DDNS with Samb4, bind9 and isc-DHCP

Author:
Stefan Kania

Ort:
SambaXP 2017 Göttingen

9. Mai 2017

Content

Kapitel 1

Introduction

In this years tutorial I would like to set up a *Samba 4* Active Directory infrastructure with two ADDCs, two *bind9* nameservers and two *isc-dhcp-Servers*. At the end of the day, you will have a failover DDNS infrastructure. Everything will be installed on a *debian jessie* with the distribution packages.

Everyone of you will find three virtual machines in your VirtualBox. Two for the Active Dirctory-Domaincontrollers and one as a client.

1.1 Host configurations

Each of the machines are already configured and packages are installed.

1.1.1 First Active Directory-Domaincontroller(ADDC)

The first ADDC will be the master for the sysvol-replication and the master for the DHCP-server.

System configuration

- ssh-access for *root* is allowed
- password for user *root* is *secret*
- hostname is *addc-01.example.net*
- IP-address is *192.168.56.11*

Network configuration

In listing 1.1.3.1 you will find the network configuration.

```
allow-hotplug eth0
iface eth0 inet static
       address 10.0.2.15
       netmask 255.255.255.0
       gateway 10.0.2.2
       dns-nameservers 8.8.8.8
       dns-search example.net


auto eth1
iface eth1 inet static
       address 192.168.56.11
       netmask 255.255.255.0
```

Listing 1.1.1.1: /etc/network/interfaces

Installed packages

I already installed some packages to the system, in listing 1.1.3.2 you will see the list of installed packages:

```
root@addc-01:~# apt install samba libpam-heimdal heimdal-clients \
                          ldb-tools winbind libpam-winbind \
                          smbclient libnss-winbind bind9 ntp \
                          xinetd rsync isc-dhcp-server
```

Listing 1.1.1.2: Installed packages

1.1.2 Second Active Directory-Domaincontroller

The second ADDC will be the slave for the sysvol-replication and the slave for the DHCP-server.

System configuration

- ssh-access for *root* is allowed
- password for user *root* is *secret*
- hostname is *addc-02.example.net*
- IP-address is *192.168.56.12*

Network configuration

in listing 1.1.3.1 you will find the network configuration.

```
allow-hotplug eth0
iface eth0 inet static
        address 10.0.2.15
        netmask 255.255.255.0
        gateway 10.0.2.2
        dns-nameservers 8.8.8.8
        dns-search example.net


auto eth1
iface eth1 inet static
        address 192.168.56.12
        netmask 255.255.255.0
```

Listing 1.1.2.1: /etc/network/interfaces

Installed packages

I already installed some packages to the system, in listing 1.1.3.2 you will see the list of installed packages:

```
root@addc-01:~# apt install samba libpam-heimdal heimdal-clients \
                            ldb-tools winbind libpam-winbind \
                            smbclient libnss-winbind bind9 ntp \
                            isc-dhcp-server rsync
```

Listing 1.1.2.2: Installed packages

1.1.3 Linux-client

The Linux-Client should be a member of the Active Directory-domain and a dhcp-client.

System configuration

- ssh-access for *root* is allowed
- password for user *root* is *secret*
- hostname is *client-01.example.net*
- IP-address is *192.168.56.13*

Network configuration

in listing 1.1.3.1 you will find the network configuration.

```
allow-hotplug eth0
iface eth0 inet static
       address 10.0.2.15
       netmask 255.255.255.0
       gateway 10.0.2.2
       dns-nameservers 8.8.8.8
       dns-search example.net


auto eth1
iface eth1 inet static
       address 192.168.56.13
       netmask 255.255.255.0
```

Listing 1.1.3.1: /etc/network/interfaces

Installed packages

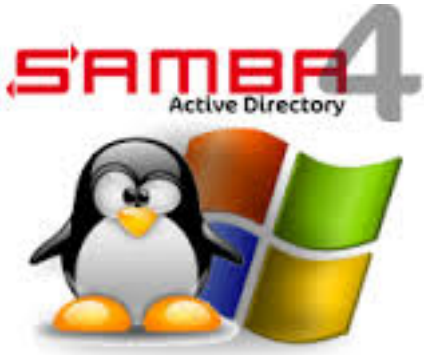
I already installed some packages to the system, in listing 1.1.3.2 you will see the list of installed packages:

```
root@addc-01:~# apt install samba libpam-heimdal winbind \
                          libpam-winbind smbclient \
                          libnss-winbind
```

Listing 1.1.3.2: Installed packages

Kapitel 2

Setting up the first ADDC



During the installation of the packages a /etc/samba/smb.conf was created, you have to delete /etc/samba/smb.conf from package installation, as in listing 2.0.3.1:

```
root@addc-01:~# rm /etc/samba/smb.conf
```

Listing 2.0.3.1: Deleting smb.conf

Now we can start to set up the first ADDC. We use the `samba-tool domain provision`-command for the provisioning. In listing 2.0.3.2 you can see the command with all the outputs:

```
root@addc-01:~# samba-tool domain provision
Realm [EXAMPLE.NET]:
 Domain [EXAMPLE]:
 Server Role (dc, member, standalone) [dc]:
 DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) [SAMBA_INTERNAL]:
 DNS forwarder IP address (write 'none' to disable forwarding) [8.8.8.8]: ^C
root@addc-01:~# rm /etc/samba/smb.conf
root@addc-01:~# samba-tool domain provision
Realm [EXAMPLE.NET]:
 Domain [EXAMPLE]:
 Server Role (dc, member, standalone) [dc]:
 DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) \
            [SAMBA_INTERNAL]: BIND9_DLZ
Administrator password:
Retype password:
Looking up IPv4 addresses
More than one IPv4 address found. Using 192.168.56.11
Looking up IPv6 addresses
No IPv6 address will be assigned
Setting up share.ldb
Setting up secrets.ldb
Setting up the registry
Setting up the privileges database
Setting up idmap db
Setting up SAM db
Setting up sam.ldb partitions and settings
Setting up sam.ldb rootDSE
Pre-loading the Samba 4 and AD schema
Adding DomainDN: DC=example,DC=net
Adding configuration container
Setting up sam.ldb schema
Setting up sam.ldb configuration data
Setting up display specifiers
Modifying display specifiers
Adding users container
Modifying users container
Adding computers container
Modifying computers container
Setting up sam.ldb data
Setting up well known security principals
Setting up sam.ldb users and groups
Setting up self join
Adding DNS accounts
Creating CN=MicrosoftDNS,CN=System,DC=example,DC=net
Creating DomainDnsZones and ForestDnsZones partitions
Populating DomainDnsZones and ForestDnsZones partitions
See /var/lib/samba/private/named.conf for an example configuration include \
                                file for BIND
and /var/lib/samba/private/named.txt for further documentation required for \
                                secure DNS updates
Setting up sam.ldb rootDSE marking as synchronized
Fixing provision GUIDs
A Kerberos configuration suitable for Samba 4 has been generated at \
                                /var/lib/samba/private/krb5.conf
Once the above files are installed, your Samba4 server will be ready to use
```

```
Server Role: active directory domain controller
Hostname: addc-01
NetBIOS Domain: EXAMPLE
DNS Domain: example.net
DOMAIN SID: S-1-5-21-2008792133-990162457-3339658904
```

Listing 2.0.3.2: Provisioning the domain

As you can see, the only change is that I chose *BIND9_DLZ* as the DNS backend. All other parameters are the default settings, taking from host-configuration.

Now change the /etc/samba/smb.conf to only use the *eth0*-device for the samba-service. In listing 2.0.3.3 you will see the new smb.conf:

```
# Global parameters
[global]
        workgroup = EXAMPLE
        realm = EXAMPLE.NET
        netbios name = ADDC-01
        server role = active directory domain controller
        server services = s3fs, rpc, nbt, wrepl, ldap, cldap, kdc, drepl, winbindd,\
                          ntp_signd, kcc, dnsupdate
        interfaces = 192.168.56.11
        bind interfaces only = yes
        wins support = yes

[netlogon]
        path = /var/lib/samba/sysvol/example.net/scripts
        read only = No

[sysvol]
        path = /var/lib/samba/sysvol
        read only = No
```

Listing 2.0.3.3: The new smb.conf

Before configuring bind9, make sure, that your new ADDC will use it's own IP as nameserver after rebooting the system. Change the /etc/network/interfaces to settings in listing 2.0.3.4:

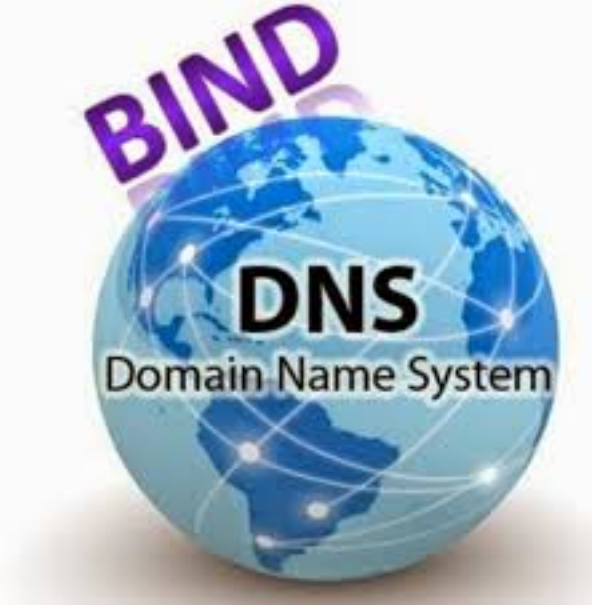
```
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
        address 10.0.2.15
        netmask 255.255.255.0
        gateway 10.0.2.2
        dns-nameservers 192.1687.56.11
        dns-search example.net


auto eth1
iface eth1 inet static
        address 192.168.56.11
        netmask 255.255.255.0
```

Listing 2.0.3.4: Setting own IP as nameserver

Now you can go to the next step, setting up the nameserver *bind9*

Kapitel 3

Configuring bind9 and ntp (first ADDC)



Samba4 gives you an internal DNS-nameserver, but using *bind9* has a lot of advantages. You can use bind9 for more then managing AD-domains. One more reason to use is that bind9 supports round robin DNS. You need a round robin DNS if you want to set up a CTDB-cluster in your network.

