Erlon Cruz
NetApp®

Tiago Pasqualini da Silva
NetApp®

Dustin Schoebrun
redhat.

Agenda

- IPv6 Introduction
- IPv4 to IPv6
- State of IPv6 in OpenStack
- IPv6 architecture in OpenStack: A practical example
- Storage Demo: IPv6 without NAT

NetApp®

redhat.

What is IPv6?

- The next generation of the IP protocol
- Defined in IETF RFC 1883 in 1995, Internet Standard with RFC 8200 in 2017
- Intended as a replacement for IPv4 but capable of living with it
- What changes with IPv6?
  - Address space
    - *IPv4 has a 32 bit address space ~ 4.3 billion addresses*
      - `xxx.xxx.xxx.xxx`
      - Where xxx is an integer from 0-255
    - *IPv6 has a 128 bit address space ~ 3.4 * 10^{38} addresses*
      - `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`
      - Where xxxx is a hexadecimal value from 0000 to ffff

NetApp®

redhat.

Advantages of IPv6

- Virtually unlimited real IP addresses
  - No need for NAT
- Huge standard subnet size
  - 2^{64} addresses, which is the square of the entire IPv4 address space
- No need to assign floating IP addresses to instances
- Routers cannot fragment an IPv6 packet
- Prefix Delegation
- Stateless Address Autoconfiguration (SLAAC)

NetApp®

redhat.

SLAAC

- Allows an IPv6 host to automatically configure itself when connected
- Uses Neighbor Discovery Protocol through ICMPv6 router discovery messages
- A router on the network will respond to this request with a router advertisement packet
  - This packet contains the requirements for address configuration, routes, and required autoconfiguration options
- DHCPv6 and static configuration are also options

Prefix Delegation

- In IPv4, home networks and enterprises typically use private addresses
  - 192.168.xxx.xxx and 10.xxx.xxx.xxx
- However, IPv6 addresses are globally accessible end-to-end
  - So home networks and enterprises now distribute globally routable addresses
  - It becomes difficult to manually provision such networks at a large scale
- DHCPv6 uses Prefix Delegation to assign an address prefix and will automate the configuration and creation of the publicly routable addresses on the network.
  - It does so by assigning a subnet to the router, for example a /64 address space.
  - Will advertise the addresses it allows to the hosts on the network, via SLACC or DHCPv6

Quagga

- Routing protocol network suite that provides implementation of several routing protocols
  - OSPF, RIP, BGP, and others
- GPL licensed
- Allows users to use software-defined networking on their systems
- We will use Quagga to handle the the creation of routes in our OpenStack environment.
  - We set up BGP with Quagga
  - BGP = Border Gateway Protocol

NetApp®

redhat.

IPv6 in OpenStack

NetApp®

redhat.

What is the state of IPv6 in OpenStack?

- IPv6 features have been worked on since the beginning (**B**exar)
- Support was gradually being worked on across projects and releases
- Before Grizzly the configuration process was very obscure
- Icehouse, Juno and Kilo increased adoption

NetApp®

redhat.

What is the state of IPv6 in OpenStack?

- Nova
  - Diablo: initial grow of OpenStack, low maturity of IPv6
  - Grizzly: support for IPv6 in RPC services
  - Incremental fixes, new features and documentation until today

NetApp®

redhat.

What is the state of IPv6 in OpenStack?

- Neutron
  - Grizzly: support for IPv6 in RPC services
  - Juno: support for SLAAC
  - Kilo: support for multiple IPv6 prefixes on internal router ports
  - Liberty: improvements on IPv6 HA routers
  - Last releases: bug fixes and small improvements

NetApp®

redhat.

What is the state of IPv6 in OpenStack?

- Cinder
  - No big changes needed in core
  - Grizzly: support for IPv6 in RPC services
  - Liberty: support for iSER IPv6
  - Last releases: small bug fixes

NetApp®

redhat.

What is the state of IPv6 support in Manila?

- Pike
  - Initial IPv6 implementation
    - *Support for IPv6 access rules and export locations*
    - *Support for IPv6 in network plugins in neutron*
    - *No third party vendor support*
  - Not thoroughly tested
- Queens
  - Vendors adding support and fixing bugs
  - CI scenario tests added
  - Devstack plugins support

NetApp®

redhat.

