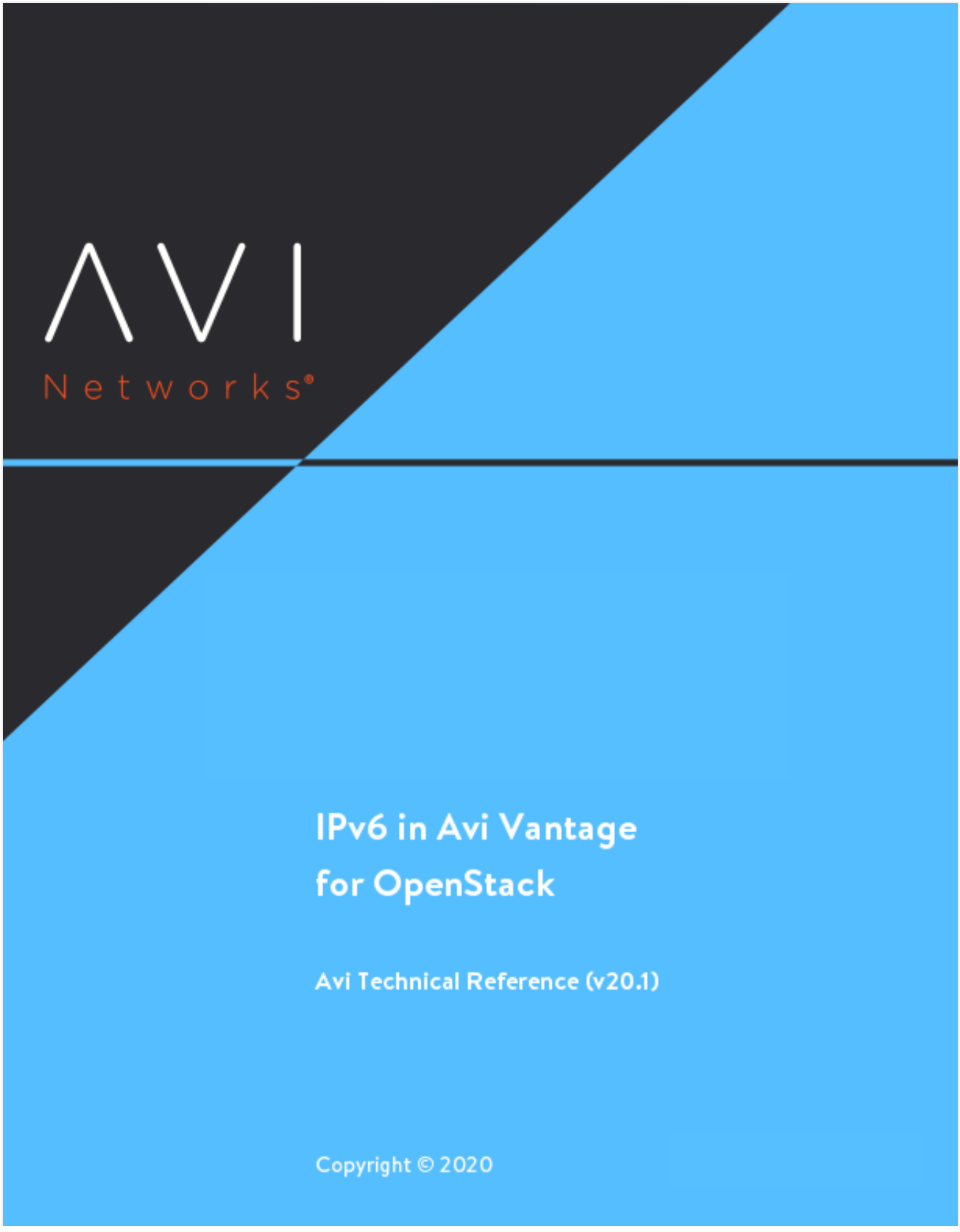


IPv6 in Avi Vantage for OpenStack

Avi Technical Reference (v20.1)

IPv6 in Avi Vantage for OpenStack

view online

Overview

Starting with release 18.1.1, OpenStack integration with Avi Vantage is IPv6 capable. The integration discussed in this article has been tested for OpenStack Ocata which focuses on resolving scalability and performance issues.

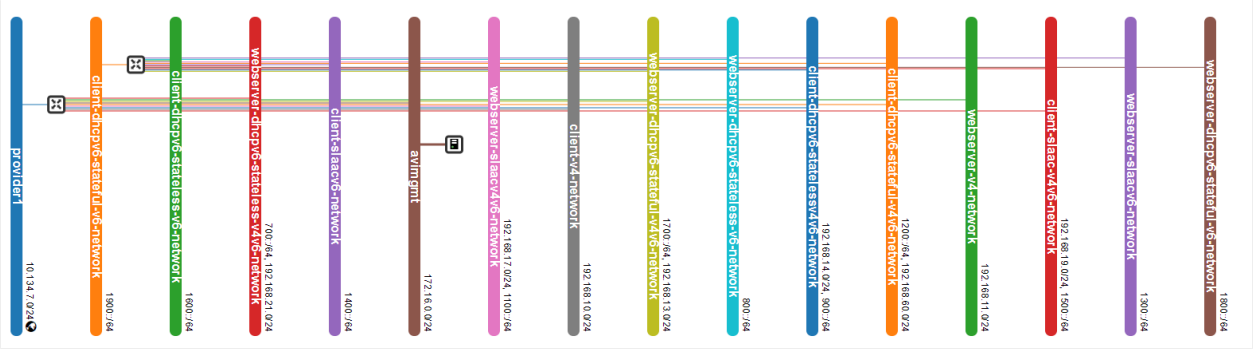
Software Defined Networking (SDN) support is as follows:

Contrail	Yes

Setup Information

In a general setup, the Avi Controller is spun up in the management network (*avimgmt* in this case). The Controller is used for testing against OpenStack no-access cloud as well as a generic OpenStack cloud. Server and client instances are spun up in Stateless address auto configuration (SLAAC) based client and server networks respectively. These instances are in either Dynamic Host Configuration Protocol version 6 (DHCPv6) stateful based client or server networks. More information regarding this are discussed in the following sections.

