Red Hat Enterprise Linux 9

Managing networking infrastructure services

A guide to managing networking infrastructure services in Red Hat Enterprise Linux 9

Last Updated: 2022-07-21

Red Hat Enterprise Linux 9 Managing networking infrastructure services

A guide to managing networking infrastructure services in Red Hat Enterprise Linux 9

Legal Notice

Abstract

This document describes how to set up and manage networking core infrastructure services, such as DNS and DHCP, on Red Hat Enterprise Linux 9.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better.

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.
- For submitting feedback via Bugzilla, create a new ticket:
  1. Go to the Bugzilla website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.

CHAPTER 1. CONFIGURING AND MANAGING A BIND DNS SERVER

DNS (Domain Name System) is a distributed database system that associates hostnames with their respective IP addresses. **BIND** (Berkeley Internet Name Domain) consists of a set of DNS-related programs. It contains a name server called **named**. The **/etc/named.conf** is the main configuration file in the BIND configuration. This section focuses on installing, configuring, and managing **BIND** on the DNS server.

1.1. INSTALLING BIND

The installation of the **bind-utils** package ensures the **BIND** utilities are available on the system.

Procedure

1. Install **BIND**:

```
# dnf install bind bind-utils
```

2. Enable and start the **named** service:

```
# systemctl enable --now named
```

Verification steps

- Verify the status of the **named** service:

```
# systemctl status named
```

1.2. CONFIGURING BIND AS A CACHING NAME SERVER

The following procedure demonstrates configuring **BIND** as a caching name server.

Prerequisites

- The **bind** package is installed.

Procedure

1. Ensure to take backup of the original configuration file.

```
# cp /etc/named.conf /etc/named.conf.orig
```

2. Edit the **/etc/named.conf** file with the following changes:
   - In the options section, uncomment the **listen-on**, **listen-on-v6**, and **directory** parameters:

```
acl clients {192.0.2.0/24;};

options {
        listen-on port 53 { any; };
```

```
        listen-on-v6 port 53 { any; };

        directory       "/var/named";
```

- Set the **allow-query** parameter to your network address. Only the hosts on your local network can query the DNS server:

```
    allow-query     { localhost; clients; };
    allow-recursion { localhost; clients; };
    recursion yes;
    allow-update { none; };
    allow-transfer { localhost; };
};

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

- Use the package shipped file as:

```
include "/etc/named.rfc1912.zones";
```

- Create an extra include for any custom zone configuration:

```
include "/etc/named/example.zones";
```

3. Create the **/etc/named/example.zones** file and add the following zone configuration:

```
//forward zone
zone "example.com" IN {
        type master;
        file "example.com.zone";

};

//backward zone
zone "2.0.192.in-addr.arpa" IN {
        type master;
        file "example.com.rzone";

};
```

- type: It defines the zone's role of the server.
- master: It is an authoritative server and maintains the master copy of the zone data.
- file: It specifies the zone's database file.

4. Go to DNS data directory **/var/named/**:

```
# cd /var/named/
# ls

data  dynamic  named.ca  named.empty  named.localhost  named.loopback  slaves
```

5. Create the **/var/named/*example.com*.zone** file with your forward zone parameters:

```
$TTL    86400
@ IN SOA example.com. root (
  42        ; serial
  3H        ; refresh
  15M       ; retry
  1W        ; expiry
  1D )      ; minimum

 IN NS ns.example.com.

ns            IN A        192.0.2.1
station1      IN A        192.0.2.101
station2      IN A        192.0.2.102
station3      IN A        192.0.2.103
```

6. Create the **/var/named/*example.com*.rzone** file with your reverse zone parameters:

```
$TTL    86400
@ IN SOA example.com. root.example.com. (
  1997022700 ; serial
  28800      ; refresh
  14400      ; retry
  3600000    ; expire
  86400 )    ; minimum

 IN NS ns.example.com.

101    IN    PTR    station1.example.com.
102    IN    PTR    station2.example.com.
103    IN    PTR    station3.example.com.
```

7. Set secure permissions on the zone files:

```
# chown root:named /var/named/example.com.zone /var/named/example.com.rzone
# chmod 640 /var/named/example.com.zone /var/named/example.com.rzone
```

8. Restart BIND:

```
# systemctl restart named
```

Verification steps

- Verify the forward zone file:

```
# named-checkzone example.com /var/named/example.com.zone
```

```
zone example.com/IN: loaded serial xxxxxxx
OK
```

- Verify the reverse zone file:

```
# named-checkzone 2.0.192.in-addr.arpa /var/named/example.com.rzone

zone 2.0.192.in-addr.arpa/IN: loaded serial xxxxxxx
OK
```

- Verify the configuration:

