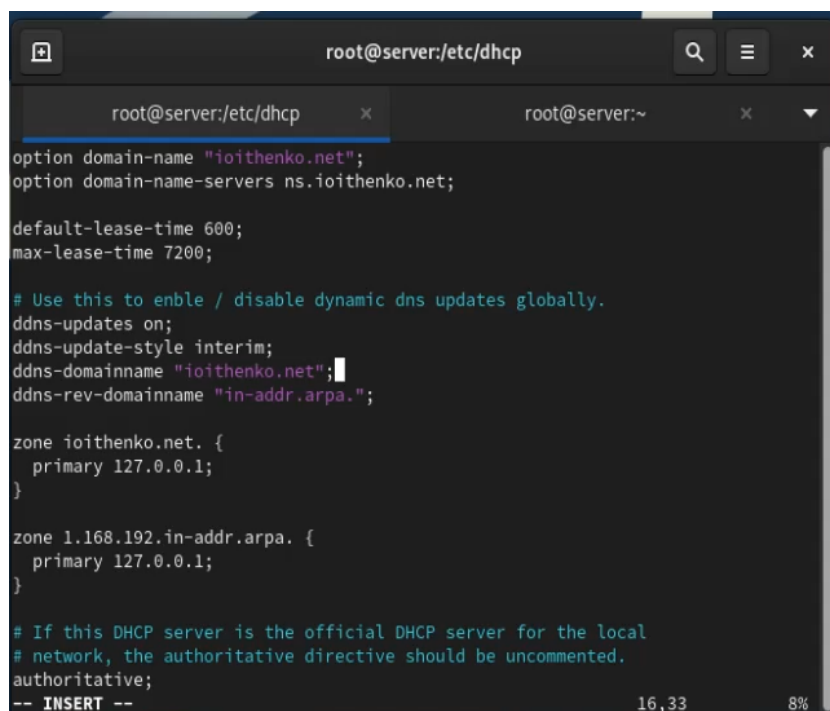
Редактируем файл /etc/named/ioithenko.net (рис. 2.18).



```
//  
// 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 "ioithenko.net" IN {  
    type master;  
    file "master/fz/ioithenko.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; };  
};  
:wq
```

Рис. 2.18: Редактирование файла /etc/named/ioithenko.net

Перезапускаем DNS-сервер. Редактируем файл /etc/dhcp/dhcpd.conf (рис. 2.19).



```
root@server:/etc/dhcp
option domain-name "ioithenko.net";
option domain-name-servers ns.ioithenko.net;

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

# Use this to enable / disable dynamic dns updates globally.
ddns-updates on;
ddns-update-style interim;
ddns-domainname "ioithenko.net";
ddns-rev-domainname "in-addr.arpa.";

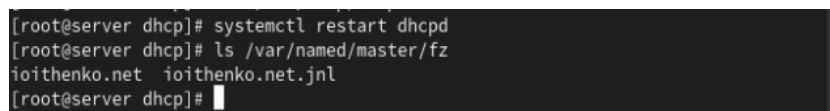
zone ioithenko.net. {
    primary 127.0.0.1;
}

zone 1.168.192.in-addr.arpa. {
    primary 127.0.0.1;
}

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
-- INSERT --
```

Рис. 2.19: Редактирование файла /etc/dhcp/dhcpd.conf

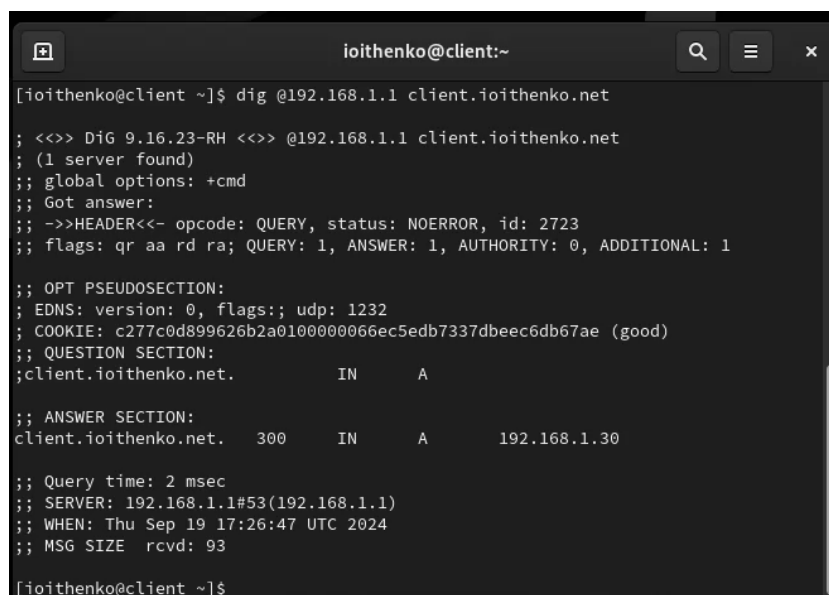
Перезапускаем DHCP-сервер. В каталоге прямой DNS-зоны появился файл ioithenko.net.jnl (рис. 2.20).



```
[root@server dhcp]# systemctl restart dhcpd
[root@server dhcp]# ls /var/named/master/fz
ioithenko.net  ioithenko.net.jnl
[root@server dhcp]#
```