3.1 Configuring bind9

First you have to set some options in /etc/bind/named.conf.options as you can see in listing 7.0.2.1:

```
forwarders {
       8.8.8.8;
};
tkey-gssapi-keytab "/var/lib/samba/private/dns.keytab";
```

Listing 3.1.1: Changes in /etc/bind/named.con.options

The file dns.keytab was created during the provisioning. Bind9 needs this file to authenticate against the ADDC Kerberos.

Now you have to tell bind9 to read and write all DNS-informations to a AD-Zone. For this you have to edit /etc/bind/named.conf.local as you can see in listing 7.0.2.2:

```
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
include "/var/lib/samba/private/named.conf";
```

Listing 3.1.2: Changes in /etc/bind/named.conf.local

The file /var/lib/samba/private/named.conf was also created during the provisioning. In this file you will find an entry that points to the right version of your installed bind9-version.

The last thing you have to do, is check the filesystem-permission, so that bind9 has access to all necessary files. See listing 3.1.3 for all files you should check:

```
ls -ld /var/lib/samba/private/
drwxr-xr-x 6 root root 4096 Feb 23 18:39 /var/lib/samba/private/

ls -l /var/lib/samba/private/named.conf
-rw-r--r-- 1 root root 678 Feb 23 18:39 /var/lib/samba/private/named.conf

ls -ld /var/lib/samba/private/dns
drwxrwx--- 3 root bind 4096 Feb 23 18:39 /var/lib/samba/private/dns

ls -ld /var/lib/samba/private/dns.keytab
-rw-r----- 1 root bind 737 Feb 23 18:39 /var/lib/samba/private/dns.keytab

ls -l /var/lib/samba/private/dns/
total 2948
-rw-rw---- 1 root bind 3014656 Feb 23 18:39 sam.ldb
drwxrwx--- 2 root bind 4096 Feb 23 18:39 sam.ldb.d

ls -l /var/lib/samba/private/dns/sam.ldb.d/
total 25184
-rw-rw---- 1 root bind 7884800 Feb 23 18:39 CN=CONFIGURATION,DC=EXAMPLE,DC=NET.ldb
-rw-rw---- 1 root bind 7700480 Feb 23 18:39 \
                    CN=SCHEMA,CN=CONFIGURATION,DC=EXAMPLE,DC=NET.ldb
-rw-rw---- 2 root bind 4247552 Feb 23 18:38 DC=DOMAINDNSZONES,DC=EXAMPLE,DC=NET.ldb
-rw-rw---- 1 root bind 1286144 Feb 23 18:39 DC=EXAMPLE,DC=NET.ldb
-rw-rw---- 2 root bind 4247552 Feb 23 18:38 DC=FORESTDNSZONES,DC=EXAMPLE,DC=NET.ldb
-rw-rw---- 2 root bind 421888 Feb 23 18:39 metadata.tdb
```

Listing 3.1.3: Check file permissions

Now you can reboot your system.

After the reboot you should do the tests from listing 3.1.4:

```
root@addc-01:~# ps ax | grep samba
  846 ? Ss 0:00 /usr/sbin/samba -D
  874 ? S 0:00 /usr/sbin/samba -D
  875 ? S 0:00 /usr/sbin/samba -D
  876 ? S 0:00 /usr/sbin/samba -D
  877 ? S 0:00 /usr/sbin/samba -D
  878 ? S 0:00 /usr/sbin/samba -D
  879 ? S 0:00 /usr/sbin/samba -D
  880 ? S 0:00 /usr/sbin/samba -D
  881 ? S 0:00 /usr/sbin/samba -D
  882 ? S 0:00 /usr/sbin/samba -D
  883 ? S 0:00 /usr/sbin/samba -D
  884 ? S 0:00 /usr/sbin/samba -D
  885 ? S 0:00 /usr/sbin/samba -D
  938 pts/0 S+ 0:00 grep samba

root@addc-01:~# ps ax | grep named
  546 ? Ssl 0:00 /usr/sbin/named -f -u bind

 root@addc-01:~# host addc-01
addc-01.example.net has address 192.168.56.11

root@addc-01:~# host -t srv _ldap._tcp.example.net
_ldap._tcp.example.net has SRV record 0 100 389 addc-01.example.net.

root@addc-01:~# host -t srv _gc._tcp.example.net
_gc._tcp.example.net has SRV record 0 100 3268 addc-01.example.net.

root@addc-01:~# host -t srv _kerberos._tcp.example.net
_kerberos._tcp.example.net has SRV record 0 100 88 addc-01.example.net.
```

Listing 3.1.4: Testing the first ADDC

Later in this tutorial we would like to create a PTR-record for every client who connects to the AD, so at this point you have to create a *reverse-zone*. Listing 3.1.5 is showing the process:

```
root@addc-01:~# kinit administrator
administrator@EXAMPLE.NET's Password:

root@addc-01:~# samba-tool dns zonecreate addc-01 56.168.192.in-addr.arpa -k yes
Zone 56.168.192.in-addr.arpa created successfully
```

Listing 3.1.5: Create a reverse-zone

3.2 Setting up the timeserver

Now the Active Directory-Domaincontroller is working. But there is one more thing you have to do – you need a timeserver on your ADDC. The timeserver is needed by the Windows-clients to set the correct time. But the time-package a timeserver is sending to a windows-client must be signed. So you have to set up the timeserver to use the ADDC to sign the time-packages. In listing 3.2.1 you will see the configuration for the timeserver *ntp*. All settings must be made in /etc/ntp.conf:

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
server 0.pool.ntp.org iburst prefer
server 1.pool.ntp.org iburst prefer
driftfile /var/lib/ntp/ntp.drift
logfile /var/log/ntp
ntpsigndsocket /var/lib/samba/ntp_signd/
```

```
restrict default kod nomodify notrap nopeer mssntp
restrict 127.0.0.1
restrict 0.pool.ntp.org mask 255.255.255.255 nomodify notrap nopeer noquery
restrict 1.pool.ntp.org mask 255.255.255.255 nomodify notrap nopeer noquery

# set permission for the socket
chgrp ntp /var/lib/samba/ntp_signd/
```

Listing 3.2.1: The ntp-configuration

Set the permission, so that the *ntp* can access the socket. See listing 8.1.2:

```
root@addc-01:~# chgrp ntp /var/lib/samba/ntp_signd/
```

Listing 3.2.2: Setting permissions for ntp

Now you can restart *ntp* with `systemctl restart ntp`

Kapitel 4

Setting up the first DHCP-server

Now we start the implementation of the DHCP-Server to the Active Directory. The package for the *isc-dhcp-server* is already installed on the system.

The first thing you have to do, is creating a system-user. This user must be a AD-user. This will be just a service-user. In listing 4.0.1 you will see the command to create the user:

```
root@addc-01:~# samba-tool user create dhcpduser --description="Unprivileged \
                        user for TSIG-GSSAPI DNS updates via ISC DHCP server" \
                        --random-password
User 'dhcpduser' created successfully

root@addc-01:~# samba-tool group addmembers DnsAdmins dhcpduser
Added members to group DnsAdmins

root@addc-01:~# samba-tool user setexpiry dhcpduser --noexpiry
Expiry for user 'dhcpduser' disabled.
```

Listing 4.0.1: Creating the dhcp-user

As you can see, the user will get a random password, becomes a member of the group *DnsAdmins* and –very important – the password will not expire.

Now you need a *keytab-file* so the user can authenticate via Kerberos. In listing 4.0.2 you can see the process of creating the *keytab-file*:

```
root@addc-01:~# samba-tool domain exportkeytab --principal=dhcpduser@EXAMPLE.NET \
                                            /etc/dhcp/dhcpduser.keytab

root@addc-01:~# chown root:root /etc/dhcp/dhcpduser.keytab

root@addc-01:~# chmod 400 /etc/dhcp/dhcpduser.keytab
```

Listing 4.0.2: Creating a keytab-file

Create a directory with `mkdir -p /etc/dhcp/bin` in which you will copy the update-script. Now copy the update-script /root/dhcp-dyndns.sh to /etc/dhcp/bin. In listing 4.0.3 you can see the script:

```
#!/bin/bash

# /etc/bin/dhcp-dyndns.sh

# This script is for secure DDNS updates on Samba 4
# Version: 0.8.7

# DNS domain
domain=$(hostname -d)
if [ -z ${domain} ]; then
    echo "Cannot obtain domain name, is DNS set up correctly?"
    echo "Cannot continue... Exiting."
    logger "Cannot obtain domain name, is DNS set up correctly?"
    logger "Cannot continue... Exiting."
    exit 1
fi

