vRealize Operations Manager Load Balancing

Configuration Guide
Version 6.1 and 6.2

TECHNICAL WHITE PAPER

DECEMBER 2015

VERSION 1.0

Table of Contents

Revision History

DATE	VERSION	DESCRIPTION
December 2015	1.0	Initial version.
February 2016	1.1	Minor updates to include vRealize Operations Manager version 6.2

Introduction

This document describes the configuration of the load balancing modules of F5 Networks BIG-IP software (F5) and NSX load balancers for vRealize Operations Manager 6.1 and 6.2. This document is not an installation guide, but a load-balancing configuration guide that supplements the vRealize Operations Manager installation and configuration documentation available in the vRealize Operations Manager Documentation Center.

This information is for the following products and versions.

PRODUCT	VERSION	DOCUMENTATION
vRealize Operations Manager	6.1 and 6.2	http://pubs.vmware.com/vrealizeoperationsmanager-6/index.jsp
F5 BIG IP	11.5	https://support.f5.com/kb/en-us.html
NSX	6.1.3	https://pubs.vmware.com/NSX-6/index.jsp#Welcome/welcome.html
HA Proxy	1.5.x	http://www.haproxy.org/
CentOS	v6.x, v7,x	http://wiki.centos.org/Documentation
RHEL	v6.x	https://access.redhat.com/documentation/en-US/index.html
Keepalived	v1.2.13-4.el6	http://www.keepalived.org/

Load Balancing Concepts

Load balancers distribute work among servers in high availability (HA) deployments. The system administrator backs up the load balancers on a regular basis at the same time as other components.

Follow your site policy for backing up load balancers, keeping in mind the preservation of network topology and vRealize Operations Manager backup planning.

Following are the advantages of using a load balancer in front of the vRealize Operations Manager cluster:

- Utilizing a load balancer ensures that the deployed cluster is properly balanced for performance of UI traffic.
- Allows all nodes in the cluster to equally participate in the handling of UI sessions and traffic.
- Provides high availability if any admin or data node fails, by directing UI traffic only to serving nodes in the cluster.
- Provides simpler access for the users. Instead of accessing each node individually the user only needs one URL to access the entire cluster and not be concerned with which node is available.
- Provides load balancing, high availability and ease of configuration for the End Point Operations (EPOps) agents.

Selecting a Load Balancer

There are no specific requirements for selecting a load balancer platform for vRealize Operations Manager. Majority of Load Balancers available today support complex web servers and SSL. You can use a load balancer in front of a vRealize Operations Manager cluster as long as certain parameters and configuration variables are followed. HAProxy was chosen for this example due to its ease of deployment, open source availability, stability, capability handling SSL sessions, and performance. Following are some of the parameters that should be considered for configuring other brands of load balancers:

- You must use TCP Mode, HTTP mode is not supported.
- It is not recommended to use round-robin balancing mode
- Cookie persistence does not work
- SSL pass-through is used, SSL termination is not supported
- Hash type balancing is recommended to ensure that the same client IP address always reaches the same node, if the node is available

- Health checks should be performed for at least 3 pages presented in the UI

How to Handle SSL UI Certificates with a Load Balancer

In all the default installations of vRealize Operations Manager nodes a default self-signed VMware certificate is included. You can implement your own SSL certificate from an internal Certificate Authority or external Certificate Authority. For more information on the certificate installation procedures, see Requirements for Custom vRealize Operations Manager SSL Certificates.

In addition to these configuration variables it is important to understand how SSL certificates are distributed in a cluster. If you upload a certificate to a node in the cluster, for example: the master node, the certificate will then be pushed to all nodes in the cluster. To handle UI sessions by all the nodes in the cluster you must upload an SSL certificate that contains all of the DNS names (optional: IP addresses and DNS names) in the **Subject Alternative Name** field of the uploaded certificate. The common name should be the Load Balancer DNS name. The subject alternative names are used to support access to the admin UI page.

