IPv6 in Cloud Deployments

Shannon McFarland – CCIE #5245
Distinguished Engineer – Cisco
@eyepv6

2017 North American IPv6 Summit
April 25-26, 2017
LinkedIn Headquarters
Sunnyvale, CA

Agenda

- General Cloud + IPv6 Stuff
- State of the Union – IPv6 in Public and Private Clouds
- A look at OpenStack IPv6 Support
- Conclusion

Top Customer Concerns

- Execs want out of the on-prem DC/private Cloud business >> IT has to maintain an on-prem/private Cloud deployment to match the realities of the business:
  - Public Cloud providers often mandate NAT
  - IPv6 support is oriented around the provider load balancer
  - Development/Orchestration/Automation tools are not up to par for IPv6
  - Monitoring/logging/analytics are not on par with IPv4 support
  - Security is always an afterthought
  - IPv6 network performance is still lacking
- Good news: IPv6 support across the various Data Center components is pretty good and the public Cloud support is getting better

General Cloud + IPv6 Stuff

It’s The End Of The World As We Know It

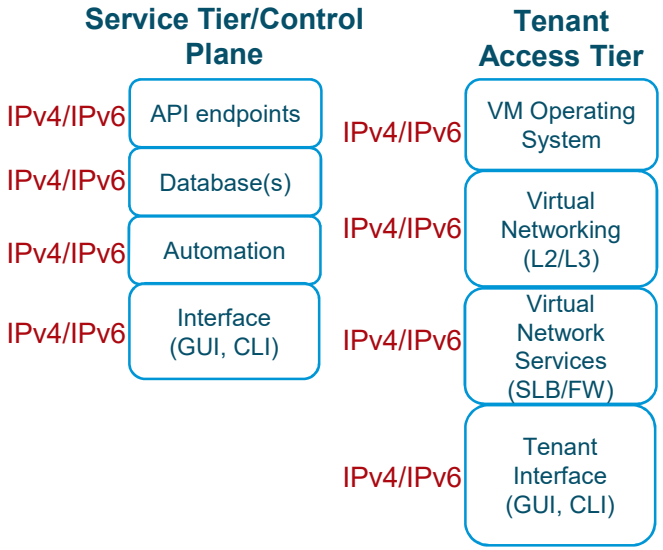
- IANA and RIRs are out or almost out of IPv4 addresses:
  - https://www.arin.net/knowledge/ipv6_info_center.html
  - https://www.ripe.net/publications/ipv6-info-centre
  - https://www.apnic.net/community/ipv6-program
  - http://afrinic.net/services/ipv6-programme
  - http://portalipv6.lacnic.net/en/
- It’s easy to get IPv6 addressing and the general deployment of IPv6 on your infrastructure is much easier to do than it used to be - no excuses not to do it

The Hard Stuff – IPv6 + Cloud

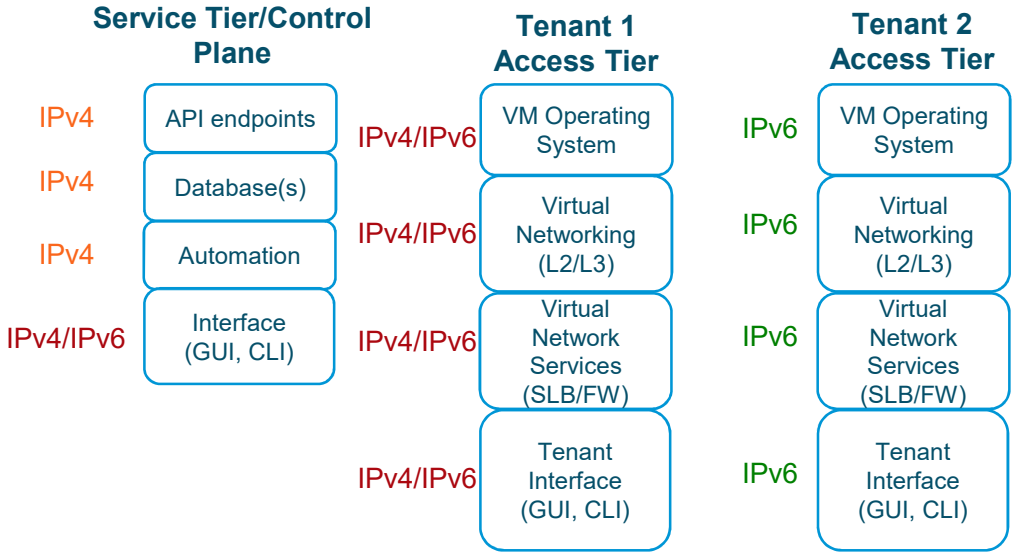
- Inside of a Cloud stack you have a lot of moving parts and they all ride on IP:
  - API endpoints
  - Provisioning, Orchestration and Management services
  - Boatload of protocols and databases and high-availability components
  - Virtual networking services <> Physical networking
- It has been a bumpy road to getting a solid IPv6 implementation in any Cloud stack
- Focus has been on tenant-facing support vs the actual Cloud control plane
- Two common approaches for IPv6 support:
  - **Dual-Stack everything** (Service Tier + Tenant Access Tier [Tenant management interface along with VM network access])
  - **Conditional Dual-Stack** (Tenant Access Tier only – API endpoints & DBs are still IPv4)

Cloud Stack – IP Version Options

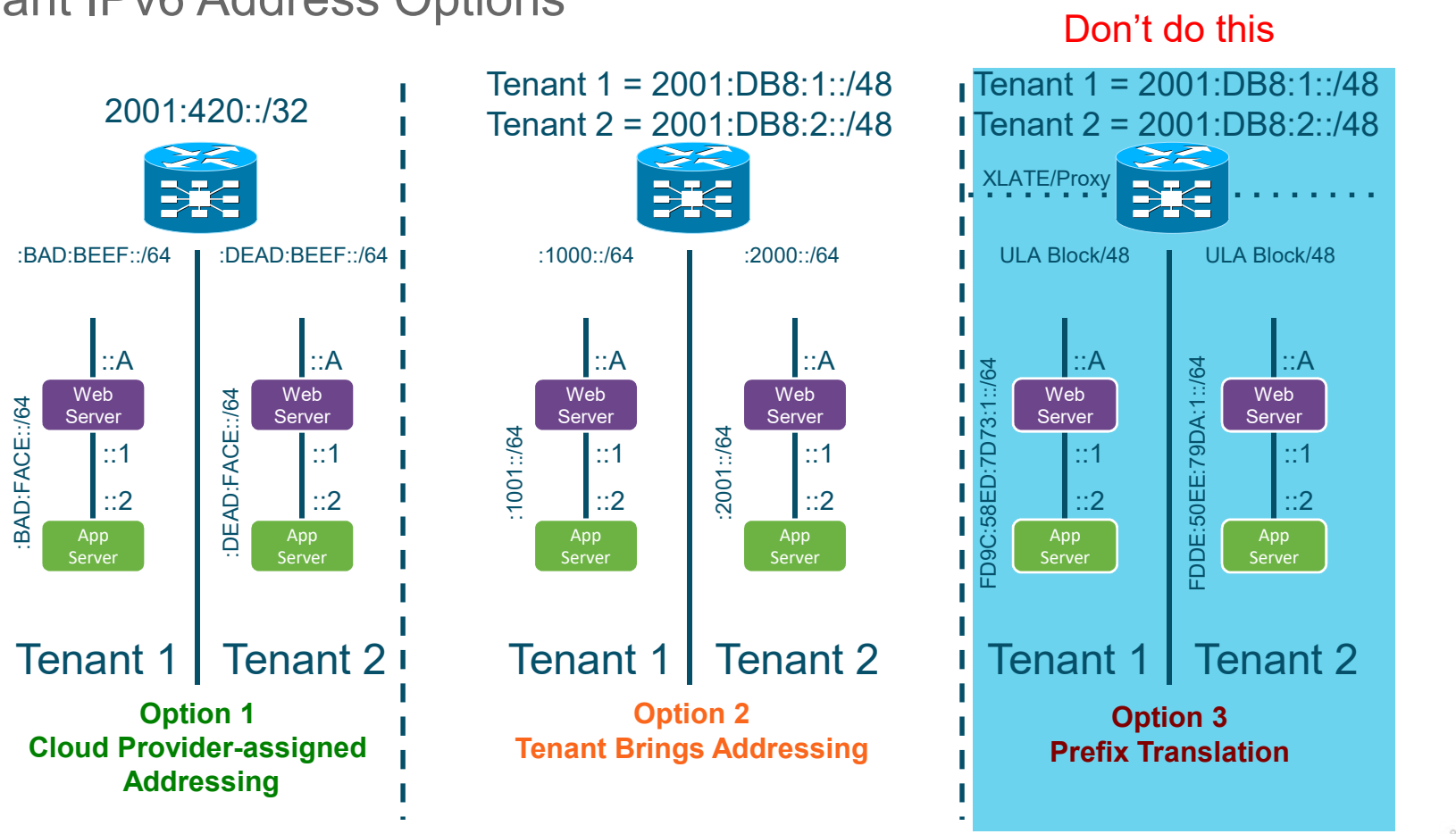
Dual-Stack Everything



Conditional Dual-Stack



Tenant IPv6 Address Options



State of the Union – IPv6 in Public and Private Clouds

Amazon AWS

- Getting started with IPv6 for VPC: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/get-started-ipv6.html
- Migrating to IPv6: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-migrate-ipv6.html
- Region support: https://aws.amazon.com/blogs/aws/aws-ipv6-update-global-support-spanning-15-regions-multiple-aws-services/
- Other stuff (CloudFront, WAF, S3): https://aws.amazon.com/about-aws/whats-new/2016/10/ipv6-support-for-cloudfront-waf-and-s3-transfer-acceleration/