Рис. 2.20: Успешный перезапуск DHCP-сервера

На виртуальной машине client открываем терминал и с помощью утилиты dig убеждаемся в наличии DNS-записи о клиенте в прямой DNS-зоне (рис. 2.21).



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

; <<> DiG 9.16.23-RH <<> @192.168.1.1 client.ioithenko.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2723
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: c277c0d899626b2a0100000066ec5edb7337dbeec6db67ae (good)
;; QUESTION SECTION:
;client.ioithenko.net.      IN      A

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

;; Query time: 2 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Thu Sep 19 17:26:47 UTC 2024
;; MSG SIZE rcvd: 93

[ioithenko@client ~]$
```

Рис. 2.21: Проверка DNS-записи о клиенте в прямой DNS-зоне

Давайте разберем вывод команды dig построчно.

1. 3 «» DIG 9.16.23-RH «» [192.168.1.1?] client.ioithenko.net

- DIG 9.16.23-RH: Это версия утилиты dig, используемой для выполнения DNS-запросов.
- [192.168.1.1?]: Запрос отправляется на DNS-сервер с IP-адресом 192.168.1.1.
- client.ioithenko.net: Это доменное имя, для которого выполняется запрос.

2. (1 server found)

- Указывает, что был найден один сервер для обработки запроса.

3. 33 global options:

- Это строка может указывать на глобальные параметры, используемые в запросе, но конкретные опции не указаны.

4. 33 Got answer:

- Указывает, что сервер вернул ответ на запрос.

5. 3} ~»HEADER«- opcode: QUERY, status: NOERROR, id: 2723

- opcode: QUERY: Тип операции — это стандартный запрос.
- status: NOERROR: Статус ответа — нет ошибок, запрос выполнен успешно.
- id: 2723: Идентификатор запроса, который может быть использован для сопоставления запросов и ответов.

6. 33 flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

- flags:
 - qr: Указывает, что это ответ (query response).
 - aa: Сервер является авторитетным для данного ответа.
 - rd: Запрос был сделан с рекурсией (recursive).
 - ra: Сервер поддерживает рекурсию.
- QUERY: 1: В запросе была одна запись.
- ANSWER: 1: В ответе содержится одна запись.
- AUTHORITY: 0: Указано количество авторитетных записей.
- ADDITIONAL: 1: В ответе также есть одна дополнительная запись.

7. 53 OPT PSEUDOSECTION!3 EDNS: version: 0, flags:: udp: 1232

- OPT PSEUDOSECTION: Раздел для расширенных DNS (EDNS).
- version: 0: Версия EDNS.
- flags:: Здесь нет дополнительных флагов.
- udp: 1232: Максимальный размер UDP-пакета для ответа — 1232 байта.

8. 3 COOKIE: c277c0d899626b2a0100000066ecSedb7337dbeecédb6Tae (good)

- Это cookie, используемое для идентификации сессии DNS. Указание на то, что cookie “good” может означать его корректность или успешность.

9. 33 QUESTION SECTION:

- Начало секции вопросов.

10. 3client.ioithenko.net. IN A

- Запрос на получение записи типа A (IPv4-адрес) для домена client.ioithenko.net.

11. 53 ANSWER SECTION: client.ioithenko.net. 300 IN A 192.168.1.30

- Ответ на запрос:
 - Доменное имя: client.ioithenko.net.
 - Время жизни (TTL): 300 секунд.
 - Запись типа A с адресом 192.168.1.30.

12. 33 Query time: 2 msec

- Время, затраченное на выполнение запроса — 2 миллисекунды.

13. 3} SERVER: 192.168.1.1#53(192.168.1.13}

- Указывает на сервер, который обработал запрос (192.168.1.1) и его порт (53). Второй адрес (192.168.1.13) может указывать на другой адрес сервера или клиента.
-

14. 33 MSG SIZE revd: 93

- Размер полученного сообщения — 93 байта.

На VM server переходим в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создаем в нём каталог dhcp, в который помещаю в соответствующие подкаталоги конфигурационные файлы DHCP:

Заменяем конфигурационные файлы DNS-сервера(рис. 2.22).

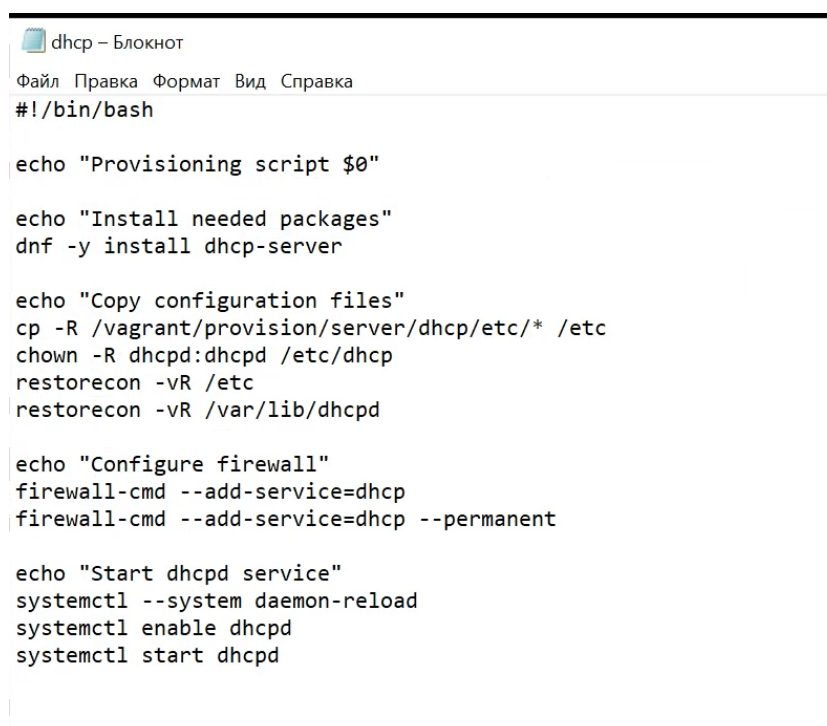
```