When the certificate is uploaded to the master node, it is pushed to all the nodes in the cluster. Currently, when you use a load balancer with vRealize Operations Manager, the only supported method is SSL pass-through, which means the SSL certificate cannot be terminated on the load balancer.

To change SSL certificate on a cluster deployment:

1. Log in to the master node by using the following link: https://<ipaddress>/admin.
2. On the top right side, click the certificate button to change the certificate.
3. Upload your PEM file and store it on the local node: /data/vcops/user/conf/ssl/uploaded_cert.pem
4. Copy the PEM file to all the nodes.
5. Unpack the PEM file contents on each node.
6. Activate the new certificates by changing some symbolic links and restart the web server (apache httpd) on each node in the cluster.

When you view the certificate on the node that you are accessing, you will see all nodes in the cluster listed in the certificate SAN.

vRealize Operations Manager Overview

The vRealize Operations Manager clusters consist of a master node, an optional replica node for high availability, optional data nodes, and optional remote collector nodes. You can access and interact with the product by using the product UI available on the master and data nodes. The remote collector nodes do not contain a product UI and are used for data collection only. The product UI is powered by a Tomcat instance that resides across each node, but is not load balanced out of the box. You can scale up vRealize Operations Manager environment by adding nodes when the environment grows larger.

vRealize Operations Manager supports high availability by enabling a replica node for the vRealize Operations Manager master node. A high availability replica node can take over the functions that a master node provides. When a problem occurs with the master node, fail-over to the replica node is automatic and requires only 2 to 3 minutes of vRealize Operations Manager downtime. Data stored on the master node is always backed up on the replica node. In addition, with high availability enabled, the cluster can survive the loss of a data node without losing any data.

NODE ROLE	FUNCTIONS
Master Node	It is the initial, required node in the cluster. All other nodes are managed by the master node. It contains the product UI. In a single-node installation, the master node performs data collection and analysis as it is the only node where vRealize Operations Manager adapters are installed.

