Red Hat Enterprise Linux 8

네트워킹 인프라 서비스 관리

Red Hat Enterprise Linux 8에서 네트워킹 인프라 서비스 관리 가이드

Last Updated: 2022-07-25

Red Hat Enterprise Linux 8 네트워킹 인프라 서비스 관리

Red Hat Enterprise Linux 8에서 네트워킹 인프라 서비스 관리 가이드

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

법적 공지

초록

이 문서에서는 Red Hat Enterprise Linux 8에서 DNS 및 DHCP와 같은 네트워킹 핵심 인프라 서비스를 설정하고 관리하는 방법을 설명합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 CTO Chris Wright의 메시지 를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 이를 개선할 수 있는지 알려 주십시오.

- 특정 문구에 대한 간단한 의견 작성 방법은 다음과 같습니다.
  1. 문서가 *Multi-page HTML* 형식으로 표시되는지 확인합니다. 또한 문서 오른쪽 상단에 **피드백** 버튼이 있는지 확인합니다.
  2. 마우스 커서를 사용하여 주석 처리하려는 텍스트 부분을 강조 표시합니다.
  3. 강조 표시된 텍스트 아래에 표시되는 **피드백 추가** 팝업을 클릭합니다.
  4. 표시된 지침을 따릅니다.
- Bugzilla를 통해 피드백을 제출하려면 새 티켓을 생성합니다.
  1. Bugzilla 웹 사이트로 이동합니다.
  2. 구성 요소로 **문서**를 사용합니다.
  3. **설명** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
  4. **버그 제출**을 클릭합니다.

1장. BIND DNS 서버 구성 및 관리

DNS(Domain Name System)는 호스트 이름을 해당 IP 주소와 연결하는 분산 데이터베이스 시스템입니다. **BIND** (Berkeley Internet Name Domain)는 DNS 관련 프로그램 세트로 구성됩니다. **라는** 이름 서버가 포함되어 있습니다. **/etc/named.conf** 는 BIND 구성의 기본 구성 파일입니다. 이 섹션에서는 DNS 서버에서 **BIND** 를 설치, 구성 및 관리하는 방법을 중점적으로 설명합니다.

1.1. BIND 설치

bind-utils 패키지를 설치하면 시스템에서 **BIND** 유틸리티를 사용할 수 있습니다.

절차

1. **BIND** 설치:

```
# yum install bind bind-utils
```

2. **named** 서비스를 활성화하고 시작합니다.

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

검증 단계

- **named** 서비스의 상태를 확인합니다.

```
# systemctl status named
```

1.2. BIND를 캐싱 이름 서버로 구성

다음 절차에서는 **BIND** 를 캐싱 이름 서버로 구성하는 방법을 보여줍니다.

사전 요구 사항

- **bind** 패키지가 설치되어 있습니다.

절차

1. 원래 구성 파일의 백업을 수행해야 합니다.

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

2. 다음 변경 사항으로 **/etc/named.conf** 파일을 편집합니다.
   - options 섹션에서 listen-on , **listen-on -v6** 및 **디렉터리** 매개변수의 주석을 제거합니다.

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

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

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

        directory       "/var/named";
```

- **allow-query** 매개변수를 네트워크 주소로 설정합니다. 로컬 네트워크의 호스트만 DNS 서버를 쿼리할 수 있습니다.

```
    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;
        };
};
```

- 제공된 패키지 파일을 다음과 같이 사용합니다.

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

- 사용자 지정 영역 구성에 대한 추가 포함을 생성합니다.

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

3. **/etc/named/example.zones** 파일을 생성하고 다음 영역 구성을 추가합니다.

```
//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: 서버의 영역 역할을 정의합니다.
- master: 권한 있는 서버이며 영역 데이터의 마스터 사본을 유지합니다.
- file: 영역의 데이터베이스 파일을 지정합니다.

4. DNS 데이터 디렉토리 **/var/named/**:로 이동합니다.

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

```
# ls

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

5. forward zone **매개변수로 /var/named/*example.com*.zone** 파일을 생성합니다.

```
$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. 역방향 영역 매개변수를 사용하여 **/var/named/*example.com*.rzone** 파일을 생성합니다.

```
$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. 영역 파일에 보안 권한을 설정합니다.

```
# 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. BIND를 다시 시작합니다.

```
# systemctl restart named
```

검증 단계

- 전달 영역 파일을 확인합니다.

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

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

-

- 역방향 영역 파일을 확인합니다.

```
# 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
```

- 구성을 확인합니다.

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

구성이 올바르면 명령에서 출력을 반환하지 않습니다.

2장. 바인딩되지 않은 DNS 서버 설정

바인딩되지 않은 DNS 서버는 검증, 재귀 및 캐싱 DNS 확인자입니다. **또한 unbound 는 보안에 중점을 두고 있으며 여기에는 기본적으로 DNSSEC(Domain Name System Security Extensions)가 활성화되어 있습니다.**

2.1. UNBOUND를 캐싱 DNS 서버로 구성

기본적으로 바인딩되지 않은 DNS 서비스는 확인 및 캐시 성공 및 실패한 조회입니다. 그러면 서비스는 캐시의 동일한 레코드에 대한 요청에 응답합니다.