[root@server dhcp]# cd /vagrant/provision/server
[root@server server]# mkdir -p /vagrant/provision/server/dhcp/etc/dhcp
[root@server server]# mkdir -p /vagrant/provision/server/dhcp/etc/systemd/system
[root@server server]# cp -R /etc/dhcp/dhcpd.conf /vagrant/provision/server/dhcp/
etc/dhcp/
[root@server server]# cp -R /etc/systemd/system/dhcpd.service /vagrant/provision/
server/dhcp/etc/systemd/system
[root@server server]# cd /vagrant/provision/server/dns
[root@server dns]# cp -R /var/named/* /vagrant/provision/server/dns/var/named
cp: overwrite '/vagrant/provision/server/dns/var/named/master/fz.ioithenko.net'?
y
cp: overwrite '/vagrant/provision/server/dns/var/named/master/rz.192.168.1'? y
[root@server dns]# cp -R /etc/named/* /vagrant/provision/server/dns/etc/named
cp: overwrite '/vagrant/provision/server/dns/etc/named/ioithenko.net'? y
[root@server dns]#

```

Рис. 2.22: Заполнение DHCP

В каталоге `/vagrant/provision/server` создаем исполняемый файл `dhcp.sh` (рис. 2.23).



```

dhcp - Блокнот
Файл Правка Формат Вид Справка
#!/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

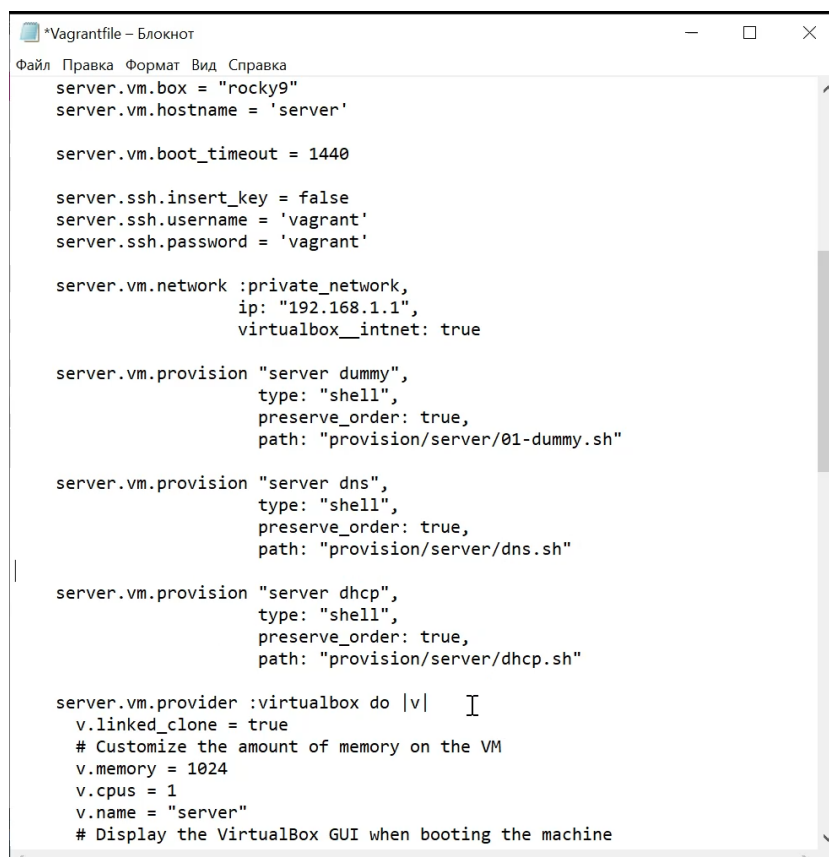
echo "Configure firewall"
firewall-cmd --add-service=dhcp
firewall-cmd --add-service=dhcp --permanent