Google Cloud Platform

- “Compute Engine currently does not support IPv6”: https://cloud.google.com/compute/docs/networks-and-firewalls
- “GCP networks do not support IPv6 at all”: https://cloud.google.com/compute/docs/networking
- Workaround/Hack/Blasphemy ☺: https://cloud.google.com/compute/docs/load-balancing/ipv6
- Google being Google, they will close the gaps quickly

Microsoft Azure

- Azure IPv6 Regions: https://azure.microsoft.com/en-us/updates/ipv6-for-azure-vms/
- Azure IPv6-enabled Load-Balancer: https://docs.microsoft.com/en-us/azure/load-balancer/load-balancer-ipv6-overview
- Long list of limitations (see LB link for full list - have a drink handy):
  - MSFT uses IPv6 NAT between Internet clients and IPv6 VMs – wha?!?!
    - “The load balancer routes the IPv6 packets to the private IPv6 addresses of the VMs using network address translation (NAT). The IPv6 Internet client cannot communicate directly with the IPv6 address of the VMs.”
    - Can’t assign public IPv6 addresses to VMs
  - Health probes from the LB is an IPv4 probe but it can take down the IPv4 and IPv6 endpoints down
  - Can’t upgrade existing VM to IPv6 – must deploy new VMs
  - VMs communicate with each other over IPv4, not IPv6 - VM-to-VM over IPv6 must traverse the LB

Open Source, Private Cloud and Container Stuff

- OpenStack: https://docs.openstack.org/ocata/networking-guide/config-ipv6.html
- Container-Focused Stuff:
  - Container Orchestration Engine (COE)-specific stuff like API, Key-Value, control plane stuff
  - Container Network Interface (CNI) – Data plane networking plugins like Contiv, Calico, Weave, Flannel
- Docker: https://docs.docker.com/engine/userguide/networking/default_network/ipv6/
- Kubernetes: It's complicated – Cisco (and other contributors) are feverishly working on IPv6 support:
  - Add support for IPv6 to bridge plugin: https://github.com/containernetworking/cni/pull/416
  - Generate MAC address for IPv6-only pods: https://github.com/containernetworking/cni/pull/394
  - Enable IPv6 in the host-local plugin: https://github.com/containernetworking/cni/pull/279
  - Adding IPv6 to cidr_set/cider_set_test: https://github.com/kubernetes/kubernetes/pull/43586
  - API: IP & CIDR fields (waiting on lower-layer IPv6 patches)
  - iptables kubelet & proxy - TBD
- Apache Mesosphere: VERY limited information – You can get an IPv6 address from the IPAM server but all of the other components such as Network Isolator Module/virtualizer, Mesos-DNS, etc.. are lacking: http://mesos.apache.org/documentation/latest/networking-for-mesos-managed-containers/

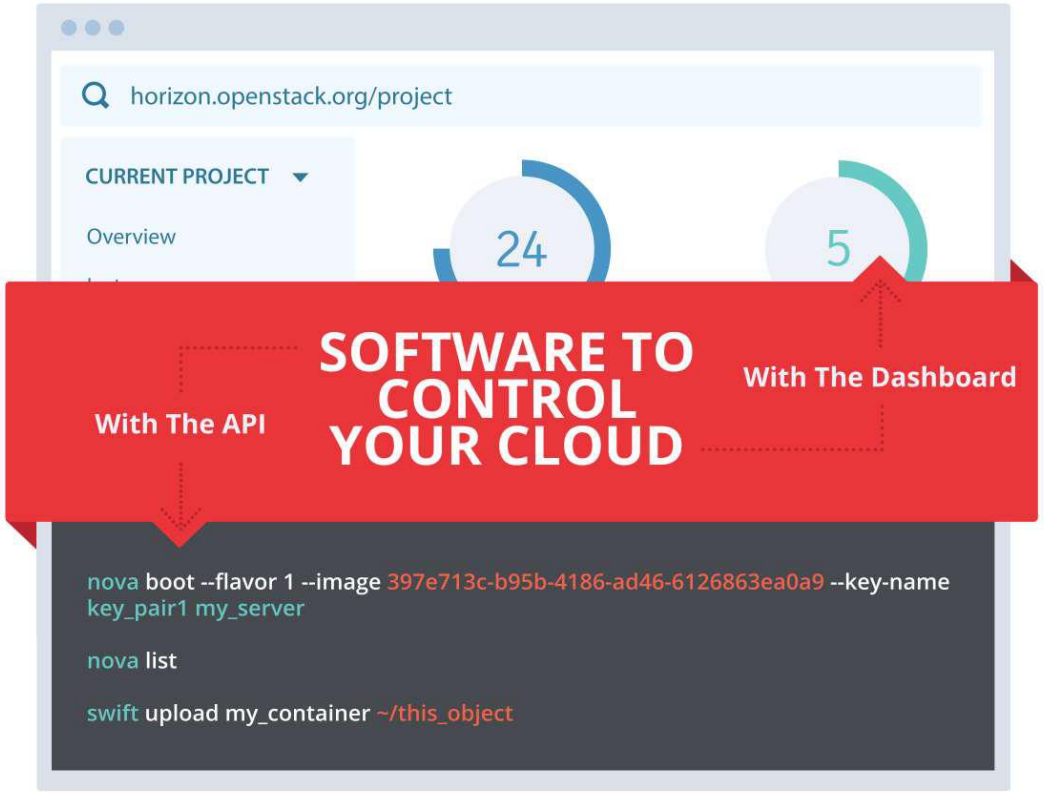
Reference Material

- OpenStack: https://docs.openstack.org/ocata/networking-guide/config-ipv6.html
  - Tenant IPv6: http://www.debug-all.com/?m=201505
  - Using Heat for IPv6 deployment: http://www.debug-all.com/?m=201506
  - IPv6 Prefix Delegation: http://www.debug-all.com/?m=201611
  - Address Scopes: http://www.debug-all.com/?m=201612
- Docker: https://docs.docker.com/engine/userguide/networking/default_network/ipv6/
  - Docker Engine with IPv6: http://www.debug-all.com/?m=201509
  - Docker Registry with IPv6: http://www.debug-all.com/?m=201511
  - Docker Swarm with IPv6 (non-swarm-mode): http://www.debug-all.com/?m=201604
  - Docker Host Mode with IPv6: http://www.debug-all.com/?m=201606

A Look at OpenStack + IPv6

OpenStack Cloud Computing Software

- Freely available, open source software allowing anyone to build their own private or public clouds
- Open source and open APIs allows the customer to avoid being locked ir to a single vendor
- Built by a growing community of contributors
- Opportunities for vendors to develop their own solutions and services



OpenStack IPv6 Support Overview

- Address Assignment:
  - SLAAC
  - DHCPv6 Stateless
  - DHCPv6 Stateful
  - IPv6 PD
- Control plane components (mostly)
- Orchestration via the Heat project
- Security Groups
- LBaaS

Neutron Addressing Schemes

Reference

ipv6_ra_mode	ipv6_address_mode	Result
SLAAC	N/S	Address using Neutron router
N/S	SLAAC	Address using external router
SLAAC	SLAAC	Address using Neutron router

Address Configuration Flags	Value
Auto	1
Managed	0
Other	0

ipv6_ra_mode	ipv6_address_mode	Result
DHCPv6-stateless	N/S	Address using Neutron router and optional information using external service
N/S	DHCPv6-stateless	Address using external router and optional information using Neutron DHCP implementation
DHCPv6-stateless	DHCPv6-stateless	Address and optional information using Neutron router and DHCP implementation respectively

Address Configuration Flags	Value
Auto	1
Managed	0
Other	1

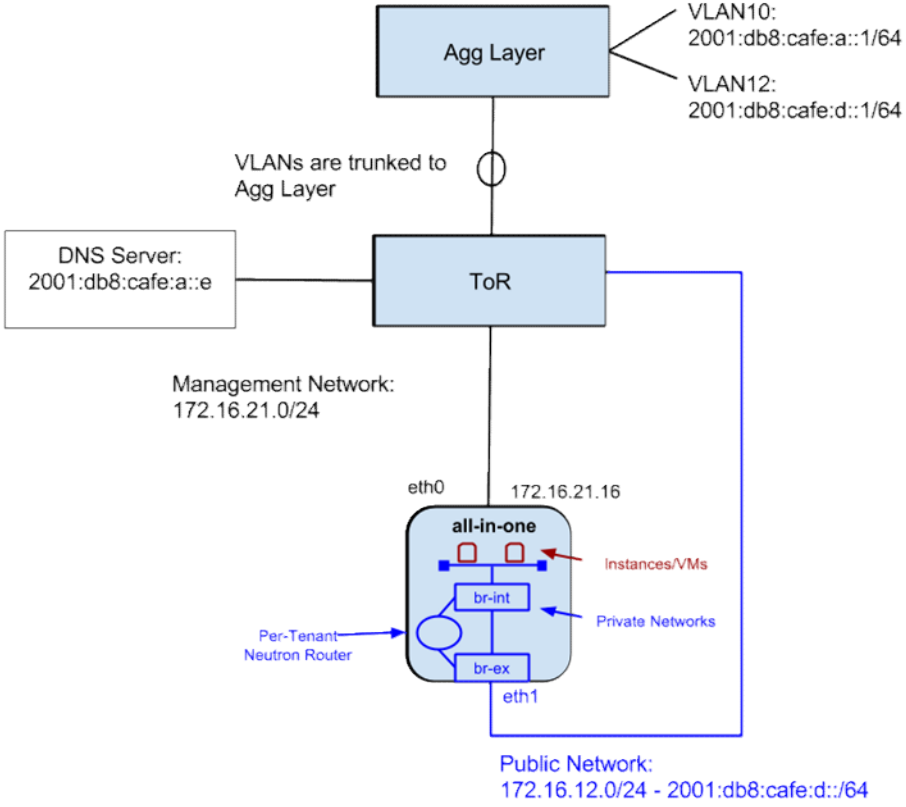
ipv6_ra_mode	ipv6_address_mode	Result
DHCPv6-stateful	N/S	Address and optional information using external service
N/S	DHCPv6-stateful	Address and optional information using Neutron DHCP implementation
DHCPv6-stateful	DHCPv6-stateful	Address and optional information using Neutron DHCP implementation

