Debian Router/Gateway in 10 Minuten

28.11.2016 27

Es gibt viele Gründe einen eigenen selbstkonfigurierten Router/Gateway einzusetzen.
In diesem Guide beschreibe ich, wie man zwei Server mit gemeinsamen internen Private-Netzwerk und Debian 8 über die gridscale RESTful-API aufsetzt.
Nur einer der beiden Server hat Zugriff aufs Internet, arbeitet als Gateway zwischen privatem Netz und Internet und versorgt gleichzeitig das interne Private-Netzwerk mit IPs via DHCP.

Wie man eine Beispielkonfiguration mit der gridscale RESTful-API aufbauen kann, erfährst du hier: Zwei Server mit internem Netz via RESTful API installieren
Für dieses Guide hier lässt man lediglich bei einem der beiden Server die Verbindung zum Internet über das Public-Netzwerk weg.

Wenn du die beiden Server statt über die API einfach im gridscale Panel bauen möchtest, kannst du den API-Teil einfach überspringen. Der Rest dieses Guides setzt einfach nur 2 mit Debian 8 installierte Server mit der Netzwerkkonfiguration wie gleich beschrieben voraus.

Wir benötigen 2 Server mit Debian 8 mit folgenden Annahmen:

1. Erstellen eines Private-Netzwerks mit dem die beiden Server verbunden werden
2. Router/Gateway mit 2 Netzwerk-Interfaces

* Public-Netzwerk (eth0)
* Private-Netzwerk (eth1)

3. geschützter Server mit 1 Netzwerk-Interface

* Private-Netzwerk (eth0)

Der Router/Gateway erhält eine öffentliche IP zugewiesen, der geschützte Server keine (dies macht dann das Router/Gateway später).

Ok, los gehts 🙂

1) Netzwerk-Interfaces konfigurieren (Router/Gateway)
Ändere die Werte für „address“, „netmask“ und „broadcast“ passend für dein internes Netzwerk auf dem Router/Gateway.
Der geschützte Server benötigt erstmal keine Änderung der Netzwerk-Konfiguration.

```
# nano -w /etc/network/interfaces

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
allow-hotplug eth0
iface eth0 inet dhcp
iface eth0 inet6 dhcp

# The internal LAN interface (eth1)
allow-hotplug eth1
iface eth1 inet static
   address 10.0.0.1
   netmask 255.255.255.0
   network 10.0.0.0
   broadcast 10.0.0.255
```

2) DNSmasq installieren und konfigurieren (Router/Gateway)

DNSmasq ist ein DNS forwarder und DHCP server. Ändere „domain“ passend zu dem FQDN deines Netzwerks und „dhcp-range“ auf den gewünschte Bereich von DHCP IP Adressen die der Router/Gateway an die Clients auf dem Private-Netzwerk vergeben soll.

```
# apt-get install dnsmasq
# nano -w /etc/dnsmasq.conf
interface=eth1
listen-address=127.0.0.1
domain=your.domain.name
dhcp-range=10.0.0.100,10.0.0.150,12h
```

3) IP Forwarding aktivieren (Router/Gateway)

Die folgende Zeile auskommentieren:

```
# nano -w /etc/sysctl.conf
net.ipv4.ip_forward=1
```

4) iptables installieren und konfigurieren (Router/Gateway)

Als erstes installieren wir ein paar benötigte Tools um gespeicherte iptables Regeln bei nächsten Reboot des Router/Gateways automatisch laden zu können.
Beide Fragen ob die aktuellen iptables Regeln gespeichert werden sollen, kann man mit „Yes“ oder „Ja“ beantworten.

```
apt-get install iptables-persistent
```

Wir editieren nun die von der gerade durchgeführten Installation angelegte Datei „/etc/iptables/rules.v4“.
Als Beispiel richten wir NAT ein, um den Servern im Private-Netzwerk Zugriff auf das Internet zu geben:

```
nano -w /etc/iptables/rules.v4
*nat
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT

*filter
-A INPUT -i lo -j ACCEPT
# ssh erlauben, damit wir uns nicht selbst aussperren
-A INPUT -i eth0 -p tcp -m tcp --dport 22 -j ACCEPT
# eingehenden Traffic erlauben der zu den ausgehenden Verbindungen,
# u.a. für Clients aus dem Private-Netzwerk
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
# alles andere eingehend verbieten
-A INPUT -i eth0 -j DROP
COMMIT
```

5) iptables Regeln aktivieren (Router/Gateway)

```
iptables-restore < /etc/iptables/rules.v4
```

6) Rebooten und prüfen ob alles klappt (Router/Gateway)
Das wars! Nach einem Reboot hast du einen einfachen Router/Gateway für dein Private-Netzwerk.

7) Und was ist jetzt aus dem „geschützten Server“ geworden?
Dieser hat mittlerweile eine IP von dnsmasq erhalten. Vom Router/Gateway aus:

```
root@router-gw:~# journalctl | grep "DHCPOFFER(eth1)"
Jan 28 18:07:16 router-gw dnsmasq-dhcp[994]: DHCPOFFER(eth1) 10.0.0.142 0a:93:33:9f:39:02
```

Also loggen wir uns von dem Router/Gateway aus per SSH (deine Workstation -> Router/Gateway -> geschützter Server) mit dem vorher verwendeten root-Passwort an und prüfen ob der geschützte Server auch wirklich ins Internet kommt und der Router/Gateway wirklich funktioniert:

```
# ssh 10.0.0.142
The authenticity of host '10.0.0.142 (10.0.0.142)' can't be established.
ECDSA key fingerprint is b5:e2:32:54:2d:b3:9c:29:51:f6:15:61:e7:b6:f8:ac.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.142' (ECDSA) to the list of known hosts.
root@10.0.0.142's password:
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan 28 18:14:58 2016

root@secure-server:~# ip a s eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen
    link/ether 0a:93:33:9f:39:02 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.142/24 brd 10.0.0.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::893:33ff:fe9f:3902/64 scope link
       valid_lft forever preferred_lft forever

root@secure-server:~# ping -c 5 www.google.de
PING www.google.de (173.194.113.88) 56(84) bytes of data.
64 bytes from fra02s21-in-f24.1e100.net (173.194.113.88): icmp_seq=1 ttl=59 time=1.05 ms
64 bytes from fra02s21-in-f24.1e100.net (173.194.113.88): icmp_seq=2 ttl=59 time=1.28 ms
64 bytes from fra02s21-in-f24.1e100.net (173.194.113.88): icmp_seq=3 ttl=59 time=0.911 ms
64 bytes from fra02s21-in-f24.1e100.net (173.194.113.88): icmp_seq=4 ttl=59 time=1.01 ms
64 bytes from fra02s21-in-f24.1e100.net (173.194.113.88): icmp_seq=5 ttl=59 time=1.14 ms

--- www.google.de ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.911/1.081/1.284/0.130 ms
```