SECTION 10 Configure DHCP Pools and Failover

DHCP pools and failover are not new features of the dhcpd. But they are new in the CLE curriculum.

Objectives

1. Configure DHCP Pools
2. Configure DHCP Failover

Objective 1 Configure DHCP Pools

The **pool** declaration can be used to specify a pool of addresses that will be treated differently than any other pool of addresses, even on the same network segment or subnet.

For example, you may want to provide a large set of addresses that can be assigned to DHCP clients that are registered to your DHCP server, while providing a smaller set of addresses, possibly with short lease times, that are available for unknown clients.

To do this, you would set up a pair of **pool** declarations:

```
subnet 10.0.0.0 netmask 255.255.255.0 {
    option routers 10.0.0.254;

# Unknown clients get this pool.
    pool {
        option domain-name-servers 10.0.0.254;
        max-lease-time 300;
        range 10.0.0.100 10.0.0.250;
        allow unknown-clients;
    }

# Known clients get this pool.
    pool {
        option domain-name-servers 10.0.0.251, 10.0.0.252;
        max-lease-time 28800;
        range 10.0.0.5 10.0.0.99;
        deny unknown-clients;
    }
}
```

It is also possible to set up entirely different subnets for known and unknown clients. “Known clients” mean, that an IP address is assigned to a MAC address using the **host** statement. Pools exist at the level of shared networks, so address ranges within pool declarations can be on different subnets.

As you can see in the preceding example, pools can have permit lists that control which clients are allowed access to the pool and which aren't.

Each entry in a pool's permit list is introduced with the allow or deny keyword. If a pool has a permit list, then only those clients that match specific entries on the permit list will be eligible to be assigned addresses from the pool.

If a pool has a deny list, then only those clients that do not match any entries on the deny list will be eligible. If both permit and deny lists exist for a pool, then only clients that match the permit list and do not match the deny list will be allowed access.

Objective 2 Configure DHCP Failover

This objective covers the following topics:

- Basics of DHCP Failover
- Configure Failover

Basics of DHCP Failover

The failover protocol allows two DHCP servers (and no more than two) to share a common address pool. Each server will have about half of the available IP addresses in the pool at any given time for allocation.

If one server fails, the other server will continue to renew leases out of the pool, and will allocate new addresses out of the half of available addresses that it had before communication with the other server was lost.

When a server starts that has not previously communicated with its failover peer, it must establish communications and synchronize with the peer before it can serve clients. This can happen either because you have just configured your DHCP servers to perform failover for the first time, or because one of your failover servers has failed and lost its database.

The initial recovery process is designed to ensure that when one failover peer loses its database and then resynchronizes, any leases that the failed server gave out before it failed will be honored. When the failed server starts up, it notices that it has no saved failover state and attempts to contact its peer.

When it has established contact, it asks the peer for a complete copy of its lease database. The peer then sends its complete database and sends a message indicating that it is done. The failed server then waits until MCLT (*Maximum Client Lead Time*) has passed. Once MCLT has passed, both servers make the transition back into normal operation.

This waiting period ensures that any leases the failed server may have given out while out of contact with its partner will have expired.

While the failed server is recovering, its partner remains in the partner-down state, which means that it is serving all clients. The failed server provides no service at all to DHCP clients until it has made the transition into normal operation.

In the case where both servers detect that they have never before communicated with their partner, they both come up in this recovery state and follow the procedure we have just described. In this case, no service will be provided to DHCP clients until MCLT has expired.

Configure Failover

In a failover configuration you have to decide which server acts as the primary and which acts as the secondary server. Both servers need to have the same configuration for the basic DHCP service they are sharing. They only differ in the failover setup. Therefore, it is recommended to have a common configuration file on both machines which is included in the file /etc/dhcpd.conf.