Network Topology

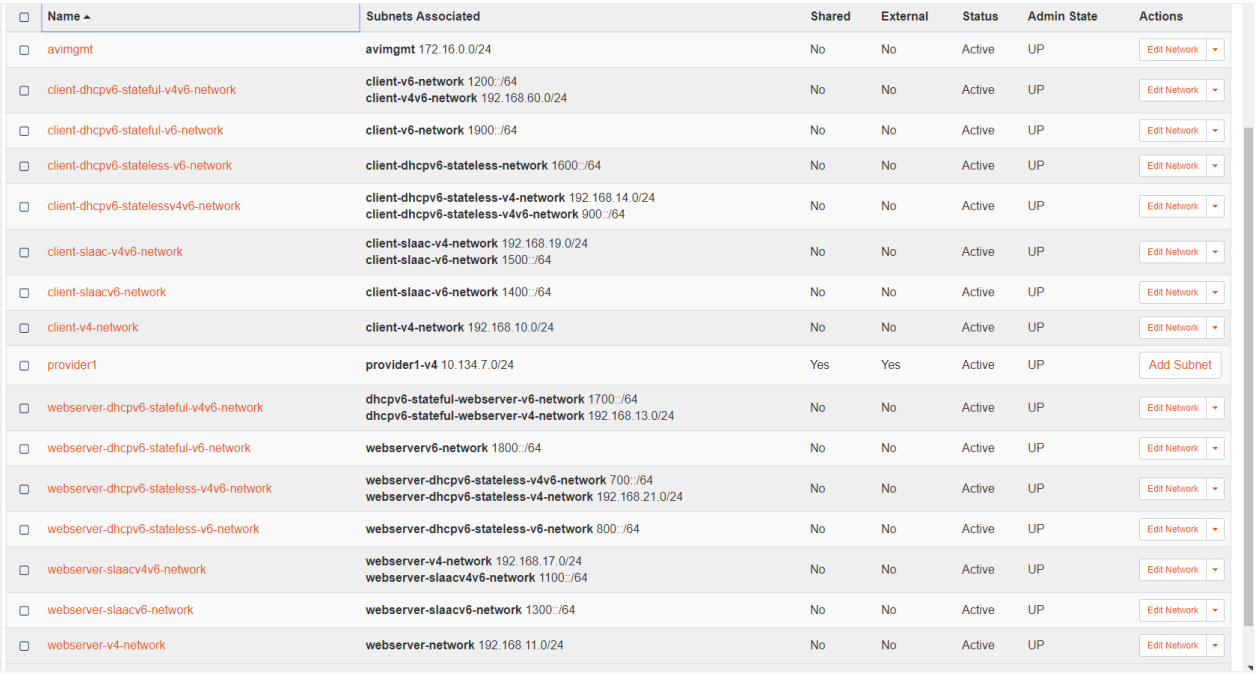


Networks Considered

As displayed in the figure, the following network combinations are considered: * IPv4 networks. * IPv6 networks, which are further broken down into SLAAC, DHCPv6 stateless, and DHCPv6 stateful. These are chosen from the Horizon UI or OpenStack CLI during the network creation. * Networks starting with *client* prefix are chosen for virtual services and networks starting with *server* prefix are chosen for backend servers.

As a result, a total of 16 networks are considered as displayed in the screenshot below.

Note: This is done to cover all possible scenarios and is not an exact requirement.



Name	Subnets Associated	Shared	External	Status	Admin State	Actions
avimgmt	**avimgmt** 172.16.0.0/24	No	No	Active	UP	Edit Network
client-dhcpv6-stateful-v4v6-network	**client-v6-network** 1200::/64 **client-v4v6-network** 192.168.60.0/24	No	No	Active	UP	Edit Network
client-dhcpv6-stateful-v6-network	**client-v6-network** 1900::/64	No	No	Active	UP	Edit Network
client-dhcpv6-stateless-v6-network	**client-dhcpv6-stateless-network** 1600::/64	No	No	Active	UP	Edit Network
client-dhcpv6-statelessv4v6-network	**client-dhcpv6-stateless-v4-network** 192.168.14.0/24 **client-dhcpv6-stateless-v4v6-network** 900::/64	No	No	Active	UP	Edit Network
client-slaac-v4v6-network	**client-slaac-v4-network** 192.168.19.0/24 **client-slaac-v6-network** 1500::/64	No	No	Active	UP	Edit Network
client-slaacv6-network	**client-slaac-v6-network** 1400::/64	No	No	Active	UP	Edit Network
client-v4-network	**client-v4-network** 192.168.10.0/24	No	No	Active	UP	Edit Network
provider1	**provider1-v4** 10.134.7.0/24	Yes	Yes	Active	UP	Add Subnet
webserver-dhcpv6-stateful-v4v6-network	**dhcpv6-stateful-webserver-v6-network** 1700::/64 **dhcpv6-stateful-webserver-v4-network** 192.168.13.0/24	No	No	Active	UP	Edit Network
webserver-dhcpv6-stateful-v6-network	**webserverv6-network** 1800::/64	No	No	Active	UP	Edit Network
webserver-dhcpv6-stateless-v4v6-network	**webserver-dhcpv6-stateless-v4v6-network** 700::/64 **webserver-dhcpv6-stateless-v4-network** 192.168.21.0/24	No	No	Active	UP	Edit Network
webserver-dhcpv6-stateless-v6-network	**webserver-dhcpv6-stateless-v6-network** 800::/64	No	No	Active	UP	Edit Network
webserver-slaacv4v6-network	**webserver-v4-network** 192.168.17.0/24 **webserver-slaacv4v6-network** 1100::/64	No	No	Active	UP	Edit Network
webserver-slaacv6-network	**webserver-slaacv6-network** 1300::/64	No	No	Active	UP	Edit Network
webserver-v4-network	**webserver-network** 192.168.11.0/24	No	No	Active	UP	Edit Network