# Samba 4 realm
REALM=$(echo ${domain^^})

# Additional nsupdate flags (-g already applied), e.g. "-d" for debug
NSUPDFLAGS="-d"

# krbcc ticket cache
export KRB5CCNAME="/tmp/dhcp-dyndns.cc"

# Kerberos principal
SETPRINCIPAL="dhcpduser@${REALM}"
# Kerberos keytab
```

```
# /etc/dhcpduser.keytab
# krbcc ticket cache
# /tmp/dhcp-dyndns.cc

TESTUSER=$(wbinfo -u | grep dhcpduser)
if [ -z "${TESTUSER}" ]; then
    echo "No AD dhcp user exists, need to create it first.. exiting."
    echo "you can do this by typing the following commands"
    echo "kinit Administrator@${REALM}"
    echo "samba-tool user create dhcpduser --random-password --description=\
         "Unprivileged user for DNS updates via ISC DHCP server\""
    echo "samba-tool user setexpiry dhcpduser --noexpiry"
    echo "samba-tool group addmembers DnsAdmins dhcpduser"
    exit 1
fi

# Check for Kerberos keytab
if [ ! -f /etc/dhcp/dhcpduser.keytab ]; then
    echo "Required keytab /etc/dhcpduser.keytab not found, it needs to be created."
    echo "Use the following commands as root"
    echo "samba-tool domain exportkeytab --principal=${SETPRINCIPAL} \
          /etc/dhcpduser.keytab"
    echo "chown dhcpd:dhcpd /etc/dhcpduser.keytab"
    echo "chmod 400 /etc/dhcpduser.keytab"
    exit 1
fi

# Variables supplied by dhcpd.conf
action=$1
ip=$2
DHCID=$3
name=${4%%.*}

usage()
{
echo "USAGE:"
echo " `basename $0` add ip-address dhcid|mac-address hostname"
echo " `basename $0` delete ip-address dhcid|mac-address"
}

_KERBEROS () {
# get current time as a number
test=$(date +%d'-'%m'-'%y' '%H':'%M':'%S)
# Note: there have been problems with this
# check that 'date' returns something like
# 04-09-15 09:38:14

# Check for valid kerberos ticket
#logger "${test} [dyndns] : Running check for valid kerberos ticket"
klist -c /tmp/dhcp-dyndns.cc -s
if [ "$?" != "0" ]; thenroot@client-01:~# net ads testjoin
Join is OK

    logger "${test} [dyndns] : Getting new ticket, old one has expired"
    kinit -F -k -t /etc/dhcp/dhcpduser.keytab -c /tmp/dhcp-dyndns.cc \
               "${SETPRINCIPAL}"
    if [ "$?" != "0" ]; then
        logger "${test} [dyndns] : dhcpd kinit for dynamic DNS failed"
        exit 1;
    fi
fi

}
```

```
# Exit if no ip address or mac-address
if [ -z "${ip}" ] || [ -z "${DHCID}" ]; then
    usage
    exit 1
fi

# Exit if no computer name supplied, unless the action is 'delete'
if [ "${name}" = "" ]; then
    if [ "${action}" = "delete" ]; then
        name=$(host -t PTR "${ip}" | awk '{print $NF}' | awk -F '.' '{print $1}')
    else
        usage
        exit 1;
    fi
fi

# Set PTR address
ptr=$(echo ${ip} | awk -F '.' '{print $4"."$3"."$2"."$1".in-addr.arpa"}')

## nsupdate ##

case "${action}" in
add)
    _KERBEROS

nsupdate -g ${NSUPDFLAGS} << UPDATE
server 127.0.0.1
realm ${REALM}
update delete ${name}.${domain} 3600 A
update add ${name}.${domain} 3600 A ${ip}
send
UPDATE
result1=$?

nsupdate -g ${NSUPDFLAGS} << UPDATE
server 127.0.0.1
realm ${REALM}
update delete ${ptr} 3600 PTR
update add ${ptr} 3600 PTR ${name}.${domain}
send
UPDATE
result2=$?
;;
delete)
     _KERBEROS

nsupdate -g ${NSUPDFLAGS} << UPDATE
server 127.0.0.1
realm ${REALM}
update delete ${name}.${domain} 3600 A
send
UPDATE
result1=$?
nsupdate -g ${NSUPDFLAGS} << UPDATE
server 127.0.0.1
realm ${REALM}
update delete ${ptr} 3600 PTR
send
UPDATE
result2=$?
;;
*)
echo "Invalid action specified"
exit 103
```

```
;;
esac

result="${result1}${result2}"

if [ "${result}" != "00" ]; then
    logger "DHCP-DNS Update failed: ${result}"
else
    logger "DHCP-DNS Update succeeded"
fi

exit ${result}
```

Listing 4.0.3: The update-script

Set the permission of the file `chmod 755 /etc/dhcp/bin/dhcp-dyndns.sh`.

Now you are at the point where you can start configuring the dhcp-server. During the installation of the isc-dhcp-server a configuration is created.

Move this file to a safe place `mv /etc/dhcp/dhcpd.conf /root/dhcpd.conf.orig`.

Now you must create a new configuration-file /etc/dhcp/dhcpd.conf. In listing 4.0.4 you see the content of the file:

```
authoritative;
ddns-update-style none;

subnet 192.168.56.0 netmask 255.255.255.0 {
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.56.255;
  option time-offset 0;
# option routers 192.168.0.1;
  option domain-name "example.net";
  option domain-name-servers 192.168.56.11;
  option netbios-name-servers 192.168.56.11;
  option ntp-servers 192.168.0.11;
  pool {
    max-lease-time 1800; # 30 minutes
    range 192.168.56.210 192.168.56.229;
  }
}

on commit {
set noname = concat("dhcp-", binary-to-ascii(10, 8, "-", leased-address));
set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
set ClientDHCID = binary-to-ascii(16, 8, ":", hardware);
set ClientName = pick-first-value(option host-name, config-option-host-name, \
    client-name, noname);
log(concat("Commit: IP: ", ClientIP, " DHCID: ", ClientDHCID, " Name: ", ClientName));
execute("/etc/dhcp/bin/dhcp-dyndns.sh", "add", ClientIP, ClientDHCID, ClientName);
}

on release {
set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
set ClientDHCID = binary-to-ascii(16, 8, ":", hardware);
log(concat("Release: IP: ", ClientIP));
execute("/etc/dhcp/bin/dhcp-dyndns.sh", "delete", ClientIP, ClientDHCID);
}

on expiry {
set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
# cannot get a ClientMac here, apparently this only works when actually receiving a packet
log(concat("Expired: IP: ", ClientIP));
# cannot get a ClientName here, for some reason that always fails
```

```
execute("/etc/dhcp/bin/dhcp-dyndns.sh", "delete", ClientIP, "", "0");
}
```

Listing 4.0.4: The new dhcpd.conf

Before you can restart the DHCP-Server you must define which interface the DHCP-server should use. Open /etc/default/isc-dhcp-server and edit it as you can see in listing 4.0.5:

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth1"
```

Listing 4.0.5: Changes in /etc/default/isc-dhcp-server

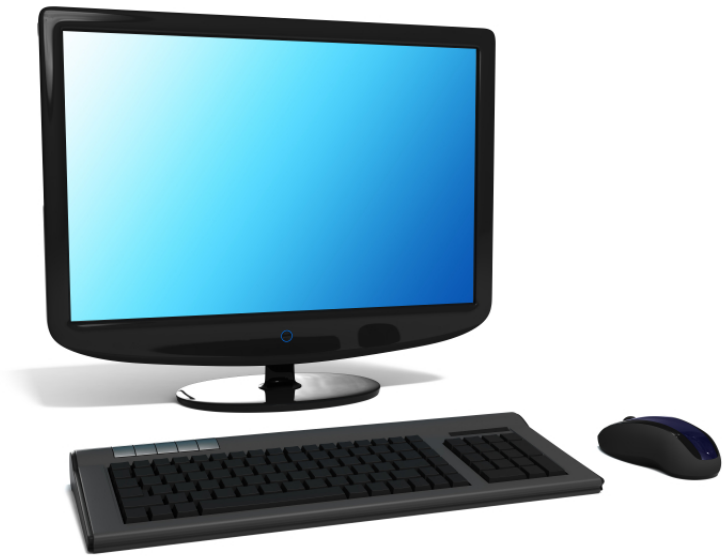
Now restart the DHCP-Server with `systemctl restart isc-dhcp-server.service`. Now you are ready to join a client and test if the client will get it's IP-configuration. Listing 4.0.6 is showing the start and the status:

```
root@addc-01:~# systemctl restart isc-dhcp-server.service

root@addc-01:~# systemctl status isc-dhcp-server.service
 isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server)
   Active: active (running) since Mon 2017-02-27 19:16:40 CET; 2s ago
  Process: 1600 ExecStop=/etc/init.d/isc-dhcp-server stop (code=exited, status=0/SUCCESS)
  Process: 1607 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/isc-dhcp-server.service
           1616 /usr/sbin/dhcpd -q -cf /etc/dhcp/dhcpd.conf -pf /var/run/dhcpd.pid eth1

Feb 27 19:16:38 addc-01 systemd[1]: Starting LSB: DHCP server...
Feb 27 19:16:38 addc-01 dhcpd[1615]: Wrote 0 leases to leases file.
Feb 27 19:16:38 addc-01 dhcpd[1616]: Server starting service.
Feb 27 19:16:40 addc-01 isc-dhcp-server[1607]: Starting ISC DHCP server: dhcpd.
Feb 27 19:16:40 addc-01 systemd[1]: Started LSB: DHCP server.
```