The configuration file for the primary server looks like this:

```
failover peer "digitalairlines" {
        primary;
        address 10.0.0.10;
        port 847;
        peer address 10.0.0.12;
        peer port 647;
        max-response-delay 180;
        mclt 1800;
        split 128;
        load balance max seconds 3;
}

# Now we include the identical configuration on both
# machines
include "/etc/dhcpd.conf.master";
```

The statements in this example peer declaration are as follows:

- **primary** and **secondary**. Determines whether the server is primary or secondary, as described earlier.
- **address**. Specifies the IP address or DNS name on which the server should listen for connections from its failover peer.
- **port**. Specifies the TCP port on which the server should listen for connections from its failover peer (default: 647 and 847). This parameter must be specified, because the failover protocol does not yet have a reserved TCP port number.
- **peer address**. Specifies the IP address or DNS name to which the server should connect to reach its failover peer.
- **peer port**. Specifies the TCP port to which the server should connect to reach its failover peer for failover messages. This parameter must be specified, because the failover protocol does not yet have a reserved TCP port number. The **peer port** can be the same as the **port**.
- **max-response-delay**. Specifies how many seconds the DHCP server waits without receiving a message from its failover peer before it assumes that connection has failed.

This number should be small enough that a transient network failure breaks the connection will not result in the servers being out of communication for a long time, but large enough that the server isn't constantly making and breaking connections.

This parameter must be specified.

- **mclt**. Defines the *Maximum Client Lead Time*. It must be specified on the primary, and can be specified also on the secondary server. This is the length of time for which a lease may be renewed by either failover peer without contacting the other.

  The longer you set this, the longer it will take for the running server to recover IP addresses after moving into PARTNER-DOWN state. The shorter you set it, the more load your servers will experience when they are not communicating.

  A value of 1800 is recommended.

- **split**. Specifies the split between the primary and secondary server for the purposes of load balancing.

  Whenever a client makes a DHCP request, the DHCP server runs a hash on the client identification. If the hash comes out to less than the split value, the primary answers. If it comes out to equal to or more than the split, the secondary answers.

  A meaningful value is 128 and can only be configured on the primary server.

- **load balance max seconds**. Specifies a cutoff after which load balancing is disabled. The cutoff is based on the number of seconds since the client sent its first DHCPDISCOVER or DHCPREQUEST message.

  The man pages recommend setting this to 3 or 5. The effect of this is that if one of the failover peers gets into a state where it is responding to failover messages but not responding to some client requests, the other failover peer will take over its client load automatically as the clients retry.

The configuration file for the secondary server looks like this:

```
failover peer "digitalairlines" {
        secondary;
        address 10.0.0.12;
        port 647;
        peer address 10.0.0.10;
        peer port 847;
        max-response-delay 180;
        load balance max seconds 3;
}

# Now we include the identical configuration on both
# machines
include "/etc/dhcpd.conf.master";
```

The differences to the primary server configuration are the statement “secondary”, the missing statements **mclt** and **split** and the interchanged values of **address**, **port**, **peer address** and **peer port**.

The main DHCP server configuration is contained in the file /etc/dhcpd.conf.master, which included in both configurations.

In order to find this file, it needs to be copied into the chroot environment of the DHCP server. The best way to achieve this is to modify the variable DHCPD_CONF_INCLUDE_FILES in /etc/sysconfig/dhcpd: DHCPD_CONF_INCLUDE_FILES="/etc/dhcpd.conf.master"

The master configuration file need to be modified:

```
ddns-update-style none;

default-lease-time 86400;
max-lease-time 86400;

option domain-name "digitalairlines.com";
option domain-name-servers 10.0.0.254;

option routers 10.0.0.254;

subnet 10.0.0.0 netmask 255.255.255.0 {
        pool {
                failover peer "digitalairlines";
                deny dynamic bootp clients;
                range 10.0.0.101 10.0.0.120;
        }
}
```

All failover configurations have to be defined in a **pool** statement. If you use several pools, you need to define the failover configuration in each of them.

In order to be aware of the name of the failover configuration, it has to be defined before the pool definition. That is why the include statement is written at the end of /etc/dhcpd.conf.