The setup thus comprises of one-arm and two-arm setups where everything can be contained in one network or different networks, respectively. In this document, the two-arm setups are used for virtual service creation (separate network) and then the traffic is sent to the server, which is on a different network. In two-arm setups, the virtual service is in one network and its back-end servers are in an another network.

IP Addressing Schema Used

Virtual services are created using either stateful DHCPv6, SLAAC, or static IPv6 addressing. In this case, the IP address is chosen from a SLAAC v6 or DHCPv6 stateful network. IPv4 is used at instances too.

Note: OpenStack does not support floating IPv6. Hence,
1. For dual stack virtual service, floating IP will always get translated (NAT) to private v4 VIP address.
2. For an IPv6 Virtual IP (VIP)as in the case of a dual stack virtual service, the IPv6 VIP, irrespective of being private or public, must be on the same interface as that of the private IPv4 VIP network, for the placement to get through.

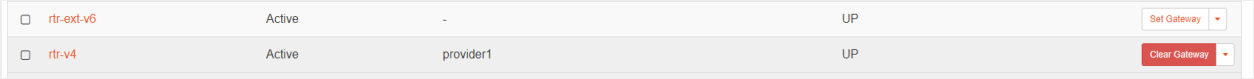
You will notice the following combinations of IPv4, IPv6, and dual stack networks covering all addressing types: * Static or manual configuration. * Stateless address auto configuration (SLAAC) - IPv6 prefix (/64) is assigned to the end node using router advertisement (RA) and the node self-constructs the interface ID (IID) portion of the address (the last /64 bits). * Stateful DHCPv6 - Similar to IPv4 DHCP, a DHCPv6 server handles the entire IPv6 addressing and configuration options. * Service Engine starts the DHCPv6 client only upon the reception of route advertisement (RA) from the designated router. * Stateless DHCPv6 - A combination of SLAAC for address assignment and DHCPv6 for option assignment, such as, Domain Name System (DNS)DNS, domain name, is used.

Networking

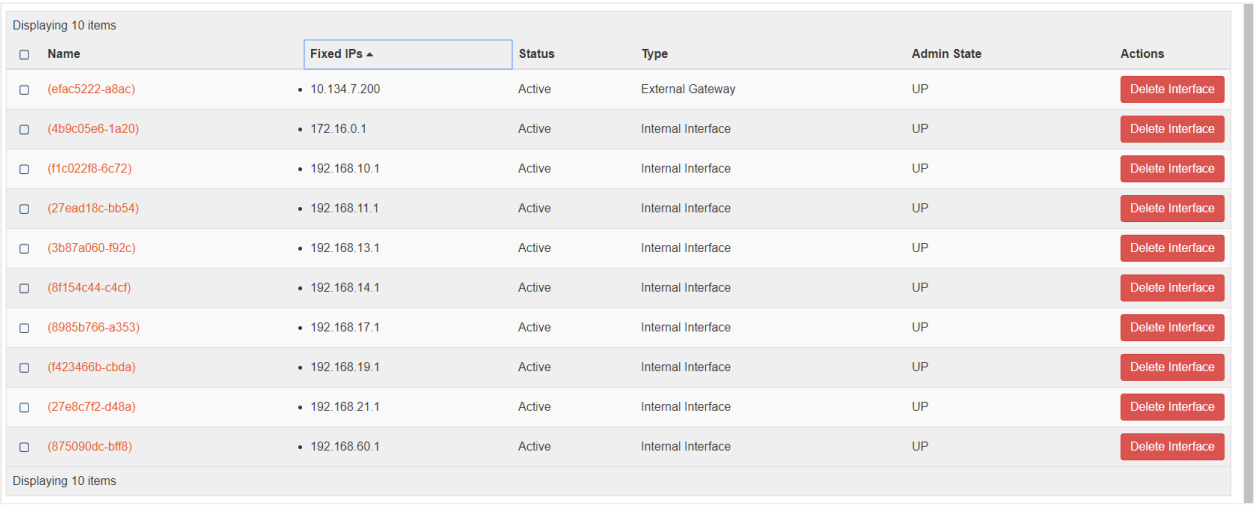
Note: This is setup-agnostic and you can alter this as per your requirements.

As noticed in the network topology above, all networks should be routed within the respective Virtual LAN (VLAN) 207 network. Here, VLAN 207 is the provider network VLAN. This routing requires a router in the network.