Address Configuration Flags	Value
Auto	0
Managed	1
Other	1

http://docs.openstack.org/mitaka/networking-guide/config-ipv6.html

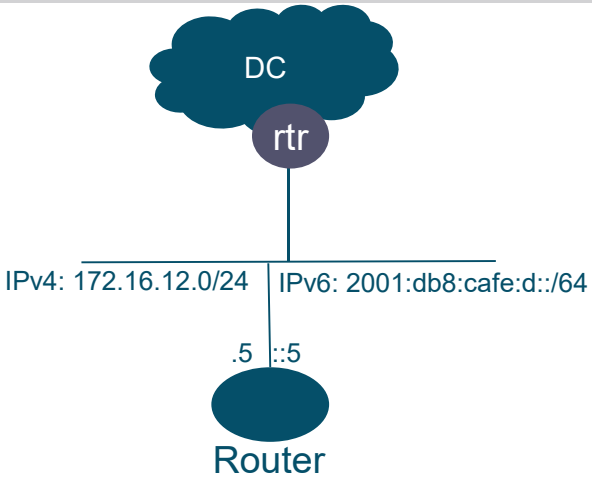
Reference

Tenant IPv6 - Neutron L3 Example



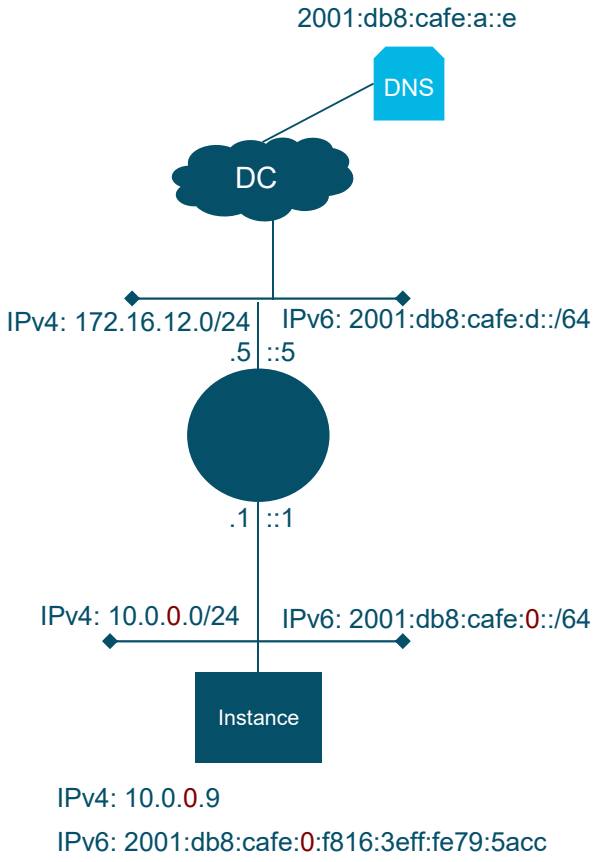
Create the Public Network/Subnet

```
neutron net-create public --router:external

neutron subnet-create --name public-subnet --allocation-pool start=172.16.12.5,
end=172.16.12.254 public 172.16.12.0/24

neutron subnet-create --ip-version=6 --name=public-v6-subnet --allocation-pool start=2001:db8:cafe:d::5,
end=2001:db8:cafe:d:ffff:ffff:ffff:fffe --disable-dhcp public 2001:db8:cafe:d::/64
```



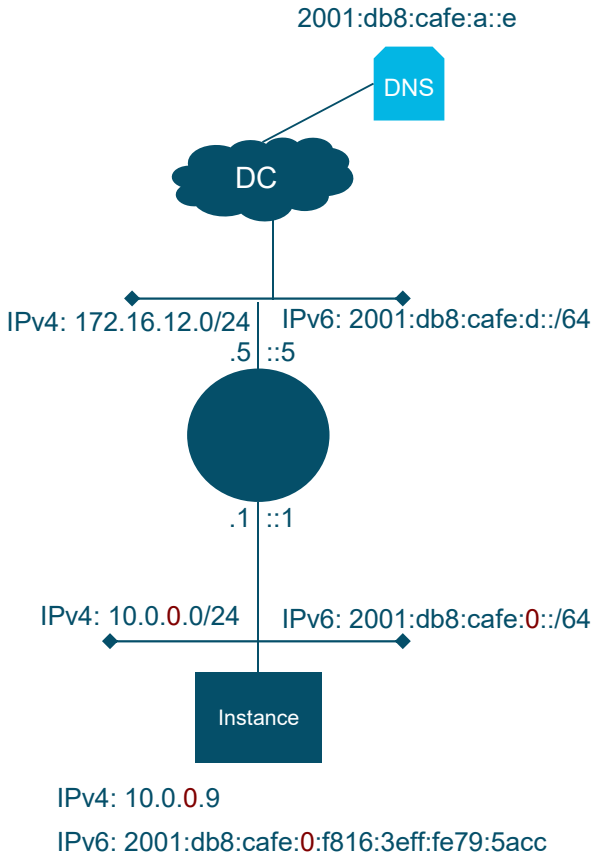
SLAAC Mode

```
neutron net-create private

neutron subnet-create --ip-version=6 --name=private_v6_subnet --ipv6-address-mode=slaac
--ipv6-ra-mode=slaac private 2001:db8:cafe::/64
+-------------------+-------------------------------------------------------------------------------+
| Field             | Value                                                                         |
+-------------------+-------------------------------------------------------------------------------+
| allocation_pools  | {"start": "2001:db8:cafe::2", "end": "2001:db8:cafe:0:ffff:ffff:ffff:fffe"} |
| cidr              | 2001:db8:cafe::/64                                                            |
| dns_nameservers   |                                                                               |
| enable_dhcp       | True                                                                          |
| gateway_ip        | 2001:db8:cafe::1                                                              |
| host_routes       |                                                                               |
| id                | 42cc3dbc-938b-4ad6-b12e-59aef7618477                                          |
| ip_version        | 6                                                                             |
| ipv6_address_mode | slaac                                                                         |
| ipv6_ra_mode      | slaac                                                                         |
| name              | private_v6_subnet                                                             |
| network_id        | 7166ce15-c581-4195-9479-ad2283193d06                                          |
| subnetpool_id     |                                                                               |
| tenant_id         | f057804eb39b4618b40e06196e16265b                                              |
+-------------------+-------------------------------------------------------------------------------+
```



Router Example

```
neutron router-create private-router

neutron router-gateway-set private-router public

neutron router-interface-add private-router private-v4-subnet

neutron router-interface-add private-router private-v6-subnet
```



SLAAC Mode Info

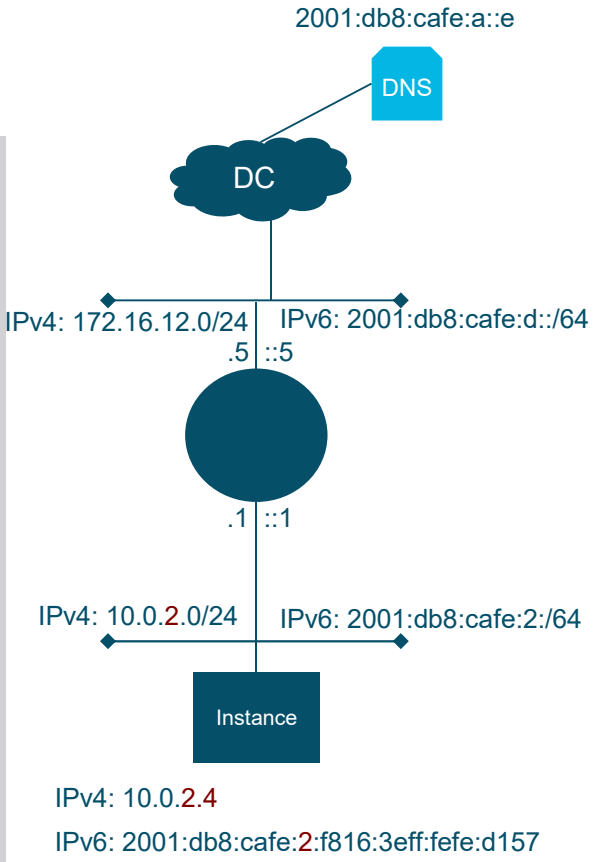
- OpenStack will not inject the IPv6 DNS entry from the subnet dns_nameservers entry
- Options
  - Manually setting the IPv6 DNS server entry in the resolv.conf file allows for correct IPv6-based name resolution
  - Bake DNS settings into your image
  - Cloud-init to inject the DNS configuration
- You do get A and AAAA records back over IPv4 transport
- Basically, it works as it should

