

Contents

1. [What is DDNS](#)
 2. [How to set up DDNS](#)
 1. [Generate a key for verification](#)
 2. [DNS Server Configuration](#)
 3. [DHCP Server Configuration](#)
 4. [Restart the servers](#)
 5. [Test it](#)
 3. [See also](#)
 4. [External links](#)
-

What is DDNS

DDNS — Dynamic DNS

DDNS is a service that can be used to automatically update DNS records if client PCs get their IP settings from a DHCP Server. These updates are usually performed by the DHCP Server. DDNS is handy if you have a DNS Server in your local network that should be able to resolve the names of your local PCs. This information should not be forwarded to outside your network, unless you use public IP addresses.



In this example configuration I use private IP addresses and therefore configure the DNS Server to not forward this information to a public DNS Server. Although, in case your set up requires it to forward this information, I'll describe also this case at the appropriate point.



You should **already have a working DHCP and DNS server set up** before following the below instructions. You can find information on how to do that at [DHCP_Server](#) and [Bind9](#).

How to set up DDNS

Generate a key for verification

We have to create a key, that will be used to secure the exchange of information between DHCP and DNS server. Only our DHCP server should be allowed to perform DNS record updates, not just anyone.

Generate the key

```
root# dnssec-keygen -a HMAC-MD5 -b 128 -r /dev/urandom -n USER DDNS_UPDATE
```

This will create two files, **Kdhcp_updater.*.key** and **Kdhcp_updater.*.private**. Copy the key from the *.private file (the line with the key should look similar to this one:

```
Key: pRP5FapFoJ95JEL06sv4PQ==
```

Everything after "**Key:** " is the actual key.

Create the file ddns.key

Now create a new file (ddns.key) with the following content (don't forget to replace <key> with your key):

```
key DDNS_UPDATE {  
    algorithm HMAC-MD5.SIG-ALG.REG.INT;  
    secret "<key>";  
};
```

in my case it would look like this:

```
key DDNS_UPDATE {  
    algorithm HMAC-MD5.SIG-ALG.REG.INT;  
    secret "pRP5FapFoJ95JEL06sv4PQ==";  
};
```

Copy the key into the correct locations

Copy this file to **/etc/bind/** and **/etc/dhcp** and adjust the file permissions as follows:

```
root# cp ddns.key /etc/bind/  
root# cp ddns.key /etc/dhcp/  
root# chown root:bind /etc/bind/ddns.key  
root# chown root:root /etc/dhcp/ddns.key  
root# chmod 640 /etc/bind/ddns.key  
root# chmod 640 /etc/dhcp/ddns.key
```

or in two lines:

```
install -o root -g bind -m 0640 ddns.key /etc/bind/ddns.key  
install -o root -g root -m 0640 ddns.key /etc/dhcp/ddns.key
```

DNS Server Configuration

Configure zones to be updated

The DNS server must be configured to allow updates for each zone that the DHCP server will be updating. In our example the clients in the example.org domain will be assigned addresses on the 192.168.2.0/24 subnet. We will need a key declaration for our key, and two zone declarations - one for the forward lookup zone and one for the reverse lookup zone. To do so add the following to the file **/etc/bind/named.conf.local**:

```
include "/etc/bind/ddns.key";  
  
zone "example.org" {  
    type master;  
    notify no;  
    file "/var/cache/bind/db.example.org";  
    allow-update { key DDNS_UPDATE; };  
};  
  
zone "2.168.192.in-addr.arpa" {  
    type master;  
    notify no;  
    file "/var/cache/bind/db.192.168.2";  
    allow-update { key DDNS_UPDATE; };  
};
```

The option **notify no** stops named (the DNS daemon) from forwarding information about the local network to external DNS servers. This is only useful if you use private addresses in your network. In case you use public addresses, you want your DNS to forward that information to public DNS servers. To configure this, just delete the line **notify no**;

Create the zone files

Then you have to create two zone files, one for the forward lookup zone (db.example.org) and one for the reverse lookup zone (db.192.168.2). These are the zones you defined previously in the file **/etc/bind/named.conf.local**.