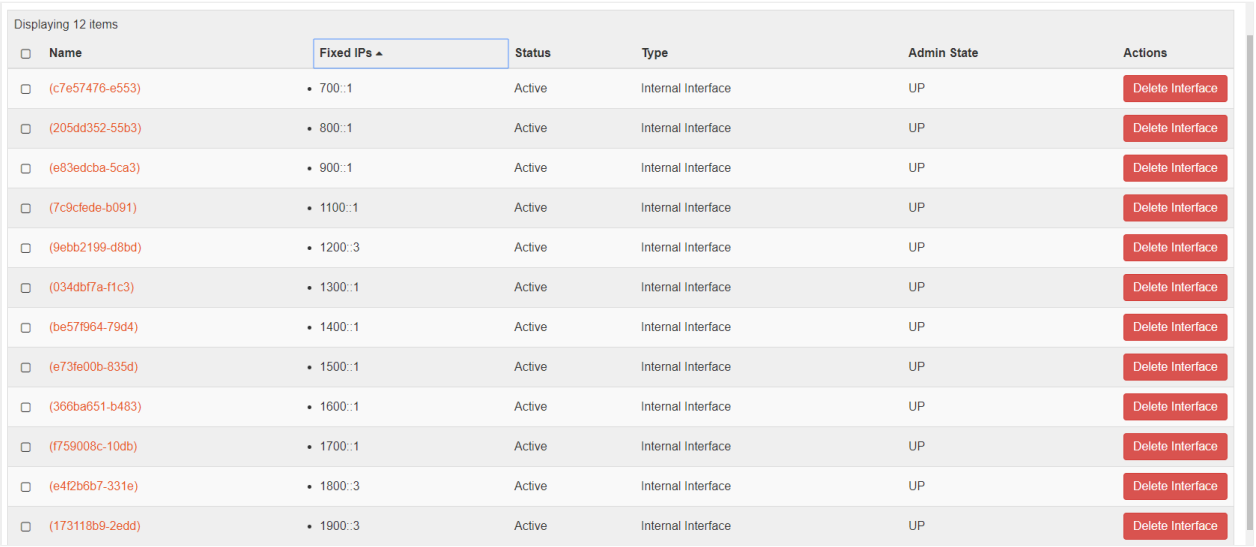
Interfaces connected to the router



Each network must have an interface that is connected to the router (IPv4 or IPv6 router) as displayed in the screenshots below.



IPv4 Interfaces in rtr-v4



IPv6 Interfaces in rtr-ext-v6

IPv6 route advertisement

A router is configured to handle IPv6 route advertisement (RAs). If DHCPv6 is the chosen IP addressing schema, RAs must be sent across the OpenStack for it work on the SEs. Similarly, after the IPv6 networks are created, pick the link-local address (LLA) for each IPv6 network and update that as the default gateway for each IPv6 network. The configuration is as shown below:

```
root@openstack-ocata:~# ip netns | grep qrouter
qrouter-3eeb3d41-8955-431c-b2bd-1934c97900a9
qrouter-017709bf-5ab1-43b5-a372-75c554c1b961
```

```
root@openstack-ocata:~# ip netns exec qrouter-017709bf-5ab1-43b5-a372-75c554c1b961 bash
```

```
ip a
```

As seen in the output below, the value *inet6 fe80::f816:3eff:fe88:18ff* is the LLA that is updated under the IPv6 network settings.

```
13: qr-c7e57476-e5@if109:
mtu 1450 qdisc noqueue state UP group default qlen 1000
link/ether fa:16:3e:88:18:ff brd ff:ff:ff:ff:ff:ff link-netnsid 0
inet6 700::1/64 scope global
valid_lft forever preferred_lft forever
inet6 fe80::f816:3eff:fe88:18ff/64 scope link
valid_lft forever preferred_lft forever
```

Security Groups

- In case of a generic OpenStack cloud, the security group rules are automatically created for the Service Engines.

Displaying 7 items

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
Egress	IPv6	Any	Any	::/0	-	Delete Rule
Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
Ingress	IPv6	ICMP	Any	::/0	-	Delete Rule
Ingress	IPv6	TCP	22 (SSH)	::/0	-	Delete Rule
Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule
Egress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule

- In case of a No-Orchestrator cloud, when the SE is spun up manually using the steps mentioned at Creating Service Engine using Heat-Templates in no access OpenStack Cloud or Installing Avi Vantage into a No-Access OpenStack Cloud, certain rules for IPv4 and IPv6 must be allowed for the traffic (The Internet Control Message Protocol (ICMP), Secure Shell (SSH) to work. Open up ports such as 22, 80, 443, and 8443. If 8443 is not configured as an exception, then the SE will not be able to SSH into the Controller and vice-versa.

Displaying 14 items

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
Ingress	IPv4	Any	Any	-	default	Delete Rule
Ingress	IPv6	Any	Any	::/0	-	Delete Rule
Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
Egress	IPv6	Any	Any	::/0	-	Delete Rule
Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
Ingress	IPv6	ICMP	Any	::/0	-	Delete Rule
Egress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
Egress	IPv6	ICMP	Any	::/0	-	Delete Rule
Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule
Egress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	443 (HTTPS)	0.0.0.0/0	-	Delete Rule
Egress	IPv4	TCP	8443	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	8443	0.0.0.0/0	-	Delete Rule

Allowed-Address-Pairs

The allowed address pair extension extends the port attribute that enables you to specify arbitrary mac_address/ip_address ((Classless Inter-Domain Routing, (CIDR)) pairs that are allowed to pass through a port regardless of the subnet associated with the network.

- In case of a generic OpenStack cloud, this is enabled by default in the cloud settings and so no further changes are required.