echo "Start dhcpd service"
systemctl --system daemon-reload
systemctl enable dhcpd
systemctl start dhcpd

```

Рис. 2.23: Создание скрипта `dhcp.sh`

Для отработки скрипта во время запуска добавляю в `Vagrantfile` в разделе конфигурации для сервера (рис. 2.24).



```
*Vagrantfile - Блокнот
Файл Правка Формат Вид Справка
server.vm.box = "rocky9"
server.vm.hostname = 'server'

server.vm.boot_timeout = 1440

server.ssh.insert_key = false
server.ssh.username = 'vagrant'
server.ssh.password = 'vagrant'

server.vm.network :private_network,
  ip: "192.168.1.1",
  virtualbox__intnet: true

server.vm.provision "server dummy",
  type: "shell",
  preserve_order: true,
  path: "provision/server/01-dummy.sh"

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

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

server.vm.provider :virtualbox do |v|
  v.linked_clone = true
  # Customize the amount of memory on the VM
  v.memory = 1024
  v.cpus = 1
  v.name = "server"
  # Display the VirtualBox GUI when booting the machine
```

Рис. 2.24: Vagrantfile

После этого выключаю VM:

vagrant halt client

vagrant halt server

3 Выводы

В результате выполнения работы я приобрела практические навыки по установке и конфигурированию DHCP-сервера.

4 Ответы на контрольные вопросы

1. В каких файлах хранятся настройки сетевых подключений?

В Linux настройки сети обычно хранятся в текстовых файлах в директории `/etc/network/` или `/etc/sysconfig/network-scripts/`.

2. За что отвечает протокол DHCP?

Протокол DHCP (Dynamic Host Configuration Protocol) отвечает за автоматическое присвоение сетевых настроек устройствам в сети, таких как IP-адресов, маски подсети, шлюза, DNS-серверов и других параметров.

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

Принцип работы протокола DHCP:

Discover (Обнаружение): Клиент отправляет в сеть запрос на обнаружение DHCP-сервера.

Offer (Предложение): DHCP-сервер отвечает клиенту, предлагая ему конфигурацию сети.

Request (Запрос): Клиент принимает предложение и отправляет запрос на использование предложенной конфигурации.

Acknowledgment (Подтверждение): DHCP-сервер подтверждает клиенту, что предложенная конфигурация принята и может быть использована.

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

Настройки DHCP-сервера обычно хранятся в файлах конфигурации, таких как `/etc/dhcp/dhcpd.conf`. Они содержат информацию о диапазонах IP-адресов, параметрах сети и других опциях DHCP.

5. Что такое DDNS? Для чего применяется DDNS?

DDNS (Dynamic Domain Name System) - это система динамического доменного имени. Она используется для автоматического обновления записей DNS, когда IP-адрес узла изменяется. DDNS применяется, например, в домашних сетях, где IP-адреса часто изменяются посредством DHCP.

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

Утилита `ifconfig` используется для получения информации о сетевых интерфейсах.

Примеры:

`ifconfig`: Показывает информацию обо всех активных сетевых интерфейсах.

`ifconfig eth0`: Показывает информацию о конкретном интерфейсе (в данном случае, `eth0`).

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

Утилита `ping` используется для проверки доступности узла в сети.

Примеры:

`ping yandex.ru`: Пингует домен `yandex.ru`.

`ping -c 4 192.168.1.1`: Пингует IP-адрес `192.168.1.1` и отправляет 4 эхо-запроса.