IPv6 Architectures in OpenStack

NetApp®

redhat.

IPv6 Dev Architecture in OpenStack

- We will show how easy it is to setup a test environment for IPv6 with Manila
  - Manila started supporting IPv6 in Queens
- For the following demo, we have used our development lab hypervisors and storage devices
  - Baremetal hypervisors running Ubuntu and KVM
  - NetApp ONTAP devices
- Hypervisors are connected to two IPv6 networks: management and data
- ONTAP are also connected to the same networks, with IPv6 interfaces

NetApp®

redhat.

IPv6 Dev Architecture in OpenStack

- OpenStack runs on VMs hosted on those hypervisors
  - Each VM connects to a bridge on the hypervisor
- We used Devstack for a simple and easy test setup
  - Just for testing purposes, no real scenario
- Each tenant has a router with gateways to private and public networks
- BGP is needed on host to route packets to the correct tenant router
  - Manila Devstack script sets up Quagga for BGP

IPv6 Dev Architecture in OpenStack

- No floating IPs are needed
- Hypervisor can access VM via private IP
  - Admin tier: fd12::/16
  - Hypervisor tier: fd12:1::/32
  - Devstack tier: fd12:1:1::/48
  - Public subnet: fd12:1:1:1::/64
  - Private subnet: fd12:1:1:0::/64
- In this same scenario, with IPv4, we would commonly use floating IPs

NetApp®

redhat.

local.conf for Devstack

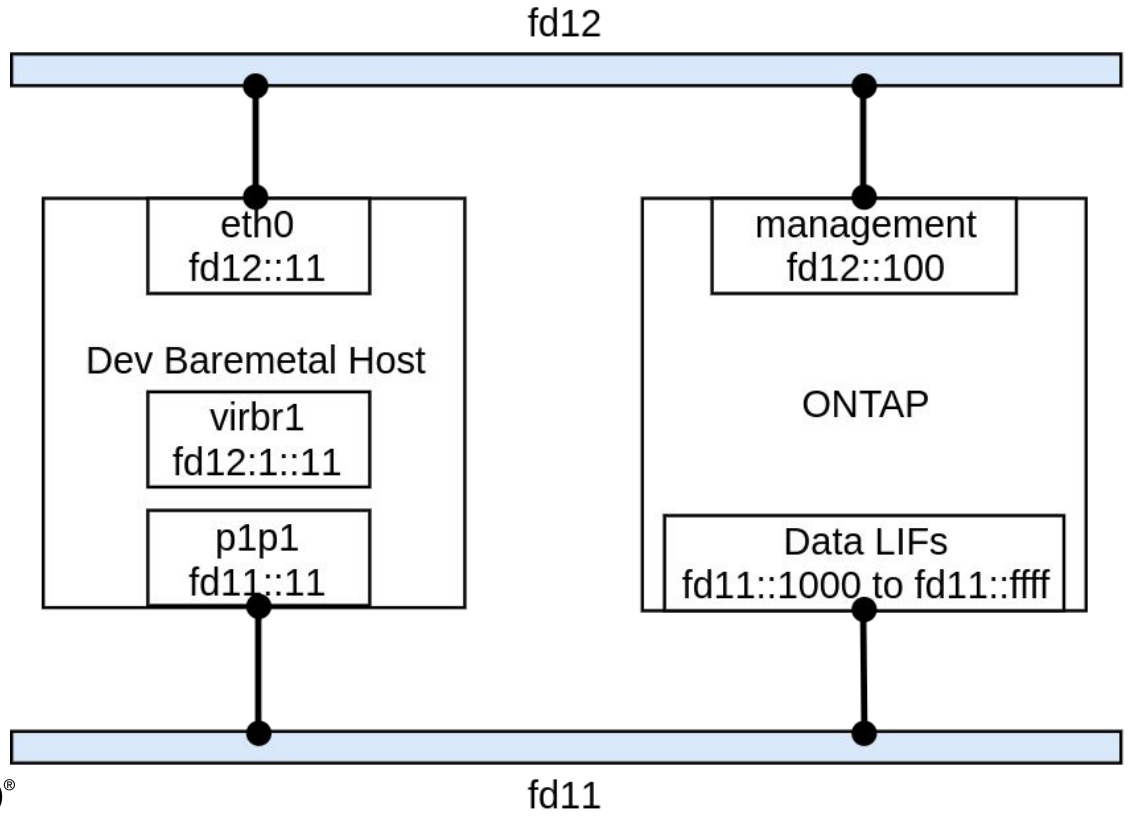
To setup Devstack for IPv6, the settings needed are:

```
SUBNETPOOL_PREFIX_V6=fd12:1:1::/48
MANILA_SETUP_IPV6=True
FLOATING_RANGE=172.24.5.0/24
PUBLIC_NETWORK_GATEWAY=172.24.5.1
IP_VERSION=4+6
NEUTRON_CREATE_INITIAL_NETWORKS=False
enable_plugin neutron-dynamic-routing https://git.openstack.org/openstack/neutron-dynamic-routing
```