절차

1. **unbound 패키지를 설치합니다.**

```
# yum install unbound
```

2. **/etc/cabundle/cabundle.conf 파일을 편집하고 server 절에서 다음과 같이 변경합니다.**

   a. **인터페이스 매개변수를 추가하여 바인딩되지 않은 서비스가 쿼리를 수신 대기하는 IP 주소를 구성합니다. 예를 들면 다음과 같습니다.**

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

이러한 설정을 사용하면 unbound 가 지정된 IPv4 및 IPv6 주소에서만 수신 대기합니다.

인터페이스를 필수로 제한하면 인터넷과 같은 인증되지 않은 네트워크의 클라이언트가 이 DNS 서버로 쿼리를 보낼 수 없습니다.

   b. **access-control 매개변수를 추가하여 클라이언트가 DNS 서비스를 쿼리할 수 있는 서브넷에서 구성합니다. 예를 들면 다음과 같습니다.**

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

3. **unbound** 서비스를 원격으로 관리하기 위한 개인 키 및 인증서를 생성합니다.

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

이 단계를 건너뛰면 다음 단계의 구성을 확인하면 누락된 파일이 보고됩니다. 그러나 바인딩되지 않은 서비스는 누락된 경우 파일을 자동으로 생성합니다.

4. 구성 파일을 확인합니다.

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

5. 들어오는 **DNS** 트래픽을 허용하도록 **firewalld** 규칙을 업데이트합니다.

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

6. **unbound** 서비스를 활성화하고 시작합니다.

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

검증

1. **localhost** 인터페이스에서 수신 대기하는 바인딩되지 않은 **DNS** 서버를 쿼리하여 도메인을 확인합니다.

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

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

처음으로 레코드를 쿼리한 후 **unbound** 는 해당 캐시에 항목을 추가합니다.

