

Provide outgoing IPv4 connectivity to Docker Container on a IPv6-only Host via clatd (464xlat)

Date: 2022-04-10
modified: 2022-04-10
tags: Linux, Docker, Container, IPv6, NAT
description: Docker Container on IPv6-Only Host with clatd (464xlat)
category: Linux
slug: provide_outgoing_ipv4_connectivity_for_docker_container_on_a_ipv6-only_host_via_clatd_464xlat
Author: Dominik Wombacher
lang: en
transid: provide_outgoing_ipv4_connectivity_for_docker_container_on_a_ipv6-only_host_via_clatd_464xlat
Status: published

My experiences are based on a recent [Harbor](#) setup on [Rocky Linux](#), but should be easily transferable to any other Application and Linux Distribution.

The journey began after Harbor was up and running and the vulnerability Scanner Trivy, used to Scan pushed Images, failed to get it's updates from a IPv4-only address.

So the goal was to get outgoing IPv4 working, incoming traffic is handled by a HAProxy and not further described.

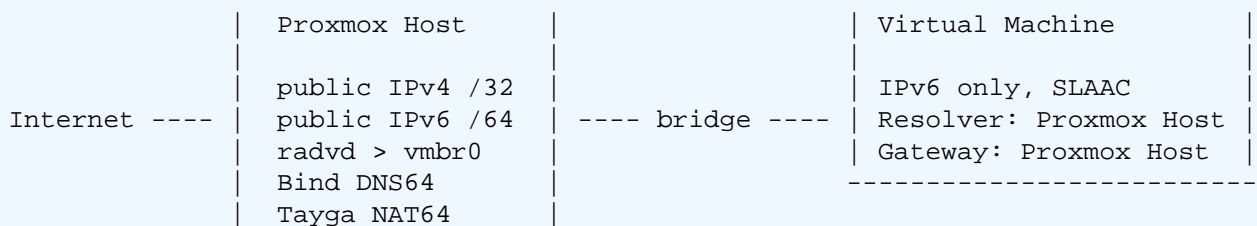
Some background about the Infrastructure: The Proxmox VE Server that host the VM has public IPv4 and IPv6 connectivity, is running radvd to advertise a /64 prefix to a bridge interface, Bind / Named to provide DNS64 and Tayga for NAT64.

This Setup allows the VM to automatically configure IPv6 via SLAAC and also talk to IPv4 targets, e.g. github.com, out of the box. DNS Resolution is done by Bind with enabled DNS64 option on the Proxmox Host, which also act as Gateway for the VM.

If no AAAA Record is available for a site, Bind will return one where the IPv4 address is embedded in a IPv6 address. In my setup I defined `64:ff9b::/96` for that purpose, example result for github.com (IPv4: 140.82.121.4):

```
dig github.com aaaa +short
64:ff9b::8c52:7903
```

Tayga, running on the Proxmox Host, will receive this IPv6 traffic from the VM and perform NAT64, viola github.com is accessible from the IPv6-only VM. More about that setup in a earlier [Post](#).




```
cd ~
git clone https://github.com/toreanderson/clatd
sudo make -C clatd install installdeps

sudo systemctl enable --now clatd

# Configure global IPv4 DNS Server, used by all Container
# Privacy friendly Server by Freifunk München https://ffmuc.net/wiki/doku.php?id=knb:dohdot
/etc/docker/daemon.json
...
{
  "dns" : [ "5.1.66.255", "185.150.99.255" ]
}
...

sudo systemctl restart docker
```

Drawback: That way Container will not be able to reach IPv6-only Systems, the target need to be reachable via IPv4.

At least in my case that's acceptable, main goal is to ensure Trivy can download updates, if you need outgoing IPv6, you have to go either with [ipv6nat](#) or the docker build-in IPv6 feature.

There seem to be no perfect one-size-fits-all Solution, so as often in IT, you have to pick the right tool for the job.