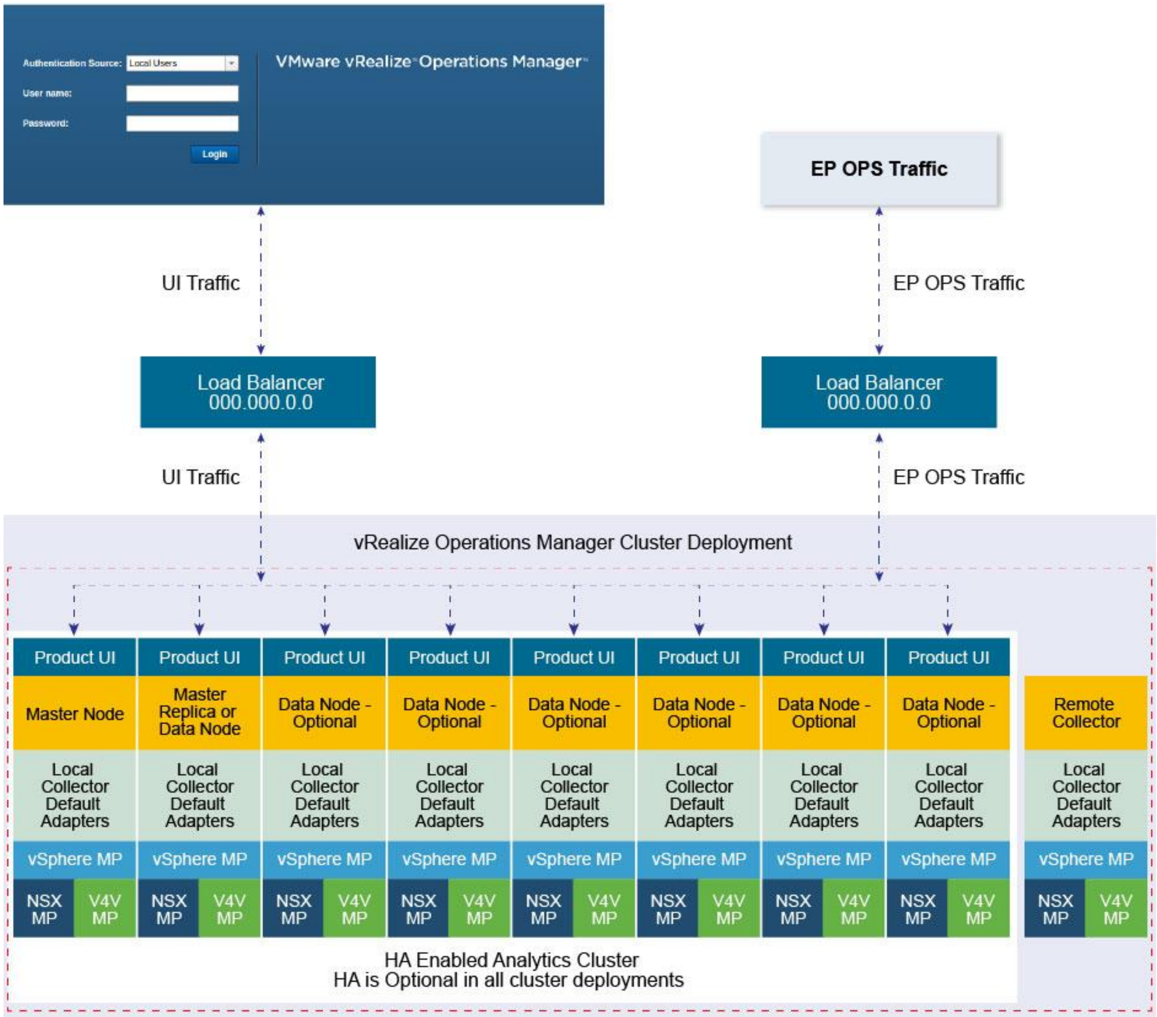
Data Node	In larger deployments, only data nodes have adapters installed to perform collection and analysis. It contains the product UI.
Replica Node	To enable high availability, the cluster requires that you convert a data node in to a replica of the master node. It does not contain product UI.

vRealize Operations Manager Architecture

Currently, the vRealize Operations Manager 6.0 release supports the maximum of 8-nodes in the analytics cluster. Remote collectors are not considered part of the analytics clusters as they do not participate in any type of data calculations or processing. EPOps traffic is load balanced to the same cluster.

NOTE: The load balancer cannot decrypt the traffic, hence cannot differentiate between EPOps and analytics traffic.

Following is a basic architecture overview of a vRealize Operations Manager 8-node cluster with high availability enabled.



Authentication Source: Local Users

User name:
Password:
Login
VMware vRealize Operations Manager
EP OPS Traffic
UI Traffic
EP OPS Traffic
Load Balancer 000.000.0.0
Load Balancer 000.000.0.0
UI Traffic
EP OPS Traffic
vRealize Operations Manager Cluster Deployment
HA Enabled Analytics Cluster
HA is Optional in all cluster deployments

Product UI	Product UI	Product UI	Product UI	Product UI	Product UI	Product UI	Product UI	
Master Node	Master Replica or Data Node	Data Node - Optional	Data Node - Optional	Data Node - Optional	Data Node - Optional	Data Node - Optional	Data Node - Optional	Remote Collector
Local Collector Default Adapters	Local Collector Default Adapters	Local Collector Default Adapters	Local Collector Default Adapters	Local Collector Default Adapters	Local Collector Default Adapters	Local Collector Default Adapters	Local Collector Default Adapters	Local Collector Default Adapters
vSphere MP	vSphere MP	vSphere MP	vSphere MP	vSphere MP	vSphere MP	vSphere MP	vSphere MP	vSphere MP
NSX MP V4V MP	NSX MP V4V MP	NSX MP V4V MP	NSX MP V4V MP	NSX MP V4V MP	NSX MP V4V MP	NSX MP V4V MP	NSX MP V4V MP	NSX MP V4V MP

FIGURE 1. VREALIZE OPERATIONS MANAGER 8-NODES CLUSTER WITH HIGH AVAILABILITY

Configuring End Point Operations Agents

End Point Operations agents are used to gather operating system metrics to monitor availability of remote platforms and applications. This metrics are sent to the vRealize Operations Manager server. You can configure additional load balancers to separate analytics traffic from EPOps traffic.