2. 이전 쿼리를 반복합니다.

```
# dig @localhost www.example.com
```

```
...
www.example.com.    85332    IN    A    198.51.100.34

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

캐시된 항목으로 인해 항목이 만료될 때까지 동일한 레코드에 대한 추가 요청이 훨씬 더 빨라집니다.

다음 단계

- 이 DNS 서버를 사용하도록 네트워크에서 클라이언트를 구성합니다. 예를 들어 nmcli 유틸리티를 사용하여 NetworkManager 연결 프로필에서 DNS 서버의 IP를 설정합니다.

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

추가 리소스

- unbound.conf(5) man page

3장. DHCP 서비스 제공

DHCP(Dynamic host Configuration Protocol)는 클라이언트에 IP 정보를 자동으로 할당하는 네트워크 프로토콜입니다.

이 섹션에서는 dhcpd 서비스에 대한 일반적인 정보와 DHCP 서버 및 DHCP 릴레이를 설정하는 방법을 설명합니다.

프로시저에 IPv4 및 IPv6 네트워크에서 DHCP를 제공하는 데 다른 단계가 필요한 경우 이 장의 섹션에 두 프로토콜의 프로시저가 포함됩니다.

3.1. 고정 IP 주소와 동적 IP 주소 지정의 차이점

고정 IP 주소 지정

장치에 고정 IP 주소를 할당하면 수동으로 변경하지 않는 한 이 주소는 시간이 지남에 따라 변경되지 않습니다. 원하는 경우 고정 IP 주소 지정을 사용합니다.

- DNS 및 인증 서버와 같은 서버에 대한 네트워크 주소 일관성을 보장하기 위해
- 다른 네트워크 인프라와 독립적으로 작동하는 대역 외 관리 장치를 사용하려면 다음을 수행합니다.

동적 IP 주소 지정

동적 IP 주소를 사용하도록 장치를 구성하면 시간이 지남에 따라 주소가 변경될 수 있습니다. 이러한 이유로 동적 주소는 호스트를 재부팅한 후 IP 주소가 다를 수 있으므로 네트워크에 연결되는 장치에 일반적으로 사용됩니다.

동적 IP 주소는 보다 유연하고 설정하기 쉽고 관리할 수 있습니다. DHCP(Dynamic Host Control Protocol)는 호스트에 네트워크 구성을 동적으로 할당하는 기존 방법입니다.

참고

정적 또는 동적 IP 주소를 사용할 시기를 정의하는 엄격한 규칙은 없습니다. 사용자의 요구, 기본 설정 및 네트워크 환경에 따라 달라집니다.

3.2. DHCP 트랜잭션 단계

DHCP는 Discovery, Offer, Request, Acknowledgement, DORA 프로세스라고도 하는 네 단계로 작동합니다. DHCP는 이 프로세스를 사용하여 클라이언트에 IP 주소를 제공합니다.

검색

DHCP 클라이언트는 메시지를 전송하여 네트워크에서 DHCP 서버를 검색합니다. 이 메시지는 네트워크 및 데이터 링크 계층에서 브로드캐스트됩니다.

offer

DHCP 서버는 클라이언트에서 메시지를 수신하고 DHCP 클라이언트에 IP 주소를 제공합니다. 이 메시지는 데이터 링크 계층에서 유니캐스트이지만 네트워크 계층에서 브로드캐스트됩니다.

요청

DHCP 클라이언트는 제공된 IP 주소에 대한 DHCP 서버를 요청합니다. 이 메시지는 데이터 링크 계층에서 유니캐스트이지만 네트워크 계층에서 브로드캐스트됩니다.

감사 인사

DHCP 서버에서 DHCP 클라이언트에 확인을 보냅니다. 이 메시지는 데이터 링크 계층에서 유니캐스트이지만 네트워크 계층에서 브로드캐스트됩니다. DHCP DORA 프로세스의 최종 메시지입니다.

3.3. DHCPV4 및 DHCPV6에 DHCPD를 사용할 때의 차이점

dhcpd 서비스는 한 서버에서 DHCPv4 및 DHCPv6를 모두 제공할 수 있도록 지원합니다. 그러나 각 프로토콜에 DHCP를 제공하려면 별도의 구성 파일이 있는 dhcpd 의 별도의 인스턴스가 필요합니다.

DHCPv4

- 구성 파일: /etc/dhcp/dhcpd.conf
- systemd 서비스 이름: dhcpd

DHCPv6

- 구성 파일: /etc/dhcp/dhcpd6.conf

- systemd 서비스 이름: dhcpd6

3.4. DHCPD 서비스의 리스 데이터베이스

DHCP 리스는 dhcpd 서비스가 클라이언트에 네트워크 주소를 할당하는 기간입니다. dhcpd 서비스는 DHCP 리스를 다음 데이터베이스에 저장합니다.

- DHCPv4의 경우: /var/lib/dhcpd/dhcpd.leases
- DHCPv6의 경우: /var/lib/dhcpd/dhcpd6.leases

주의

데이터베이스 파일을 수동으로 업데이트하면 데이터베이스가 손상될 수 있습니다.

리스 데이터베이스에는 MAC(Media Access Control) 주소에 할당된 IP 주소 또는 리스가 만료되는 타임스탬프와 같이 할당된 리스에 대한 정보가 포함되어 있습니다. 리스 데이터베이스의 모든 타임스탬프는 UTC(Universal Time)에 있습니다.

dhcpd 서비스는 정기적으로 데이터베이스를 다시 생성합니다.

1. 서비스는 기존 파일의 이름을 변경합니다.
   - /var/lib/dhcpd/dhcpd.leases to /var/lib/dhcpd/dhcpd.leases~
   - /var/lib/dhcpd/dhcpd6.leases to /var/lib/dhcpd/dhcpd6.leases~
2. 이 서비스는 알려진 모든 리스를 새로 생성된 /var/lib/dhcpd/dhcpd.leases 및

/var/lib/dhcpd/dhcpd6.leases 파일에 씁니다.

추가 리소스

- dhcpd.leases(5) man page
- 손상된 리스 데이터베이스 복원

3.5. DHCPV6과 RADVD 비교

IPv6 네트워크에서 라우터 알림 메시지만 IPv6 기본 게이트웨이에 대한 정보를 제공합니다. 결과적으로 기본 게이트웨이 설정이 필요한 서브넷에서 DHCPv6을 사용하려면 라우터 알림 데몬(radvd)과 같은 라우터 알림 서비스를 추가로 구성해야 합니다.

radvd 서비스는 라우터 알림 패킷의 플래그를 사용하여 DHCPv6 서버의 가용성을 발표합니다.

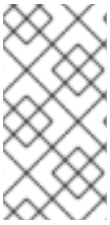
이 섹션에서는 DHCPv6 및 radvd 를 비교하고 radvd 구성에 대한 정보를 제공합니다.

	DHCPv6	radvd
기본 게이트웨이에 대한 정보 제공	제공되지 않음	제공됨
개인 정보 보호를 위해 임의의 주소 보장	제공됨	제공되지 않음
추가 네트워크 구성 옵션 전송	제공됨	제공되지 않음
미디어 액세스 제어(MAC) 주소를 IPv6 주소에 매핑	제공됨	제공되지 않음

3.6. IPV6 라우터를 위한 RADVD 서비스 구성

라우터 광고 데몬(radvd)은 IPv6 상태 비저장 자동 구성에 필요한 라우터 알림 메시지를 전송합니다. 이를 통해 사용자는 주소, 설정, 경로를 자동으로 구성하고 이러한 알림을 기반으로 기본 라우터를 선택할 수 있습니다.

이 섹션의 절차는 radvd 를 구성하는 방법을 설명합니다.



참고

radvd 서비스에서만 **/64** 접두사를 설정할 수 있습니다. 다른 접두사를 사용하려면 **DHCPv6**을 사용합니다.

사전 요구 사항

- **root** 사용자로 로그인합니다.

절차

1. **radvd** 패키지를 설치합니다.

```
# yum install radvd
```

2. **/etc/radvd.conf** 파일을 편집하고 다음 구성을 추가합니다.

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

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

이러한 설정은 **2001:db8:0:1::/64** 서브넷의 **enp1s0** 장치에 라우터 알림 메시지를 전송하도록 **radvd** 를 구성합니다. **AdvManaged flags on** 설정은 클라이언트가 **DHCP** 서버에서 **IP** 주소를 수신해야 함을 정의하고, **Adv OtherConfigflag** 매개 변수는 클라이언트가 **DHCP** 서버에서 주소 이외의 정보를 수신해야 함을 정의합니다.

3. 선택적으로 시스템이 부팅될 때 **radvd** 가 자동으로 시작되도록 구성합니다.

```
# systemctl enable radvd
```

4. **radvd** 서비스를 시작합니다.

```
# systemctl start radvd
```

5. 선택적으로, 라우터 광고 패키지의 콘텐츠와 구성된 값 **radvd send**을 표시합니다.