SLAAC Mode – Sniffer Capture

```
15:08:01.520353 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 16) fe80::f816:3eff:fe79:5acc > ff02::2: [icmp6 sum ok] ICMP6, router solicitation, length 16
         source link-address option (1), length 8 (1): fa:16:3e:79:5a:cc
           0x0000:  fa16 3e79 5acc

15:08:01.520667 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 56) fe80::f816:3eff:fec3:17b4 > ff02::1: [icmp6 sum ok] ICMP6, router advertisement, length 56
       hop limit 64, Flags [none], pref medium, router lifetime 30s, reachable time 0s, retrans time 0s
         prefix info option (3), length 32 (4): 2001:db8:cafe::/64, Flags [onlink, auto], valid time 86400s, pref. time 14400s
           0x0000:  40c0 0001 5180 0000 3840 0000 0000 2001
           0x0010:  0db8 cafe 0000 0000 0000 0000 0000
         source link-address option (1), length 8 (1): fa:16:3e:c3:17:b4
           0x0000:  fa16 3ec3 17b4
15:08:02.256004 IP6 (hlim 1, next-header Options (0) payload length: 36) fe80::f816:3eff:fe79:5acc > ff02::16: HBH (rtalert: 0x0000) (padn) [icmp6 sum ok] ICMP6, multicast listener report v2, 1 group record(s) [gaddr ff02::1:ff79:5acc is_ex { }]
15:08:02.484047 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 24) :: > ff02::1:ff79:5acc: [icmp6 sum ok] ICMP6, neighbor solicitation, length 24, who has 2001:db8:cafe:0:f816:3eff:fe79:5acc
```

Stateless DHCPv6 Mode

```
neutron net-create private-dhcpv6-stateless

neutron subnet-create --ip-version=6 --name=private_dhcpv6_stateless_subnet
--ipv6-address-mode=dhcpv6-stateless --ipv6-ra-mode=dhcpv6-stateless private-dhcpv6-stateless
2001:db8:cafe:2::/64 --dns-nameserver 2001:db8:cafe:a::e
+-------------------+-------------------------------------------------------------------------------------+
| Field             | Value                                                                               |
+-------------------+-------------------------------------------------------------------------------------+
| allocation_pools  | {"start": "2001:db8:cafe:2::2", "end": "2001:db8:cafe:2:ffff:ffff:ffff:fffe"} |
| cidr              | 2001:db8:cafe:2::/64                                                                |
| dns_nameservers   | 2001:db8:cafe:a::e                                                                  |
| enable_dhcp       | True                                                                                |
| gateway_ip        | 2001:db8:cafe:2::1                                                                  |
| host_routes       |                                                                                     |
| id                | e63e72d5-493a-4a49-8f7d-8846c2bc7a8f                                                |
| ip_version        | 6                                                                                   |
| ipv6_address_mode | dhcpv6-stateless                                                                    |
| ipv6_ra_mode      | dhcpv6-stateless                                                                    |
| name              | private_dhcpv6_stateless_subnet                                                     |
| network_id        | 27618d5e-318c-46a4-b6a2-a155beed9643                                                |
| subnetpool_id     |                                                                                     |
| tenant_id         | f057804eb39b4618b40e06196e16265b                                                    |
+-------------------+-------------------------------------------------------------------------------------+
```



Reference

DHCPv6 Stateless Mode Info

- Enable client for DHCPv6 Stateless:

Ubuntu

```
/etc/network/interfaces
auto eth0
iface eth0 inet dhcp
iface eth0 inet6 auto
dhcp 1
```

CentOS/RHEL/Fedora

```
/etc/sysconfig/network-scripts/ifcfg-xxxx
IPV6INIT="yes"
DHCPV6C="yes”
DHCPV6C_OPTIONS="-S"
```

- Then you get addressing and options:

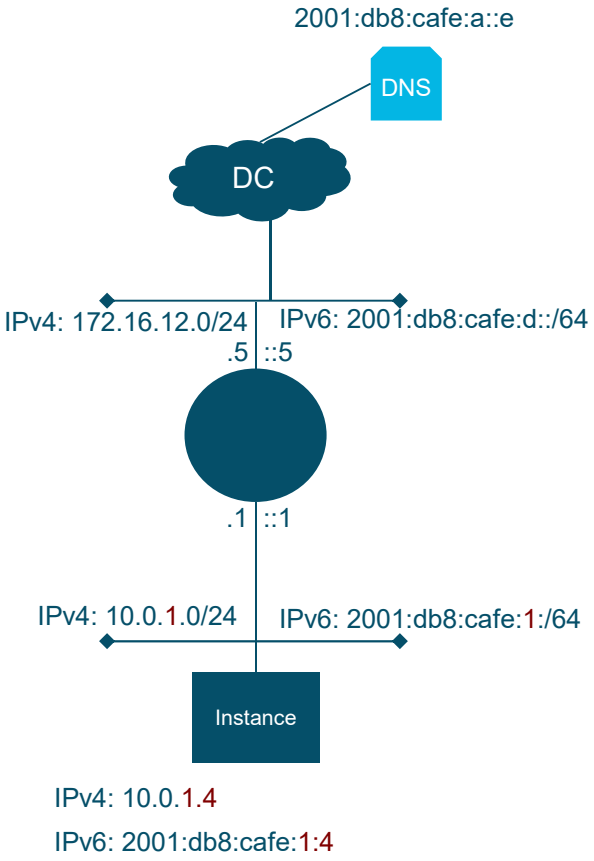
```
ubuntu@dhcpv6-stateless-4:~$ more /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 172.16.10.14
nameserver 2001:db8:cafe:a::e
search openstacklocal
```

DHCPv6 Stateless Mode – Sniffer Capture

```
15:43:23.911172 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 56) fe80::f816:3eff:fec1:bc52 > ff02::1: [icmp6 sum ok] ICMP6, router advertisement, length 56
       hop limit 64, Flags [other stateful], pref medium, router lifetime 30s, reachable time 0s, retrans time 0s
         prefix info option (3), length 32 (4): 2001:db8:cafe:2::/64, Flags [onlink, auto], valid time 86400s, pref. time 14400s
           0x0000:  40c0 0001 5180 0000 3840 0000 0000 2001
           0x0010:  0db8 cafe 0002 0000 0000 0000 0000
         source link-address option (1), length 8 (1): fa:16:3e:c1:bc:52
           0x0000:  fa16 3ec1 bc52
15:43:25.353331 IP6 (hlim 1, next-header UDP (17) payload length: 44) fe80::f816:3eff:fefe:d157.546 > ff02::1:2.547: [udp sum ok] dhcp6 inf-req (xid=d2dbc8 (client-ID hwaddr type 1 fa163efed157) (option-request DNS-server DNS-search-list Client-FQDN SNTP-servers) (elapsed-time 94))
15:43:25.353578 IP6 (class 0xc0, hlim 64, next-header UDP (17) payload length: 88) fe80::f816:3eff:fe2d:a6de.547 > fe80::f816:3eff:fefe:d157.546: [udp sum ok] dhcp6 reply (xid=d2dbc8 (client-ID hwaddr type 1 fa163efed157) (server-ID hwaddr type 1 fa163e2da6de) (DNS-search-list openstacklocal.) (DNS-server 2001:db8:cafe:a::e) (lifetime 86400))
```

Stateful DHCPv6 Mode

```
neutron net-create private-dhcpv6

neutron subnet-create --ip-version=6 --name=private_dhcpv6_subnet --ipv6-address-mode=dhcpv6-stateful
--ipv6-ra-mode=dhcpv6-stateful private-dhcpv6 2001:db8:cafe:1::/64 --dns-nameserver 2001:db8:cafe:a::e
+-------------------+-------------------------------------------------------------------------------+
| Field             | Value                                                                         |
+-------------------+-------------------------------------------------------------------------------+
| allocation_pools  | {"start": "2001:db8:cafe:1::2", "end": "2001:db8:cafe:1:ffff:ffff:ffff:fffe"} |
| cidr              | 2001:db8:cafe:1::/64                                                          |
| dns_nameservers   | 2001:db8:cafe:a::e                                                            |
| enable_dhcp       | True                                                                          |
| gateway_ip        | 2001:db8:cafe:1::1                                                            |
| host_routes       |                                                                               |
| id                | 545ea206-9d14-4dca-8bae-7940719bdab5                                          |
| ip_version        | 6                                                                             |
| ipv6_address_mode | dhcpv6-stateful                                                               |
| ipv6_ra_mode      | dhcpv6-stateful                                                               |
| name              | private_dhcpv6_subnet                                                         |
| network_id        | 55ed8333-2876-400a-92c1-ef49bc10aa2b                                          |
| subnetpool_id     |                                                                               |
| tenant_id         | f057804eb39b4618b40e06196e16265b                                              |
+-------------------+-------------------------------------------------------------------------------+
```



Reference

DHCPv6 Stateful Mode Info

- Enable client for DHCPv6:

Ubuntu

```
/etc/network/interfaces
auto eth0
iface eth0 inet dhcp
iface eth0 inet6 dhcp
```

CentOS/RHEL/Fedora

```
/etc/sysconfig/network-scripts/ifcfg-xxxx
IPV6INIT="yes"
DHCPV6C="yes"
```

- Then you get addressing and options:

```
ubuntu@dhcpv6-1:~$ more /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 172.16.10.14
nameserver 2001:db8:cafe:a::e
search openstacklocal
```