Listing 4.0.6: Status of DHCP-Server

Kapitel 5

Joining the Client



After the first ADDC is running, it is time, to set up a client that will be a member of the ADDC. The client should also get it's IP-settings via the DHCP-Server. The first thing you have to do is creating a new /etc/samba/smb.conf with the content of the listing 5.0.1:

```
[global]
      workgroup = example
      realm = EXAMPLE.NET
      security = ADS
      winbind refresh tickets = Yes
      template shell = /bin/bash
      idmap config * : range = 10000 - 19999
      idmap config EXAMPLE : backend = rid
      idmap config EXAMPLE : range = 1000000 - 1999999
```

Listing 5.0.1: Client smb.conf

The nameserver of the client-configuration must be the IP-address from the ADDC *192.168.56.11*. So you have to change the settings in /etc/network/interfaces. Copy the /etc/krb5.conf from the first ADDC to your client. Then you can join the client to your domain as you can see in listing 5.0.2:

```
root@client-01:~# net ads join -U administrator
Enter administrator's password:
Using short domain name -- EXAMPLE
Joined 'CLIENT-01' to dns domain 'example.net'

root@client-01:~# net ads testjoin
Join is OK
```

Listing 5.0.2: Join the client

To get the users from Active Directory to your Linux-client you have to edit /etc/nsswich.conf as you can see in listing 5.0.3:

```
passwd: compat winbind
group: compat winbind
```

Listing 5.0.3: The new nsswitch.conf

Test the settings with `getent` as you can see in listing 5.0.4:

```
root@client-01:~# getent passwd EXAMPLE\\administrator
EXAMPLE\administrator:*:1000500:1000513:Administrator:/home/EXAMPLE/administrator:/bin/bash
```

Listing 5.0.4: Looking for users

Change to use DHCP

Now you can configure the client to use DHCP to get it's IP-settings. Listing 5.0.5 is showing the new /etc/network/interfaces:

```
allow-hotplug eth0
iface eth0 inet static
       address 10.0.2.15
       netmask 255.255.255.0
       gateway 10.0.2.2
# dns-nameservers 192.168.56.11
# dns-search example.net


auto eth1
iface eth1 inet dhcp

#iface eth1 inet static
# address 192.168.56.13
# netmask 255.255.255.0
```

Listing 5.0.5: The new interfaces

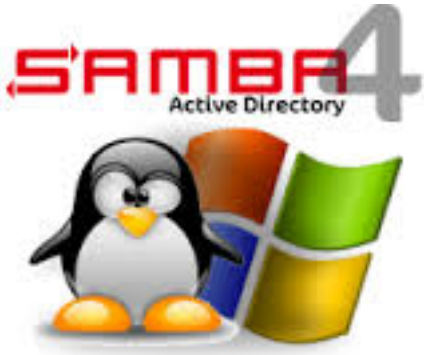
Now you can reboot the client. After the reboot you can lookup the client either in the forward-zone and the reverse-zone as you can see in listing 5.0.6:

```
root@client-01:~# host client-01
client-01.example.net has address 192.168.56.210

root@client-01:~# host 192.168.56.210
210.56.168.192.in-addr.arpa domain name pointer client-01.example.net.
```

Listing 5.0.6: Resolving the client

Kapitel 6

Setting up the second ADDC



All packages you need to set up the second ADDC are already installed on the virtual machine with the IP-Address *192.168.56.12*.

6.0.1 Creating DNS-record

Before you start setting up the second ADDC you must create all DNS-records for the second DC on the first DC. In listing **??** you can see all commands to add the entries to both zones:

```
root@addc-01:~# kinit administrator
administrator@EXAMPLE.NET's Password:

root@addc-01:~# samba-tool dns add addc-01 example.net addc-02 A 192.168.56.12 -k yes
Record added successfully

root@addc-01:~# samba-tool dns add addc-01 56.168.192.in-addr.arpa 11 PTR \
                           addc-01.example.net -k yes
Record added successfully

root@addc-01:~# samba-tool dns add addc-01 56.168.192.in-addr.arpa 12 PTR \
                               addc-02.example.net -k yes
Record added successfully
```

Listing 6.0.1.1: Adding DNS-entries

6.0.2 Join the second ADDC to domain

Now copy /etc/krb5.conf from the first ADDC to the second ADDC and change the nameserver-setting, so that the new ADDC will use the first ADDC as a nameserver. After you have copied the KRB5.CONF and change the nameserver setting, you can join the new ADDC to your domain, as you can see in listing 6.0.2.1:

```
root@addc-02:~# rm /etc/samba/smb.conf

root@addc-02:~# samba-tool domain join --dns-backend=BIND9_DLZ example.net DC \
                                     --realm=example.net -Uadministrator
Finding a writeable DC for domain 'example.net'
Found DC addc-01.example.net
Password for [WORKGROUP\administrator]:
workgroup is EXAMPLE
realm is example.net
checking sAMAccountName
Adding CN=ADDC-02,OU=Domain Controllers,DC=example,DC=net
Adding CN=ADDC-02,CN=Servers,CN=Default-First-Site-Name,CN=Sites,\
               CN=Configuration,DC=example,DC=net
Adding CN=NTDS Settings,CN=ADDC-02,CN=Servers,CN=Default-First-Site-Name,\
               CN=Sites,CN=Configuration,DC=example,DC=net
Adding SPNs to CN=ADDC-02,OU=Domain Controllers,DC=example,DC=net
Setting account password for ADDC-02$
Enabling account
Adding DNS account CN=dns-ADDC-02,CN=Users,DC=example,DC=net with dns/ SPN
Setting account password for dns-ADDC-02
Calling bare provision
Looking up IPv4 addresses
More than one IPv4 address found. Using 192.168.56.12
Looking up IPv6 addresses
No IPv6 address will be assigned
Setting up share.ldb
Setting up secrets.ldb
Setting up the registry
Setting up the privileges database
Setting up idmap db
```

```
Setting up SAM db
Setting up sam.ldb partitions and settings
Setting up sam.ldb rootDSE
Pre-loading the Samba 4 and AD schema
A Kerberos configuration suitable for Samba 4 has been generated at \
          /var/lib/samba/private/krb5.conf
Provision OK for domain DN DC=example,DC=net
Starting replication
Schema-DN[CN=Schema,CN=Configuration,DC=example,DC=net] objects[402/1550] \
        linked_values[0/0]
Schema-DN[CN=Schema,CN=Configuration,DC=example,DC=net] objects[804/1550] \
        linked_values[0/0]
Schema-DN[CN=Schema,CN=Configuration,DC=example,DC=net] objects[1206/1550] \
        linked_values[0/0]
Schema-DN[CN=Schema,CN=Configuration,DC=example,DC=net] objects[1550/1550] \
        linked_values[0/0]
Analyze and apply schema objects
Partition[CN=Configuration,DC=example,DC=net] objects[402/1616] linked_values[0/0]
Partition[CN=Configuration,DC=example,DC=net] objects[804/1616] linked_values[0/0]
Partition[CN=Configuration,DC=example,DC=net] objects[1206/1616] linked_values[0/0]
Partition[CN=Configuration,DC=example,DC=net] objects[1608/1616] linked_values[0/0]
Partition[CN=Configuration,DC=example,DC=net] objects[1616/1616] linked_values[28/0]
Replicating critical objects from the base DN of the domain
Partition[DC=example,DC=net] objects[98/98] linked_values[23/0]
Partition[DC=example,DC=net] objects[312/214] linked_values[24/0]
Done with always replicated NC (base, config, schema)
Replicating DC=DomainDnsZones,DC=example,DC=net
Partition[DC=DomainDnsZones,DC=example,DC=net] objects[47/47] linked_values[0/0]
Replicating DC=ForestDnsZones,DC=example,DC=net
Partition[DC=ForestDnsZones,DC=example,DC=net] objects[18/18] linked_values[0/0]
Committing SAM database
Sending DsReplicaUpdateRefs for all the replicated partitions
Setting isSynchronized and dsServiceName
Setting up secrets database
See /var/lib/samba/private/named.conf for an example configuration include file for BIND
and /var/lib/samba/private/named.txt for further documentation required \
                                for secure DNS updates
Joined domain EXAMPLE (SID S-1-5-21-2008792133-990162457-3339658904) as a DC
```

Listing 6.0.2.1: Joining the second ADDC

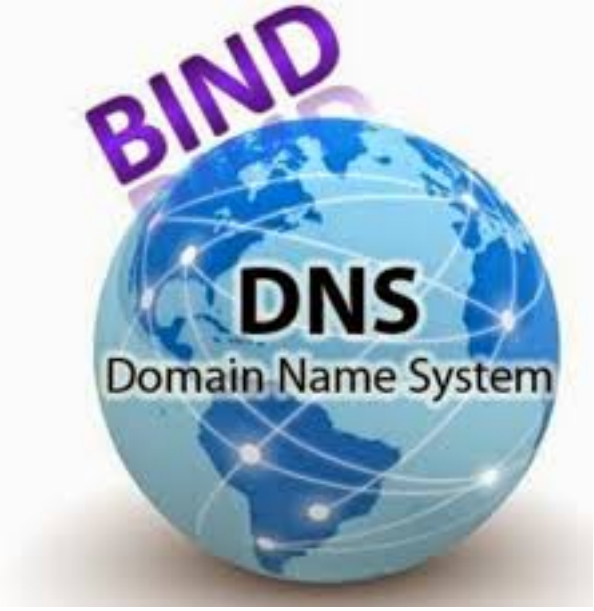
Change the /etc/samba/smb.conf so that the Samba-service will only use the IP-Address *192.168.56.12*, you see the new smb.conf in listing 6.0.2.2:

```
# Global parameters
[global]
       workgroup = EXAMPLE
       realm = example.net
       netbios name = ADDC-02
       server role = active directory domain controller
       server services = s3fs, rpc, nbt, wrepl, ldap, cldap, kdc, drepl, winbindd, \
                         ntp_signd, kcc, dnsupdate
       interfaces = 192.168.56.12
       bind interfaces only = yes

[netlogon]
       path = /var/lib/samba/sysvol/example.net/scripts
       read only = No

[sysvol]
       path = /var/lib/samba/sysvol
       read only = No
```

Listing 6.0.2.2: Changing smb.conf

Kapitel 7

Configuring bind9 (second ADDC)



Before you can restart your second ADDC you have to configure the bind9-nameserver.