Displaying 2 items

Name	Network Address	IP Version	Gateway IP	Actions
client-v6-network	1200::/64	IPv6	fe80::f816:3eff:fefe:3936	Edit Subnet
client-v4v6-network	192.168.60.0/24	IPv4	192.168.60.1	Edit Subnet

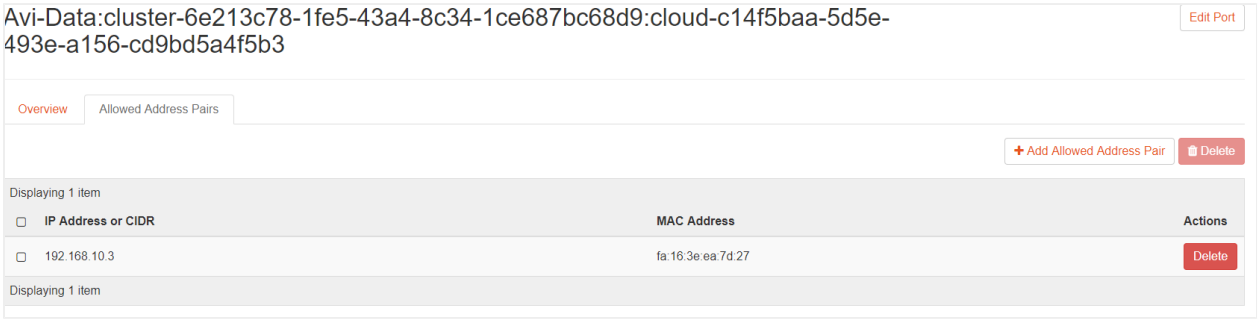
Displaying 2 items

- In case of a No-Orchestrator cloud, this can be done using OpenStack CLI or via Horizon.
  - The allowed-address-pairs neutron extension allows traffic with specific CIDRs to exit from a port. Avi Vantage uses this extension to place VIPs on Service Engine (SE) data. Thereby, VIPs on SE data ports allow VIP traffic to exit through these data ports.
  - Add allowed-address-pairs on the SE ports so that the security groups do not drop the packets. For the MLS /OVS plugin, you can add the allowed-address-pairs with 0.0.0.0/0 and with (if required for IPv6 or dual stack) ::/0 once for each of the SE ports or specific VIP address.
  - In case of Contrail used as SDN type with Avi Vantage as explained here the VIP addresses are added to the fixed address list. For differentiating interface IP, you should add the same to AAP with /24 or /120 prefix based on whether it is IPv4 or IPv6.

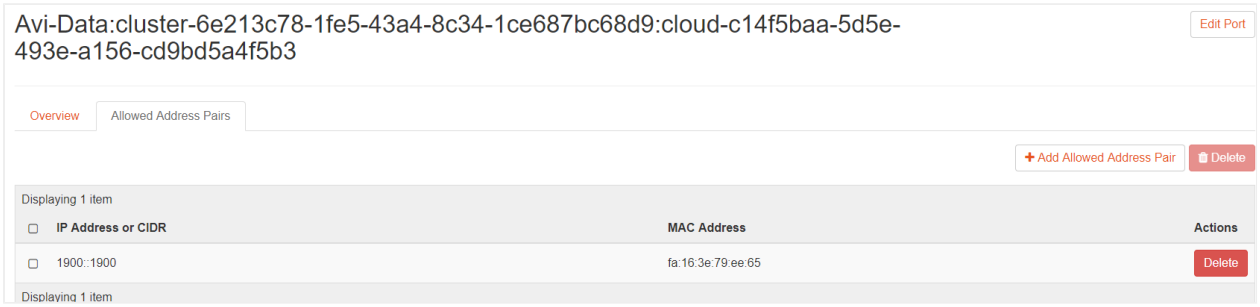
```
neutron port-update da0e1e9a-312d-41c2-b15f-f10ac344ef03 --allowed-address-pairs type=dict list=true ip_address=192.168

neutron port-update da0e1e9a-312d-41c2-b15f-f10ac344ef03 --allowed-address-pairs type=dict list=true ip_address=2001::6
```

If true, then the allowed-address-pairs extension will be used. If the underlying network plugin does not support this feature, then the VIP traffic will not work.



AAP Entry for IPv4 (No-Orchestrator Cloud)



AAP Entry for IPv6 (No-Orchestrator Cloud)

Configuring Avi Vantage

Points to Consider

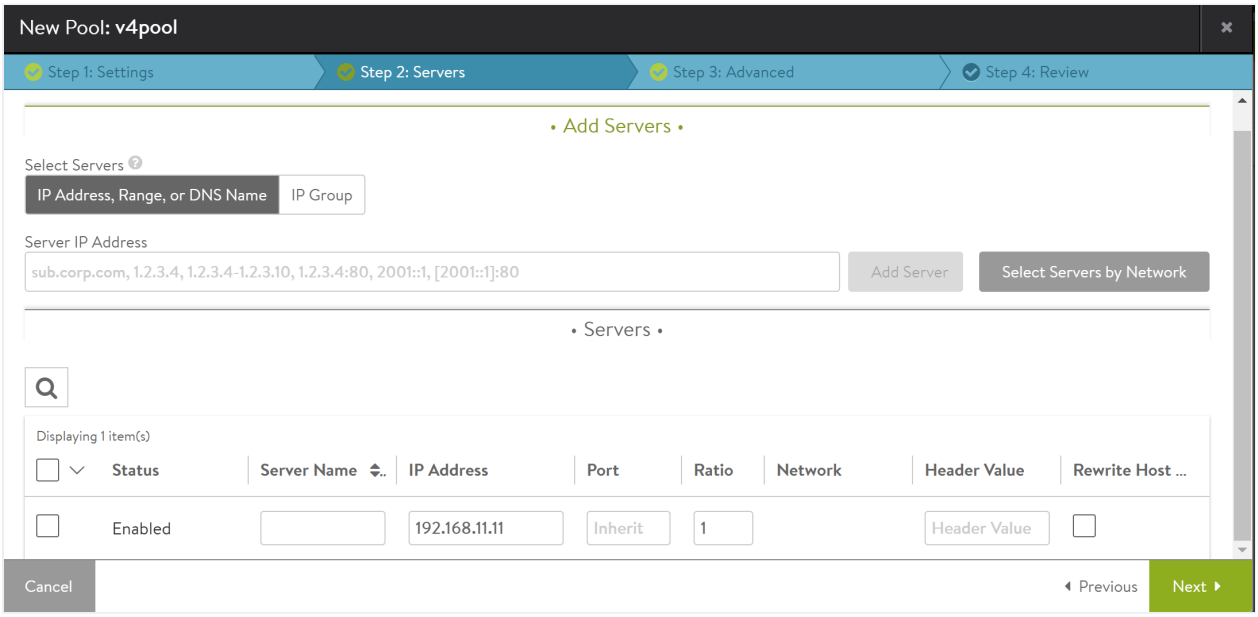
- Install Avi Vantage for OpenStack, by following the instructions at Installing Avi Vantage for OpenStack.
- Use suitable IP addressing mechanism for virtual services and pools IPv6 configuration. The example here uses SLAACv6.
- Obtain subnet IDs using OpenStack UI by navigating to openstack-horizon/networks/subnets/id, or on OpenStack CLI as explained below:

```
root@openstack-ocata:/root# openstack subnet show client-v4-network
+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| allocation_pools  | 192.168.10.2-192.168.10.254          |
| cidr              | 192.168.10.0/24                      |
| created_at        | 2017-12-08T13:06:54Z                 |
| description       |                                      |
| dns_nameservers   |                                      |
| enable_dhcp       | True                                 |
| gateway_ip        | 192.168.10.1                         |
| host_routes       |                                      |
| id                | 337c70de-3be5-4072-8e7f-04d61ee6ceb5 |
| ip_version        | 4                                    |
```