You can copy the sample file **db.empty** and then just add your changes.

```
root# cp db.empty db.example.org
root# cp db.empty db.192.168.2
```

The following listing shows the contents of **db.empty**, which is (after the previous step) the same as **db.example.org** and **db.192.168.2**.

```
; BIND reverse data file for empty rfc1918 zone
;
; DO NOT EDIT THIS FILE - it is used for multiple zones.
; Instead, copy it, edit named.conf, and use that copy.
;
$TTL 86400
@ IN SOA localhost. root.localhost. (
        1      ; Serial
        604800 ; Refresh
        86400  ; Retry
        2419200 ; Expire
        86400 ) ; Negative Cache TTL
;
@ IN NS localhost.
```

So let's first edit **db.example.org**. It should afterwards look like this:

```
; Zone file for example.org
;
$TTL 86400
@ IN SOA example.org. root.example.org. (
        1      ; Serial
        604800 ; Refresh
        86400  ; Retry
        2419200 ; Expire
        86400 ) ; Negative Cache TTL
;
@ IN NS ns.example.org.
ns IN A 192.168.2.1
```

and then **db.192.168.2**:

```

; Zone file for 192.168.2

;
$TTL 86400
@ IN SOA example.org. root.example.org. (
        1      ; Serial
        604800 ; Refresh
        86400  ; Retry
        2419200 ; Expire
        86400 ) ; Negative Cache TTL

;
@ IN NS ns.example.org.
1 IN PTR ns.example.org.

```

The trailing dot is important, because it makes the name a FQDN (Fully Qualified Domain Name). In this case the nameserver's IP address is **192.168.2.1** and it's name is **ns**. Adjust the files according to your network's configuration.

Create symbolic links

Finally we need to create links from `/var/cache/bind` to the actual zone files in `/etc/bind`. This is because `/etc/bind` is not writeable for `bind`, but `/var/cache/bind` is.

```

root# cd /var/cache/bind
root# ln -s /etc/bind/db.example.org .
root# ln -s /etc/bind/db.192.168.2 .

```

Be careful about the permissions of the directory that contains these links. The user/group `bind` should be able to write to it. If the permissions are like in the following listing, it should be ok.

```

drwxrwxr-x 2 root bind 4.0K Jul 25 14:54 bind/

```

DHCP Server Configuration

Additionally you have to configure your DHCP server to update these zones. To do so, you need to edit the file `/etc/dhcp/dhcpd.conf`. This file consists of different sections. In the global section, which is everything that is not inside a Subnet-, or group block, you have to add the following:

```

option domain-name "example.org";

```

```
ddns-updates      on;
ddns-update-style interim;
ignore            client-updates;
update-static-leases on;
```

option domain-name

This options specifies the domain name, which is also used for DDNS.

ddns-update-style

This option should always be interim. The only other option is `adhoc`, but that one is outdated .

client-updates

If you use the option **allow client-updates**, the clients are allowed to register their name at the DNS server by themselves. We don't want this, so I added **ignore client-updates** (which is already the default option), just to show this explicitly.

update-static-leases

By default the DHCP-Server doesn't update the DNS entries of static leases. If you want it to update them, you need to set this option to `on`. It can be that this causes some problems, that's why the manpage of `dhcpcd.conf` doesn't recommend the use of it. If you experience problems, turn it off, but then you have to configure these hosts statically not only for DHCP, but also for DNS.

You have to add the following lines for the zones that shall be updated by your DHCP server.

```
include "/etc/dhcp/ddns.key";

zone example.org. {
    primary 127.0.0.1;
    key DDNS_UPDATE;
}

zone 2.168.192.in-addr.arpa. {
    primary 127.0.0.1;
    key DDNS_UPDATE;
}
```

The primary statement specifies the IP address of the name server whose zone information is to be updated. In this case DHCP and DNS server are running on the same machine, that's why we put 127.0.0.1 there. If that's not the case, you have to put the IP address of the machine that the DNS server is running on there. The zone descriptions have to end with a period.