Failover is not supported on address allocation pools that contain addresses allocated to bootp clients. Therefore, the statement
deny dynamic bootp clients;
has to be defined.

When starting the DHCP server on the primary, you will see messages like these in /var/log/messages:

```
Jul  4 11:52:29 da10 dhcpd: failover peer digitalairlines:
I move from recover to startup
Jul  4 11:52:44 da10 dhcpd: failover peer digitalairlines:
I move from startup to recover
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
peer moves from unknown-state to recover
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
requesting full update from peer
Jul  4 11:54:57 da10 dhcpd: Sent update request all message
to digitalairlines
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
peer moves from recover to recover
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
requesting full update from peer
Jul  4 11:54:57 da10 dhcpd: Sent update done message to
digitalairlines
Jul  4 11:54:57 da10 dhcpd: Update request all from
digitalairlines: nothing pending
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
peer update completed.
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
I move from recover to recover-done
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
I move from recover-done to normal
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
peer moves from recover-done to normal
Jul  4 11:54:57 da10 dhcpd: pool 800e4138 10.0.0/24 total
20  free 20  backup 0  lts -10
Jul  4 11:54:57 da10 dhcpd: pool 800e4138 10.0.0/24  total
20  free 20  backup 0  lts 10
```

On the secondary server (which is started later), messages like these will appear in /var/log/messages:

```
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:
I move from recover to startup
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:
peer moves from unknown-state to recover
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:
requesting full update from peer
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:
I move from startup to recover
Jul  4 11:55:52 da12 dhcpd: Sent update request all message
to digitalairlines
Jul  4 11:55:52 da12 dhcpd: Sent update done message to
digitalairlines
Jul  4 11:55:52 da12 dhcpd: Update request all from
digitalairlines: nothing pending
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:
peer update completed.
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:
I move from recover to recover-done
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:
peer moves from recover to recover-done
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:
I move from recover-done to normal
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:
peer moves from recover-done to normal
Jul  4 11:55:52 da12 dhcpd: pool 800e40f8 10.0.0/24 total
20  free 20  backup 0  lts 10
Jul  4 11:55:52 da12 dhcpd: pool response: 10 leases
```


When a client requests an IP address, you will see messages like this in /var/log/messages:

```
Jul  4 11:59:26 da10 dhcpd: DHCPREQUEST for 192.168.5.16
from 00:04:ac:d6:58:96 via eth0: ignored (not
authoritative).
Jul  4 11:59:30 da10 dhcpd: DHCPREQUEST for 192.168.5.16
from 00:04:ac:d6:58:96 via eth0: ignored (not
authoritative).
Jul  4 11:59:36 da10 dhcpd: pool 800e4138 10.0.0/24 total
20  free 10  backup 10  lts 0
Jul  4 11:59:36 da10 dhcpd: DHCPDISCOVER from
00:04:ac:d6:58:96 via eth0
Jul  4 11:59:36 da10 dhcpd: DHCPREQUEST for 10.0.0.111
(10.0.0.12) from 00:04:ac:d6:58:96 via eth0: lease owned by
peer
Jul  4 11:59:37 da10 dhcpd: DHCPOFFER on 10.0.0.110 to
00:04:ac:d6:58:96 (linux-5ncs) via eth0
```

In this case, the client send a DHCPREQUEST in order to get same the IP address as the last time (192.168.5.16). This address is not available from a range definition, so the server refuses to offer this address. The the client send a DHCPDISCOVER to detect a DHCP server.

The next message from the client is a DHCPREQUEST for the IP address 10.0.0.111, this address is provided from the secondary server (“lease owned by peer”). The last message is the DHCPOFFER from the secondary server.

