

Daniel Herrmann Techblog

(Cisco) Networking, VMWare, PHP, MySQL and more...

NAT Cloud in EVE-NG Community Edition

By Daniel Herrmann | 1. April 2018

9 Comments

[EVE-NG](#) is a great network simulation tool available for free (Community Edition) which supports running all kind of network nodes (ASAv, NX-OSv, CSR1000v, Arista vEOS, and so on). There is a long list of supported images [on their site](#).

I used it to prepare for my CCIE R&S switching exam and am now using it to go for CCIE Datacenter. Two things however bugged me from the first time using it:

- There is no NAT cloud to easily connect devices to the Internet (imagine you've built a Linux image, added it to EVE-NG and now realize you forgot to install something)
- Links are not hot-connectable, you need to shut down the devices to connect a link.

Both of this is available in the Professional edition. Apparently, the hot-add feature has [already been implemented as addition to the community edition](#), however it looks like the author decided to create an own [fork of EVE-NG](#) with a [nice feature list](#) which should be available in Summer 2018. And the best thing is, it will be [free and open source](#).

However, it is not there yet and I want to share with you how to create a NAT cloud in the EVE-NG community edition. Essentially, this is a virtual network with a DHCP server, which will allow NAT connections over the management interface of the EVE-NG VM for Internet access.

Interface, DHCP Server and NAT

First of all, we need to create a network which can be used in the topology. I'll be using the predefined pnet9 interface (Cloud 9 network) for this, but any other interface will do.

```
1 ip address add 192.168.255.1/24 dev pnet9
2 echo 1 > /proc/sys/net/ipv4/ip_forward
3 iptables -t nat -A POSTROUTING -o pnet0 -s 192.168.255.0/24 -j MASQUERADE
```

This will assign an IP address to the device, enable IP forwarding in the kernel and establish an iptables rule to NAT the traffic to the pnet0 interface, which has the management IP address assigned. Technically, the pnet devices are bridges, but for the sake of this note this does not matter.

DHCP Server

We need an DHCP server on this interface lest we have to configure all the IP addressing for the Internet connection manually.

```
1 # Install DHCP Server
2 apt-get install isc-dhcp-server
```

Next, we need to modify the DHCP server configs to look as follows:

```
1 INTERFACES="pnet9"

1 authoritative;
2
3 # EVE-NG NAT Interface
4 subnet 192.168.255.0 netmask 255.255.255.0 {
5     range 192.168.255.10 192.168.255.240;
6     interface pnet9;
7     default-lease-time 600;
8     max-lease-time 7200;
9     option domain-name "lab.your-domain.org";
10    option domain-name-servers 8.8.8.8;
11    option broadcast-address 192.168.255.255;
12    option subnet-mask 255.255.255.0;
13    option routers 192.168.255.1;
14 }
```

Finally, start the DHCP server and enable it to start during boot

```
1 systemctl start isc-dhcp-server
2 systemctl enable isc-dhcp-server
```

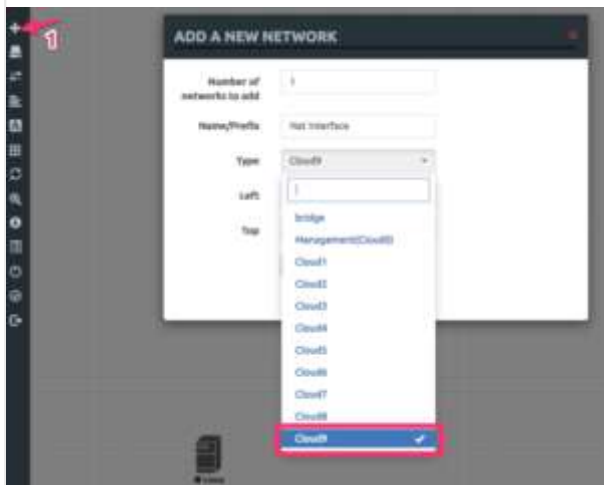
Thats about it. Whenever you use the Cloud 9 interface, there is a DHCP server running which allows Internet access.

Testing it

Create a small lab, for this example I only have a Linux node based on a Debian image. Next, add a network (+ icon on top of the screen, then network), give it a name and select the **Cloud9** interface in the type dropdown.

This will place a small cloud icon in your lab topology, which you can use to connect your topology nodes to the Internet. You can connect as many nodes as you want, there is no limit in terms of the available Ethernet interfaces.