```
# radvdump
```

추가 리소스

- **radvd.conf(5) man page**
- **/usr/share/doc/radvd/radvd.conf.example**
- IPv6 라우터 알림에서 64비트 이외의 접두사 길이를 사용할 수 있습니까?

3.7. DHCP 서버의 네트워크 인터페이스 설정

기본적으로 **dhcpd** 서비스는 서비스의 구성 파일에 정의된 서브넷에 **IP** 주소가 있는 네트워크 인터페이스에서만 요청합니다.

예를 들어 다음 시나리오에서는 **dhcpd** 가 **enp0s1** 네트워크 인터페이스에서만 수신 대기합니다.

- **/etc/dhcp/dhcpd.conf** 파일에 **192.0.2.0/24** 네트워크의 서브넷 정의만 있습니다.
- **enp0s1** 네트워크 인터페이스는 **192.0.2.0/24** 서브넷에 연결되어 있습니다.
- **enp7s0** 인터페이스는 다른 서브넷에 연결되어 있습니다.

DHCP 서버에 동일한 네트워크에 연결된 여러 네트워크 인터페이스가 포함되어 있지만 서비스는 특정 인터페이스에서만 수신 대기해야 하는 경우에만 이 섹션의 절차를 따릅니다.

IPv4, IPv6 또는 두 프로토콜 모두에 **DHCP**를 제공할지 여부에 따라 다음 절차를 참조하십시오.

- IPv4 네트워크

- IPv6 네트워크

사전 요구 사항

- root 사용자로 로그인합니다.
- dhcp-server 패키지가 설치되어 있어야 합니다.

절차

- IPv4 네트워크의 경우:

1. /usr/lib/systemd/system/dhcpd.service 파일을 /etc/systemd/system/ 디렉터리에 복사합니다.

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

/usr/lib/systemd/system/dhcpd.service 파일을 편집하지 마십시오. dhcp-server 패키지의 향후 업데이트에서는 변경 사항을 재정의할 수 있습니다.

2. /etc/systemd/system/dhcpd.service 파일을 편집하고 인터페이스의 이름을 추가합니다. dhcpd 는 ExecStart 매개변수의 명령에 수신해야 합니다.

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

이 예제에서는 dhcpd 가 *enp0s1* 및 enp7s0 인터페이스에서만 수신 대기하도록 구성합니다.

3. systemd 관리자 구성을 다시 로드합니다.

```
# systemctl daemon-reload
```

4. dhcpd 서비스를 다시 시작합니다.

```
# systemctl restart dhcpd.service
```

- IPv6 네트워크의 경우:

1. /usr/lib/systemd/system/dhcpd6.service 파일을 /etc/systemd/system/ 디렉터리에 복사합니다.

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

/usr/lib/systemd/system/dhcpd6.service 파일을 편집하지 마십시오. dhcp-server 패키지의 향후 업데이트에서는 변경 사항을 재정의할 수 있습니다.

2. /etc/systemd/system/dhcpd6.service 파일을 편집하고 ExecStart 매개 변수의 명령에 dhcpd 가 수신해야 하는 인터페이스의 이름을 추가합니다.

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

이 예제에서는 dhcpd 가 *enp0s1* 및 enp7s0 인터페이스에서만 수신 대기하도록 구성합니다.

3. systemd 관리자 구성을 다시 로드합니다.

```
# systemctl daemon-reload
```

4. dhcpd6 서비스를 다시 시작합니다.

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

3.8. DHCP 서버에 직접 연결된 서브넷의 DHCP 서비스 설정

DHCP 서버가 DHCP 요청에 응답해야 하는 서브넷에 직접 연결된 경우 다음 절차를 사용하십시오. 서버의 네트워크 인터페이스에 이 서브넷의 IP 주소가 할당된 경우입니다.

IPv4, IPv6 또는 두 프로토콜 모두에 DHCP를 제공할지 여부에 따라 다음 절차를 참조하십시오.

- IPv4 네트워크
- IPv6 네트워크

사전 요구 사항

- root 사용자로 로그인합니다.
- dhcp-server 패키지가 설치되어 있어야 합니다.

절차

- IPv4 네트워크의 경우:

1. /etc/dhcp/dhcpd.conf 파일을 편집합니다.

a. 선택적으로 다른 지시문에 이러한 설정이 포함되어 있지 않은 경우 dhcpd 에서 기본값으로 사용하는 글로벌 매개 변수를 추가합니다.

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

이 예에서는 example.com 으로 연결의 기본 도메인 이름을 설정하고 기본 리스 시간을 86400 초(1일)로 설정합니다.

b. 새 줄에 authoritative 문을 추가합니다.

```
authoritative;
```

중요

authoritative 문이 없으면 dhcpd 서비스는 클라이언트에서 풀 외부에 있는 주소를 요청하는 경우 DHCPNAK 를 사용하여 DHCPREQUEST 메시지를 응답하지 않습니다.

c.

서버 인터페이스에 직접 연결된 각 **IPv4** 서브넷에 대해 서브넷 선언을 추가합니다.