NetApp®

redhat.

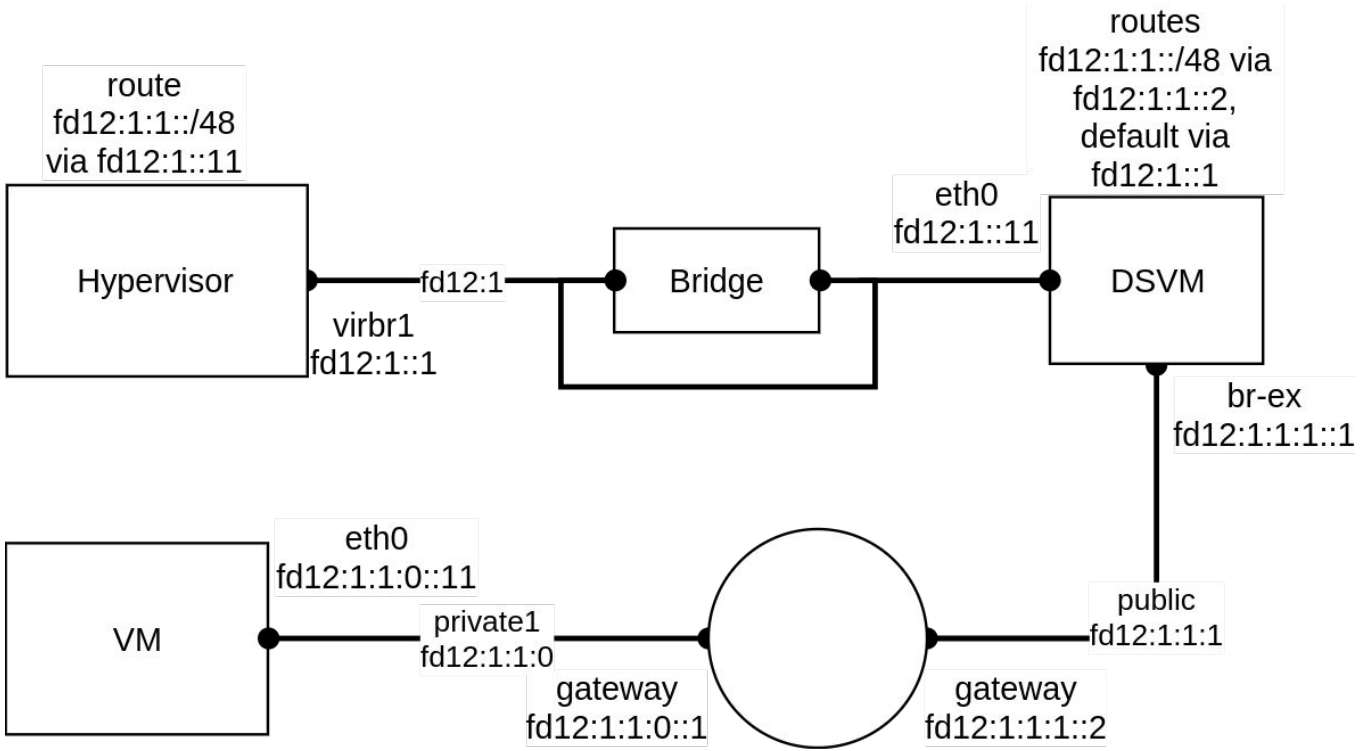
IPv6 Dev Architecture in OpenStack



NetApp®

redhat.