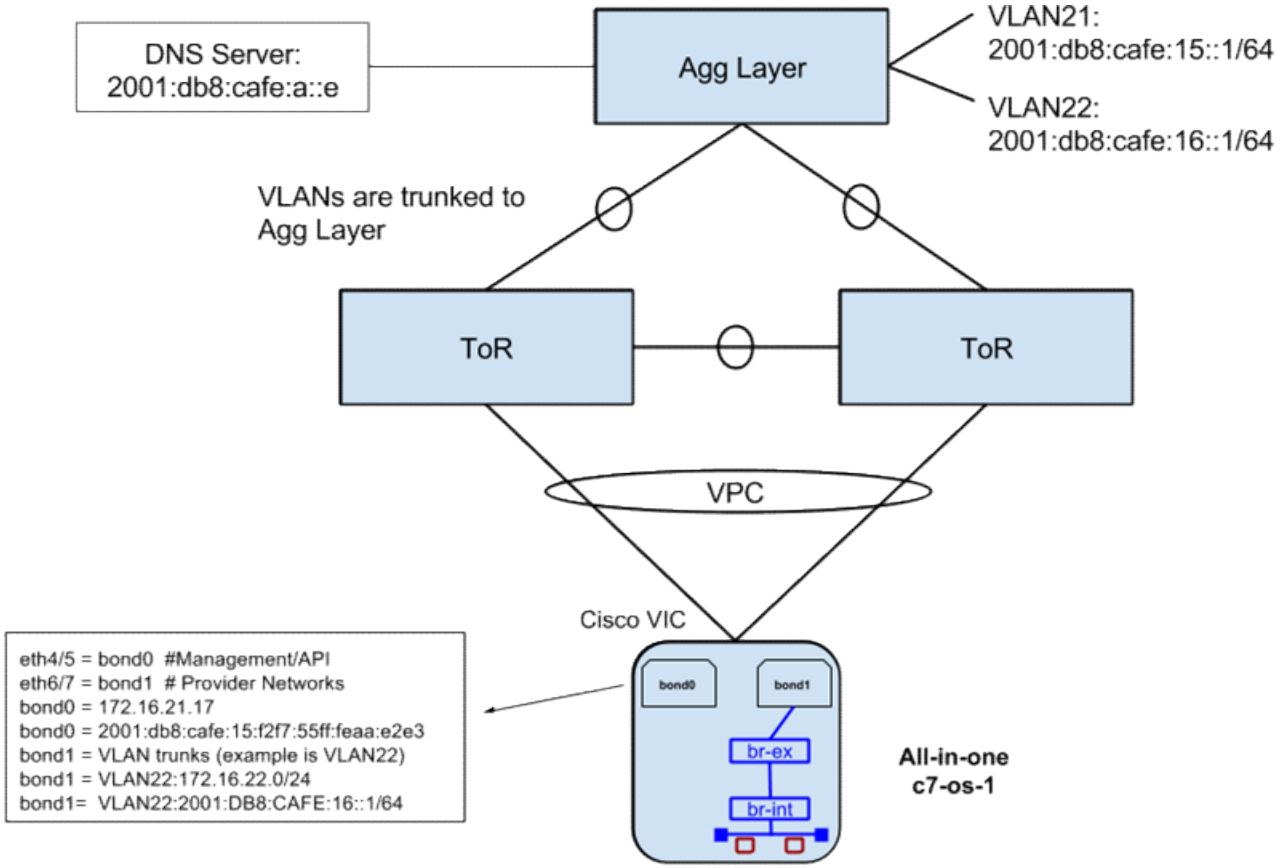
DHCPv6 Stateful Mode – Sniffer Capture

```
14:56:02.671930 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 24) fe80::f816:3eff:fe77:e5a0 > ff02::1: [icmp6 sum ok] ICMP6, router
advertisement, length 24
       hop limit 64, Flags [managed], pref medium, router lifetime 30s, reachable time 0s, retrans time 0s
         source link-address option (1), length 8 (1): fa:16:3e:77:e5:a0
           0x0000:  fa16 3e77 e5a0


14:56:08.042878 IP6 (hlim 1, next-header UDP (17) payload length: 64) fe80::f816:3eff:fe22:386b.546 > ff02::1:2.547: [udp sum ok] dhcp6 solicit
(xid=85680b (client-ID hwaddr/time type 1 time 482446373 fa163e22386b) (option-request DNS-server DNS-search-list Client-FQDN SNTP-servers) (elapsed-
time 101) (IA_NA IAID:1042430059 T1:3600 T2:5400))


14:56:08.143267 IP6 (class 0xc0, hlim 64, next-header UDP (17) payload length: 175) fe80::f816:3eff:fe06:176f.547 > fe80::f816:3eff:fe22:386b.546: [udp
sum ok] dhcp6 advertise (xid=85680b (client-ID hwaddr/time type 1 time 482446373 fa163e22386b) (server-ID hwaddr type 1 fa163e06176f) (IA_NA
IAID:1042430059 T1:43200 T2:75600 (IA_ADDR 2001:db8:cafe:1::4 pltime:86400 vltime:86400)) (status-code success) (preference 255) (DNS-search-list
openstacklocal.) (DNS-server 2001:db8:cafe:a::e) (Client-FQDN))


14:56:08.143719 IP6 (hlim 1, next-header UDP (17) payload length: 106) fe80::f816:3eff:fe22:386b.546 > ff02::1:2.547: [udp sum ok] dhcp6 request
(xid=9cb172 (client-ID hwaddr/time type 1 time 482446373 fa163e22386b) (server-ID hwaddr type 1 fa163e06176f) (option-request DNS-server DNS-search-
list Client-FQDN SNTP-servers) (elapsed-time 0) (IA_NA IAID:1042430059 T1:3600 T2:5400 (IA_ADDR 2001:db8:cafe:1::4 pltime:7200 vltime:7500)))


14:56:08.143897 IP6 (class 0xc0, hlim 64, next-header UDP (17) payload length: 186) fe80::f816:3eff:fe06:176f.547 > fe80::f816:3eff:fe22:386b.546: [udp
sum ok] dhcp6 reply (xid=9cb172 (client-ID hwaddr/time type 1 time 482446373 fa163e22386b) (server-ID hwaddr type 1 fa163e06176f) (IA_NA
IAID:1042430059 T1:3600 T2:6300 (IA_ADDR 2001:db8:cafe:1::4 pltime:7200 vltime:7500)) (status-code success) (DNS-search-list openstacklocal.) (DNS-
server 2001:db8:cafe:a::e) (Client-FQDN))
```

Address Assignment: Provider Networks

Reference

Provider Networks

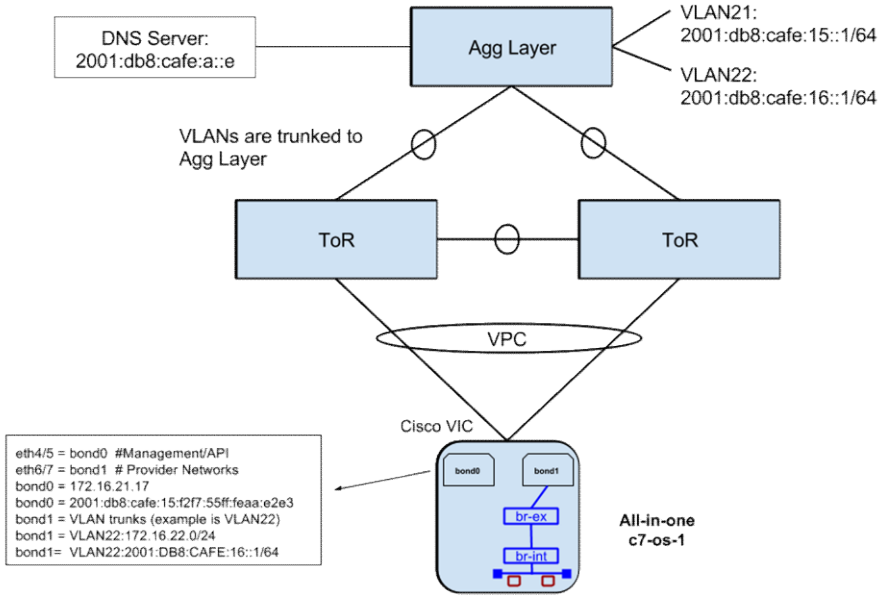


Provider Network - Examples

```
neutron net-create --router:external --provider:physical_network provider --provider:network_type vlan --
provider:segmentation_id=22 --shared external-net


neutron subnet-create external-net 172.16.22.0/24 --name external-subnet --gateway 172.16.22.1 --allocation-pool
start=172.16.22.5,end=172.16.22.254


# SLAAC
neutron subnet-create external-net --ip-version=6 --ipv6-address-mode=slaac --ipv6-ra-mode=slaac --name=external-subnet-v6
--allocation-pool start=2001:db8:cafe:16::5,end=2001:db8:cafe:16:ffff:ffff:ffff:fffe 2001:db8:cafe:16::/64


# Stateless DHCPv6
neutron subnet-create external-net --ip-version=6 --ipv6-address-mode=dhcpv6-stateless --ipv6-ra-mode=dhcpv6-stateless --
name=external-subnet-v6 --allocation-pool start=2001:db8:cafe:16::5,end=2001:db8:cafe:16:ffff:ffff:ffff:fffe
2001:db8:cafe:16::/64 --dns-nameserver 2001:db8:cafe:a::e


# Stateful DHCPv6
neutron subnet-create external-net --ip-version=6 --ipv6-address-mode=dhcpv6-stateful --ipv6-ra-mode=dhcpv6-stateful --
name=external-subnet-v6 --allocation-pool start=2001:db8:cafe:16::5,end=2001:db8:cafe:16:ffff:ffff:ffff:fffe
2001:db8:cafe:16::/64 --dns-nameserver 2001:db8:cafe:a::e
```

```
# SLAAC
interface Vlan22
 description Provider Network trunked for C7-os-1
 ip address 172.16.22.2 255.255.255.0
 ipv6 address 2001:DB8:CAFE:16::1/64
 standby version 2
 standby 2 ipv6 autoconfig
 standby 2 timers msec 250 msec 750
 standby 2 priority 110
 standby 2 preempt
 standby 2 authentication OPEN

# Stateless DHCPv6
interface Vlan22
 description Provider Network trunked for C7-os-1
 ip address 172.16.22.2 255.255.255.0
 ipv6 address 2001:DB8:CAFE:16::1/64
 ipv6 nd other-config-flag
 standby version 2
 standby 2 ipv6 autoconfig
 standby 2 timers msec 250 msec 750
 standby 2 priority 110
 standby 2 preempt
 standby 2 authentication OPEN

# Stateful DHCPv6
interface Vlan22
 description Provider Network trunked for C7-os-1
 ip address 172.16.22.2 255.255.255.0
 ipv6 address 2001:DB8:CAFE:16::1/64
 ipv6 nd managed-config-flag
 standby version 2
 standby 2 ipv6 autoconfig
 standby 2 timers msec 250 msec 750
 standby 2 priority 110
 standby 2 preempt
 standby 2 authentication OPEN
```