```
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;
}
```

이 예제에서는 **192.0.2.0/24** 네트워크에 대한 서브넷 선언을 추가합니다. 이 구성을 사용하면 **DHCP** 서버에서 이 서브넷에서 **DHCP** 요청을 전송하는 클라이언트에 다음 설정을 할당합니다.

- **range** 매개변수에 정의된 범위에서 사용 가능한 **IPv4** 주소
- 이 서브넷에 대한 **DNS** 서버의 **IP: 192.0.2.1**
- 이 서브넷의 기본 게이트웨이**: 192.0.2.1**
- 이 서브넷의 브로드캐스트 주소**: 192.0.2.255**
- 이 서브넷의 클라이언트는 최대 임대 시간입니다. 이 서브넷의 클라이언트는 **IP**를 해제하고 서버에 새 요청을 보냅니다. **172800** 초 **(2**일**)**

2.

선택적으로 시스템이 부팅될 때 **dhcpd** 가 자동으로 시작되도록 구성합니다.

```
# systemctl enable dhcpd
```

3.

dhcpd 서비스를 시작합니다.

```
# systemctl start dhcpd
```

- **IPv6** 네트워크의 경우**:**

1. **/etc/dhcp/dhcpd6.conf** 파일을 편집합니다.

a. 선택적으로 다른 지시문에 이러한 설정이 포함되어 있지 않은 경우 **dhcpd** 에서 기본값으로 사용하는 글로벌 매개 변수를 추가합니다.

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

이 예에서는 **example.com** 으로 연결의 기본 도메인 이름을 설정하고 기본 리스 시간을 **86400** 초(1일)로 설정합니다.

b. 새 줄에 **authoritative** 문을 추가합니다.

```
authoritative;
```

중요

authoritative 문이 없으면 **dhcpd** 서비스는 클라이언트에서 풀 외부에 있는 주소를 요청하는 경우 **DHCPNAK** 를 사용하여 **DHCPREQUEST** 메시지를 응답하지 않습니다.

c. 서버 인터페이스에 직접 연결된 각 **IPv6** 서브넷에 대해 서브넷 선언을 추가합니다.

```
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;
}
```

이 예제에서는 **2001:db8:0:1::/64** 네트워크에 대한 서브넷 선언을 추가합니다. 이 구성을 사용하면 **DHCP** 서버에서 이 서브넷에서 **DHCP** 요청을 전송하는 클라이언트에 다음 설정을 할당합니다.

- **range6** 매개변수에 정의된 범위에서 사용 가능한 **IPv6** 주소입니다.
- 이 서브넷의 **DNS** 서버의 **IP**는 **2001:db8:0:1::1** 입니다.

- 이 서브넷의 클라이언트가 IP를 해제한 후 서버에 새 요청을 전송하는 최대 리스 시간은 172800 초(2일)입니다.

  IPv6에서는 라우터 알림 메시지를 사용하여 기본 게이트웨이를 식별해야 합니다.

2. 선택적으로 시스템이 부팅될 때 dhcpd6 이 자동으로 시작되도록 구성합니다.

```
# systemctl enable dhcpd6
```

3. dhcpd6 서비스를 시작합니다.

```
# systemctl start dhcpd6
```

추가 리소스

- dhcp-options(5) man page
- dhcpd.conf(5) 매뉴얼 페이지의 권한 있는 문 섹션
- /usr/share/doc/dhcp-server/dhcpd.conf.example
- /usr/share/doc/dhcp-server/dhcpd6.conf.example

3.9. DHCP 서버에 직접 연결되지 않은 서브넷의 DHCP 서비스 설정

DHCP 서버가 DHCP 요청에 응답해야 하는 서브넷에 직접 연결되지 않은 경우 다음 절차를 사용하십시오. 이는 서버가 제공해야 하는 서브넷에 직접 연결된 DHCP 서버의 인터페이스가 없기 때문에 DHCP 릴레이 에이전트가 DHCP 서버로 요청을 전달하는 경우입니다.

IPv4, IPv6 또는 두 프로토콜 모두에 DHCP를 제공할지 여부에 따라 다음 절차를 참조하십시오.

- IPv4 네트워크

- IPv6 네트워크

사전 요구 사항

- root 사용자로 로그인합니다.
- dhcp-server 패키지가 설치되어 있어야 합니다.

절차

- IPv4 네트워크의 경우:

1. /etc/dhcp/dhcpd.conf 파일을 편집합니다.

a. 선택적으로 다른 지시문에 이러한 설정이 포함되어 있지 않은 경우 dhcpd 에서 기본값으로 사용하는 글로벌 매개 변수를 추가합니다.

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

이 예에서는 example.com 으로 연결의 기본 도메인 이름을 설정하고 기본 리스 시간을 86400 초(1일)로 설정합니다.

b. 새 줄에 authoritative 문을 추가합니다.

```
authoritative;
```

중요

authoritative 문이 없으면 dhcpd 서비스는 클라이언트에서 풀 외부에 있는 주소를 요청하는 경우 DHCPNAK 를 사용하여 DHCPREQUEST 메시지를 응답하지 않습니다.

c. 서버 인터페이스에 직접 연결되지 않은 IPv4 서브넷에 대해 다음과 같은 공유 네트워크 선언을 추가합니다.

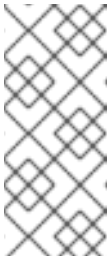
```
shared-network example {
  option domain-name-servers 192.0.2.1;
  ...