```
# named-checkconf /etc/named.conf
```

If the configuration is correct, the command does not return any output.

CHAPTER 2. SETTING UP AN UNBOUND DNS SERVER

The **unbound** DNS server is a validating, recursive, and caching DNS resolver. Additionally, **unbound** focuses on security and has, for example, Domain Name System Security Extensions (DNSSEC) enabled by default.

2.1. CONFIGURING UNBOUND AS A CACHING DNS SERVER

By default, the **unbound** DNS service resolves and caches successful and failed lookups. The service then answers requests to the same records from its cache.

Procedure

1. Install the **unbound** package:

```
# dnf install unbound
```

2. Edit the **/etc/unbound/unbound.conf** file, and make the following changes in the **server** clause:

   a. Add **interface** parameters to configure on which IP addresses the **unbound** service listens for queries, for example:

```
interface: 127.0.0.1
interface: 192.0.2.1
interface: 2001:db8:1::1
```

   With these settings, **unbound** only listens on the specified IPv4 and IPv6 addresses.

   Limiting the interfaces to the required ones prevents clients from unauthorized networks, such as the internet, from sending queries to this DNS server.

   b. Add **access-control** parameters to configure from which subnets clients can query the DNS service, for example:

```
access-control: 127.0.0.0/8 allow
access-control: 192.0.2.0/24 allow
access-control: 2001:db8:1::/64 allow
```

3. Create private keys and certificates for remotely managing the **unbound** service:

```
# systemctl restart unbound-keygen
```

   If you skip this step, verifying the configuration in the next step will report the missing files. However, the **unbound** service automatically creates the files if they are missing.

4. Verify the configuration file:

```
# unbound-checkconf
unbound-checkconf: no errors in /etc/unbound/unbound.conf
```

5. Update the firewalld rules to allow incoming DNS traffic:

```
# firewall-cmd --permanent --add-service=dns
# firewall-cmd --reload
```

6. Enable and start the **unbound** service:

```
# systemctl enable --now unbound
```

Verification

1. Query the **unbound** DNS server listening on the **localhost** interface to resolve a domain:

```
# dig @localhost www.example.com
...
www.example.com.    86400    IN    A    198.51.100.34

;; Query time: 330 msec
...
```

After querying a record for the first time, **unbound** adds the entry to its cache.

2. Repeat the previous query:

```
# dig @localhost www.example.com
...
www.example.com.    85332    IN    A    198.51.100.34

;; Query time: 1 msec
...
```

Because of the cached entry, further requests for the same record are significantly faster until the entry expires.

Next steps

- Configure clients in your network to use this DNS server. For example, use the **nmcli** utility to set the IP of the DNS server in a NetworkManager connection profile:

```
# nmcli connection modify Example_Connection ipv4.dns 192.0.2.1
# nmcli connection modify Example_Connection ipv6.dns 2001:db8:1::1
```

Additional resources

- **unbound.conf(5)** man page

CHAPTER 3. PROVIDING DHCP SERVICES

The dynamic host configuration protocol (DHCP) is a network protocol that automatically assigns IP information to clients.

This section explains general information on the **dhcpd** service, as well as how to set up a DHCP server and DHCP relay.

If a procedure requires different steps for providing DHCP in IPv4 and IPv6 networks, the sections in this chapter contain procedures for both protocols.

3.1. THE DIFFERENCE BETWEEN STATIC AND DYNAMIC IP ADDRESSING

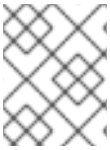
Static IP addressing

When you assign a static IP address to a device, the address does not change over time unless you change it manually. Use static IP addressing if you want:

- To ensure network address consistency for servers such as DNS, and authentication servers.
- To use out-of-band management devices that work independently of other network infrastructure.

Dynamic IP addressing

When you configure a device to use a dynamic IP address, the address can change over time. For this reason, dynamic addresses are typically used for devices that connect to the network occasionally because the IP address can be different after rebooting the host.
Dynamic IP addresses are more flexible, easier to set up, and administer. The Dynamic Host Control Protocol (DHCP) is a traditional method of dynamically assigning network configurations to hosts.



NOTE

There is no strict rule defining when to use static or dynamic IP addresses. It depends on user's needs, preferences, and the network environment.

3.2. DHCP TRANSACTION PHASES

The DHCP works in four phases: Discovery, Offer, Request, Acknowledgement, also called the DORA process. DHCP uses this process to provide IP addresses to clients.

Discovery

The DHCP client sends a message to discover the DHCP server in the network. This message is broadcasted at the network and data link layer.

Offer

The DHCP server receives messages from the client and offers an IP address to the DHCP client. This message is unicast at the data link layer but broadcast at the network layer.

Request

The DHCP client requests the DHCP server for the offered IP address. This message is unicast at the data link layer but broadcast at the network layer.

Acknowledgment

The DHCP server sends an acknowledgment to the DHCP client. This message is unicast at the data link layer but broadcast at the network layer. It is the final message of the DHCP DORA process.

3.3. THE DIFFERENCES WHEN USING DHCPD FOR DHCPV4 AND DHCPV6

The **dhcpd** service supports providing both DHCPv4 and DHCPv6 on one server. However, you need a separate instance of **dhcpd** with separate configuration files to provide DHCP for each protocol.

DHCPv4

- Configuration file: **/etc/dhcp/dhcpd.conf**
- Systemd service name: **dhcpd**

DHCPv6

- Configuration file: **/etc/dhcp/dhcpd6.conf**
- Systemd service name: **dhcpd6**

3.4. THE LEASE DATABASE OF THE DHCPD SERVICE

A DHCP lease is the time period for which the **dhcpd** service allocates a network address to a client. The **dhcpd** service stores the DHCP leases in the following databases:

- For DHCPv4: **/var/lib/dhcpd/dhcpd.leases**
- For DHCPv6: **/var/lib/dhcpd/dhcpd6.leases**

WARNING

Manually updating the database files can corrupt the databases.

The lease databases contain information about the allocated leases, such as the IP address assigned to a media access control (MAC) address or the time stamp when the lease expires. Note that all time stamps in the lease databases are in Coordinated Universal Time (UTC).

The **dhcpd** service recreates the databases periodically:

1. The service renames the existing files:
   - **/var/lib/dhcpd/dhcpd.leases** to **/var/lib/dhcpd/dhcpd.leases~**
   - **/var/lib/dhcpd/dhcpd6.leases** to **/var/lib/dhcpd/dhcpd6.leases~**
2. The service writes all known leases to the newly created **/var/lib/dhcpd/dhcpd.leases** and **/var/lib/dhcpd/dhcpd6.leases** files.

Additional resources

Additional resources

- **dhcpd.leases(5)** man page
- Restoring a corrupt lease database

3.5. COMPARISON OF DHCPV6 TO RADVD

In an IPv6 network, only router advertisement messages provide information on an IPv6 default gateway. As a consequence, if you want to use DHCPv6 in subnets that require a default gateway setting, you must additionally configure a router advertisement service, such as Router Advertisement Daemon (**radvd**).

The **radvd** service uses flags in router advertisement packets to announce the availability of a DHCPv6 server.

This section compares DHCPv6 and **radvd**, and provides information about configuring **radvd**.

	DHCPv6	radvd
Provides information on the default gateway	no	yes
Guarantees random addresses to protect privacy	yes	no
Sends further network configuration options	yes	no
Maps media access control (MAC) addresses to IPv6 addresses	yes	no

3.6. CONFIGURING THE RADVD SERVICE FOR IPV6 ROUTERS

The router advertisement daemon (**radvd**) sends router advertisement messages that are required for IPv6 stateless autoconfiguration. This enables users to automatically configure their addresses, settings, routes, and to choose a default router based on these advertisements.

The procedure in this section explains how to configure **radvd**.

NOTE

You can only set **/64** prefixes in the **radvd** service. To use other prefixes, use DHCPv6.

Prerequisites

- You are logged in as the **root** user.

Procedure

1. Install the **radvd** package:

```
# dnf install radvd
```

2. Edit the **/etc/radvd.conf** file, and add the following configuration:

```
interface enp1s0
{
  AdvSendAdvert on;
  AdvManagedFlag on;
  AdvOtherConfigFlag on;

  prefix 2001:db8:0:1::/64 {
  };
};
```

These settings configures **radvd** to send router advertisement messages on the **enp1s0** device for the **2001:db8:0:1::/64** subnet. The **AdvManagedFlag on** setting defines that the client should receive the IP address from a DHCP server, and the **AdvOtherConfigFlag** parameter set to **on** defines that clients should receive non-address information from the DHCP server as well.

3. Optionally, configure that **radvd** automatically starts when the system boots:

```
# systemctl enable radvd
```

4. Start the **radvd** service:

```
# systemctl start radvd
```

5. Optionally, display the content of router advertisement packages and the configured values **radvd** sends:

```
# radvdump
```

Additional resources

- **radvd.conf(5)** man page
- **/usr/share/doc/radvd/radvd.conf.example**
- Can I use a prefix length other than 64 bits in IPv6 Router Advertisements?

3.7. SETTING NETWORK INTERFACES FOR THE DHCP SERVERS

By default, the **dhcpd** service processes requests only on network interfaces that have an IP address in the subnet defined in the configuration file of the service.

For example, in the following scenario, **dhcpd** listens only on the **enp0s1** network interface:

- You have only a **subnet** definition for the 192.0.2.0/24 network in the **/etc/dhcp/dhcpd.conf** file.
- The **enp0s1** network interface is connected to the 192.0.2.0/24 subnet.
- The **enp7s0** interface is connected to a different subnet.

Only follow the procedure in this section if the DHCP server contains multiple network interfaces connected to the same network but the service should listen only on specific interfaces.

Depending on whether you want to provide DHCP for IPv4, IPv6, or both protocols, see the procedure for:

- IPv4 networks
- IPv6 networks

Prerequisites

- You are logged in as the **root** user.
- The **dhcp-server** package is installed.

Procedure

- For IPv4 networks:
  1. Copy the **/usr/lib/systemd/system/dhcpd.service** file to the **/etc/systemd/system/** directory:

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

     Do not edit the **/usr/lib/systemd/system/dhcpd.service** file. Future updates of the **dhcp-server** package can override the changes.
  2. Edit the **/etc/systemd/system/dhcpd.service** file, and append the names of the interface, that **dhcpd** should listen on to the command in the **ExecStart** parameter:

     ```
     ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid $DHCPDARGS enp0s1 enp7s0
     ```

     This example configures that **dhcpd** listens only on the ***enp0s1*** and **enp7s0** interfaces.
  3. Reload the **systemd** manager configuration:

     ```
     # systemctl daemon-reload
     ```
  4. Restart the **dhcpd** service:

     ```
     # systemctl restart dhcpd.service
     ```
- For IPv6 networks:
  1. Copy the **/usr/lib/systemd/system/dhcpd6.service** file to the **/etc/systemd/system/** directory:

     ```
     # cp /usr/lib/systemd/system/dhcpd6.service /etc/systemd/system/
     ```

     Do not edit the **/usr/lib/systemd/system/dhcpd6.service** file. Future updates of the **dhcp-server** package can override the changes.
  2. Edit the **/etc/systemd/system/dhcpd6.service** file, and append the names of the interface, that **dhcpd** should listen on to the command in the **ExecStart** parameter:

```
ExecStart=/usr/sbin/dhcpd -f -6 -cf /etc/dhcp/dhcpd6.conf -user dhcpd -group dhcpd --no-pid $DHCPDARGS enp0s1 enp7s0
```

This example configures that **dhcpd** listens only on the ***enp0s1*** and **enp7s0** interfaces.

3. Reload the **systemd** manager configuration:

```
# systemctl daemon-reload
```

4. Restart the **dhcpd6** service:

```
# systemctl restart dhcpd6.service
```

3.8. SETTING UP THE DHCP SERVICE FOR SUBNETS DIRECTLY CONNECTED TO THE DHCP SERVER

Use the following procedure if the DHCP server is directly connected to the subnet for which the server should answer DHCP requests. This is the case if a network interface of the server has an IP address of this subnet assigned.

Depending on whether you want to provide DHCP for IPv4, IPv6, or both protocols, see the procedure for:

- IPv4 networks
- IPv6 networks

Prerequisites

- You are logged in as the **root** user.
- The **dhcp-server** package is installed.

Procedure

- For IPv4 networks:
  1. Edit the **/etc/dhcp/dhcpd.conf** file:
     a. Optionally, add global parameters that **dhcpd** uses as default if no other directives contain these settings:

```
option domain-name "example.com";
default-lease-time 86400;
```

This example sets the default domain name for the connection to **example.com**, and the default lease time to **86400** seconds (1 day).

     b. Add the **authoritative** statement on a new line:

```
authoritative;
```

IMPORTANT

Without the **authoritative** statement, the **dhcpd** service does not answer **DHCPREQUEST** messages with **DHCPNAK** if a client asks for an address that is outside of the pool.

c. For each IPv4 subnet directly connected to an interface of the server, add a **subnet** declaration:

```
subnet 192.0.2.0 netmask 255.255.255.0 {
  range 192.0.2.20 192.0.2.100;
  option domain-name-servers 192.0.2.1;
  option routers 192.0.2.1;
  option broadcast-address 192.0.2.255;
  max-lease-time 172800;
}
```

This example adds a subnet declaration for the 192.0.2.0/24 network. With this configuration, the DHCP server assigns the following settings to a client that sends a DHCP request from this subnet:

- A free IPv4 address from the range defined in the **range** parameter
- IP of the DNS server for this subnet: **192.0.2.1**
- Default gateway for this subnet: **192.0.2.1**
- Broadcast address for this subnet: **192.0.2.255**
- The maximum lease time, after which clients in this subnet release the IP and send a new request to the server: **172800** seconds (2 days)

2. Optionally, configure that **dhcpd** starts automatically when the system boots:

```
# systemctl enable dhcpd
```

3. Start the **dhcpd** service:

```
# systemctl start dhcpd
```

- For IPv6 networks:

1. Edit the **/etc/dhcp/dhcpd6.conf** file:

a. Optionally, add global parameters that **dhcpd** uses as default if no other directives contain these settings:

```
option dhcp6.domain-search "example.com";
default-lease-time 86400;
```

This example sets the default domain name for the connection to **example.com**, and the default lease time to **86400** seconds (1 day).

b. Add the **authoritative** statement on a new line:

```
authoritative;
```

IMPORTANT

Without the **authoritative** statement, the **dhcpd** service does not answer **DHCPREQUEST** messages with **DHCPNAK** if a client asks for an address that is outside of the pool.

c. For each IPv6 subnet directly connected to an interface of the server, add a **subnet** declaration:

```
subnet6 2001:db8:0:1::/64 {
  range6 2001:db8:0:1::20 2001:db8:0:1::100;
  option dhcp6.name-servers 2001:db8:0:1::1;
  max-lease-time 172800;
}
```

This example adds a subnet declaration for the 2001:db8:0:1::/64 network. With this configuration, the DHCP server assigns the following settings to a client that sends a DHCP request from this subnet:

- A free IPv6 address from the range defined in the **range6** parameter.
- The IP of the DNS server for this subnet is **2001:db8:0:1::1**.
- The maximum lease time, after which clients in this subnet release the IP and send a new request to the server is **172800** seconds (2 days).
  Note that IPv6 requires uses router advertisement messages to identify the default gateway.

2. Optionally, configure that **dhcpd6** starts automatically when the system boots:

```
# systemctl enable dhcpd6
```

3. Start the **dhcpd6** service:

```
# systemctl start dhcpd6
```

Additional resources

- **dhcp-options(5)** man page
- The **The authoritative statement** section in the **dhcpd.conf(5)** man page
- **/usr/share/doc/dhcp-server/dhcpd.conf.example**
- **/usr/share/doc/dhcp-server/dhcpd6.conf.example**

3.9. SETTING UP THE DHCP SERVICE FOR SUBNETS THAT ARE NOT DIRECTLY CONNECTED TO THE DHCP SERVER

Use the following procedure if the DHCP server is not directly connected to the subnet for which the server should answer DHCP requests. This is the case if a DHCP relay agent forwards requests to the

DHCP server, because none of the DHCP server's interfaces is directly connected to the subnet the server should serve.

Depending on whether you want to provide DHCP for IPv4, IPv6, or both protocols, see the procedure for:

- IPv4 networks
- IPv6 networks

Prerequisites

- You are logged in as the **root** user.
- The **dhcp-server** package is installed.

Procedure

- For IPv4 networks:

1. Edit the **/etc/dhcp/dhcpd.conf** file:

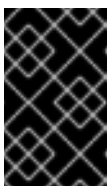
   a. Optionally, add global parameters that **dhcpd** uses as default if no other directives contain these settings:

   ```
   option domain-name "example.com";
   default-lease-time 86400;
   ```

   This example sets the default domain name for the connection to **example.com**, and the default lease time to **86400** seconds (1 day).

   b. Add the **authoritative** statement on a new line:

   ```
   authoritative;
   ```

   IMPORTANT

   Without the **authoritative** statement, the **dhcpd** service does not answer **DHCPREQUEST** messages with **DHCPNAK** if a client asks for an address that is outside of the pool.

   c. Add a **shared-network** declaration, such as the following, for IPv4 subnets that are not directly connected to an interface of the server:

   ```
   shared-network example {
     option domain-name-servers 192.0.2.1;
     ...

     subnet 192.0.2.0 netmask 255.255.255.0 {
       range 192.0.2.20 192.0.2.100;
       option routers 192.0.2.1;
     }

     subnet 198.51.100.0 netmask 255.255.255.0 {
       range 198.51.100.20 198.51.100.100;
   ```

```
    option routers 198.51.100.1;
  }
  ...
}
```

This example adds a shared network declaration, that contains a **subnet** declaration for both the 192.0.2.0/24 and 198.51.100.0/24 networks. With this configuration, the DHCP server assigns the following settings to a client that sends a DHCP request from one of these subnets:

- The IP of the DNS server for clients from both subnets is: **192.0.2.1**.
- A free IPv4 address from the range defined in the **range** parameter, depending on from which subnet the client sent the request.
- The default gateway is either **192.0.2.1** or **198.51.100.1** depending on from which subnet the client sent the request.

d. Add a **subnet** declaration for the subnet the server is directly connected to and that is used to reach the remote subnets specified in **shared-network** above:

```
subnet 203.0.113.0 netmask 255.255.255.0 {
}
```

NOTE

If the server does not provide DHCP service to this subnet, the **subnet** declaration must be empty as shown in the example. Without a declaration for the directly connected subnet, **dhcpd** does not start.

2. Optionally, configure that **dhcpd** starts automatically when the system boots:

```
# systemctl enable dhcpd
```

3. Start the **dhcpd** service:

```
# systemctl start dhcpd
```

- For IPv6 networks:

1. Edit the **/etc/dhcp/dhcpd6.conf** file:

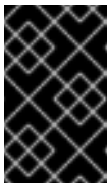
a. Optionally, add global parameters that **dhcpd** uses as default if no other directives contain these settings:

```
option dhcp6.domain-search "example.com";
default-lease-time 86400;
```

This example sets the default domain name for the connection to **example.com**, and the default lease time to **86400** seconds (1 day).

b. Add the **authoritative** statement on a new line:

```
authoritative;
```



IMPORTANT

Without the **authoritative** statement, the **dhcpd** service does not answer **DHCPREQUEST** messages with **DHCPNAK** if a client asks for an address that is outside of the pool.

c. Add a **shared-network** declaration, such as the following, for IPv6 subnets that are not directly connected to an interface of the server:

```
shared-network example {
  option domain-name-servers 2001:db8:0:1::1:1
  ...

  subnet6 2001:db8:0:1::1:0/120 {
    range6 2001:db8:0:1::1:20 2001:db8:0:1::1:100
  }

  subnet6 2001:db8:0:1::2:0/120 {
    range6 2001:db8:0:1::2:20 2001:db8:0:1::2:100
  }
  ...
}
```

This example adds a shared network declaration that contains a **subnet6** declaration for both the 2001:db8:0:1::1:0/120 and 2001:db8:0:1::2:0/120 networks. With this configuration, the DHCP server assigns the following settings to a client that sends a DHCP request from one of these subnets:

- The IP of the DNS server for clients from both subnets is **2001:db8:0:1::1:1**.
- A free IPv6 address from the range defined in the **range6** parameter, depending on from which subnet the client sent the request.
  Note that IPv6 requires uses router advertisement messages to identify the default gateway.

d. Add a **subnet6** declaration for the subnet the server is directly connected to and that is used to reach the remote subnets specified in **shared-network** above:

```
subnet6 2001:db8:0:1::50:0/120 {
}
```

NOTE

If the server does not provide DHCP service to this subnet, the **subnet6** declaration must be empty as shown in the example. Without a declaration for the directly connected subnet, **dhcpd** does not start.

2. Optionally, configure that **dhcpd6** starts automatically when the system boots:

```
# systemctl enable dhcpd6
```

3. Start the **dhcpd6** service:

```
# systemctl start dhcpd6
```

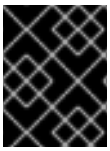
-

Additional resources

- **dhcp-options(5)** man page
- The **The authoritative statement** section in the **dhcpd.conf(5)** man page
- **/usr/share/doc/dhcp-server/dhcpd.conf.example**
- **/usr/share/doc/dhcp-server/dhcpd6.conf.example**
- Setting up a DHCP relay agent

3.10. ASSIGNING A STATIC ADDRESS TO A HOST USING DHCP

Using a **host** declaration, you can configure the DHCP server to assign a fixed IP address to a media access control (MAC) address of a host. For example, use this method to always assign the same IP address to a server or network device.



IMPORTANT

If you configure a fixed IP address for a MAC address, the IP address must be outside of the address pool you specified in the **fixed-address** and **fixed-address6** parameters.

Depending on whether you want to configure fixed addresses for IPv4, IPv6, or both protocols, see the procedure for:

- IPv4 Networks
- IPv6 Networks

Prerequisites

- The **dhcpd** service is configured and running.
- You are logged in as the **root** user.

Procedure

- For IPv4 networks:
  1. Edit the **/etc/dhcp/dhcpd.conf** file:
     a. Add a **host** declaration:

```
host server.example.com {
 hardware ethernet 52:54:00:72:2f:6e;
 fixed-address 192.0.2.130;
}
```

This example configures the DHCP server to always assigns the **192.0.2.130** IP address to the host with the **52:54:00:72:2f:6e** MAC address.

The **dhcpd** service identifies systems by the MAC address specified in the **fixed-address** parameter, and not by the name in the **host** declaration. As a consequence,

you can set this name to any string that does not match other **host** declarations. To configure the same system for multiple networks, use a different name, otherwise, **dhcpd** fails to start.

b. Optionally, add further settings to the **host** declaration that are specific for this host.

2. Restart the **dhcpd** service:

```
# systemctl start dhcpd
```

- For IPv6 networks:

1. Edit the **/etc/dhcp/dhcpd6.conf** file:

a. Add a **host** declaration:

```
host server.example.com {
 hardware ethernet 52:54:00:72:2f:6e;
 fixed-address6 2001:db8:0:1::200;
}
```

This example configures the DHCP server to always assign the **2001:db8:0:1::20** IP address to the host with the **52:54:00:72:2f:6e** MAC address.

The **dhcpd** service identifies systems by the MAC address specified in the **fixed-address6** parameter, and not by the name in the **host** declaration. As a consequence, you can set this name to any string, as long as it is unique to other **host** declarations. To configure the same system for multiple networks, use a different name because, otherwise, **dhcpd** fails to start.

b. Optionally, add further settings to the **host** declaration that are specific for this host.

2. Restart the **dhcpd6** service:

```
# systemctl start dhcpd6
```

Additional resources

- **dhcp-options(5)** man page
- **/usr/share/doc/dhcp-server/dhcpd.conf.example**
- **/usr/share/doc/dhcp-server/dhcpd6.conf.example**

3.11. USING A GROUP DECLARATION TO APPLY PARAMETERS TO MULTIPLE HOSTS, SUBNETS, AND SHARED NETWORKS AT THE SAME TIME

Using a **group** declaration, you can apply the same parameters to multiple hosts, subnets, and shared networks.

Note that the procedure in this section describes using a **group** declaration for hosts, but the steps are the same for subnets and shared networks.

Depending on whether you want to configure a group for IPv4, IPv6, or both protocols, see the procedure for:

- IPv4 networks
- IPv6 networks

Prerequisites

- The **dhcpd** service is configured and running.
- You are logged in as the **root** user.

Procedure

- For IPv4 networks:
  1. Edit the **/etc/dhcp/dhcpd.conf** file:
     a. Add a **group** declaration:

```
group {
  option domain-name-servers 192.0.2.1;

  host server1.example.com {
    hardware ethernet 52:54:00:72:2f:6e;
    fixed-address 192.0.2.130;
  }

  host server2.example.com {
    hardware ethernet 52:54:00:1b:f3:cf;
    fixed-address 192.0.2.140;
  }
}
```

     This **group** definition groups two **host** entries. The **dhcpd** service applies the value set in the **option domain-name-servers** parameter to both hosts in the group.

     b. Optionally, add further settings to the **group** declaration that are specific for these hosts.

  2. Restart the **dhcpd** service:

```
# systemctl start dhcpd
```

- For IPv6 networks:
  1. Edit the **/etc/dhcp/dhcpd6.conf** file:
     a. Add a **group** declaration:

```
group {
  option dhcp6.domain-search "example.com";

  host server1.example.com {
    hardware ethernet 52:54:00:72:2f:6e;
```

```
    fixed-address 2001:db8:0:1::200;
  }

  host server2.example.com {
    hardware ethernet 52:54:00:1b:f3:cf;
    fixed-address 2001:db8:0:1::ba3;
  }
}
```

This **group** definition groups two **host** entries. The **dhcpd** service applies the value set in the **option dhcp6.domain-search** parameter to both hosts in the group.

b. Optionally, add further settings to the **group** declaration that are specific for these hosts.

2. Restart the **dhcpd6** service:

```
# systemctl start dhcpd6
```

Additional resources

- **dhcp-options(5)** man page
- **/usr/share/doc/dhcp-server/dhcpd.conf.example**
- **/usr/share/doc/dhcp-server/dhcpd6.conf.example**

3.12. RESTORING A CORRUPT LEASE DATABASE

If the DHCP server logs an error that is related to the lease database, such as **Corrupt lease file - possible data loss!**,you can restore the lease database from the copy the **dhcpd** service created. Note that this copy might not reflect the latest status of the database.

WARNING

If you remove the lease database instead of replacing it with a backup, you lose all information about the currently assigned leases. As a consequence, the DHCP server could assign leases to clients that have been previously assigned to other hosts and are not expired yet. This leads to IP conflicts.

Depending on whether you want to restore the DHCPv4, DHCPv6, or both databases, see the procedure for:

- Restoring the DHCPv4 lease database
- Restoring the DHCPv6 lease database

Prerequisites

- You are logged in as the **root** user.

- The lease database is corrupt.

Procedure

- Restoring the DHCPv4 lease database:
  1. Stop the **dhcpd** service:

     ```
     # systemctl stop dhcpd
     ```

  2. Rename the corrupt lease database:

     ```
     # mv /var/lib/dhcpd/dhcpd.leases /var/lib/dhcpd/dhcpd.leases.corrupt
     ```

  3. Restore the copy of the lease database that the **dhcp** service created when it refreshed the lease database:

     ```
     # cp -p /var/lib/dhcpd/dhcpd.leases~ /var/lib/dhcpd/dhcpd.leases
     ```

     IMPORTANT

     If you have a more recent backup of the lease database, restore this backup instead.

  4. Start the **dhcpd** service:

     ```
     # systemctl start dhcpd
     ```

- Restoring the DHCPv6 lease database:
  1. Stop the **dhcpd6** service:

     ```
     # systemctl stop dhcpd6
     ```

  2. Rename the corrupt lease database:

     ```
     # mv /var/lib/dhcpd/dhcpd6.leases /var/lib/dhcpd/dhcpd6.leases.corrupt
     ```

  3. Restore the copy of the lease database that the **dhcp** service created when it refreshed the lease database:

     ```
     # cp -p /var/lib/dhcpd/dhcpd6.leases~ /var/lib/dhcpd/dhcpd6.leases
     ```

     IMPORTANT

     If you have a more recent backup of the lease database, restore this backup instead.

  4. Start the **dhcpd6** service:

     ```
     # systemctl start dhcpd6
     ```

Additional resources

- The lease database of the dhcpd service

3.13. SETTING UP A DHCP RELAY AGENT

The DHCP Relay Agent (**dhcrelay**) enables the relay of DHCP and BOOTP requests from a subnet with no DHCP server on it to one or more DHCP servers on other subnets. When a DHCP client requests information, the DHCP Relay Agent forwards the request to the list of DHCP servers specified. When a DHCP server returns a reply, the DHCP Relay Agent forwards this request to the client.

Depending on whether you want to set up a DHCP relay for IPv4, IPv6, or both protocols, see the procedure for:

- IPv4 networks
- IPv6 networks

Prerequisites

- You are logged in as the **root** user.

Procedure

- For IPv4 networks:

1. Install the **dhcp-relay** package:

```
# dnf install dhcp-relay
```

2. Copy the **/lib/systemd/system/dhcrelay.service** file to the **/etc/systemd/system/** directory:

```
# cp /lib/systemd/system/dhcrelay.service /etc/systemd/system/
```

Do not edit the **/usr/lib/systemd/system/dhcrelay.service** file. Future updates of the **dhcp-relay** package can override the changes.

3. Edit the **/etc/systemd/system/dhcrelay.service** file, and append the **-i *interface*** parameter, together with a list of IP addresses of DHCPv4 servers that are responsible for the subnet:

```
ExecStart=/usr/sbin/dhcrelay -d --no-pid -i enp1s0 192.0.2.1
```

With these additional parameters, **dhcrelay** listens for DHCPv4 requests on the **enp1s0** interface and forwards them to the DHCP server with the IP **192.0.2.1**.

4. Reload the **systemd** manager configuration:

```
# systemctl daemon-reload
```

5. Optionally, configure that the **dhcrelay** service starts when the system boots:

```
# systemctl enable dhcrelay.service
```

6. Start the **dhcrelay** service:

```
# systemctl start dhcrelay.service
```

- For IPv6 networks:

1. Install the **dhcp-relay** package:

```
# dnf install dhcp-relay
```

2. Copy the **/lib/systemd/system/dhcrelay.service** file to the **/etc/systemd/system/** directory and name the file **dhcrelay6.service**:

```
# cp /lib/systemd/system/dhcrelay.service /etc/systemd/system/dhcrelay6.service
```

Do not edit the **/usr/lib/systemd/system/dhcrelay.service** file. Future updates of the **dhcp-relay** package can override the changes.

3. Edit the **/etc/systemd/system/dhcrelay6.service** file, and append the **-l *receiving_interface*** and **-u *outgoing_interface*** parameters:

```
ExecStart=/usr/sbin/dhcrelay -d --no-pid -l enp1s0 -u enp7s0
```

With these additional parameters, **dhcrelay** listens for DHCPv6 requests on the **enp1s0** interface and forwards them to the network connected to the **enp7s0** interface.

4. Reload the **systemd** manager configuration:

```
# systemctl daemon-reload
```

5. Optionally, configure that the **dhcrelay6** service starts when the system boots:

```
# systemctl enable dhcrelay6.service
```

6. Start the **dhcrelay6** service:

```
# systemctl start dhcrelay6.service
```

Additional resources

- **dhcrelay(8)** man page