The steps to configure EPOps load balancer are described as required throughout this document.

You must shut down that the load balancer while upgrading or shutting down vRealize Operations Manager cluster. The load balancer should be restarted after the cluster is upgraded.

In the case of EPOps balancing, the overall latency between agent, load balancer, and cluster should be lower than 20 millisecond. If the latency is higher, you must install a remote collector and direct the agents directly to it.

HAProxy Installation and Configuration

HAProxy offers high availability, load balancing, and proxying for TCP and HTTP-based applications.

Prerequisites

Following are the prerequisites to ensure a functional load balancer configuration and deployment.

- Fully Patched CentOS or Redhat Linux VM
- CPU: 2 or 4 vCPU
- Memory: 4GB
- Disk space: 50GB
- HAProxy 1.5.x

  NOTE: HAProxy 1.6 is supported, however it may require some changes that are out of scope for this document.
- Fully functioning DNS with both forward and reverse lookups
- All nodes in the vRealize Operations Manager cluster operating correctly
- HAProxy deployed in same datacenter and preferably on the same cluster as vRealize Operations Manager
- HAProxy deployed on same subnet, also known as a one arm configuration, as vRealize Operations Manager cluster
  NOTE: Multiple subnet deployment has not been tested.
- HAProxy not deployed on the same ESX hosts as vRealize Operations Manager cluster to ensure availability
- Minimum 2-node deployment of vRealize Operations Manager cluster
- Deployment does not require high availability to be enabled, but it is recommended that you enable high availability
- One master node and at least one data node is required for using a load balancer beneficially

Installation and Configuration of Single-Node HAProxy on CentOS 6.5 or RHEL

A single-node HAProxy deployment is the basic model for majority of environments that require the use of a proxy server in front of vRealize Operations Manager cluster. For installing a single-node HAProxy deployment on single-node of CentOS, you must complete the following steps:

1. Perform a package update on the system to ensure all the packages are up-to-date:

   ```
   yum update (update all packages)
   ```

2. Verify that the system Hostname is valid:

   ```
   view /etc/sysconfig/network
   ```

3. Verify the network settings for the primary network interface:

   ```
   view /etc/sysconfig/network-scripts/ifcfg-eth0
   ```

4. If the VM is cloned, ensure to clean the old persistent rules:

   ```
   /etc/udev/rules.d/70-persistent-net.rules
   ```

5. Restart network service to make any additional changes on network settings:

   ```
   service network restart
   ```

6. Download the HAProxy:

   ```
   yum install wget
   wget http://www.haproxy.org/download/1.5/src/haproxy-1.5.11.tar.gz
   ```

7. Install core build essentials for building and compiling HAProxy:

   ```
   yum install build-essential openssl-devel make gcc-c++ gcc zlib-devel
   ```

8. Unzip HAProxy:

   ```
   cd
   ```

9. Change directories to HAProxy extract location:

   ```
   cd extracted directory
   ```

10. Compile HAProxy:

    ```
    make TARGET=linux26 USE_OPENSSL=1 USE_ZLIB=1
    ```

11. (Optional) Add prefix for make install command if you want to install into a custom directory:

    ```
    make PREFIX=/apps/opt/haproxy-ssl install
    ```

12. Install the binary:

    ```
    make install
    ```

13. Create directory for configuration and executables:

    ```
    mkdir /etc/haproxy
    ```

14. Move the initialization script example into startup directory:

    ```
    cp ./examples/haproxy.init /etc/init.d/haproxy
    ```

15. Create the HAProxy configuration file:

    ```
    touch /etc/haproxy/haproxy.cfg instead of:
    vi /etc/haproxy/haproxy.cfg
    :wq
    ```

