By Andrew Colin Kissa
Published: 2008-06-25 19:14

How To Set Up DHCP Failover On Centos 5.1

Version 1.0.0
Author: Andrew Colin Kissa <andrew [at] topdog [dot] za [dot] net>
Last edited 18/06/2008**Introduction**

This tutorial will walk you through setting up DHCP fail over on CentOS 5.1 using the default ISC dhcp server, this can easily be adapted to any other Linux distro out there. You will most likely need Failover in environments where network down time can not be tolerated. My home is running a DLNA setup so I need my devices to be able to obtain network parameters at all times.

Since DHCP and DNS often go hand in hand i will be configuring a local DNS server which allows for dynamic updates, such that hostnames will be automatically updated to DNS when ever a lease is granted to a client.

My configuration with use the following please substitute to reflect your own network.

- Domain name - home.topdog-software.com
- Network - 192.168.1.0/24
- DHCP servers - 192.168.1.2,192.168.1.3
- Gateway - 192.168.1.254
- DNS servers - 192.168.1.2,192.168.1.3

Install required Packages

- DHCP

```
# yum install dhcp -y
```

- DNS

```
# yum install bind bind-chroot caching-nameserver -y
```

- NTP

```
# yum install ntp -y
```

ConfigurationDHCP

Backup your original config on the Master 192.168.1.2:

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

Edit the DHCP configuration `/etc/dhcpd.conf` on the master 192.168.1.2 and add the following, read the comments to understand the options:

```
authoritative;                          # server is authoritative
option domain-name "home.topdog-software.com";       # the domain name issued
option domain-name-servers 192.168.1.2,192.168.1.3;      # name servers issued
option netbios-name-servers 192.168.1.2;            # netbios servers
allow booting;                          # allow for booting over the network
allow bootp;                       # allow for booting
next-server 192.168.1.2;                  # TFTP server for booting
filename "pxelinux.0";                  # kernel for network booting
ddns-update-style interim;               # setup dynamic DNS updates
ddns-updates on;
ddns-domainname "home.topdog-software.com";          # domain name for DDNS updates
key rndckey {
    algorithm     hmac-md5;
    secret        "xxxxxxxxxx";           # get from the /etc/rndc.key file
}
zone home.topdog-software.com                 # forward zone to update
```

```
{
    primary 127.0.0.1;                        # update on the local machine
    key rndckey;                              # key to use for the update
}
zone 1.168.192.in-addr.arpa                   # reverse zone to update
{
    primary 127.0.0.1;                        # update on the local machine
    key rndckey;                              # key for update
}
failover peer "home-net" {                    # fail over configuration
    primary;                                  # This is the primary
    address 192.168.1.2;                      # primarys ip address
    port 647;
    peer address 192.168.1.3;                 # peer's ip address
    peer port 647;
    max-response-delay 60;
    max-unacked-updates 10;
    mclt 3600;
    split 128;
    load balance max seconds 3;
}
subnet 192.168.1.0 netmask 255.255.255.0      # zone to issue addresses from
{
    pool {
        failover peer "home-net";             # pool for dhcp leases with failover bootp not allowed
        deny dynamic bootp clients;
        option routers 192.168.1.254;
        range 192.168.1.25 192.168.1.50;
    }
    pool {                                    # accomodate our bootp clients here no replication and failover
        option routers 192.168.1.254;
        range 192.168.1.51 192.168.1.55;
    }
```

HowtoForge

```
    allow unknown-clients;
    ignore client-updates;
}
```

Back up your original config on the Slave 192.168.1.3:

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

Edit the DHCP configuration `/etc/dhcpd.conf` on the slave 192.168.1.3 and add the following, read the comments to understand the options:

```
authoritative;                          # server is authoritative
option domain-name "home.topdog-software.com";          # the domain name issued
option domain-name-servers 192.168.1.2,192.168.1.3;     # name servers issued
option netbios-name-servers 192.168.1.2;              # netbios servers
allow booting;                          # allow for booting over the network
allow bootp;                    # allow for booting
next-server 192.168.1.2;                # TFTP server for booting
filename "pxelinux.0";                  # kernel for network booting
ddns-update-style interim;              # setup dynamic DNS updates
ddns-updates on;
ddns-domainname "home.topdog-software.com";          # domain name for DDNS updates
key rndckey {
    algorithm    hmac-md5;
    secret      "xxxxxxxxxx";           # get from the /etc/rndc.key file on the master
}
zone home.topdog-software.com                 # forward zone to update
{
    primary 192.168.1.2;                # update on the local machine
    key rndckey;                # key to use for the update
```

```
}
zone 1.168.192.in-addr.arpa                 # reverse zone to update
{
    primary 192.168.1.2;                    # update on the local machine
    key rndckey;                            # key for update
}
failover peer "home-net" {                  # fail over configuration
    secondary;                              # This is the secondary
    address 192.168.1.3;                    # our ip address
    port 647;
    peer address 192.168.1.2;               # primary's ip address
    peer port 647;
    max-response-delay 60;
    max-unacked-updates 10;
    mclt 3600;
    load balance max seconds 3;
}
subnet 192.168.1.0 netmask 255.255.255.0        # zone to issue addresses from
{
    pool {
        failover peer "home-net";           # pool for dhcp leases with failover bootp not allowed
        deny dynamic bootp clients;
        option routers 192.168.1.254;
        range 192.168.1.25 192.168.1.50;
    }
    pool {                                  # accomodate our bootp clients here no replication and failover
        option routers 192.168.1.254;
        range 192.168.1.51 192.168.1.55;
    }
    allow unknown-clients;
    ignore client-updates;
}
```

DNS

Back up the the Bind configuration on the master:

```
# cp /var/named/chroot/etc/named.caching-nameserver.conf /var/named/chroot/etc/named.caching-nameserver.conf.orig
```

Edit the configuration to reflect the config below.

```
options {
    directory       "/var/named";
    dump-file       "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    query-source    port 53;
    query-source-v6 port 53;
    allow-query     { localhost; localnets; };
};
include "/etc/rndc.key";
include "/etc/named.rfc1912.zones";
zone "home.topdog-software.com" {
    type master;
    file "data/home.topdog-software.com.hosts";
    allow-transfer { 192.168.1.3; };
    allow-update { key "rndckey"; };
    allow-query { any; };
};
zone "1.168.192.in-addr.arpa" {
    type master;
    file "data/1.168.192.in-addr.arpa.hosts";
    allow-transfer { 192.168.1.3; };
    allow-update { key "rndckey"; };
    allow-query { any; };
```

```
};
```

Back up the the Bind configuration on the slave:

```
# cp /var/named/chroot/etc/named.caching-nameserver.conf /var/named/chroot/etc/named.caching-nameserver.conf.orig
```

Edit the configuration to reflect the config below.

```
options {
    directory       "/var/named";
    dump-file       "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    query-source    port 53;
    query-source-v6 port 53;
    allow-query     { localhost; localnets; };
};
include "/etc/rndc.key";
include "/etc/named.rfc1912.zones";
zone "home.topdog-software.com" {
    type slave;
    masters { 192.168.1.2; };
    file "data/home.topdog-software.com.hosts";
};
zone "1.168.192.in-addr.arpa" {
    type slave;
    masters { 192.168.1.2; };
    file "data/1.168.192.in-addr.arpa.hosts";
};
```

Create the zone files on the master

- */var/named/chroot/var/named/data/home.topdog-software.com.hosts*

```
$ORIGIN .
$TTL 38400
home.topdog-software.com IN SOA ns1.home.topdog-software.com. andrew.topdog.za.net. (
                2008061629 ; serial
                10800     ; refresh (3 hours)
                3600      ; retry (1 hour)
                604800    ; expire (1 week)
                38400     ; minimum (10 hours 40 minutes)
                )
          NS     ns1.home.topdog-software.com.
          NS     ns2.home.topdog-software.com.
ns1    IN   A 192.168.1.2
ns2    IN   A 192.168.1.3
```

- */var/named/chroot/var/named/data/1.168.192.in-addr.arpa.hosts*

```
$ORIGIN .
$TTL 38400     ; 10 hours 40 minutes
1.168.192.in-addr.arpa  IN SOA  ns1.home.topdog-software.com. andrew.topdog.za.net. (
                2008061644 ; serial
                10800     ; refresh (3 hours)
                3600      ; retry (1 hour)
                604800    ; expire (1 week)
                38400     ; minimum (10 hours 40 minutes)
                )
          NS     ns1.home.topdog-software.com.
          NS     ns2.home.topdog-software.com.
2   IN  PTR ns1.home.topdog-software.com.
```

3   IN PTR ns2.home.topdog-software.com.

NTP

NTP is required because the two DHCP servers need to be in sync for fail over as well as DDNS to take place. You can run a full fledged NTP server if you want, i will only provide you with instructions on using cron to sync NTP to an external NTP server every hour. You need to do this on BOTH servers.

