

# Eigener DynDNS Dienst einrichten

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 Domain (die bei Hostpoint registriert wird) braucht man **zwei Namensserver** welche sagen können welche IP hinter der Domain steckt. Diese Server sind die autoritiven Nameserver. Laut den Vorgaben von Hostpoint, sollten hier die eingetragenen Nameserver *nicht* genau die gleiche IP-Adresse haben und auch nicht ähnliche Adressen (gleicher Netzbereich).

Da wir nun, unserem root-Server hierfür nicht extra zwei verschiedene IP-Adressen einrichten wollen, nur damit Hostpoint zufrieden ist, brauchen wir also noch einen zweiten Nameserver der nichts anderes macht als die Daten unseres Nameservers zu übernehmen. Dies ist der sogenannte **Slave-Nameserver**. Diesen kann man bei Hostpoint einfach per Klick erstellen bzw. einfügen und zwar indem man bei den Domain Einstellungen den obersten Nameserver durch seinen eigenen ersetzt. Somit werden die zwei unteren Nameserver (ns2.hostpoint.ch und ns3.hostpoint.ch) automatisch zu Slave-Nameservern.

Für diese Anleitung gehe ich davon aus dass ein frisch installiertes **Ubuntu 16.04**, dass später als **root-Server meinroot.ch** Konfiguriert wird, einsatzbereit und zur Verfügung steht. Weiter, sollte man bereits im Besitz einer eigenen Domain bei Hostpoint sein. z.B. wie meine → **Domain blackgate.org**.

---

## Installation

Zuallererst wird auf meinroot.ch den DNS-Server installiert:

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

Als nächstes, wird in Bind eine neue Zonendatei für blackgate.org erstellt, diese liegt in /etc/bind/db.blackgate.org und sieht wie folgt aus:

```
$ORIGIN .
$TTL 60; 1 minute
blackgate.org      IN SOA  ns.meinroot.ch. dn.meinroot.ch. (
                    2017033101; serial
                    7200; refresh (2 hours)
                    1800; retry (30 minutes)
                    604800; expire (1 week)
                    21600; minimum (6 hours)
                    )
                    NS   ns.hostpoint.ch.
                    NS   ns.meinroot.ch.
                    NS   ns2.hostpoint.ch.
                    A    83.150.6.68
                    MX   10 mail.blackgate.org.blackgate.org.
$ORIGIN blackgate.org.
mail                CNAME  blackgate.org.
www                 CNAME  blackgate.org.
```

Jetzt erzeugen wir einen Update-Key, damit unser Server die Datei später auch automatisch updaten darf:

```
# /usr/sbin/ddns-confgen -z blackgate.org
```

Diesen Key schreiben wir in die Keydatei "/etc/bind/ddns-key.blackgate.org.conf":

```
key "ddns-key.blackgate.org" {
    algorithm hmac-sha256;
    secret "CTQpYK6Dfdsf334fsd08LSY7ApziY3423fds8zw=";
};
```

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

```
// Update-Keyfile
include "/etc/bind/ddns-key.blackgate.org.conf";
```

Dannach wird der Key auch noch in "/etc/bind/named.conf.local" hinterlegt:

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

Nun muss noch erlaubt werden, dass die Hostpoint-Nameserver überhaupt einen Zonentransfer machen dürfen. Dafür, wird in der Datei "/etc/bind/named.conf.options" folgendes eintragen:

```
// Zonen-Transfer fuer Hostpoint Nameserver
allow-transfer { 217.26.51.254; 217.26.53.254; 217.26.48.126; };
```

Jetzt brauchen wir noch ein Script, welches dann anschliessend von aussen das Update des DynDNS-Eintrages vornimmt - und zusätzlich auch gleich noch zu Log-Zwecken einen Eintrag in eine MySQL-Tabelle durchführt:

Dafür gibt es folgendes Skript, welches hier → `/root/bin/update_dyndns.sh` abgelegt wird:

[/root/bin/update\\_dyndns.sh](#)

```
#!/bin/bash

NS="127.0.0.1"
NSKEY="/etc/bind/ddns-key.blackgate.org.conf"
TMPFILE="/var/www/html/new_dynamic_ip.txt"
ZONE="blackgate.org"
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/new_dynamic_ip.txt` und kümmert sich sowohl um das Update des Nameservers, als auch ums weiterleiten der Zonendaten an den Hostpoint-Slave Nameserver.

Diese Shell-Datei, wird dann automatisiert von Zeit zu Zeit, per [cronjob](#) von root ausgeführt. Zum erstellen, des cronjobs, wird die `/etc/crontab` Datei editiert und folgender Eintrag eingefügt:

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

Jetzt wird nur noch ein PHP-Skript gebraucht, welches per URL erreichbar ist und die **Datei** `/var/www/html/new_dynamic_ip.txt` mit der übergebenen IP beschreibt. Dazu, wird folgendes PHP Skript gebraucht:

[/var/www/html/ip\\_update.php](#)

```
<?php

//
$username = "michael";
$pass = "Me1nP4sswort123!";
$dyntxt = "/var/www/html/new_dynamic_ip.txt";
$db = "dyndns";

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

    //Dieser Teil mit der Datenbank ist optional und kann (Wenn man
    es nicht nutzen möchte) bis zur Unteren Markierung auskommentiert
    werden!
    $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);
    //Bis hier und NICHT WEITER, kann auskommentiert werden!

    echo "success";
}

?>
```

Der Username und das Passwort werden nun vom Script genutzt, das nicht jeder x-beliebige ein IP update veranlassen kann. Weiter werden diese credentials auch gleichzeitig zum anmelden an der MySQL Datenbank genutzt. Es muss also auch der entsprechende MySQL-User und die Tabelle existieren.

**SQL-Datei zum erstellen der MySQL Struktur:**

[create\\_table.sql](#)

```
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;
```

Als nächstes, wird bei **Hostpoint** einen **Slave-DNS** Eintrag bestellt; eingerichtet. Hierbei wird dann die **IP von meinroot.ch** eingetragen.

Dannach, hinterlegen wir noch als Nameserver 1 unseren meinroot.ch und als Nameserver 2 irgendeinen der Hostpoint-Nameserver (ns.hostpoint.ch, ns2.hostpoint.ch, ns3.hostpoint.ch)

Nun sollte man auch schon bereits zum ersten mal die URL zum testen aufrufen können. Zum überprüfen, was die Nameserver nun gerade als IP für blackgate.org rausgeben dienen folgende Befehle:

```
# dig @127.0.0.1 blackgate.org  
# dig @ns.meinroot.ch blackgate.org  
# dig @ns.hostpoint.ch blackgate.org  
# dig @ns2.hostpoint.ch blackgate.org
```

Der EWB- oder Swiscom-Router, bekommt nun folgende Konfiguration unter DynDNS:

```
Dynamic DNS-Anbieter: Benutzerdefiniert  
Update-URL:  
https://ns.meinroot.ch/ip_update.php?username=<username>&password=<pass>&domain=<domain>&ipv4=<ipaddr>&ipv6=<ip6addr>  
Domainname: blackgate.org  
Benutzername: michael  
Kennwort: MeInP4sswort123!
```

Nach diesem Schritt, sollte nun der Router selbständig alle paar Minuten die *externe IP Adresse* auf Änderungen prüfen und falls welche festgestellt würden, würde die neue IP in das Webverzeichnis des neu installierten DNS-root-servers hinterlegt.

Von da aus, wird dann die IP-Adresse von unserem Shell-Skript, dass jede Minute via Crontab ausgeführt wird wieder auf eine Änderung überprüft; falls eine festgestellt wird. Wird die neue externe IP Adresse in unseren lokalen DNS Server eingetragen welches sich dann automatisch mit dem DNS Server von Hostpoint synchronisiert.

**Somit bleibt nun meine Dynamische IP Adresse immer aktuell und meine Domain bleibt**

**immer erreichbar!** 😎

Last update: **2017/03/31 14:25**