First you have to set some options in /etc/bind/named.conf.options as you can see in listing 7.0.2.1:

```
forwarders {
       8.8.8.8;
};
tkey-gssapi-keytab "/var/lib/samba/private/dns.keytab";
```

Listing 7.0.2.1: Changes in /etc/bind/named.con.options

The file dns.keytab was created during the provisioning. Bind9 needs this file to authenticate against the ADDC Kerberos.

Now you have to tell bind9 to read and write all DNS-informations to a AD-Zone. For this you have to edit /etc/bind/named.conf.local as you can see in listing 7.0.2.2:

```
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
include "/var/lib/samba/private/named.conf";
```

Listing 7.0.2.2: Changes in /etc/bind/named.conf.local

The file /var/lib/samba/private/named.conf was also created during the provisioning. In this file you will find an entry that points to the right version of your installed bind9-version.

The last thing you have to do, is check the filesystem-permission, so that bind9 has access to all necessary files. See listing 7.0.2.3 for all files you should check:

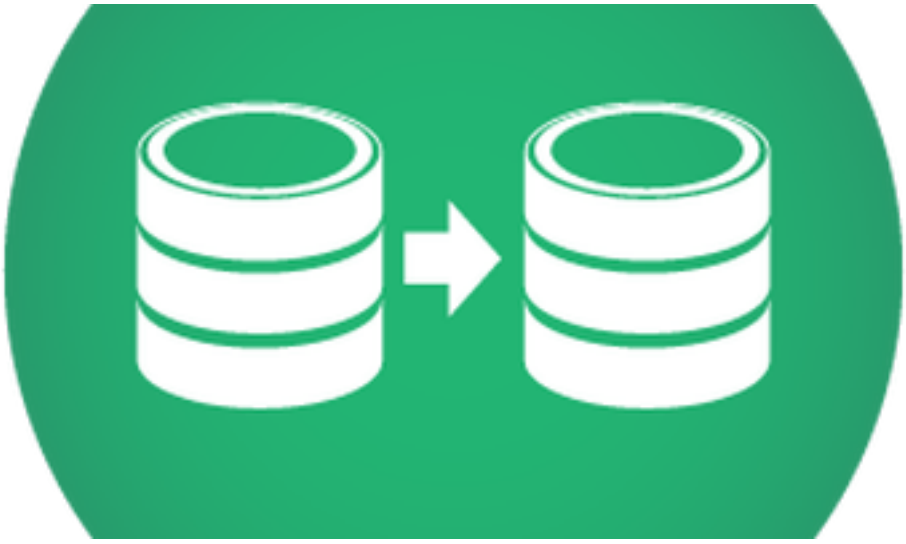
```
ls -ld /var/lib/samba/private/
drwxr-xr-x 6 root root 4096 Feb 23 18:39 /var/lib/samba/private/

ls -l /var/lib/samba/private/named.conf
-rw-r--r-- 1 root root 678 Feb 23 18:39 /var/lib/samba/private/named.conf

ls -ld /var/lib/samba/private/dns
drwxrwx--- 3 root bind 4096 Feb 23 18:39 /var/lib/samba/private/dns

ls -ld /var/lib/samba/private/dns.keytab
-rw-r----- 1 root bind 737 Feb 23 18:39 /var/lib/samba/private/dns.keytab

ls -l /var/lib/samba/private/dns/
total 2948
-rw-rw---- 1 root bind 3014656 Feb 23 18:39 sam.ldb
drwxrwx--- 2 root bind 4096 Feb 23 18:39 sam.ldb.d

ls -l /var/lib/samba/private/dns/sam.ldb.d/
total 25184
-rw-rw---- 1 root bind 7884800 Feb 23 18:39 CN=CONFIGURATION,DC=EXAMPLE,DC=NET.ldb
-rw-rw---- 1 root bind 7700480 Feb 23 18:39 CN=SCHEMA,CN=CONFIGURATION,\
                                            DC=EXAMPLE,DC=NET.ldb
-rw-rw---- 2 root bind 4247552 Feb 23 18:38 DC=DOMAINDNSZONES,DC=EXAMPLE,DC=NET.ldb
-rw-rw---- 1 root bind 1286144 Feb 23 18:39 DC=EXAMPLE,DC=NET.ldb
-rw-rw---- 2 root bind 4247552 Feb 23 18:38 DC=FORESTDNSZONES,DC=EXAMPLE,DC=NET.ldb
-rw-rw---- 2 root bind 421888 Feb 23 18:39 metadata.tdb
```

Listing 7.0.2.3: Check file permissions

Now you can reboot your system.

Kapitel 8

Testing database replication



Because we are using *bind9* as nameserver you must check if all necessary DNS-records where created during the join. Do all the tests from listing 8.0.2.1on both ADDCs:

```
root@addc-01:~# host addc-01
addc-01.example.net has address 192.168.56.11
root@addc-01:~# host addc-02
addc-02.example.net has address 192.168.56.12
root@addc-01:~# host -t srv _ldap._tcp.example.net
_ldap._tcp.example.net has SRV record 0 100 389 addc-01.example.net.

root@addc-02:~# host addc-01
addc-01.example.net has address 192.168.56.11
root@addc-02:~# host addc-02
addc-02.example.net has address 192.168.56.12
root@addc-02:~# host -t srv _ldap._tcp.example.net
_ldap._tcp.example.net has SRV record 0 100 389 addc-01.example.net.
```

Listing 8.0.2.1: Test DNS-records

As you can see, you have just one service-record for the LDAP-service. But you must have an entry for every ADDC, otherwise the clients can't login if the missing ADDC is the only one. Let's fix this problem: First thing you should do, is rebooting both systems, to start all services in the right order. Then test again if you have both ADDC listed for the services. In listing 8.0.2.2 you see the right result of the test:

```
root@addc-01:~# host addc-01
addc-01.example.net has address 192.168.56.11

root@addc-01:~# host addc-02
addc-02.example.net has address 192.168.56.12

root@addc-01:~# host -t srv _ldap._tcp.example.net
_ldap._tcp.example.net has SRV record 0 100 389 addc-01.example.net.
_ldap._tcp.example.net has SRV record 0 100 389 addc-02.example.net.


root@addc-02:~# host addc-01
addc-01.example.net has address 192.168.56.11

root@addc-02:~# host addc-02
addc-02.example.net has address 192.168.56.12

root@addc-02:~# host -t srv _ldap._tcp.example.net
_ldap._tcp.example.net has SRV record 0 100 389 addc-02.example.net.
_ldap._tcp.example.net has SRV record 0 100 389 addc-01.example.net.
```

Listing 8.0.2.2: Test DNS-records

8.0.3 Missing cname-record

Attention !

The following part is only needed if you still not having all DCs in the list after rebooting all DCs.

Look if you have all *objecguid*-entries for all ADDC. Use the command you can see in listing 8.0.3.1:

```
root@addc-01:~# ldbsearch -H /var/lib/samba/private/sam.ldb '(invocationId=*)' \
                        --cross-ncs objectguid
# record 1
dn: CN=NTDS Settings,CN=ADDC-01,CN=Servers,CN=Default-First-Site-Name,\
                    CN=Sites,CN=Configuration,DC=example,DC=net
objectGUID: 0f921c18-fe76-4d37-86ff-5968d87e42fc

# record 2
```

```
dn: CN=NTDS Settings,CN=ADDC-02,CN=Servers,CN=Default-First-Site-Name,\
                    CN=Sites,CN=Configuration,DC=example,DC=net
objectGUID: d87407b9-1d16-4c31-ac0d-7e5824565df7

# returned 2 records
# 2 entries
# 0 referrals


root@addc-02:~# ldbsearch -H /var/lib/samba/private/sam.ldb '(invocationId=*)' \
                        --cross-ncs objectguid
# record 1
dn: CN=NTDS Settings,CN=ADDC-01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,\
                    CN=Configuration,DC=example,DC=net
objectGUID: 0f921c18-fe76-4d37-86ff-5968d87e42fc

# record 2
dn: CN=NTDS Settings,CN=ADDC-02,CN=Servers,CN=Default-First-Site-Name,CN=Sites,\
                    CN=Configuration,DC=example,DC=net
objectGUID: d87407b9-1d16-4c31-ac0d-7e5824565df7

# returned 2 records
# 2 entries
# 0 referrals
```

Listing 8.0.3.1: Listing all objectguids

For all ADDCs there must be a *cname-record* pointing to the *objectGUID*. So now you should check if all cname-records were created. Listing 8.0.3.2 is showing the command:

```
root@addc-01:~# host -t CNAME 0f921c18-fe76-4d37-86ff-5968d87e42fc._msdcs.example.net
0f921c18-fe76-4d37-86ff-5968d87e42fc._msdcs.example.net is an alias \
                      for addc-01.example.net.

root@addc-01:~# host -t CNAME d87407b9-1d16-4c31-ac0d-7e5824565df7._msdcs.example.net
d87407b9-1d16-4c31-ac0d-7e5824565df7._msdcs.example.net is an alias \
                      for addc-02.example.net.

root@addc-02:~# host -t CNAME 0f921c18-fe76-4d37-86ff-5968d87e42fc._msdcs.example.net
0f921c18-fe76-4d37-86ff-5968d87e42fc._msdcs.example.net is an alias \
                      for addc-01.example.net.

root@addc-02:~# host -t CNAME d87407b9-1d16-4c31-ac0d-7e5824565df7._msdcs.example.net
d87407b9-1d16-4c31-ac0d-7e5824565df7._msdcs.example.net is an alias \
                      for addc-02.example.net.
```

Listing 8.0.3.2: List the cname-records

If one record is missing, you can create the record, as you can see in listing 8.0.3.3. Use the ADDC on which the record is missing:

```
samba-tool dns add addc1 _msdcs.example.net d87407b9-1d16-4c31-ac0d-7e5824565df7 \
                      CNAME addc2.example.net -k yes
Record added successfully
```

Listing 8.0.3.3: Create the missing cname-record

Now restart samba and bind9 or better reboot the system. After the system is up, test again for the cname-record. Now you should see the cname-record.