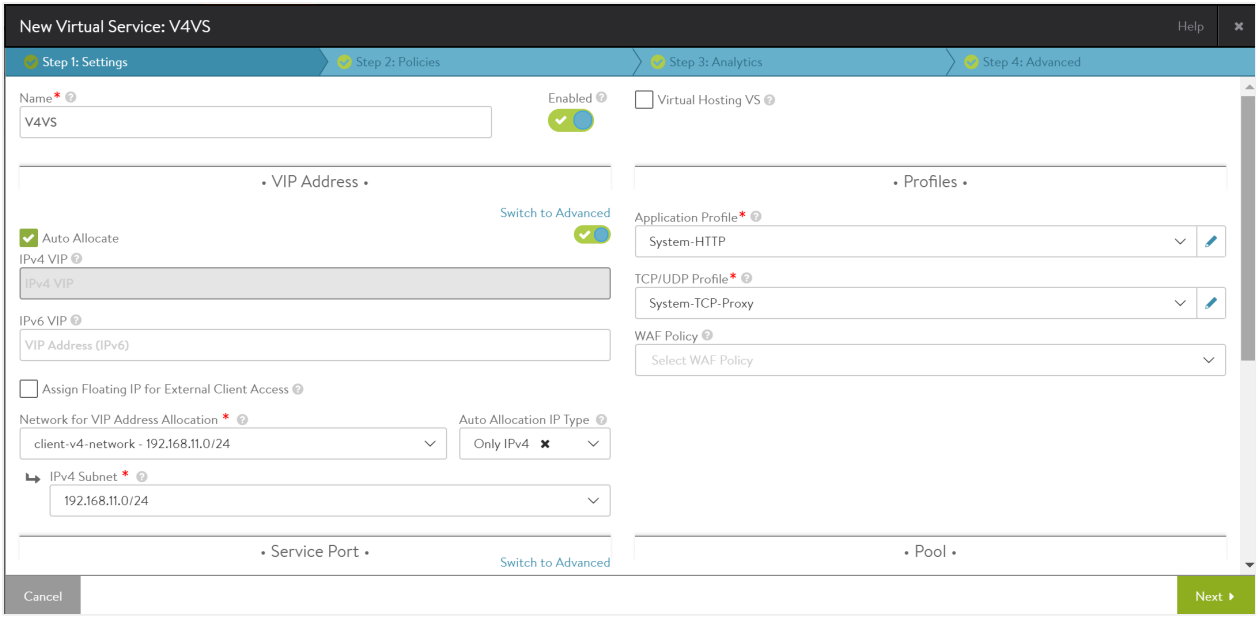
```
| ipv6_address_mode | None                                 |
| ipv6_ra_mode      | None                                 |
| name              | client-v4-network                    |
| network_id        | 85b861a8-7d87-4e03-8cfe-444082ff1412 |
| project_id        | 39155680f7d24b628d9752057527ccb9     |
| revision_number   | 4                                    |
| segment_id        | None                                 |
| service_types     |                                      |
| subnetpool_id     | None                                 |
| updated_at        | 2017-12-11T18:13:45Z                 |
+-------------------+--------------------------------------+
```

IPv4 pool and IPv4 virtual service

To configure the IPv4 pool, navigate to Applications > Pools and click on Create Pool. Under the Servers tab, provide an IPv4 Server IP Address.