Provider Router Example



IPv6 Only

IPv6-Only Instances

- IPv6-only works out-of-the-box for IP connectivity but you are hosed on metadata
- It is a real drag to have to deploy an IPv4 subnet along with IPv6 just to get basic functionality working such as FQDN, SSH keys and other metadata
- The metadata service ONLY supports IPv4
- A wish list bug to work on this issue expired: https://bugs.launchpad.net/neutron/+bug/1460177 ☹
- Workarounds:
  - Build all/most of what you want in the image itself
  - Use config-drive

Basic IPv6-only Config-Drive Example

```
[root@c7-os-1 latest]# cat user_data
#cloud-config
fqdn: v6onlyinstance.example.com
users:
  - name: cloud-user
    ssh-authorized-keys:
      - ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC4W4RPlOBiY14iJwW9kd3Chys5bUBjy2VKJkFa5az8JHcVvOh3LO5BHdhc6WryT+blmx9LKGyVSc0rfzSEAfQ91dXJCHuhl5BNk9pLibs3oe8s/1r/v
jtxQopKIIGN3PYuisvpZVLeP1kRhddIdLvuZcQm82L4VPUAOzLqbFdhsu/Y2lU5WyiTiI5VNJwwbzzc67BFHz2ov2bdBgCfFWyUQMikiyIrAv5hVcqADv7XAqY4P5sJaOaHAcNcCfMtY8RbtE
MSIyw8fey1erY4ZiknTAn/eU52mc1819xR4CwI9wYqYdpVyiNULRWH9opK30dqhhthgElzCax+WqmxMXGP root@c7-os-1.example.com


[root@c7-os-1 ~]# nova boot --flavor m1.small --image rh7-stateless --key-name new-aio-key --security-groups default --nic net-name=external-net
rhv6-only-drive --config-drive true --user-data user_data.yaml

[root@c7-os-1 ~]# nova list
+--------------------------------------+-----------------+--------+------------+-------------+-----------------------------------------------------
| ID                                   | Name            | Status | Task State | Power State | Networks
+--------------------------------------+-----------------+--------+------------+-------------+-----------------------------------------------------
| 2244a346-a34b-4ab6-905f-71dc207a92e6 | rhv6-only-drive | ACTIVE | -          | Running     | external-net=2001:db8:cafe:16:f816:3eff:feec:3c59
+--------------------------------------+-----------------+--------+------------+-------------+-----------------------------------------------------
[root@c7-os-1 ~]# ssh cloud-user@2001:db8:cafe:16:f816:3eff:feec:3c59
. . .
[cloud-user@v6onlyinstance ~]$ ip a
. . .
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether fa:16:3e:ec:3c:59 brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:cafe:16:f816:3eff:feec:3c59/64 scope global dynamic
       valid_lft 2591952sec preferred_lft 604752sec
    inet6 fe80::f816:3eff:feec:3c59/64 scope link
       valid_lft forever preferred_lft forever
[cloud-user@v6onlyinstance ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
search openstacklocal. example.com
nameserver 2001:db8:cafe:a::e
[cloud-user@v6onlyinstance ~]$ cat /etc/hostname
v6onlyinstance.example.com
```

Summary

- IPv6 is here – get over it
- IPv6 with L3-L7 is not your biggest concern in Cloud deployments – Development/Orchestration/Automation tools/processes are your biggest concern
- It is not hard to serve Internet or internal-facing clients your application magic over IPv6 – The bigger issues are:
  - Control plane support
  - IPv6 only for EVERY Cloud component
  - Protocol agnostic vs. protocol aware
- Get started now and don't be paralyzed by the perceived workload – Start small, start simple and do one app in one part of the Cloud – Rinse>repeat

Reference Slides

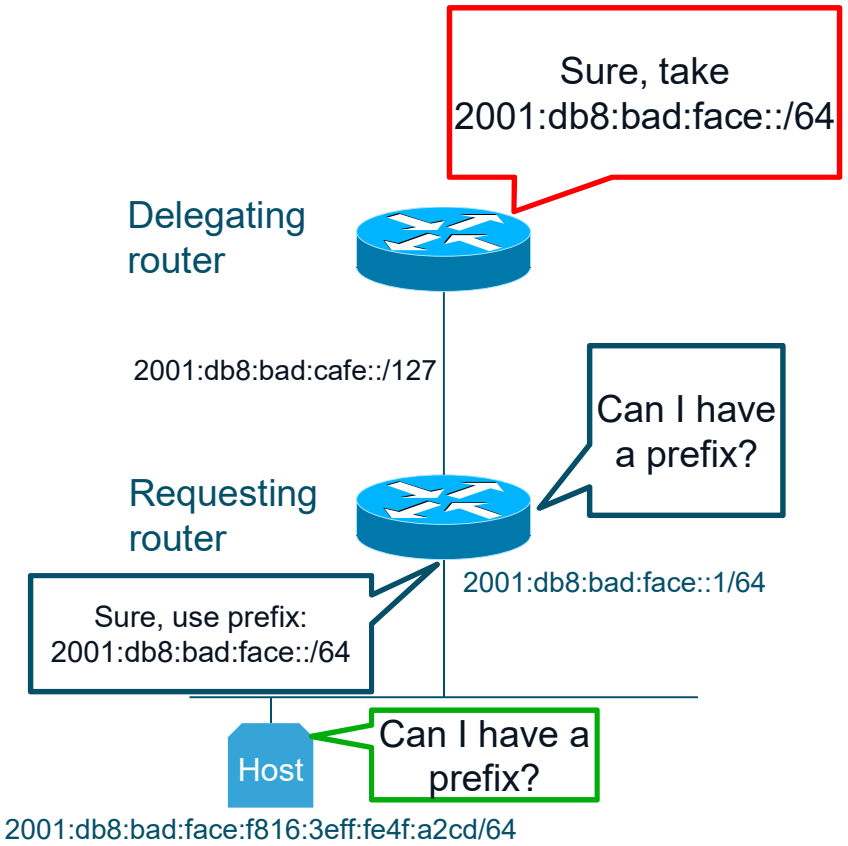
IPv6 Prefix Delegation

IPv6 PD in OpenStack

- Super important feature in OpenStack:
  - No IPv6 NAT support in OpenStack (Yee Haw!)
  - Can assign prefixes to tenants via:
    - **Manual assignment** (“Sally, here is your prefix”) - HORRIBLE method due to human errors in tracking and assignment (fat finger syndrome)
    - **IPv6 Subnet Pool Support** - Pool of addresses that tenants can pull from - http://docs.openstack.org/newton/networking-guide/config-subnet-pools.html
      - No overlap between two subnets
      - Stable address management across projects
    - **IPv6 Prefix Delegation** - Allows for upstream router/server assignment of prefixes - http://docs.openstack.org/newton/networking-guide/config-ipv6.html
    - Leverages default IPv6 subnet pool support
- Does not support HA of the PD agent (future work)

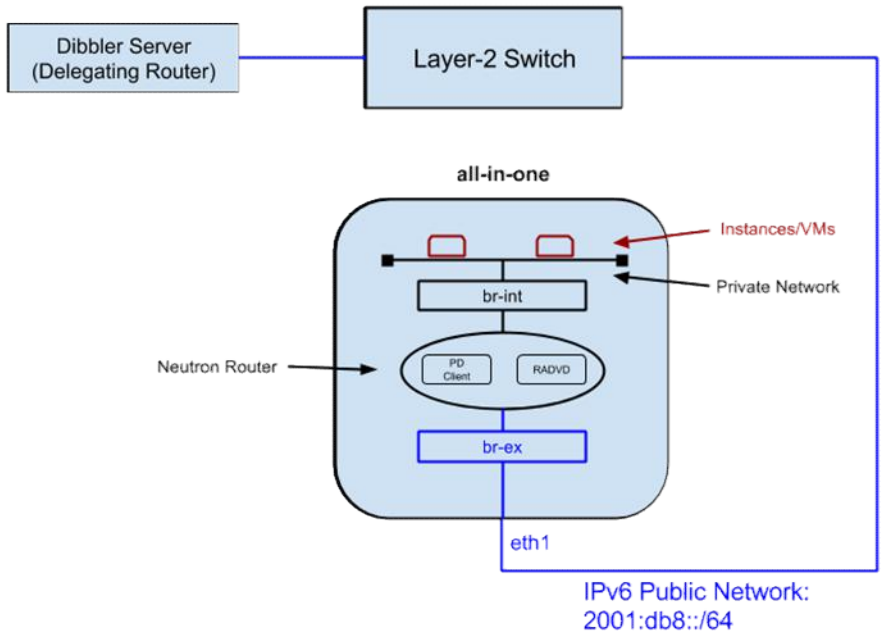
IPv6 Prefix Delegation

- RFC 3633 & 3769
- IPv6 PD was originally designed to allow a downstream router (Requesting router) to request a prefix from an upstream router (Delegating router) and use the assigned IPv6 prefix for the subscriber-side networks
- Basically, it is for routers asking other routers/relays/servers for a whole prefix that can be used to ‘seed’ downstream networks with an IPv6 prefix of their own