8.0.4 Testing the Replication

Now that the second ADDC is running you should do some tests, to see if the replication of the AD-database is working properly. In listing 8.0.4.1 you will see all the test:

```
root@addc-02:~# samba-tool drs kcc addc-01
Consistency check on addc-01 successful.

root@addc-02:~# samba-tool drs kcc addc-02
Consistency check on addc-02 successful.

root@addc-02:~# samba-tool drs showrepl
Default-First-Site-Name\ADDC-02
DSA Options: 0x00000001
DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
DSA invocationId: 2d41a55a-4c92-4e01-952e-15b58d752a85

==== INBOUND NEIGHBORS ====

CN=Schema,CN=Configuration,DC=example,DC=net
       Default-First-Site-Name\ADDC-01 via RPC
              DSA object GUID: 0f921c18-fe76-4d37-86ff-5968d87e42fc
              Last attempt @ Wed Mar 1 18:09:55 2017 CET was successful
              0 consecutive failure(s).
              Last success @ Wed Mar 1 18:09:55 2017 CET

CN=Configuration,DC=example,DC=net
       Default-First-Site-Name\ADDC-01 via RPC
              DSA object GUID: 0f921c18-fe76-4d37-86ff-5968d87e42fc
              Last attempt @ Wed Mar 1 18:09:55 2017 CET was successful
              0 consecutive failure(s).
              Last success @ Wed Mar 1 18:09:55 2017 CET

DC=ForestDnsZones,DC=example,DC=net
       Default-First-Site-Name\ADDC-01 via RPC
              DSA object GUID: 0f921c18-fe76-4d37-86ff-5968d87e42fc
              Last attempt @ Wed Mar 1 18:09:55 2017 CET was successful
              0 consecutive failure(s).
              Last success @ Wed Mar 1 18:09:55 2017 CET

DC=example,DC=net
       Default-First-Site-Name\ADDC-01 via RPC
              DSA object GUID: 0f921c18-fe76-4d37-86ff-5968d87e42fc
              Last attempt @ Wed Mar 1 18:09:55 2017 CET was successful
              0 consecutive failure(s).
              Last success @ Wed Mar 1 18:09:55 2017 CET

DC=DomainDnsZones,DC=example,DC=net
       Default-First-Site-Name\ADDC-01 via RPC
              DSA object GUID: 0f921c18-fe76-4d37-86ff-5968d87e42fc
              Last attempt @ Wed Mar 1 18:09:55 2017 CET was successful
              0 consecutive failure(s).
              Last success @ Wed Mar 1 18:09:55 2017 CET

==== OUTBOUND NEIGHBORS ====

==== KCC CONNECTION OBJECTS ====

Connection --
       Connection name: 0ec21c18-9a76-4eff-b24a-cf36c840e340
       Enabled : TRUE
       Server DNS name : addc-01.example.net
       Server DN name : CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
                        CN=Default-First-Site-Name,CN=Sites,CN=Configuration,\
                        DC=example,DC=net
              TransportType: RPC
              options: 0x00000001
Warning: No NC replicated for Connection!
```

```
root@addc-01:~# samba-tool drs showrepl
Default-First-Site-Name\ADDC-01
DSA Options: 0x00000001
DSA object GUID: 0f921c18-fe76-4d37-86ff-5968d87e42fc
DSA invocationId: f4d8a084-9584-4095-b799-5fa46ffdb0fb

==== INBOUND NEIGHBORS ====

DC=ForestDnsZones,DC=example,DC=net
       Default-First-Site-Name\ADDC-02 via RPC
              DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
              Last attempt @ Wed Mar 1 18:30:32 2017 CET was successful
              0 consecutive failure(s).
              Last success @ Wed Mar 1 18:30:32 2017 CET

DC=DomainDnsZones,DC=example,DC=net
       Default-First-Site-Name\ADDC-02 via RPC
              DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
              Last attempt @ Wed Mar 1 18:30:32 2017 CET was successful
              0 consecutive failure(s).
              Last success @ Wed Mar 1 18:30:32 2017 CET

DC=example,DC=net
       Default-First-Site-Name\ADDC-02 via RPC
              DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
              Last attempt @ Wed Mar 1 18:30:32 2017 CET was successful
              0 consecutive failure(s).
              Last success @ Wed Mar 1 18:30:32 2017 CET

CN=Schema,CN=Configuration,DC=example,DC=net
       Default-First-Site-Name\ADDC-02 via RPC
              DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
              Last attempt @ Wed Mar 1 18:30:32 2017 CET was successful
              0 consecutive failure(s).
              Last success @ Wed Mar 1 18:30:32 2017 CET

CN=Configuration,DC=example,DC=net
       Default-First-Site-Name\ADDC-02 via RPC
              DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
              Last attempt @ Wed Mar 1 18:30:32 2017 CET was successful
              0 consecutive failure(s).
              Last success @ Wed Mar 1 18:30:32 2017 CET

==== OUTBOUND NEIGHBORS ====

DC=ForestDnsZones,DC=example,DC=net
       Default-First-Site-Name\ADDC-02 via RPC
              DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
              Last attempt @ NTTIME(0) was successful
              0 consecutive failure(s).
              Last success @ NTTIME(0)

DC=DomainDnsZones,DC=example,DC=net
       Default-First-Site-Name\ADDC-02 via RPC
              DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
              Last attempt @ NTTIME(0) was successful
              0 consecutive failure(s).
              Last success @ NTTIME(0)

DC=example,DC=net
       Default-First-Site-Name\ADDC-02 via RPC
              DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
```

```
                Last attempt @ NTTIME(0) was successful
                0 consecutive failure(s).
                Last success @ NTTIME(0)

CN=Schema,CN=Configuration,DC=example,DC=net
        Default-First-Site-Name\ADDC-02 via RPC
                DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
                Last attempt @ NTTIME(0) was successful
                0 consecutive failure(s).
                Last success @ NTTIME(0)

CN=Configuration,DC=example,DC=net
        Default-First-Site-Name\ADDC-02 via RPC
                DSA object GUID: d87407b9-1d16-4c31-ac0d-7e5824565df7
                Last attempt @ NTTIME(0) was successful
                0 consecutive failure(s).
                Last success @ NTTIME(0)

==== KCC CONNECTION OBJECTS ====

Connection --
        Connection name: 796d5dad-3e67-4b47-a94e-55332d1fa8f2
        Enabled : TRUE
        Server DNS name : addc-02.example.net
        Server DN name : CN=NTDS Settings,CN=ADDC-02,CN=Servers,\
                         CN=Default-First-Site-Name,CN=Sites,\
                         CN=Configuration,DC=example,DC=net
                TransportType: RPC
                options: 0x00000001
Warning: No NC replicated for Connection!
```

Listing 8.0.4.1: Testing replication

Another way to test if replication is working is creating a new user in *addc-01* and see if the new user is shown on *addc-02* and vice versa.

8.1 Setting up the timeserver

The second Active Directory-Domaincontroller must also be a time-server for your Windows-clients . The timeserver is needed by the Windows-clients to set the correct time. But the time-package a timeserver is sending to a windows-client must be signed. So you have to set up the timeserver to use the ADDC to sign the time-packages. In listing 3.2.1 you will see the configuration for the timeserver *ntp*. All settings are must be made in /etc/ntp.conf:

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
server 0.pool.ntp.org iburst prefer
server 1.pool.ntp.org iburst prefer
driftfile /var/lib/ntp/ntp.drift
logfile /var/log/ntp
ntpsigndsocket /var/lib/samba/ntp_signd/
restrict default kod nomodify notrap nopeer mssntp
restrict 127.0.0.1
restrict 0.pool.ntp.org mask 255.255.255.255 nomodify notrap nopeer noquery
restrict 1.pool.ntp.org mask 255.255.255.255 nomodify notrap nopeer noquery