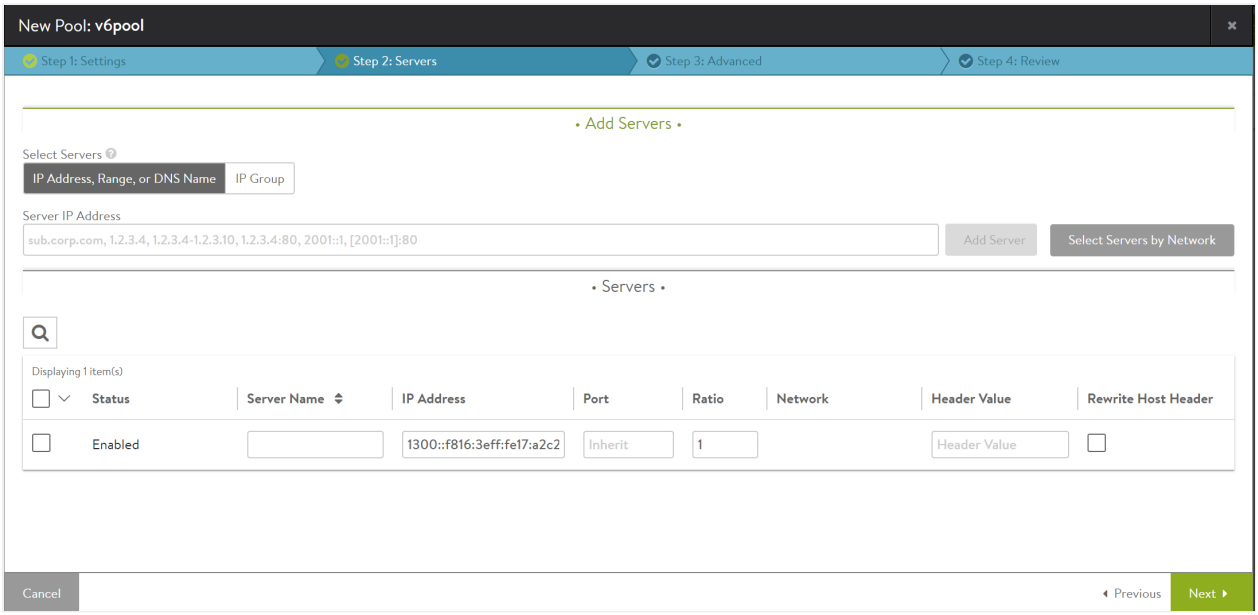


To configure an IPv4 virtual service, navigate to Applications > Virtual Service and click on Create Virtual Service (Advanced Setup). Click on the Auto Allocate checkbox. Choose *Only IPv4* under Auto Allocation IP Type and populate the Network for VIP Address Allocation field. Select the required subnet from the drop-down list for IPv4 Subnet.

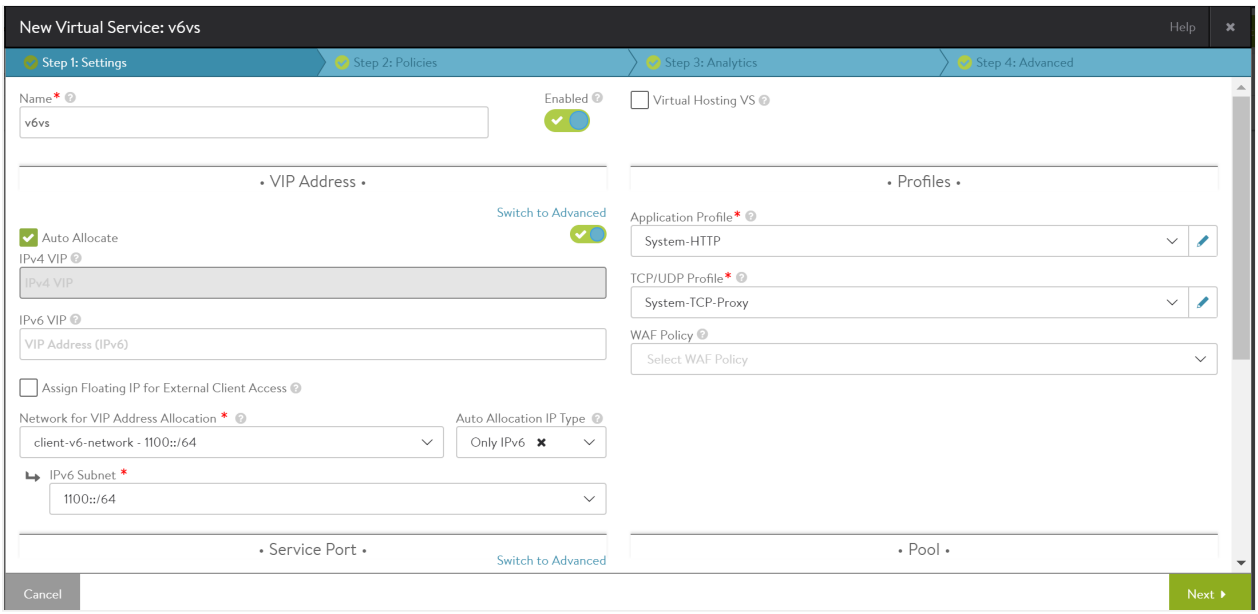


IPv6 pool and IPv6 virtual service

To configure the IPv6 pool, navigate to Applications > Pools and click on Create Pool. Under the Servers tab, provide an IPv6 Server IP Address.