  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;
  }
  ...
}
```

이 예에서는 **192.0.2.0/24** 및 **198.51.100.0/24** 네트워크에 대한 서브넷 선언이 포함된 공유 네트워크 선언을 추가합니다. 이 구성을 사용하면 **DHCP** 서버에서 다음 서브넷 중 하나에서 **DHCP** 요청을 전송하는 클라이언트에 다음 설정을 할당합니다.

- 두 서브넷 모두에서 클라이언트의 **DNS** 서버의 **IP**는 **192.0.2.1** 입니다.
- 클라이언트가 요청을 보낸 서브넷에 따라 **range** 매개변수에 정의된 범위에서 사용 가능한 **IPv4** 주소입니다.
- 기본 게이트웨이는 클라이언트가 요청을 보낸 서브넷에 따라 **192.0.2.1** 또는 **198.51.100.1** 입니다.

d. 서버가 직접 연결되고 위의 **shared-network** 에 지정된 원격 서브넷에 도달하는 데 사용되는 서브넷에 서브넷 선언을 추가합니다.

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



참고

서버가 이 서브넷에 **DHCP** 서비스를 제공하지 않으면 예제에 표시된 대로 서브넷 선언이 비어 있어야 합니다. 직접 연결된 서브넷에 대한 선언이 없으면 **dhcpd** 가 시작되지 않습니다.

2. 선택적으로 시스템이 부팅될 때 **dhcpd** 가 자동으로 시작되도록 구성합니다.

```
# systemctl enable dhcpd
```

3. **dhcpd** 서비스를 시작합니다.

```
# systemctl start dhcpd
```

- **IPv6** 네트워크의 경우:

1. **/etc/dhcp/dhcpd6.conf** 파일을 편집합니다.

a. 선택적으로 다른 지시문에 이러한 설정이 포함되어 있지 않은 경우 **dhcpd** 에서 기본값으로 사용하는 글로벌 매개 변수를 추가합니다.

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

이 예에서는 **example.com** 으로 연결의 기본 도메인 이름을 설정하고 기본 리스 시간을 **86400** 초(**1**일)로 설정합니다.

b. 새 줄에 **authoritative** 문을 추가합니다.

```
authoritative;
```

중요

authoritative 문이 없으면 **dhcpd** 서비스는 클라이언트에서 풀 외부에 있는 주소를 요청하는 경우 **DHCPNAK** 를 사용하여 **DHCPREQUEST** 메시지를 응답하지 않습니다.

c. 서버 인터페이스에 직접 연결되지 않은 **IPv6** 서브넷에 대해 다음과 같은 공유 네트워크 선언을 추가합니다.

```
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
  }
  ...
}
```

이 예제에서는 **2001:db8:0:1::1:0/120** 및 **2001:db8:0:1::2:0/120** 네트워크 둘 다에 **subnet6** 선언이 포함된 공유 네트워크 선언을 추가합니다. 이 구성을 사용하면 **DHCP** 서버에서 다음 서브넷 중 하나에서 **DHCP** 요청을 전송하는 클라이언트에 다음 설정을 할당합니다.

- 두 서브넷 모두에서 클라이언트의 **DNS** 서버의 **IP**는 **2001:db8:0:1::1:1** 입니다.

- 클라이언트가 요청을 보낸 서브넷에 따라 **range6** 매개변수에 정의된 범위에서 사용 가능한 **IPv6** 주소입니다.

  IPv6에서는 라우터 알림 메시지를 사용하여 기본 게이트웨이를 식별해야 합니다.

d. 서버가 직접 연결되고 위의 **shared-network** 에 지정된 원격 서브넷에 도달하는 데 사용되는 서브넷에 **subnet6** 선언을 추가합니다.

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

참고

서버가 이 서브넷에 **DHCP** 서비스를 제공하지 않으면 예제에 표시된 대로 **subnet6** 선언이 비어 있어야 합니다. 직접 연결된 서브넷에 대한 선언이 없으면 **dhcpd** 가 시작되지 않습니다.

2. 선택적으로 시스템이 부팅될 때 **dhcpd6** 이 자동으로 시작되도록 구성합니다.

```
# systemctl enable dhcpd6
```

3. **dhcpd6** 서비스를 시작합니다.

-

```
# systemctl start dhcpd6
```

추가 리소스

- dhcp-options(5) man page
- dhcpd.conf(5) 매뉴얼 페이지의 권한 있는 문 섹션
- /usr/share/doc/dhcp-server/dhcpd.conf.example
- /usr/share/doc/dhcp-server/dhcpd6.conf.example
- DHCP 릴레이 에이전트 설정

3.10. DHCP를 사용하여 호스트에 고정 주소 할당

호스트 선언을 사용하여 호스트의 MAC(Media Access Control) 주소에 고정 IP 주소를 할당하도록 DHCP 서버를 구성할 수 있습니다. 예를 들어 이 방법을 사용하여 항상 동일한 IP 주소를 서버 또는 네트워크 장치에 할당합니다.

중요

MAC 주소에 고정 IP 주소를 구성하는 경우 고정 주소 및 fixed-address 6 매개변수에 지정한 주소 풀의 IP 주소 외부에 있어야 합니다.

IPv4, IPv6 또는 두 프로토콜 모두에 대해 고정 주소를 구성할지 여부에 따라 다음 절차를 참조하십시오.

- IPv4 Networks
- IPv6 네트워크

사전 요구 사항

- dhcpd 서비스가 구성되어 실행 중입니다.
- root 사용자로 로그인합니다.

절차

- IPv4 네트워크의 경우:

1. /etc/dhcp/dhcpd.conf 파일을 편집합니다.

a. 호스트 선언을 추가합니다.

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

이 예제에서는 DHCP 서버를 구성하여 항상 192.0.2.130 IP 주소를 52:54:00:72:2f:6e MAC 주소가 있는 호스트에 할당합니다.

dhcpd 서비스는 호스트 선언의 이름이 아닌 고정 주소 매개변수에 지정된 MAC 주소로 시스템을 식별합니다. 결과적으로 이 이름을 다른 호스트 선언과 일치하지 않는 문자열로 설정할 수 있습니다. 여러 네트워크에 대해 동일한 시스템을 구성하려면 다른 이름을 사용합니다. 그러지 않으면 dhcpd 가 시작되지 않습니다.

b. 선택적으로 이 호스트에 특정된 호스트 선언에 추가 설정을 추가합니다.

2. dhcpd 서비스를 다시 시작합니다.