IPv6 Dev Architecture in OpenStack



NetApp®

redhat.

Dev Environment Demo

NetApp®

redhat.

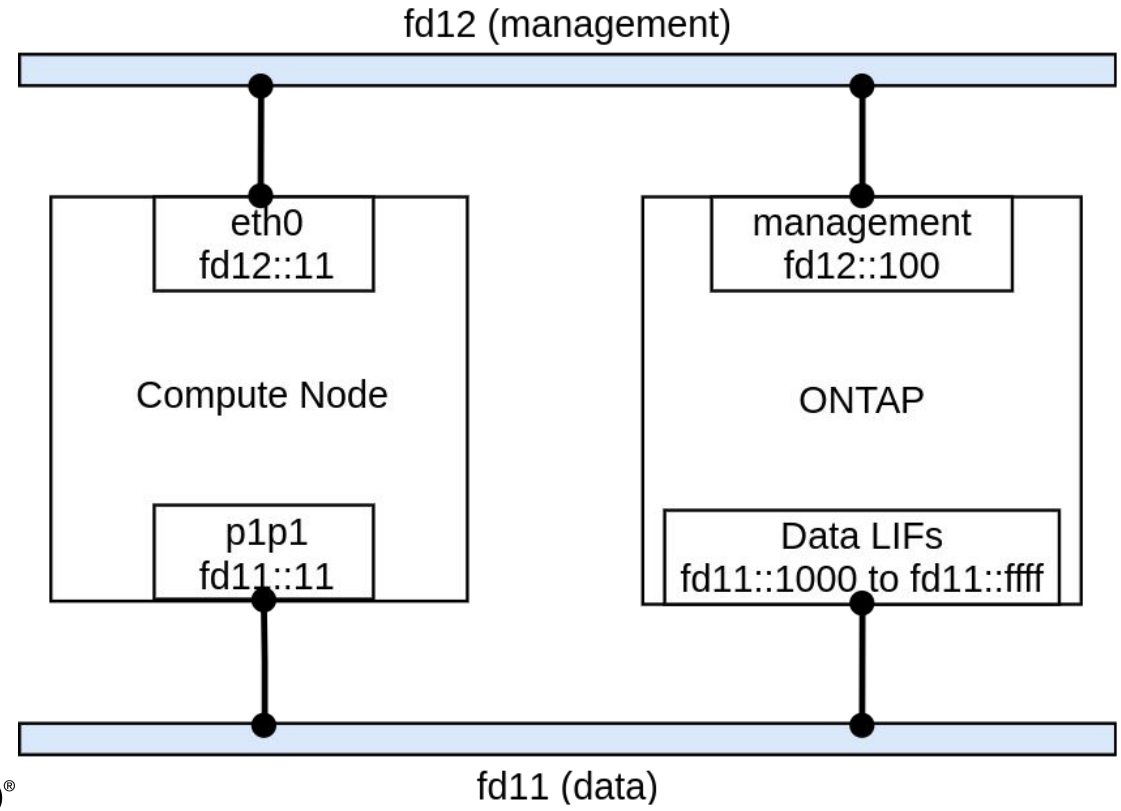
IPv6 Production Architecture in OpenStack

- In a production environment, this devstack architecture would need changes
  - Mostly network related
- On the dev environment, data packets are being routed on the hypervisor
  - In a real production environment, compute nodes would be connected to both networks
- OpenStack would be deployed directly on the baremetal nodes
- Most of the OpenStack configurations would remain unchanged

NetApp®

redhat.