On the secondary server, the corresponding messages look like this:

```
Jul  4 12:00:20 da12 dhcpd: DHCPREQUEST for 149.44.85.16
from 00:04:ac:d6:58:96 via eth1: ignored (not
authoritative).
Jul  4 12:00:25 da12 dhcpd: DHCPREQUEST for 149.44.85.16
from 00:04:ac:d6:58:96 via eth1: ignored (not
authoritative).
Jul  4 12:00:30 da12 dhcpd: pool 800e40f8 10.0.0/24 total
20  free 10  backup 10  lts 0
Jul  4 12:00:30 da12 dhcpd: DHCPDISCOVER from
00:04:ac:d6:58:96 via eth1
Jul  4 12:00:31 da12 dhcpd: DHCPOFFER on 10.0.0.111 to
00:04:ac:d6:58:96 (linux-5ncs) via eth1
Jul  4 12:00:31 da12 dhcpd: DHCPREQUEST for 10.0.0.111
(10.0.0.12) from 00:04:ac:d6:58:96 (linux-5ncs) via eth1
Jul  4 12:00:31 da12 dhcpd: DHCPACK on 10.0.0.111 to
00:04:ac:d6:58:96 (linux-5ncs) via eth1
```

On this server, the DHCPOFFER message is printed as this server offers the IP address. The final message is DHCPACK from the client.

The DHCP log file /var/lib/dhcp/db/dhcpd.leases contains information like this:

```
failover peer "digitalairlines" state {
  my state recover at 2 2006/07/04 09:57:39;
  partner state unknown-state at 2 2006/07/04 09:57:39;
}

failover peer "digitalairlines" state {
  my state recover at 2 2006/07/04 09:57:39;
  partner state recover at 2 2006/07/04 09:57:39;
}
failover peer "digitalairlines" state {
  my state recover-done at 2 2006/07/04 09:57:56;
  partner state recover at 2 2006/07/04 09:57:39;
}

failover peer "digitalairlines" state {
  my state recover-done at 2 2006/07/04 09:57:56;
  partner state recover-done at 2 2006/07/04 09:57:39;
}

failover peer "digitalairlines" state {
  my state normal at 2 2006/07/04 09:57:56;
  partner state recover-done at 2 2006/07/04 09:57:39;
}
failover peer "digitalairlines" state {
  my state normal at 2 2006/07/04 09:57:56;
  partner state normal at 2 2006/07/04 09:57:39;
}
lease 10.0.0.120 {
  starts 2 2006/07/04 09:57:56;
  tstp 2 2006/07/04 09:57:56;
  binding state backup;
}
lease 10.0.0.119 {
  starts 2 2006/07/04 09:57:56;
  tstp 2 2006/07/04 09:57:56;
  binding state backup;
}
...
```

```
...
lease 10.0.0.120 {
  starts 2 2006/07/04 09:57:56;
  tstp 2 2006/07/04 09:57:56;
  tsfp 2 2006/07/04 09:57:56;
  binding state backup;
}
lease 10.0.0.119 {
  starts 2 2006/07/04 09:57:56;
  tstp 2 2006/07/04 09:57:56;
  tsfp 2 2006/07/04 09:57:56;
  binding state backup;
}

...
lease 10.0.0.111 {
  starts 2 2006/07/04 10:00:31;
  ends 2 2006/07/04 10:30:31;
  tstp 2 2006/07/04 09:57:56;
  tsfp 2 2006/07/04 10:45:31;
  cltt 2 2006/07/04 10:00:31;
  binding state active;
  next binding state expired;
  hardware ethernet 00:04:ac:d6:58:96;
  uid "\001\000\004\254\326X\226";
}
```