```
# systemctl start dhcpd
```

- IPv6 네트워크의 경우:

1. /etc/dhcp/dhcpd6.conf 파일을 편집합니다.

a. 호스트 선언을 추가합니다.

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

이 예에서는 **DHCP** 서버가 항상 **2001:db8:0:1::20 IP** 주소를 **52:54:00:72:2f:6e MAC** 주소를 갖는 호스트에 할당하도록 구성합니다.

dhcpd 서비스는 호스트 선언의 이름이 아닌 **fixed-address6** 매개변수에 지정된 **MAC** 주소로 시스템을 식별합니다. 결과적으로 다른 호스트 선언에 고유한 경우 이 이름을 모든 문자열로 설정할 수 있습니다. 여러 네트워크에 대해 동일한 시스템을 구성하려면 다른 이름을 사용합니다. 그러지 않으면 **dhcpd** 가 시작되지 않습니다.

b. 선택적으로 이 호스트에 특정된 호스트 선언에 추가 설정을 추가합니다.

2. **dhcpd6** 서비스를 다시 시작합니다.

```
# systemctl start dhcpd6
```

추가 리소스

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

3.11. 그룹 선언을 사용하여 여러 호스트, 서브넷 및 공유 네트워크에 동시에 매개 변수 적용

그룹 선언을 사용하면 동일한 매개 변수를 여러 호스트, 서브넷 및 공유 네트워크에 적용할 수 있습니다.

이 섹션의 절차에서는 호스트에 대한 그룹 선언을 사용하여 설명하지만 단계는 서브넷 및 공유 네트워크에 대해 동일합니다.

IPv4, IPv6 또는 두 프로토콜 모두에 대해 그룹을 구성할지 여부에 따라 다음에 대한 절차를 참조하십시오.

- IPv4 네트워크
- IPv6 네트워크

사전 요구 사항

- dhcpd 서비스가 구성되어 실행 중입니다.
- root 사용자로 로그인합니다.

절차

- IPv4 네트워크의 경우:

1. /etc/dhcp/dhcpd.conf 파일을 편집합니다.

a. 그룹 선언을 추가합니다.

```
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;
  }
}
```

이 그룹 정의에서는 두 개의 호스트 항목을 그룹화합니다. dhcpd 서비스는 옵션 domain-name-servers 매개 변수에 설정된 값을 그룹의 두 호스트에 적용합니다.

b. 선택적으로 이러한 호스트에 고유한 그룹 선언에 추가 설정을 추가합니다.

2. **dhcpd** 서비스를 다시 시작합니다.

```
# systemctl start dhcpd
```

- **IPv6** 네트워크의 경우:

1. **/etc/dhcp/dhcpd6.conf** 파일을 편집합니다.

a. 그룹 선언을 추가합니다.

```
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;
  }
}
```

이 그룹 정의에서는 두 개의 호스트 항목을 그룹화합니다. **dhcpd** 서비스는 옵션 **dhcp6.domain-search** 매개 변수에 설정된 값을 그룹의 두 호스트에 적용합니다.

b. 선택적으로 이러한 호스트에 고유한 그룹 선언에 추가 설정을 추가합니다.

2. **dhcpd6** 서비스를 다시 시작합니다.

```
# systemctl start dhcpd6
```

추가 리소스

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

3.12. 손상된 리스 데이터베이스 복원

DHCP 서버가 Corrupt lease file - 가능한 데이터 손실 과 같은 리스 데이터베이스와 관련된 오류를 기록하는 경우, 생성된 dhcpd 서비스 사본에서 리스 데이터베이스를 복원할 수 있습니다. 이 복사본은 데이터베이스의 최신 상태를 반영하지 않을 수 있습니다.

주의

백업으로 교체하지 않고 리스 데이터베이스를 제거하면 현재 할당된 리스에 대한 모든 정보가 손실됩니다. 결과적으로 DHCP 서버는 이전에 다른 호스트에 할당되었으며 아직 만료되지 않은 클라이언트에 리스를 할당할 수 있었습니다. 이로 인해 IP 충돌이 발생합니다.

DHCPv4, DHCPv6 또는 두 데이터베이스를 모두 복원할지 여부에 따라 다음 절차를 참조하십시오.

- DHCPv4 리스 데이터베이스 복원
- DHCPv6 리스 데이터베이스 복원

사전 요구 사항

- root 사용자로 로그인합니다.
- 리스 데이터베이스가 손상되었습니다.

절차

- **DHCPv4** 리스 데이터베이스 복원:

1. **dhcpd** 서비스를 중지합니다.

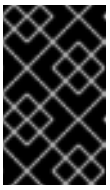
```
# systemctl stop dhcpd
```

2. 손상된 리스 데이터베이스의 이름을 변경합니다.

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

3. **lease** 데이터베이스를 새로 고칠 때 **dhcp** 서비스에서 생성한 리스 데이터베이스의 복사본을 복원합니다.

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



중요

최근 리스 데이터베이스 백업이 있는 경우 이 백업을 대신 복원합니다.

4. **dhcpd** 서비스를 시작합니다.