IPv6 PD - Example Topology

- OpenStack All-in-One host is connected to an L2-switch
- Dibbler server connected to the L2-switch
- Pre-defined IPv6 prefix on “public network”
- We need an IPv6 prefix assigned to the tenant interface of the Neutron router



IPv6 PD - Configuration

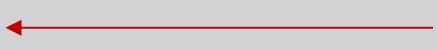
/etc/dibbler/server.conf

```
pd-class {
     pd-pool 2001:db8:face::/48
     pd-length 64
 }
```

/etc/neutron/neutron.conf

```
ipv6_pd_enabled = True
```

```
neutron net-create public --provider:network_type flat --provider:physical_network public --router:external

neutron subnet-create public --ip-version 6 --name public-v6-subnet  2001:db8:bad:cafe::/64

neutron router-create pd-rtr

neutron router-gateway-set pd-rtr public

neutron net-create ipv6-pd

neutron subnet-create ipv6-pd --name ipv6-pd-1 --ip_version 6 --ipv6_ra_mode slaac --ipv6_address_mode slaac --use-default-subnetpool

# Look for "subnetpool_id | prefix_delegation" in the output

neutron router-interface-add pd-rtr ipv6-pd-1
```



Magic happens here

Reference

Logs and tcpdumps and stuff

Neutron L3 Log

```
2016-10-17 15:03:46.822 DEBUG neutron.agent.linux.dibbler [-] Enable IPv6 PD for router 561ed48c-182c-4073-b157-77130280d5b9 subnet
3bc82226-816f-4d71-983e-7429d3d5475a ri_ifname qr-98120bdd-d1 from (pid=56056) enable /opt/stack/neutron/neutron/agent/linux/dibbler.py:123

2016-10-17 15:03:46.824 DEBUG neutron.agent.linux.utils [-] Running command (rootwrap daemon): ['ip', 'netns', 'exec', 'qrouter-561ed48c-
182c-4073-b157-77130280d5b9', 'dibbler-client', 'start', '-w', '/opt/stack/data/neutron/pd/561ed48c-182c-4073-b157-77130280d5b9:3bc82226-
816f-4d71-983e-7429d3d5475a:qr-98120bdd-d1'] from (pid=56056) execute_rootwrap_daemon /opt/stack/neutron/neutron/agent/linux/utils.py:100

2016-10-17 15:03:46.847 DEBUG neutron.agent.linux.dibbler [-] dibbler client enabled for router 561ed48c-182c-4073-b157-77130280d5b9 subnet
3bc82226-816f-4d71-983e-7429d3d5475a ri_ifname qr-98120bdd-d1 from (pid=56056) enable /opt/stack/neutron/neutron/agent/linux/dibbler.py:129
```

```
15:03:46.852214 IP6 (flowlabel 0x7bf3b, hlim 1, next-header UDP (17) payload length: 60) fe80::f816:3eff:feff:ccb0.dhcpv6-client >
ff02::1:2.dhcpv6-server: [udp sum ok] dhcp6 solicit (xid=800a54 (client-ID vid 000022b83bc82226) (IA_PD IAID:1 T1:4294967295 T2:4294967295)
(elapsed-time 0))

15:03:46.853654 IP6 (flowlabel 0x80c10, hlim 64, next-header UDP (17) payload length: 134) fe80::20c:29ff:fe87:2f6b.dhcpv6-server >
fe80::f816:3eff:feff:ccb0.dhcpv6-client: [udp sum ok] dhcp6 advertise (xid=800a54 (IA_PD IAID:1 T1:2000 T2:3000 (IA_PD-prefix
2001:db8:face:2ff2::/64 pltime:3600 vltime:7200) (status-code success)) (server-ID hwaddr/time type 1 time 529454623 000c29872f6b) (client-
ID vid 000022b83bc82226) (preference 0))

15:03:47.955793 IP6 (flowlabel 0x7bf3b, hlim 1, next-header UDP (17) payload length: 107) fe80::f816:3eff:feff:ccb0.dhcpv6-client >
ff02::1:2.dhcpv6-server: [udp sum ok] dhcp6 request (xid=561e28 (client-ID vid 000022b83bc82226) (IA_PD IAID:1 T1:4294967295 T2:4294967295
(IA_PD-prefix 2001:db8:face:2ff2::/64 pltime:3600 vltime:7200)) (server-ID hwaddr/time type 1 time 529454623 000c29872f6b) (elapsed-time
0))

15:03:47.956239 IP6 (flowlabel 0x80c10, hlim 64, next-header UDP (17) payload length: 134) fe80::20c:29ff:fe87:2f6b.dhcpv6-server >
fe80::f816:3eff:feff:ccb0.dhcpv6-client: [udp sum ok] dhcp6 reply (xid=561e28 (IA_PD IAID:1 T1:2000 T2:3000 (IA_PD-prefix
2001:db8:face:2ff2::/64 pltime:3600 vltime:7200) (status-code success)) (server-ID hwaddr/time type 1 time 529454623 000c29872f6b) (client-
ID vid 000022b83bc82226) (preference 0))
```

Reference

Dibbler Server Log

```
03:46 Server Notice    Received SOLICIT on br-ex/8, trans-id=0x800a54, 3 opts: 1 25 8 (non-relayed)
. . .
03:46 Server Debug     Adding client (DUID=00:02:00:00:22:b8:3b:c8:22:26:81:6f:4d:71:98:3e:74:29:d3:d5:47:5a) to addrDB.
03:46 Server Debug     PD: Adding PD (iaid=1) to addrDB.
03:46 Server Debug     PD: Adding 2001:db8:face:2ff2:: prefix to PD (iaid=1) to addrDB.
. . .
03:46 Server Notice    Sending ADVERTISE on br-ex/8,transID=0x800a54, opts: 25 2 1 7, 0 relay(s).
. . .
03:47 Server Notice    Received REQUEST on br-ex/8, trans-id=0x561e28, 4 opts: 1 25 2 8 (non-relayed)
. . .
03:47 Server Debug     Checking prefix 2001:db8:face:2ff2:: against reservations ...
03:47 Server Debug     PD: Requested prefix (2001:db8:face:2ff2::) is free, great!
03:47 Server Debug     Adding client (DUID=00:02:00:00:22:b8:3b:c8:22:26:81:6f:4d:71:98:3e:74:29:d3:d5:47:5a) to addrDB.
03:47 Server Debug     PD: Adding PD (iaid=1) to addrDB.
03:47 Server Debug     PD: Adding 2001:db8:face:2ff2:: prefix to PD (iaid=1) to addrDB.
03:47 Server Debug     PD: Prefix usage for class 0 increased to 2.
03:47 Server Info      PD: assigned prefix(es):2001:db8:face:2ff2::/64
```

Output truncated for clarity

IPv6 with Heat

Orchestrating IPv6 Deployment with Heat

- Heat with IPv6 uses similar parameters and resource conventions as with IPv4

IPv4	IPv6

`private_net_cidr:`
`  type: string`
`  description: Tenant IPv4 network address (CIDR notation)`
`  default: 10.10.30.0/24`
	`private_net_v6:`
	`  type: string`
	`  description: Tenant IPv6 subnet address`
	`  default: 2001:db8:cafe:1e::/64`

- https://github.com/shmcfarl/my-heat-templates

IPv6-only Example

https://github.com/shmcfarl/my-heat-templates/blob/master/v6-only-SLAAC.yaml

Parameters

```
private_net_v6:
    type: string
    description: Private IPv6 subnet address
    default: 2001:db8:cafe:1e::/64
  private_net_v6_gateway:
    type: string
    description: Private network gateway address
    default: 2001:db8:cafe:1e::1
  private_net_v6_pool_start:
    type: string
    description: Start of private network IP address
allocation pool
    default: 2001:db8:cafe:1e::2
  private_net_v6_pool_end:
    type: string
    description: End of private network IP address
allocation pool
    default: 2001:db8:cafe:1e:ffff:ffff:ffff:fffe
```

Resources

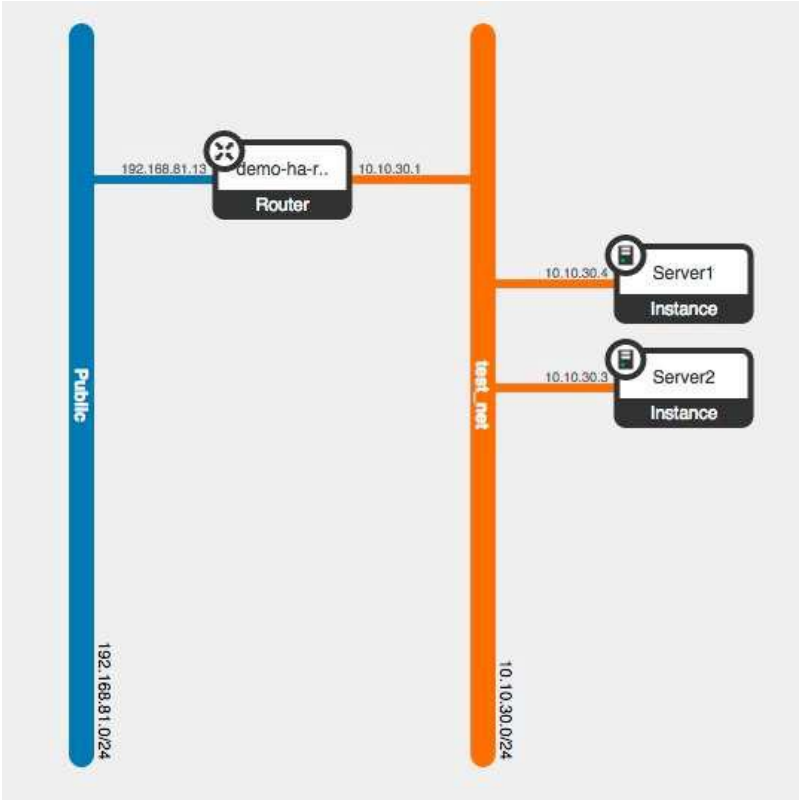
```
private_v6_subnet:
    type: OS::Neutron::Subnet
    properties:
      ip_version: 6
      ipv6_address_mode: slaac
      ipv6_ra_mode: slaac
      network: { get_resource: private_net }
      cidr: { get_param: private_net_v6 }
      gateway_ip: { get_param: private_net_v6_gateway
}
      allocation_pools:
        - start: { get_param:
private_net_v6_pool_start }
          end: { get_param: private_net_v6_pool_end }
```

```
router_interface_v6:
    type: OS::Neutron::RouterInterface
    properties:
      router: { get_resource: router }
      subnet: { get_resource:
private_v6_subnet }

server_security_group:
    type: OS::Neutron::SecurityGroup
    properties:
      description: Heat-deployed security
group.
      name: heat-security-group
      rules: [
        {remote_ip_prefix: "::/0",
        ethertype: IPv6,
        protocol: tcp,
        port_range_min: 22,
        port_range_max: 22},
        {remote_ip_prefix: "::/0",
        ethertype: IPv6,
        protocol: icmp},
        {remote_ip_prefix: "::/0",
        ethertype: IPv6,
        protocol: tcp,
        port_range_min: 80,
        port_range_max: 80}]
```