The file /var/lib/dhcp/db/dhcpd.leases on the secondary contains information like the following:

```
failover peer "digitalairlines" state {
  my state recover at 2 2006/07/04 09:58:50;
  partner state unknown-state at 2 2006/07/04 09:58:50;
  mclt 0;
}

failover peer "digitalairlines" state {
  my state recover at 2 2006/07/04 09:58:50;
  partner state recover at 2 2006/07/04 09:58:50;
  mclt 1800;
}
failover peer "digitalairlines" state {
  my state recover-done at 2 2006/07/04 09:58:51;
  partner state recover at 2 2006/07/04 09:58:50;
  mclt 1800;
}

failover peer "digitalairlines" state {
  my state recover-done at 2 2006/07/04 09:58:51;
  partner state recover-done at 2 2006/07/04 09:58:50;
  mclt 1800;
}
failover peer "digitalairlines" state {
  my state normal at 2 2006/07/04 09:58:51;
  partner state recover-done at 2 2006/07/04 09:58:50;
  mclt 1800;
}

failover peer "digitalairlines" state {
  my state normal at 2 2006/07/04 09:58:51;
  partner state normal at 2 2006/07/04 09:58:50;
  mclt 1800;
}
lease 10.0.0.120 {
  starts 2 2006/07/04 09:57:56;
  tsfp 2 2006/07/04 09:57:56;
  binding state backup;
}
```

When the secondary server fails (e.g. the server is shut down) and recovers later again, messages like the following will appear in /var/log/messages on the primary:

```
Jul  4 12:16:25 da10 dhcpd: peer digitalairlines:
disconnected
Jul  4 12:16:25 da10 dhcpd: failover peer digitalairlines:
I move from normal to communications-interrupted
Jul  4 12:20:06 da10 dhcpd: failover peer digitalairlines:
peer moves from normal to normal
Jul  4 12:20:06 da10 dhcpd: failover peer digitalairlines:
I move from communications-interrupted to normal
Jul  4 12:20:06 da10 dhcpd: pool 800e4138 10.0.0/24 total
20  free 10  backup 9  lts 0
```

On the secondary, the startup messages look like this:

```
Jul  4 12:21:01 da12 dhcpd: failover peer digitalairlines:
I move from normal to startup
Jul  4 12:21:01 da12 dhcpd: failover peer digitalairlines:
peer moves from normal to communications-interrupted
Jul  4 12:21:01 da12 dhcpd: failover peer digitalairlines:
I move from startup to normal
Jul  4 12:21:01 da12 dhcpd: failover peer digitalairlines:
peer moves from communications-interrupted to normal
Jul  4 12:21:01 da12 dhcpd: pool 800e40f8 10.0.0/24 total
20  free 10  backup 9  lts 0
```

In order to not get confused about the time stamps, make sure that you synchronize the time stamps of all your servers using the network time protocol (ntp).

The client log file /var/lib/dhcpcd/dhcpcd-eth0.info does only contain the information about the client's configuration and the DHCP server that provided the information:

```
IPADDR=10.0.0.111
NETMASK=255.255.255.0
NETWORK=10.0.0.0
BROADCAST=10.0.0.255
GATEWAY=10.0.0.254
DOMAIN='digitalairlines.com'
DNS=10.0.0.254
DHCPSID=10.0.0.12
DHCPGIADDR=0.0.0.0
DHCPSIADDR=0.0.0.0
DHCPCHADDR=00:04:AC:D6:58:96
DHCPSHADDR=00:04:AC:D6:55:F4
DHCPSNAME=''
LEASETIME=1800
RENEWALTIME=900
REBINDTIME=1575
INTERFACE='eth0'
CLASSID='Linux 2.6.16.20-0.12-default i686'
CLIENTID=00:04:AC:D6:58:96
```

Summary

Objective	Summary
1. Configure DHCP Pools	The **pool** declaration can be used to specify a pool of addresses that will be treated differently than any other pool of addresses, even on the same network segment or subnet.
2. Configure DHCP Failover	The failover protocol allows two DHCP servers to share a common address pool. If one server fails, the other server will continue to renew leases out of the pool and will allocate new addresses. The failover protocol defines a primary server role and a secondary server role. In order to configure failover, you need to write a peer declaration in the file /etc/dhcpd.conf that configures the failover protocol, and you need to write peer references in each pool declaration for which you want to do failover.