# set permission for the socket
chgrp ntp /var/lib/samba/ntp_signd/
```

Listing 8.1.1: The ntp-configuration

Set the permission, so that the *ntp* can access the socket. See listing 8.1.2:

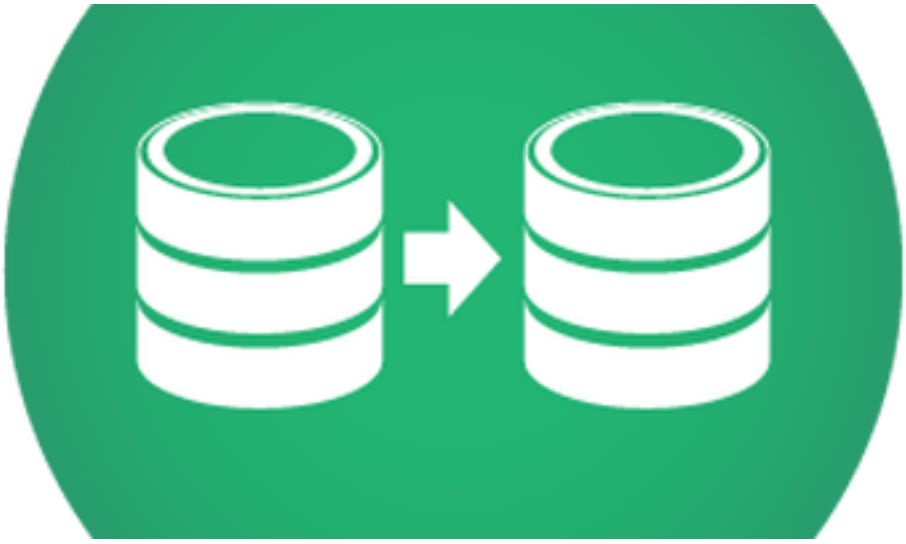
```
root@addc-02:~# chgrp ntp /var/lib/samba/ntp_signd/
```

Listing 8.1.2: Setting permissions for ntp

Now you can restart *ntp* with `systemctl restart ntp`

Kapitel 9

Configuring sysvol-replication



Now it's time to configure the *sysvol*-replication. Inside the sysvol-share samba is storing all the files for the *grouppolicies(GPO)* and the *logon scripts*. These files must be the same on all ADDCs in your domain, because every user can login to all ADDCs. Microsoft is using a special protocol to replicate the sysvol-share. This protocol is yet not implemented in Samba 4 yet. For this reason we will use *rsync* together with *xinetd* to do the sysvol-replication.

The replication of the sysvol-share is a single master replication, so we will make one of our ADDCs the master and the other one will be the slave, who will pull the new files from the master. But which ADDC should be the master? Let's do it the Microsoft-way and chose the ADDC which holds the *PdcEmulationMasterRole*. To find out which of your ADDCs holds this role you can user `samba-tool` as you can see in listing 9.0.1:

```
root@addc-02:~# samba-tool fsmo show
InfrastructureMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
                                CN=Default-First-Site-Name,CN=Sites,\
                                CN=Configuration,DC=example,DC=net
RidAllocationMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
                                CN=Default-First-Site-Name,CN=Sites,\
                                CN=Configuration,DC=example,DC=net
PdcEmulationMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
                                CN=Default-First-Site-Name,CN=Sites,\
                                CN=Configuration,DC=example,DC=net
DomainNamingMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
                                CN=Default-First-Site-Name,CN=Sites,\
                                CN=Configuration,DC=example,DC=net
SchemaMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
                                CN=Default-First-Site-Name,CN=Sites,\
                                CN=Configuration,DC=example,DC=ne
```

Listing 9.0.1: Showing the fsmo-roles

As you can see, the first ADDC is holding all fsmo-roles, so we will chose this ADDC as the master for the sysvol-replication.

Rsync should run as a service so we have to use *xinetd* to start rsync as a daemon. To activate rsync you have to write a startscript for xinetd. The script must be located in /etc/xinetd.d. The name of the script mus be rsync. In listing 9.0.2 you can see the script /etc/xinetd.d/rsync:

```
service rsync
{
   disable = no
   only_from = 192.168.56.12
   socket_type = stream
   wait = no
   user = root
   server = /usr/bin/rsync
   server_args = --daemon
   log_on_failure += USERID
}
```

Listing 9.0.2: The xinetd script for rsync

In the next step you must create a start-script /etc/rsyncd.conf for rsync. In listing 9.0.3 you can see the script:

```
[sysvol]
path = /var/lib/samba/sysvol/
comment = Samba sysvol
uid = root
gid = root
read only = yes
auth users = sysvol-repl
secrets file = /etc/samba/rsync.secret
```

Listing 9.0.3: Script to start rsync

The *auth users* are just rsync-user you don't have to create the user as a systemuser. The secrets file must contain the *auth users* and the password, as you can see in listing 9.0.4:

```
sysvol-repl:secret
```

Listing 9.0.4: The secrets file for rsync

The file must belong to *root* and must have the permission *600*. If you grant access to *other*, rsync will not work.

After you have created the two files, you can restart *xinetd*. After restarting the service take a look at the logfile and see if the service is started correctly. In listing 9.0.5 you will see the start an a part of the log-file:

```
root@addc-01:~# systemctl restart xinetd.service

root@addc-01:~# journalctl -xn
-- Logs begin at Thu 2017-03-02 10:19:14 CET, end at Thu 2017-03-02 12:05:11 CET. --
Mar 02 12:05:11 addc-01 xinetd[1506]: removing daytime
Mar 02 12:05:11 addc-01 xinetd[1506]: removing daytime
Mar 02 12:05:11 addc-01 xinetd[1506]: removing discard
Mar 02 12:05:11 addc-01 xinetd[1506]: removing discard
Mar 02 12:05:11 addc-01 xinetd[1506]: removing echo
Mar 02 12:05:11 addc-01 xinetd[1506]: removing echo
Mar 02 12:05:11 addc-01 xinetd[1506]: removing time
Mar 02 12:05:11 addc-01 xinetd[1506]: removing time
Mar 02 12:05:11 addc-01 xinetd[1506]: xinetd Version 2.3.15 started with libwrap \
                                      loadavg options compiled in.
Mar 02 12:05:11 addc-01 xinetd[1506]: Started working: 1 available service
```

Listing 9.0.5: Restart xinetd

You should see *1 available service*.

Now we can switch to the second ADDC to start rsync as a slave for the first time. You can write the command directly on the commandline, but it's better to put the command into a script and start the script. If you put the command into a script, it's easier to start the replication via cron and it's possible to add more commands to execute during the replication process. In listing 9.0.6 you will see the script to start the replication. Don't forget to make the script executable:

```
root@addc-02:~# vi /root/sysvol-repl.bash
----------------------------------------
!#/bin/bash

rsync --dry-run -XAavz --delete-after --password-file=/etc/samba/rsync.pass \
      rsync://sysvol-repl@addc-01:/sysvol /var/lib/samba/sysvol
```

Listing 9.0.6: Script to start the rsync-replication

Here you can see two things: The script will run with the option *–dry-run*, this will be used just once, to test if the right files will be replicated. After the test you must remove this parameter. The second thing you see, inside the script, is the name of a file /etc/samba/rsync.pass. In this file you must write the password for the *auth users*. The owner of the file must be *root* and you must set the permission to *600*. After you have created the script and the password-file you can start the script the first time. See listing 9.0.7 for the output:

```
root@addc-02:~# ./sysvol-repl.bash
receiving file list ... done
./
example.net/
example.net/Policies/
example.net/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/
example.net/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/GPT.INI
example.net/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHINE/
example.net/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/USER/
```

```
example.net/Policies/{6AC1786C-016F-11D2-945F-00C04FB984F9}/
example.net/Policies/{6AC1786C-016F-11D2-945F-00C04FB984F9}/GPT.INI
example.net/Policies/{6AC1786C-016F-11D2-945F-00C04FB984F9}/MACHINE/
example.net/Policies/{6AC1786C-016F-11D2-945F-00C04FB984F9}/USER/
example.net/scripts/

sent 59 bytes received 1,351 bytes 940.00 bytes/sec
total size is 40 speedup is 0.03 (DRY RUN)
```

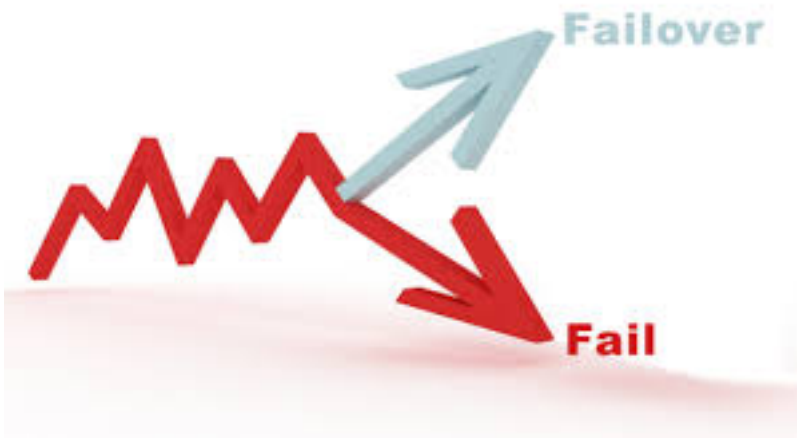
Listing 9.0.7: Output of the first run

In the last line you can see, that it was a *DRY RUN*. After you have checked that the right files and directories will be replicated, you can remove the *–dry-run* parameter from the script and run the replication.

After the replication is finished, you look inside the directory /var/lib/samba/sysvol and you will see, that all files are replicated. This will end the sysvol-replication set up.

Kapitel 10

Setting up DHCP failover



Now let's go to the last part of this years tutorial: Making the DHCP-server fault-tolerant. Ad the beginning you must copy the script and the keytab-file, you created for the first ADDC to the second ADDC. Listing 10.0.1 is showing all copy commands:

```
root@addc-02:~# scp addc-01:/etc/dhcp/dhcpduser.keytab /etc/dhcp/
root@addc-01's password:
dhcpduser.keytab
100% 337 0.3KB/s 00:00
root@addc-02:~# chmod 400 /etc/dhcp/dhcpduser.keytab

root@addc-02:~# mv /etc/dhcp/dhcpd.conf /root/dhcpd.conf.orig

root@addc-02:~# scp addc-01:/etc/dhcp/dhcpd.conf /etc/dhcp/
root@addc-01's password:
dhcpd.conf

root@addc-02:~# mkdir -p /etc/dhcp/bin

root@addc-02:~# scp addc-01:/etc/dhcp/bin/dhcp-dyndns.sh /etc/dhcp/bin/
root@addc-01's password:
dhcp-dyndns.sh
```

Listing 10.0.1: Copy the scripts

Before you can restart the DHCP-Server you must define which interface the DHCP-server should use. To change the setting, open /etc/default/isc-dhcp-server and edit it as you can see in listing 10.0.2:

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth1"
```

Listing 10.0.2: Changes in /etc/default/isc-dhcp-server

At this point don't start the slave.