The complete **dhcpcd.conf** file after the previous steps and with a basic configuration for the subnet 192.168.2.0/24:

```
authoritative;
option domain-name "example.org";
option domain-name-servers ns.example.org;

ddns-updates on;
ddns-update-style interim;
ignore client-updates;
update-static-leases on;

default-lease-time 600;
max-lease-time 7200;
log-facility local7;

include "/etc/dhcp/ddns.key";

zone EXAMPLE.ORG. {
    primary 127.0.0.1;
    key DDNS_UPDATE;
}

zone 2.168.192.in-addr.arpa. {
    primary 127.0.0.1;
    key DDNS_UPDATE;
}

subnet 192.168.2.0 netmask 255.255.255.0 {
    range 192.168.2.10 192.168.2.100;
    option routers 192.168.2.1;
}
```

Restart the servers

```
root# /etc/init.d/isc-dhcp-server restart
root# /etc/init.d/bind9 restart
```

Test it

Now that everything is set up it's time to test it.

The easiest way to do so, is to connect a PC to your network and then check **/var/log/syslog** at the server. You can do so by issuing the following command:

```
root# tail -f /var/log/syslog
```

If everything works fine, you should find something similar to the following lines in this command's output:

```
named[1724]: client 127.0.0.1#59506: signer "ddns_update" approved
named[1724]: client 127.0.0.1#59506: updating zone 'example.org/IN': adding an RR at 'joob.example.c
named[1724]: client 127.0.0.1#59506: updating zone 'example.org/IN': adding an RR at 'joob.example.c
dhcpd: Added new forward map from joob.example.org to 192.168.61.36
named[1724]: client 127.0.0.1#48511: signer "ddns_update" approved
named[1724]: client 127.0.0.1#48511: updating zone '2.168.192.in-addr.arpa/IN': deleting rrsset at '36.2
named[1724]: client 127.0.0.1#48511: updating zone '2.168.192.in-addr.arpa/IN': adding an RR at '36.2
dhcpd: added reverse map from 36.2.168.192.in-addr.arpa. to joob.example.org
```

If not, you might want to do your own research on the Internet and if you find a solution, you might want to add information about it to this document.

Another way to test it

Go to your client computers and enable them to take an IP from a DHCP server. With the following command check if your client computer name is updated in DNS. It will resolve your name with the newly allocated IP.

```
nslookup yourcomputername.example.com
```

Good Luck with your newly created Dynamic DNS Server.

See also

See [Bind9](#) and [DHCP Server](#) for more information on DNS and DHCP.

External links

Julien Valroff wrote a great article about a DDNS setup similar to services like DynDNS.org or Zonedit.com. You can find it [here](#).

Here are some external resources which will allow you to study the DDNS methods in detail:

- <https://www.isc.org/software/bind> - ISC BIND homepage
- <https://www.isc.org/software/dhcp> - ISC DHCP homepage
- <https://web.archive.org/web/20020421014410/http://www.oceanwave.com/technical-resources/unix-admin/nsupdate.html> - A DDNS Server Using BIND and Nsupdate
- <http://web.archive.org/web/20120130200400/http://ops.ietf.org/dns/dynupd/secure-ddns-howto.html> - Secure dynamic DNS howto
- <http://dag.wieers.com/howto/bits/bind-ddns.php> - Bind Dynamic DNS (DDNS) updates using nsupdate
- <https://web.archive.org/web/20180831164247/http://linux.yyz.us/nsupdate/> - nsupdate: Painless Dynamic DNS

DNS:

- <http://www.aboutdebian.com/dns.htm> - How To Set Up Linux DNS Services
- <http://www.rscott.org/dns/> - DNS Oversimplified

DDNS:

- <http://www.debianadmin.com/howto-setup-dhcp-server-and-dynamic-dns-with-bind-in-debian.html> - Howto setup DHCP Server and Dynamic DNS with BIND in Debian
- https://www.debian-administration.org/article/Configuring_Dynamic_DNS_DHCP_on_Debian_Stable - Configuring Dynamic DNS & DHCP on Debian Stable
- <https://www.debian.org/how-to-set-a-lan-gateway-with-dhcp-dynamic-dns-and-iptables-debian-etch/> - How-To: Set up a LAN gateway with DHCP, Dynamic DNS and iptables on Debian Etch
- <http://www.randombugs.com/linux/linux-isc-dhcp-server-dynamic-dns-updates-debian-ubuntu.html> - Linux ISC DHCP Server and Dynamic DNS updates under Debian and Ubuntu

[CategoryNetwork](#) [CategorySoftware](#) [CategorySystemAdministration](#)