All devices connected to the cloud should have DHCP enabled. They will receive their address from DHCP in the pool 192.168.255.10 – 192.168.255.240. If you have some devices which can or should not receive their address



using DHCP, you can configure them with a static IP address. You should avoid using addresses from the DHCP range, though.

Let's start the lab and see what happens:

```

1 # Looks like we already got the IP address
2 root@debian:~# ip a
3 2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
4   link/ether 00:50:01:00:01:00 brd ff:ff:ff:ff:ff:ff
5   inet 192.168.255.10/24 brd 192.168.255.255 scope global ens3
6       valid_lft forever preferred_lft forever
7
8 # We also got a default route
9 root@debian:~# ip r s
10 default via 192.168.255.1 dev ens3
11 192.168.255.0/24 dev ens3 proto kernel scope link src 192.168.255.10
12
13 # And we can reach our favorite ping target
14 root@debian:~# ping 8.8.8.8
15 PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
16 64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=2.07 ms
17 64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=2.05 ms

```

Looks like NAT Cloud is working in the fee EVE-NG Community Edition.

Category: Allgemein

9 thoughts on "NAT Cloud in EVE-NG Community Edition"



anand

31. July 2018

i am a new to linux iptables dhcp configuration i did exactly what is in the blog. the service isc-dhcp-server fails.

thought of rebooting and checking all the configurations lost except the dhcpd.conf.

i am having trouble getting it to work.. is it possible for you to update the blog with complete steps (like for linux idiots) eg. iptables should be saved .. and how to keep the echo > 1 /proc/sys/net/ipv4/ip_forward to be persistent across reboots. i mean every step

**GaH**

3. September 2018

to save the iptables configuration after we add `iptables -t nat -A POSTROUTING -o pnet0 -s 192.168.255.0/24 -j MASQUERADE` →

1- `sudo apt-get install iptables-persistent`

2- `sudo netfilter-persistent save`

3- `sudo netfilter-persistent reload`

reboot the system and do the following command : `iptables -t nat -L` it should show the rule persistent

for the `/proc/sys/net/ipv4/ip_forward` to be persistent across reboots

1- `nano /etc/sysctl.conf`

2- Uncomment `net.ipv4.ip_forward=1`

Uncomment the next line to enable packet forwarding for IPv4

`net.ipv4.ip_forward=1`

3- issue the following command : `sudo sysctl -p /etc/sysctl.conf`

reboot the system and do the following command : `cat /proc/sys/net/ipv4/ip_forward` it should show 1 in the output

**Jitendra**

11. October 2018

Hi

Great article . One question plz I installed eve on google cloud . And I am facing issues to connect internet from the inside node . So this above article will sort out my issue?

**Tonio5931**

24. November 2018

Hello,

Did you find a solution for Google Cloud ?

I'm not able to make working this fantastic post.. 😞

Thks !

Antoine



Brenton Taylor

25. January 2019

Thanks Dan,

Worked great for me!

Do you know how to grant direct access to nodes from another computer in the same subnet on cloud platforms such as Azure and Google?

From the node connected to Cloud9 interface I can ping a server on the same subnet as eve server but cannot ping from server to eve node.

Tried adding a manual route for 192.168.255.0/24 through eve but no good 😞



Arshad

19. December 2019

In the same way can we configure cisco site to site vpn using two cloud but both cloud natted from the same management interface pnet0 will it work for the testing purpose.



Matti Suuronen

17. February 2020

Thanks Dan,

An old post, but this might still be relevant. A couple of additions were needed in my case:

1) To make the address of the pnet9 persist and be configured automatically after reboot, I modified the /etc/network/interfaces, notice the "static" directive:

```
iface eth9 inet manual
auto pnet9
iface pnet9 inet static
address 192.168.255.1
netmask 255.255.255.0
bridge_ports eth9
bridge_stp off
```

2) The DHCP handshake would proceed fine DHCPDISCOVER > DHCPOFFER > DHCPREQUEST, but then the server would never reply back with a DHCPACK. Turned out I needed to put the bridge interface into promiscuous mode to get this working. I added the following line into /etc/rc.local :

```
ip link set pnet9 promisc on
```

**Alex**

28. February 2020

Hi there,

Thanks for this post.

I'm wondering after this NAT done, how can I access the internal devices (inside EVE) from my real PC? Is that possible? Thanks

**Simon**

19. April 2020

@Brenton – use the VPC subnet instead of 192.168.225.0/24 used in the example above.

The subnet that was allocated by GCP for the region I setup EVE in was europe-west2-c which was 10.154.0.0/20

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.