16. Insert the HAProxy config and edit server lines with IP addresses of all nodes in the cluster:

    ```
    vi /etc/haproxy/haproxy.cfg
    :wq
    ```

17. Edit the initialization script to adjust installation location of the binary files as needed. For example, by default the file uses /usr/sbin/haproxy but in most of the cases it uses /usr/local/sbin/haproxy.

    ```
    vi /etc/init.d/haproxy
    wq
    ```

18. Change the ownership of the initialization script for correct access:

    ```
    chmod 755 /etc/init.d/haproxy
    ```

19. Add the haproxy user:

    ```
    useradd haproxy
    ```

20. Start the HAProxy Service:

    ```
    service haproxy start
    ```

21. Configure HAProxy to start on reboot of server:

    ```
    chkconfig haproxy on
    ```

Install Single-Node HAProxy on CentOS 7.0

HAProxy is also supported on CentOS 7.0 and can be obtained from yum repository already compiled or compile as shown in the Installation and Configuration of Single-Node HAProxy on CentOS 6.5 section. To install HAProxy on CentOS 7 by using yum package manager, which can then be used to configure the instance using the same configuration, complete the following steps:

1. Perform a package update on system to ensure all packages are up-to-date:

   ```
   yum update (update all packages)
   ```

2. Install HAProxy:

   ```
   yum -y install haproxy
   ```

3. Copy original HAProxy configuration to backup file:

   ```
   cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bak
   ```

4. Configure HAProxy configuration. To configure analytics balancer, see Configure HAProxy Analytics and to configure EPOps balancer, see Configure EPOps HAProxy.

5. Allow firewall traffic through for the ports needed for HAProxy to function:

   ```
   firewall-cmd --permanent --zone=public --add-port=80/tcp
   firewall-cmd --permanent --zone=public --add-port=9090/tcp
   firewall-cmd --permanent --zone=public --add-port=443/tcp
   ```

6. Reload the firewall configuration:

   ```
   systemctl reload firewalld
   ```

7. Enable HAProxy to connect to any interface:

   ```
   setsebool -P haproxy_connect_any 1
   ```

8. Enable HAProxy service:

   ```
   systemctl enable haproxy
   ```

Configure Logging for HAProxy

An administrator might want to configure logging of the HAProxy service to aid in monitoring and troubleshooting an environment. The HAProxy logger allows for the use rsyslog internally on the Linux installation to log to a local file. You can also utilize Log Insight integration to send this log to a Log Insight deployment by utilizing the new Log Insight Linux agent to greatly simplify the configuration and logging of Linux platforms. To configure basic applications logging using rsyslog locally on the server perform the following steps.

1. Configure the rsyslog configuration file to accept UDP syslog reception:

   ```
   vi /etc/rsyslog.conf
   ```

2. Uncomment the following lines:

   ```
   # Provides UDP syslog reception
   $ModLoad imudp
   $UDPServerAddress 127.0.0.1
   $UDPServerRun 514
   ```

3. Save the file:

   ```
   wq!
   ```

4. Create the HAProxy logging configuration file for specific application parameters

   ```
   vi /etc/rsyslog.d/haproxy.conf
   ```

5. Add the following line:

   ```
   if ($programname == 'haproxy') then -/var/log/haproxy.log
   ```

6. Save the file:

   ```
   wq!
   ```

7. Create HAProxy Log file and set proper permissions:

   ```
   touch /var/log/haproxy.log
   chmod 755 /var/log/haproxy.log
   ```

8. Restart the rsyslog service:

   ```
   Service rsyslog restart
   ```

Configure HAProxy

The HAProxy configuration has been tested against an 8-node vRealize Operations Manager cluster. Clusters with fewer nodes are also supported and require the same configuration. Every time the cluster is expanded and a new node is deployed you must edit the HAProxy configuration and add the IP address of the new node. After editing the configuration file the HAProxy service should always be restarted so the configuration is reloaded.