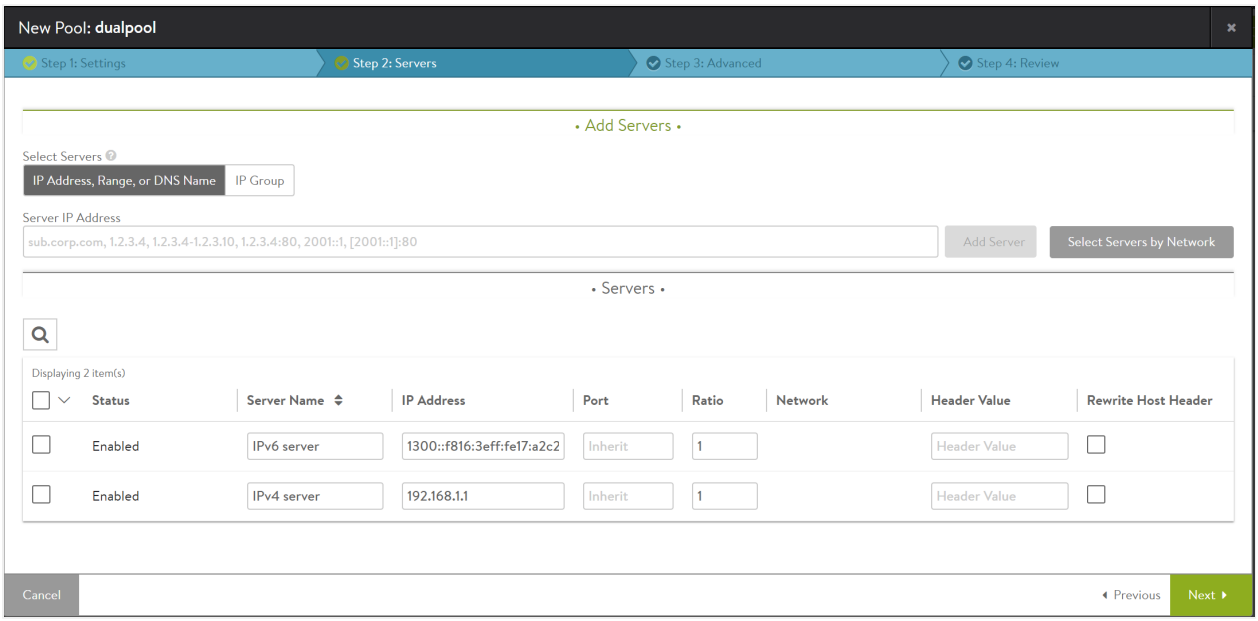
To configure an IPv6 virtual service, navigate to Applications > Virtual Service and click on Create Virtual Service (Advanced Setup). Click on the Auto Allocate checkbox. Choose *Only IPv6* under Auto Allocation IP Type and populate the Network for VIP Address Allocation field. Select the required subnet from the drop-down list for IPv6 Subnet.



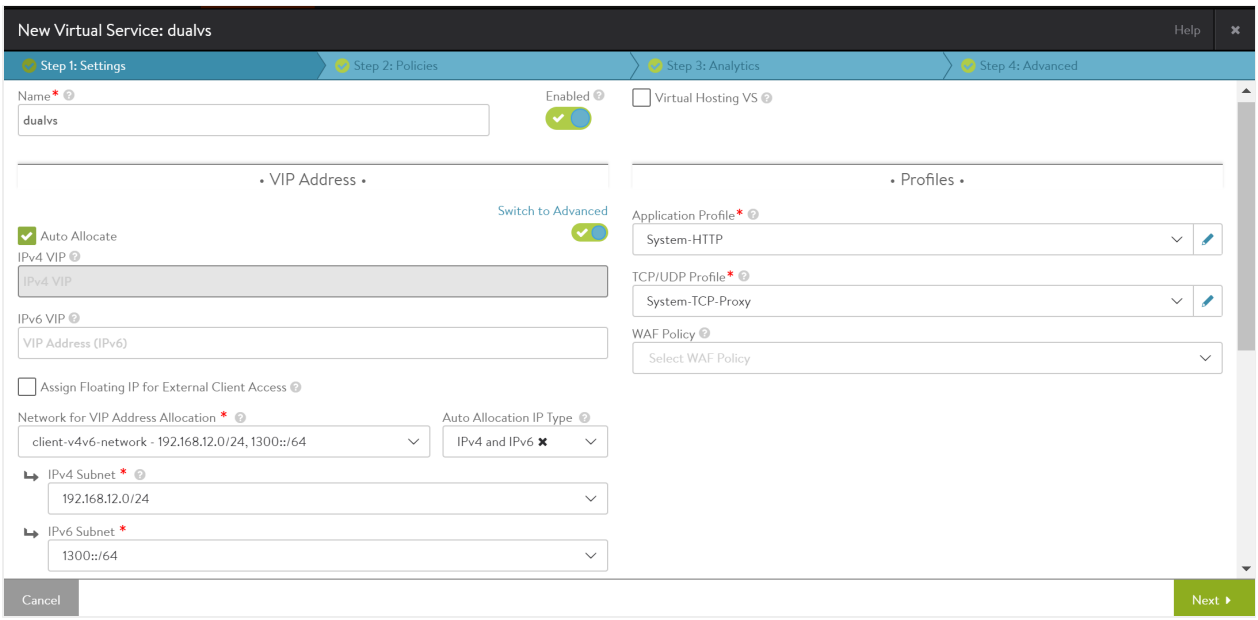
IPv4v6 pool and IPv4v6 virtual service (dual stack)

Note: In a dual stack network, if a virtual service is created with a single VIP, using either IPv4 or IPv6 address, then converting it into a dual stack by adding an additional IP address will fail.

To configure the dual stack pool, navigate to Applications > Pools and click on Create Pool. Under the Servers tab, provide either an IPv4 and IPv6 Server IP Address.



To configure a dual stack virtual service, navigate to Applications > Virtual Service and click on Create Virtual Service (Advanced Setup). Click on the Auto Allocate checkbox. Choose *IPv4 and IPv6* under Auto Allocation IP Type and populate the Network for VIP Address Allocation field. Select the required subnet from the drop-down list for IPv4 Subnet and IPv6 Subnet.



Troubleshooting

Traffic Capture

Use the tcpdump command to capture traffic for troubleshooting. The captures must be taken inside the qr interfaces' of the respective qrouter namespace in OpenStack.

Temporary IP generation is disabled in SLAACv6 mode.

In Avi Service Engine, where temporary IP generation is disabled for SLAACv6 mode, you will only have the global dynamic address.

```
5: eth2: mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
link/ether fa:16:3e:10:7a:0c brd ff:ff:ff:ff:ff:ff
inet6 1100::f816:3eff:fe10:7a0c/64 scope global dynamic
valid_lft 86306sec preferred_lft 14306sec
inet6 fe80::f816:3eff:fe10:7a0c/64 scope link
valid_lft forever preferred_lft forever
```