```
# systemctl start dhcpd
```

- **DHCPv6** 리스 데이터베이스 복원:

1. **dhcpd6** 서비스를 중지합니다.

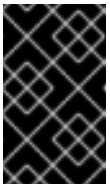
```
# systemctl stop dhcpd6
```

2. 손상된 리스 데이터베이스의 이름을 변경합니다.

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

3. **lease** 데이터베이스를 새로 고칠 때 **dhcp** 서비스에서 생성한 리스 데이터베이스의 복사본을 복원합니다.

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



중요

최근 리스 데이터베이스 백업이 있는 경우 이 백업을 대신 복원합니다.

4. **dhcpd6** 서비스를 시작합니다.

```
# systemctl start dhcpd6
```

추가 리소스

- **dhcpd** 서비스의 리스 데이터베이스

3.13. DHCP 릴레이 에이전트 설정

DHCP Relay Agent(dhcrelay)를 사용하면 **DHCP** 서버가 없는 서브넷에서 다른 서브넷의 **DHCP** 서버로 **DHCP** 및 **BOOTP** 요청을 릴레이할 수 있습니다. **DHCP** 클라이언트에서 정보를 요청하면 **DHCP** 릴레이 에이전트는 지정된 **DHCP** 서버 목록으로 요청을 전달합니다. **DHCP** 서버에서 응답을 반환하면 **DHCP** 릴레이 에이전트는 이 요청을 클라이언트에 전달합니다.

IPv4, IPv6 또는 두 프로토콜 모두에 대한 **DHCP** 릴레이를 설정할지 여부에 따라 다음 절차를 참조하십시오.

- **IPv4** 네트워크
- **IPv6** 네트워크

사전 요구 사항

- **root** 사용자로 로그인합니다.

절차

절차

- **IPv4** 네트워크의 경우**:**

1. **dhcp-relay** 패키지를 설치합니다**.**

```
# yum install dhcp-relay
```

2. **/lib/systemd/system/dhcrelay.service** 파일을 **/etc/systemd/system/** 디렉터리에 복사합니다**.**

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

/usr/lib/systemd/system/dhcrelay.service 파일을 편집하지 마십시오**. dhcp-relay** 패키지의 향후 업데이트로 인해 변경 사항을 재정의할 수 있습니다**.**

3. **/etc/systemd/system/dhcrelay.service** 파일을 편집하고 서브넷을 담당하는 **DHCPv4** 서버의 **IP** 주소 목록과 함께 **-i** *인터페이스* 매개 변수를 추가합니다**.**

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

이러한 추가 매개 변수를 사용하여 **dhcrelay** 는 **enp1s0** 인터페이스에서 **DHCPv4** 요청을 수신 대기하고 **IP 192.0.2.1** 을 사용하여 **DHCP** 서버로 전달합니다**.**

4. **systemd** 관리자 구성을 다시 로드합니다**.**

```
# systemctl daemon-reload
```

5. 선택적으로 시스템이 부팅될 때 **dhcrelay** 서비스가 시작되도록 구성합니다**.**

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

6. **dhcrelay** 서비스를 시작합니다**.**

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

- **IPv6** 네트워크의 경우:

1. **dhcp-relay** 패키지를 설치합니다.

```
# yum install dhcp-relay
```

2. **/lib/systemd/system/dhcrelay.service** 파일을 **/etc/systemd/system/** 디렉터리에 복사하고 파일 이름을 **dhcrelay6.service** 로 복사합니다.

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

/usr/lib/systemd/system/dhcrelay.service 파일을 편집하지 마십시오. **dhcp-relay** 패키지의 향후 업데이트로 인해 변경 사항을 재정의할 수 있습니다.

3. **/etc/systemd/system/dhcrelay6.service** 파일을 편집하고 **-l *receiving_interface*** 및 **-u *outgoing_interface*** 매개변수를 추가합니다.

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

이러한 추가 매개 변수를 사용하여 **dhcrelay** 는 **enp1s0** 인터페이스에서 **DHCPv6** 요청을 수신 대기하고 **enp7s0** 인터페이스에 연결된 네트워크로 전달합니다.

4. **systemd** 관리자 구성을 다시 로드합니다.

```
# systemctl daemon-reload
```

5. 선택적으로 시스템이 부팅될 때 **dhcrelay6** 서비스가 시작되도록 구성합니다.

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

6. **dhcrelay6** 서비스를 시작합니다.

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

추가 리소스

- **dhcrelay(8)** 도움말 페이지