- create a file */etc/cron.hourly/timesync* and add the following:

```
#!/bin/bash
#
ntpdate -s 0.rhel.pool.ntp.org
```

- make the file executable and run it for the first time:

```
# /etc/cron.hourly/timesync
```

Finally

Well we are done, let's fire up the services and begin testing.

- on the master:

```
# service named start
```

```
# service dhcpd start
```

- on the slave:

```
# service named start
```

```
# service dhcpd start
```

You should see the following in your logs on the master:

```
Jun 16 13:58:56 kudusoft dhcpd: failover peer home-net: I move from recover to startup
Jun 16 13:58:56 kudusoft dhcpd: dhcpd startup succeeded
Jun 16 13:58:56 kudusoft dhcpd: failover peer home-net: I move from startup to recover
Jun 16 13:59:12 kudusoft dhcpd: failover peer home-net: peer moves from unknown-state to recover
Jun 16 13:59:12 kudusoft dhcpd: failover peer home-net: requesting full update from peer
Jun 16 13:59:12 kudusoft dhcpd: Sent update request all message to home-net
Jun 16 13:59:12 kudusoft dhcpd: failover peer home-net: peer moves from recover to recover
Jun 16 13:59:12 kudusoft dhcpd: failover peer home-net: requesting full update from peer
Jun 16 13:59:12 kudusoft dhcpd: Update request all from home-net: sending update
Jun 16 13:59:12 kudusoft dhcpd: failover peer home-net: peer update completed.
Jun 16 13:59:12 kudusoft dhcpd: failover peer home-net: I move from recover to recover-done
Jun 16 13:59:13 kudusoft dhcpd: Sent update done message to home-net
Jun 16 13:59:13 kudusoft dhcpd: failover peer home-net: peer moves from recover to recover-done
Jun 16 13:59:13 kudusoft dhcpd: failover peer home-net: I move from recover-done to normal
Jun 16 13:59:13 kudusoft dhcpd: failover peer home-net: peer moves from recover-done to normal
Jun 16 13:59:14 kudusoft dhcpd: pool 914eb10 192.168.1/24 total 26  free 25  backup 0  lts -12
Jun 16 13:59:14 kudusoft dhcpd: pool 914eb10 192.168.1/24  total 26  free 25  backup 0  lts 12
```

And on the slave:

```
Jun 16 13:59:12 shaka dhcpd: Sending on   Socket/fallback/fallback-net
Jun 16 13:59:12 shaka dhcpd: failover peer home-net: I move from recover to startup
Jun 16 13:59:12 shaka dhcpd: failover peer home-net: peer moves from unknown-state to recover
Jun 16 13:59:12 shaka dhcpd: dhcpd startup succeeded
```

Jun 16 13:59:12 shaka dhcpd: failover peer home-net: requesting full update from peer

Jun 16 13:59:12 shaka dhcpd: failover peer home-net: I move from startup to recover

Jun 16 13:59:12 shaka dhcpd: Sent update request all message to home-net

Jun 16 13:59:12 shaka dhcpd: Sent update done message to home-net

Jun 16 13:59:12 shaka dhcpd: Update request all from home-net: nothing pending

Jun 16 13:59:12 shaka dhcpd: failover peer home-net: peer moves from recover to recover-done

Jun 16 13:59:14 shaka dhcpd: failover peer home-net: peer update completed.

Jun 16 13:59:14 shaka dhcpd: failover peer home-net: I move from recover to recover-done

Jun 16 13:59:14 shaka dhcpd: failover peer home-net: peer moves from recover-done to normal

Jun 16 13:59:14 shaka dhcpd: failover peer home-net: I move from recover-done to normal

Jun 16 13:59:14 shaka dhcpd: pool 9d78ad8 192.168.1/24 total 26 free 25 backup 0 lts 12

Jun 16 13:59:14 shaka dhcpd: pool response: 12 leases