

Eigener DynDNS Dienst

Eine eigene Toplevel Domain die direkt immer auf die IP des Internetanschlusses zuhause zeigt. Quasi der eigene DynDNS-Dienst – ohne die Hilfe von (kostenlosen oder kostenpflichtigen) DynDNS-Providern. Das wärs!



Für dieses Setup braucht man:

- **Eine eigene Domain** (am einfachsten über [Hostpoint](#) registriert)
- **Einen Slave-DNS Eintrag bei den Hostpoint DNS Servern**, (der auf meine aktuelle IP zeigt)
- **Einen Netzwerk lokalen root-DNS-Server** der alles managed (meinem Fall Ubuntu 16.04)

Einleitung

Für jede .de-Domain (die bei der denic registriert wird) braucht man zwei Namensserver welche sagen können welche IP hinter der Domain steckt. Dies sind die autoritiven Nameserver. Die Vorgaben der Denic möchten hier auch dass die Nameserver nicht genau die gleiche IP-Adresse haben (klar, oder?) aber auch nicht ähnliche Adressen (gleicher Netzbereich)

Da wir bei unserem root-Server nicht extra zwei IPs einrichten wollen, nur damit die Denic zufrieden ist, und da dass auch aus Ausfallgründen eher unpraktisch ist brauchen wir also noch einen zweiten Nameserver der nichts anderes macht als die Daten unsere Nameservers zu übernehmen. Dies ist der sogenannte Slave-Nameserver und man kann diesen bei Hetzner einfach per Klick bestellen (bzw. einen Eintrag auf dem Nameserver, einen ganzen Nameserver brauchen wir ja nicht!).

Für diese Anleitung gehen wir davon aus dass wir unseren root-Server meinroot.de haben und die Domain meinzu Hause.de uns gehört.

Installation

Zuallererst installieren wir auf meinroot.de den DNS-Server bind:

```
# apt-get install bind9
```

Bind bekommt eine Zonendatei für meinzu Hause.de, diese liegt in `./etc/bind/db.meinzu Hause.de` und sieht wie folgt aus:

```
$ORIGIN .
$TTL 60; 1 minute
meinzuhause.de      IN SOA  www.meinroot.de. dn.meinroot.de. (
                    2014090711; serial
                    7200; refresh (2 hours)
                    1800; retry (30 minutes)
                    604800; expire (1 week)
                    21600; minimum (6 hours)
                    )
                    NS   ns1.first-ns.de.
                    NS   www.meinroot.de.
                    NS   robotns2.second-ns.de.
                    A    89.14.9.131
                    MX   10 mail.meinzuhause.de.meinzuhause.de.
                    AAAA 2002:590e:2e2e:0:ele3:60f5:97d:9429
$ORIGIN meinzuhause.de.
mail                CNAME  meinzuhause.de.
www                 CNAME  meinzuhause.de.
```

Jetzt erzeugen wir einen Update-Key damit wir die Datei programmtechnisch updaten dürfen:

```
# /usr/sbin/ddns-confgen -z meinzuhause.de
```

Diesen Key schreiben wir in die Keydatei `./etc/bind/ddns-key.meinzuhause.de.conf`:

```
key "ddns-key.meinzuhause.de" {
    algorithm hmac-sha256;
    secret "CTQpYK6Dfdsf334fsd08LSY7ApziY3423fds8zw=";
};
```

Das Key-File müssen wir in der bind-Config `./etc/bind/named.conf` inkludieren:

```
// Update-Keyfile
include "/etc/bind/ddns-key.meinzuhause.de.conf";
```

Und den Key nochmal hinterlegen in `./etc/bind/named.conf.local`:

```
zone "meinzuhause.de" {
    type master;
    file "/etc/bind/db.meinzuhause.de";
    update-policy { grant ddns-key.meinzuhause.de zonesub ANY; };
};
```

Nun müssen wir noch erlauben dass die Hetzner-Nameserver einen Zonentransfer machen dürfen, Datei `./etc/bind/named.conf.options` eintragen:

```
// Zonen-Transfer fuer Hetzner Nameserver
allow-transfer { 213.239.242.238;213.133.105.6;193.47.99.3; };
```

Jetzt brauchen wir noch ein Script welches von außen das Updaten des DynDNS-Eintrages vornimmt – und (als extrawurst) auch gleich noch einen Eintrag in eine Mysql-Tabelle durchführt: Bühne frei für `./root/bin/dyndns.sh`:

[./root/bin/dyndns.sh](#)

```
#!/bin/bash

NS="127.0.0.1"
NSKEY="/etc/bind/ddns-key.meinzuhause.de.conf"
TMPFILE="/var/www/html/dyndns.txt"
ZONE="meinzuhause.de"
TIMEOUT=60

OLDIP=$(dig @$NS $ZONE A | grep ^$ZONE.*A | awk '{ print $5 }')
NEWIP=$(cat $TMPFILE)

echo "old: $OLDIP"
echo "new: $NEWIP"

if [ "$OLDIP" != "$NEWIP" ]; then
    echo -e "
update delete $ZONE. A
update add $ZONE. $TIMEOUT A $NEWIP
show
send" | /usr/bin/nsupdate -k $NSKEY
    rndc freeze $ZONE
    rndc thaw $ZONE
else
    echo "no update required"
fi
```

Dieses Script holt die aktuelle IP aus der Datei `./var/www/html/dyndns.txt` und kümmert sich sowohl um das Update des Nameservers, als auch ums weiterleiten der Zonendaten an die Hetzner-Slave Nameserver. Dieser Cronjob (unter root) ruft das Script minütlich auf:

```
* * * * * /root/bin/dyndns.sh > /dev/null 2>&1
```

Jetzt brauchen wir nur noch ein PHP-Script welches von außen per URL erreichbar ist und die Datei `./var/www/html/dyndns.txt` mit der übergebenen IP beschreibt, das packen wir unter `./var/www/html/dyndns.php`:

[./var/www/html/dyndns.php](#)

```
<?php

$username = "meintollerdyndnsdienst";
$password = "miteinemgeheimenpasswort";
$dyntxt = "/var/www/html/dyndns.txt";
```

```
$db = "dyndnssettings";

if($username == $_GET['username'] && $pass == $_GET['password'])
{
    $a = fopen("$dyntxt", "w");
    fwrite($a, $_GET['ipv4']);
    fclose($a);

    $mysqli = new mysqli("localhost", $username, $pass, $db);
    if ($mysqli->connect_errno)
    {
        echo "Failed to connect to MySQL: (" .
    $mysqli->connect_errno . ") " . $mysqli->connect_error;
    }

    if (!$mysqli->query("INSERT INTO dyndns(domain, ipv4, ipv6)
VALUES ('" . $_GET['domain'] . "', '" . $_GET['ipv4'] . "', '" .
$_GET['ipv6'] . "')"))
    {
        echo "Insert-Error: (" . $mysqli->errno . ") " .
    $mysqli->error;
    }

    mysqli_close($mysqli);

    echo "success";
}

?>
```

Username und Passwort werden nicht nur im Script geprüft um zu checken ob derjenige den Update machen darf, sie werden auch gleich zum anmelden an Mysql genutzt. Es muss also der entsprechende Mysql-Nutzer existieren. Das Tabellenformat ist einfach:

```
CREATE TABLE `dyndns` (
  `updatetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `domain` varchar(256) CHARACTER SET utf8 NOT NULL,
  `ipv4` varchar(25) CHARACTER SET utf8 NOT NULL,
  `ipv6` varchar(100) CHARACTER SET utf8 NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Nun müssen wir bei Hetzner einen Slave-DNS Eintrag bestellen, dort tragen wir die IP von meinroot.de ein.

Bei der Denic hinterlegen wir nun als Nameserver 1 unseren meinroot.de und als Nameserver 2 irgendeinen der Hetzner-Nameserver (ns1.first-ns.de, robotns2.second-ns.de, robotns3.second-ns.com)

Nun sollten wir alles schon gut mit dem Aufruf der URL testen und debuggen können. Zum prüfen was

die Nameserver gerade als IP für meizuhause.de rausgeben dienen folgende Befehle:

```
# dig @127.0.0.1 meizuhause.de
# dig @www.meinroot.de meizuhause.de
# dig @ns1.first-ns.de meizuhause.de
# dig @robotns2.second-ns.de meizuhause.de
```

Die Fritzbox bekommt nun folgende Konfiguration unter DynDNS:

```
Dynamic DNS-Anbieter: Benutzerdefiniert
Update-URL:
https://www.meinroot.de/dyndns.php?username=<username>&password=<pass>&domain=<domain>&ipv4=<ipaddr>&ipv6=<ip6addr>
Domainname: meizuhause.de
Benutzername: meintollerdynsdienst
Kennwort: miteinemgeheimenpasswort
```

Meine Fritzbox brachte hier noch im Systemlog alle paar Minuten folgenden (nicht tragischen) Fehler:

Dynamic DNS-Fehler: Die Dynamic DNS-Aktualisierung war erfolgreich, anschließend trat jedoch ein Fehler bei der DNS-Auflösung auf.

Wenns einen ströt wird man das los indem man sich mit Telnet auf die Fritzbox verbindet und die Datei ‚var/flash/ar7.cfg‘ so verändert dass under ‚liveday‘ ‚1d‘ steht.

-
- <https://www.thomaschristlieb.de/eigener-dyndns-dienst-mit-fritzbox-und-hetzner/>

Last update: **2017/03/31 12:03**