Configure HAProxy for vRealize Operations Manager Analytics

You can configure the HAProxy for vRealize Operations Manager analytics as follows:

```
# Configuration file to balance both web and epops
#global parameters
global

    log         127.0.0.1 local2
    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    maxconn     400
    user        haproxy
    group       haproxy
    daemon
    stats socket /var/lib/haproxy/stats
    ssl-server-verify none

#default parameters unless otherwise specified
defaults

    log global
    mode http
    option httplog
    option tcplog
    option dontlognull
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms

#listener settings for stats webpage can be optional but highly recommended
```

```
 listen stats :9090

    balance
    mode http
    stats enable
    stats auth admin:admin
    stats uri /
    stats realm Haproxy\ Statistics

#automatic redirect for http to https connections

    frontend vrops_unsecured_redirect *:80

       redirect location https://<insert_fqdn_address_here>

#front settings in this case we bind to all addresses on system or specify an interface

    frontend vrops_frontend_secure

       bind <web dedicated ip>:443
       mode tcp
       option tcplog
       default_backend vrops_backend_secure

#backend configuration of receiving servers containing tcp-checks health checks and
hashing

#needed for a proper configuration and page sessions

#adjust the server parameters to your environment

    backend vrops_backend_secure

       mode tcp
       option tcplog

    balance source
    hash-type consistent
    option tcp-check
    tcp-check connect port 443 ssl
    tcp-check send GET\ /suite-api/api/deployment/node/status\ HTTP/1.0\r\n\r\n
    tcp-check expect rstring ONLINE

server node1  <Insert node1 ip address here>:443 check inter 60s check-ssl maxconn 140
fall 6 rise 6

server node2  <Insert node2 ip address here>:443 check inter 60s check-ssl maxconn 140
fall 6 rise 6

server node3  <Insert node3 ip address here>:443 check inter 60s check-ssl maxconn 140
fall 6 rise 6

server node4  <Insert node4 ip address here>:443 check inter 60s check-ssl maxconn 140
fall 6 rise 6
```

NOTE: HAProxy 1.6 introduced strict checking of the configuration file, if you want to use HAProxy 1.6 you would have to make some changes to support the new strict validation, such as BIND address. For example, you can use:

```
bind <web dedicated ip>:443
```

Configure EPOps HAProxy

You can configure EPOps HAProxy as follows:

```
# EPOPS Load Balancer configuration.
#global parameters
global
    log         127.0.0.1 local2
    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    maxconn     2000
    user        haproxy
    group       haproxy
    daemon
    stats socket /var/lib/haproxy/stats
    ssl-server-verify none
#default parameters unless otherwise specified
defaults
    log global
    mode http
    option httplog
    option tcplog
    option dontlognull
    timeout connect 5000ms
    timeout client  50000ms
    timeout server  50000ms
#listener settings for stats webpage can be optional but highly recommended
    listen stats :9090
    balance
    mode http
    stats enable
    stats auth admin:admin
    stats uri /
    stats realm Haproxy\ Statistics

#automatic redirect for http to https connections

    frontend vrops_unsecured_redirect *:80

    redirect location <Insert https fqdn here >

    frontend epops_frontend_secure

    bind <epops dedicated ip>:443

    mode tcp

    option tcplog

    use_backend epops_backend_secure

    #adjust the server parameters to your environment

    backend epops_backend_secure

    mode tcp

    option tcplog

    balance source

    hash-type consistent

    option tcp-check

    timeout queue 20s

    tcp-check connect port 443 ssl

    tcp-check send GET\ /epops-webapp/health-check\ HTTP/1.0\r\n

    tcp-check send \r\n

    tcp-check expect string ONLINE

server node1  <Insert node1 ip address here>:443 check inter 60s check-ssl maxconn 140
fall 6 rise 6

server node2  <Insert node2 ip address here>:443 check inter 60s check-ssl maxconn 140
fall 6 rise 6

server node3  <Insert node3 ip address here>:443 check inter 60s check-ssl maxconn 140
fall 6 rise 6

server node4  <Insert node4 ip address here>:443 check inter 60s check-ssl maxconn 140
fall 6 rise 6
```

NOTE: The line “listen stats :9090” configures the status listener of HAProxy.