IPv6 with L3 High-Availability

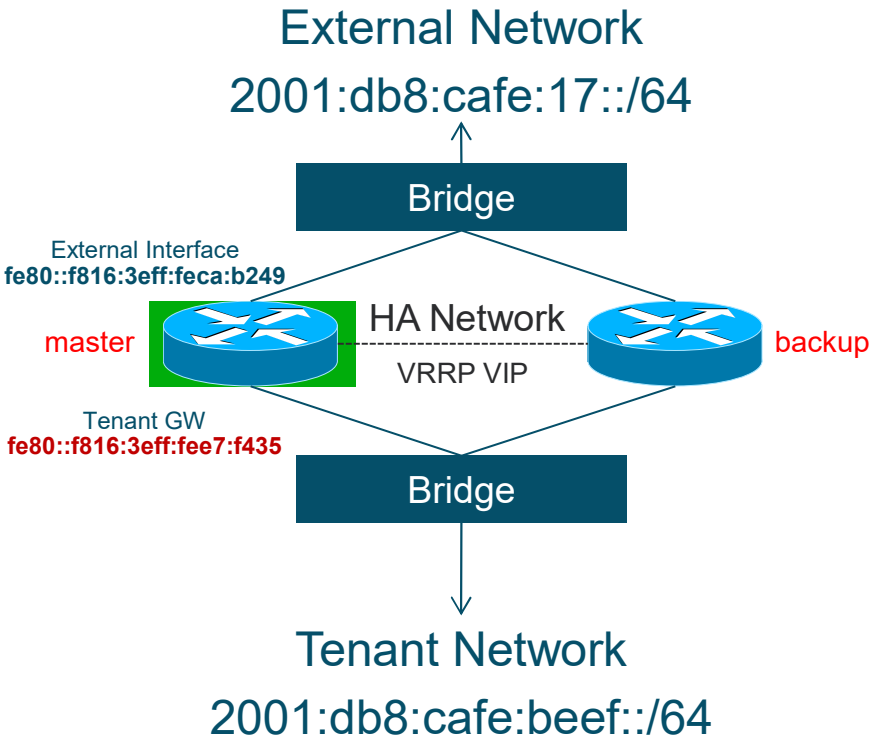
L3 HA – Tenant View

- Tenant sees one router with a single gateway IP address
- non-Admin users cannot control if the router is HA or non-HA
- From the tenant's perspective the router behaves the same in HA or non-HA mode



L3 HA – Routing View

- Tenant network has 2001:db8:cafe:beef::/64 prefix assigned
- VRRP is using a dedicated HA-only network that traverses the same tenant network type
- Router (L3 agent) on the left is the VRRP master and is the tenant IPv6 GW (fe80::f816:3eff:fee7:f435)



```
# cat /var/lib/neutron/ha_confs/0772d696-ec0f-46f3-b7d0-a984612fcdca/keepalived.conf
vrrp_instance VR_1 {
    state BACKUP
    interface ha-7cf36911-75
    virtual_router_id 1
    priority 50
    garp_master_delay 60
    nopreempt
    advert_int 2
    track_interface {
        ha-7cf36911-75
    }
    virtual_ipaddress {
        169.254.0.1/24 dev ha-7cf36911-75
    }
    virtual_ipaddress_excluded {
        10.0.0.1/24 dev qr-50deb6c5-c7
        192.168.81.102/24 dev qg-e93ae851-38
        2001:db8:cafe:17::66/64 dev qg-e93ae851-38
        2001:db8:cafe:beef::1/64 dev qr-1b8ada84-61
        fe80::f816:3eff:fe50:ab1/64 dev qr-50deb6c5-c7 scope link
        fe80::f816:3eff:feca:b249/64 dev qg-e93ae851-38 scope link
        fe80::f816:3eff:fee7:f435/64 dev qr-1b8ada84-61 scope link
    }
    virtual_routes {
        ::/0 via 2001:db8:cafe:17::1 dev qg-e93ae851-38
        0.0.0.0/0 via 192.168.81.2 dev qg-e93ae851-38
    }
```

L3 HA Interface

Track the L3 HA interface

VRRP IP address

VIP address from ‘real’ networks

LL used as VM’s GW

Default route out

```
# ip netns exec qrouter-0772d696-ec0f-46f3-b7d0-a984612fcdca ip a
9: ha-7cf36911-75: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:7e:bc:ae brd ff:ff:ff:ff:ff:ff
    inet 169.254.192.2/18 brd 169.254.255.255 scope global ha-7cf36911-75
       valid_lft forever preferred_lft forever
    inet 169.254.0.1/24 scope global ha-7cf36911-75
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe7e:bcae/64 scope link
       valid_lft forever preferred_lft forever
11: qr-50deb6c5-c7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:50:0a:b1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 scope global qr-50deb6c5-c7
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe50:ab1/64 scope link
       valid_lft forever preferred_lft forever
12: qr-1b8ada84-61: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:e7:f4:35 brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:cafe:beef::1/64 scope global nodad
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fee7:f435/64 scope link
       valid_lft forever preferred_lft forever
13: qg-e93ae851-38: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:ca:b2:49 brd ff:ff:ff:ff:ff:ff
    inet 192.168.81.102/24 scope global qg-e93ae851-38
       valid_lft forever preferred_lft forever
    inet6 2001:db8:cafe:17::66/64 scope global nodad
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:feca:b249/64 scope link
       valid_lft forever preferred_lft forever
```

L3 HA Interface

Tenant Network

External/Public Network

Reference

VRRPv2 Advertisement

```
# ip netns exec qrouter-0772d696-ec0f-46f3-b7d0-a984612fcdca tcpdump -n -i ha-7cf36911-75
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ha-7cf36911-75, link-type EN10MB (Ethernet), capture size 65535 bytes
15:17:19.021100 IP 169.254.192.2 > 224.0.0.18: VRRPv2, Advertisement, vrid 1, prio 50, authtype none, intvl 2s, length 20
15:17:21.021783 IP 169.254.192.2 > 224.0.0.18: VRRPv2, Advertisement, vrid 1, prio 50, authtype none, intvl 2s, length 20
15:17:23.023316 IP 169.254.192.2 > 224.0.0.18: VRRPv2, Advertisement, vrid 1, prio 50, authtype none, intvl 2s, length 20
15:17:25.025260 IP 169.254.192.2 > 224.0.0.18: VRRPv2, Advertisement, vrid 1, prio 50, authtype none, intvl 2s, length 20
```

Testing a failure

Check who is master:

```
[root@c7-m-aio ~]# cat /var/lib/neutron/ha_confs/0772d696-ec0f-46f3-b7d0-a984612fcdca/state
master
[root@c7-m-net-cmp ~]# cat /var/lib/neutron/ha_confs/0772d696-ec0f-46f3-b7d0-a984612fcdca/state
backup
```

Simulate a failure by shutting down the HA interface (remember this was in the ‘track’ list):

```
[root@c7-m-aio ~]# ip netns exec qrouter-0772d696-ec0f-46f3-b7d0-a984612fcdca ifconfig ha-7cf36911-75 down
```

Check that VRRP switched to the other node as master:

```
[root@c7-m-aio ~]# cat /var/lib/neutron/ha_confs/0772d696-ec0f-46f3-b7d0-a984612fcdca/state
fault
[root@c7-m-net-cmp ~]# cat /var/lib/neutron/ha_confs/0772d696-ec0f-46f3-b7d0-a984612fcdca/state
master
```

IPv6 with BGP

A quick taste of BGP + IPv6 in Neutron:

```
neutron bgp-speaker-create --ip-version 6 --local-as 65001 bgp-speaker
neutron bgp-speaker-network-add bgp-speaker public
neutron bgp-peer-create --peer-ip 2001:db8:cafe:17:20c:29ff:fe12:b124 --remote-as 65000 bgp-peer
neutron bgp-speaker-peer-add bgp-speaker bgp-peer
neutron bgp-dragent-speaker-add d2929bed-a65a-4179-8447-5def013b3113 bgp-speaker
```

The other end:

```
router bgp 65000
 bgp router-id 192.168.81.4
 neighbor 192.168.81.1 remote-as 65001
 neighbor 2001:db8:cafe:17::7 remote-as 65001
!
 address-family ipv6
 network 2001:db8:bad:face::/64
 neighbor 2001:db8:cafe:17::7 activate
 exit-address-family
```


CISCO