Now you have to add the failover configuration to both, the master and the slave configuration. Let's start with the master. Open the file /etc/dhcp/dhcpd.conf and add the failover section before the subnet configuration as showing in listing 10.0.3:

```
authoritative;
ddns-update-style none;

# Start failover configuration
failover peer "dhcp-failover" {
  primary;
  address addc-01.example.net;
  peer address addc-02.example.net;
  max-response-delay 60;
  max-unacked-updates 10;
  mclt 3600;
  split 128;
  load balance max seconds 3;
}
# End failover configuration

subnet 192.168.56.0 netmask 255.255.255.0 {
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.56.255;
  option time-offset 0;
# option routers 192.168.0.1;
  option domain-name "example.net";
  option domain-name-servers 192.168.56.11;
  option netbios-name-servers 192.168.56.11;
  option ntp-servers 192.168.0.11;
```

```
  pool {
    failover peer "dhcp-failover"; # Add for failover
    max-lease-time 1800; # 30 minutes
    range 192.168.56.210 192.168.56.229;
  }
}

on commit {
set noname = concat("dhcp-", binary-to-ascii(10, 8, "-", leased-address));
set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
set ClientDHCID = binary-to-ascii(16, 8, ":", hardware);
set ClientName = pick-first-value(option host-name, config-option-host-name,\
                client-name, noname);
log(concat("Commit: IP: ", ClientIP, " DHCID: ", ClientDHCID, " Name: ", ClientName));
execute("/etc/dhcp/bin/dhcp-dyndns.sh", "add", ClientIP, ClientDHCID, ClientName);
}

on release {
set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
set ClientDHCID = binary-to-ascii(16, 8, ":", hardware);
log(concat("Release: IP: ", ClientIP));
execute("/etc/dhcp/bin/dhcp-dyndns.sh", "delete", ClientIP, ClientDHCID);
}

on expiry {
set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
# cannot get a ClientMac here, apparently this only works when actually \
  receiving a packet
log(concat("Expired: IP: ", ClientIP));
# cannot get a ClientName here, for some reason that always fails
execute("/etc/dhcp/bin/dhcp-dyndns.sh", "delete", ClientIP, "", "0");
}
```

Listing 10.0.3: Changing the master configuration

One parameter should be mentioned *split 128;*. This parameter must only be set at the master-dhcp-server. It can be a value between *0* and *255*. This value will manage a load-balancing of the two servers. A value of *128* will use both DHCP-server equal. See the man-page of dhcpd.conf for more details.

Now edit the configuration on the slave as followed in listing 10.0.4:

```
authoritative;
ddns-update-style none;

# Start failover configuration
failover peer "dhcp-failover" {
  secondary;
  address addc-02.example.net;
  peer address addc-01.example.net;
  max-response-delay 60;
  max-unacked-updates 10;
  mclt 3600;
  load balance max seconds 3;
}
# End failover configuration

subnet 192.168.56.0 netmask 255.255.255.0 {
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.56.255;
  option time-offset 0;
# option routers 192.168.0.1;
  option domain-name "example.net";
  option domain-name-servers 192.168.56.11;
  option netbios-name-servers 192.168.56.11;
```

```
  option ntp-servers 192.168.0.11;
  pool {
    failover peer "dhcp-failover"; # add vor failover
    max-lease-time 1800; # 30 minutes
    range 192.168.56.210 192.168.56.229;
  }
}

on commit {
set noname = concat("dhcp-", binary-to-ascii(10, 8, "-", leased-address));
set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
set ClientDHCID = binary-to-ascii(16, 8, ":", hardware);
set ClientName = pick-first-value(option host-name, config-option-host-name, \
               client-name, noname);
log(concat("Commit: IP: ", ClientIP, " DHCID: ", ClientDHCID, " Name: ", ClientName));
execute("/etc/dhcp/bin/dhcp-dyndns.sh", "add", ClientIP, ClientDHCID, ClientName);
}

on release {
set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
set ClientDHCID = binary-to-ascii(16, 8, ":", hardware);
log(concat("Release: IP: ", ClientIP));
execute("/etc/dhcp/bin/dhcp-dyndns.sh", "delete", ClientIP, ClientDHCID);
}

on expiry {
set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
# cannot get a ClientMac here, apparently this only works when actually \
  receiving a packet
log(concat("Expired: IP: ", ClientIP));
# cannot get a ClientName here, for some reason that always fails
execute("/etc/dhcp/bin/dhcp-dyndns.sh", "delete", ClientIP, "", "0");
}
```

Listing 10.0.4: Failover configuration slave

The communication between two isc-dhcp-server is managed over *Object Management Application Programming Interface(OMAPI)*. This API is controlling the operation of the DHCP Failover Protocol. This API will be configured in the next steps.

First, create a dnssec-key for the communication, as shown in listing 10.0.5:

```
root@addc-01:~# dnssec-keygen -a HMAC-MD5 -b 512 -n USER DHCP_OMAPI
Kdhcp_omapi.+157+28776

root@addc-01:~# ls
Kdhcp_omapi.+157+28776.key Kdhcp_omapi.+157+28776.private
```

Listing 10.0.5: Creating a dnssec key

Next you have to extract the the key from the private-file, as shown in listing 10.0.6 and copy it to both, the master- and the slave-configuration:

```
root@addc-01:~# cat Kdhcp_omapi.+*.private |grep ^Key|cut -d ' ' -f2-

DhUhXJc10YbaC1q7AK/rdOkp6U3ZkBZU1Cyegc9Q88V95ouxIQlV0Io0cxlYjpk/Ibi+A==
```

Listing 10.0.6: Cut the key

You have to set up a special section on both, the master and the slave, inside the dhcpd.conf for the OMAPI-communication as shown in listing 10.0.7:

```
omapi-port 7911;
omapi-key omapi_key;
```

```
key omapi_key {
     algorithm hmac-md5;
     secret "DhUhXJc10YbaC1q7AK/rdOkp6U3ZkBZU1Cyegc9Q88V95ouxIQlV0Io0cxlYjpk/Ibi+A==";
}
```

Listing 10.0.7: The OMAPI-Section

Now you can restart both DHCP-server. If you take a look at the logfile while starting the service, you will see the lines from listing 10.0.8:

```
Mar 02 17:13:33 addc-02 dhcpd[2384]: Server starting service.
Mar 02 17:13:33 addc-02 dhcpd[2384]: failover peer dhcp-failover: peer moves from \
                                     normal to communications-interrupted
Mar 02 17:13:33 addc-02 dhcpd[2384]: failover peer dhcp-failover: I move from \
                                     startup to normal
Mar 02 17:13:33 addc-02 dhcpd[2384]: balancing pool 7f0f23387300 192.168.56.0/24 \
                                     total 20 free 10 backup 9 lts 0 max-own (+/-)2
Mar 02 17:13:33 addc-02 dhcpd[2384]: balanced pool 7f0f23387300 192.168.56.0/24 total 20 \
                                     free 10 backup 9 lts 0 max-misbal 3
Mar 02 17:13:33 addc-02 dhcpd[2384]: failover peer dhcp-failover: peer moves from \
                                     communications-interrupted to normal
Mar 02 17:13:33 addc-02 dhcpd[2384]: failover peer dhcp-failover: Both servers normal
Mar 02 17:13:35 addc-02 isc-dhcp-server[2375]: Starting ISC DHCP server: dhcpd.
Mar 02 17:13:35 addc-02 systemd[1]: Started LSB: DHCP server.
Mar 02 17:14:02 addc-02 dhcpd[2384]: peer dhcp-failover: disconnected
Mar 02 17:14:02 addc-02 dhcpd[2384]: failover peer dhcp-failover: I move from normal \
                                     to communications-interrupted
Mar 02 17:14:02 addc-02 dhcpd[2384]: failover peer dhcp-failover: peer moves \
                                     from normal to normal
Mar 02 17:14:02 addc-02 dhcpd[2384]: failover peer dhcp-failover: I move from \
                                     communications-interrupted to normal
Mar 02 17:14:02 addc-02 dhcpd[2384]: failover peer dhcp-failover: Both servers normal
Mar 02 17:14:02 addc-02 dhcpd[2384]: balancing pool 7f0f23387300 192.168.56.0/24 \
                                     total 20 free 10 backup 9 lts 0 max-own (+/-)2
Mar 02 17:14:02 addc-02 dhcpd[2384]: balanced pool 7f0f23387300 192.168.56.0/24 \
                                     total 20 free 10 backup 9 lts 0 max-misbal 3
```

Listing 10.0.8: Logfile from starting the DHCP-server

10.1 Conclusion

Now you have two ADDCs with bind9 as the nameserver. The DHCP-Server is also fault tolerant and has a load-balancing.

Index