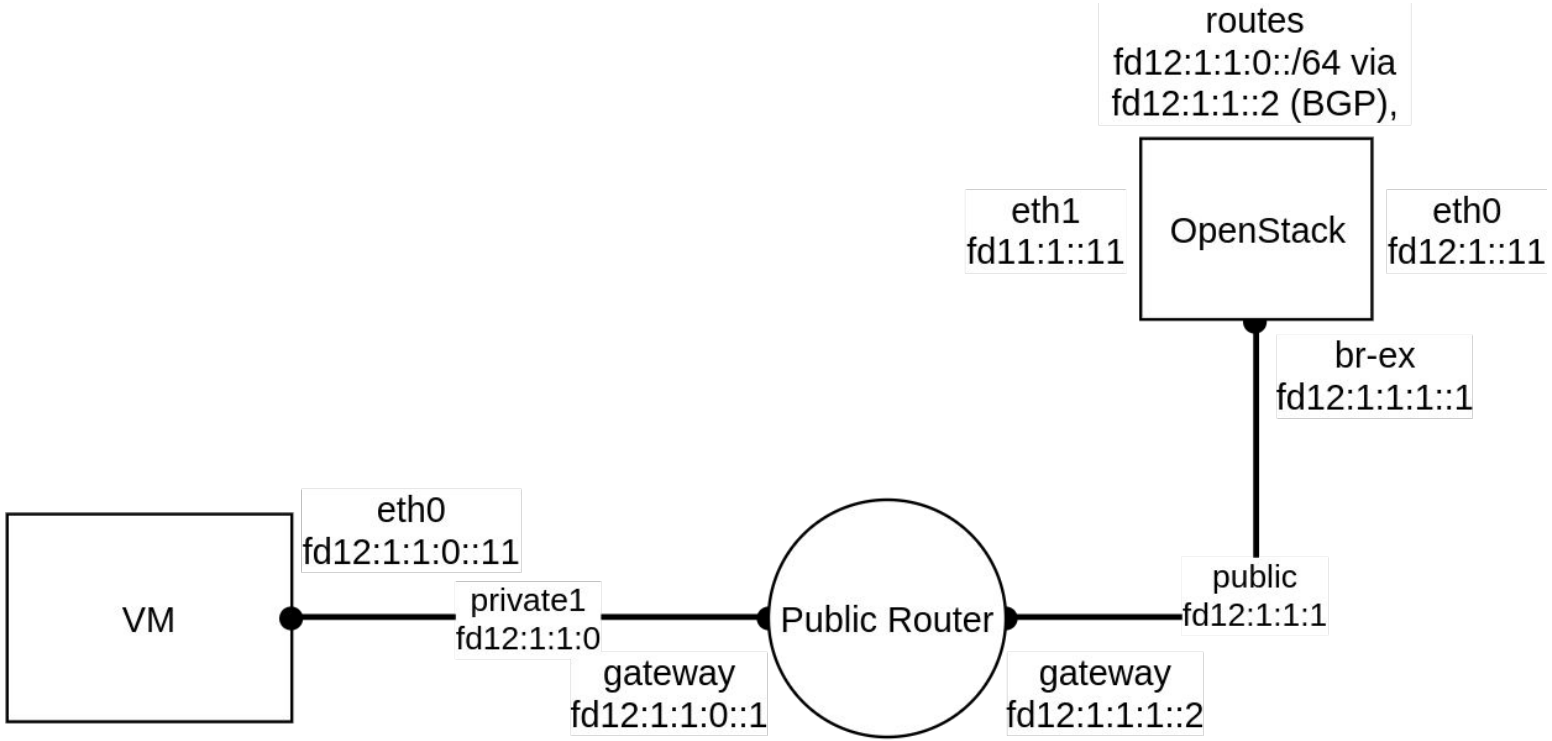
IPv6 Production Architecture in OpenStack



NetApp

redhat

IPv6 Production Architecture in OpenStack



NetApp

redhat

IPv6 and TripleO

- Configuring TripleO to use IPv6 is incredibly simple
  - Select the TripleO Heat Templates that deploy with IPv6 rather than with IPv4
- Configures the API endpoints and services to use IPv6 to communicate
  - Uses IPv6 connection and address pools rather than IPv4 pools

NetApp®

redhat.