Verify HAProxy Configuration

1. When the configuration is completed, connect to http://haproxy_ip_address:9090 by using the username and password used to configure HAProxy. In the above example, username: admin and password: admin.

2. Verify that all the nodes rows are shown in green.

Advanced Cofiguration: HAProxy with Keepalived

In some circumstances and deployments, dual highly available HAProxy is required. In a single-node deployment HAProxy becomes the single point of failure in the deployment and adds potential reliability concerns. Also, if the HAProxy needs patches, updates, or other maintenance, the HAProxy becomes a single point of downtime. To remediate this concern, deployment of two HAProxys and Keepalived is used to ensure one node is always available. The configuration of the HAProxy can be exactly same across nodes, simply adjusting for local node IP addresses. In most cases the first deployed HAProxy virtual machine can simply be cloned and used as the secondary node.

Failover of a failed HAProxy node by using Keepalived has been tested to occur in less than 5 seconds depending on the network variables. The failover period was rarely noticed by the user or effecting the UI session, during the limited testing. Keepalived uses Linux Virtual Router Redundancy Protocol (VRRP) and multicast advertisements from the master node. If the master node stops sending advertisements the backup proceeds to send a gratuitous ARP to the network and taking ownership of the VIP address and owns the hardware address that master previously owned. The master and the backup monitor each other with multicast events at a rate of once per second.

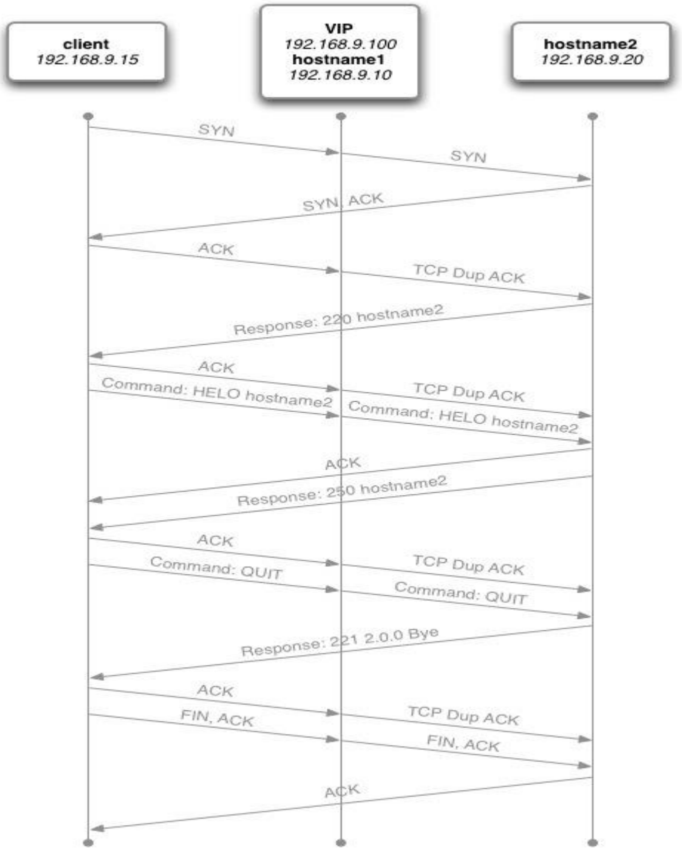


Figure 2. HAProxy with Keepalived

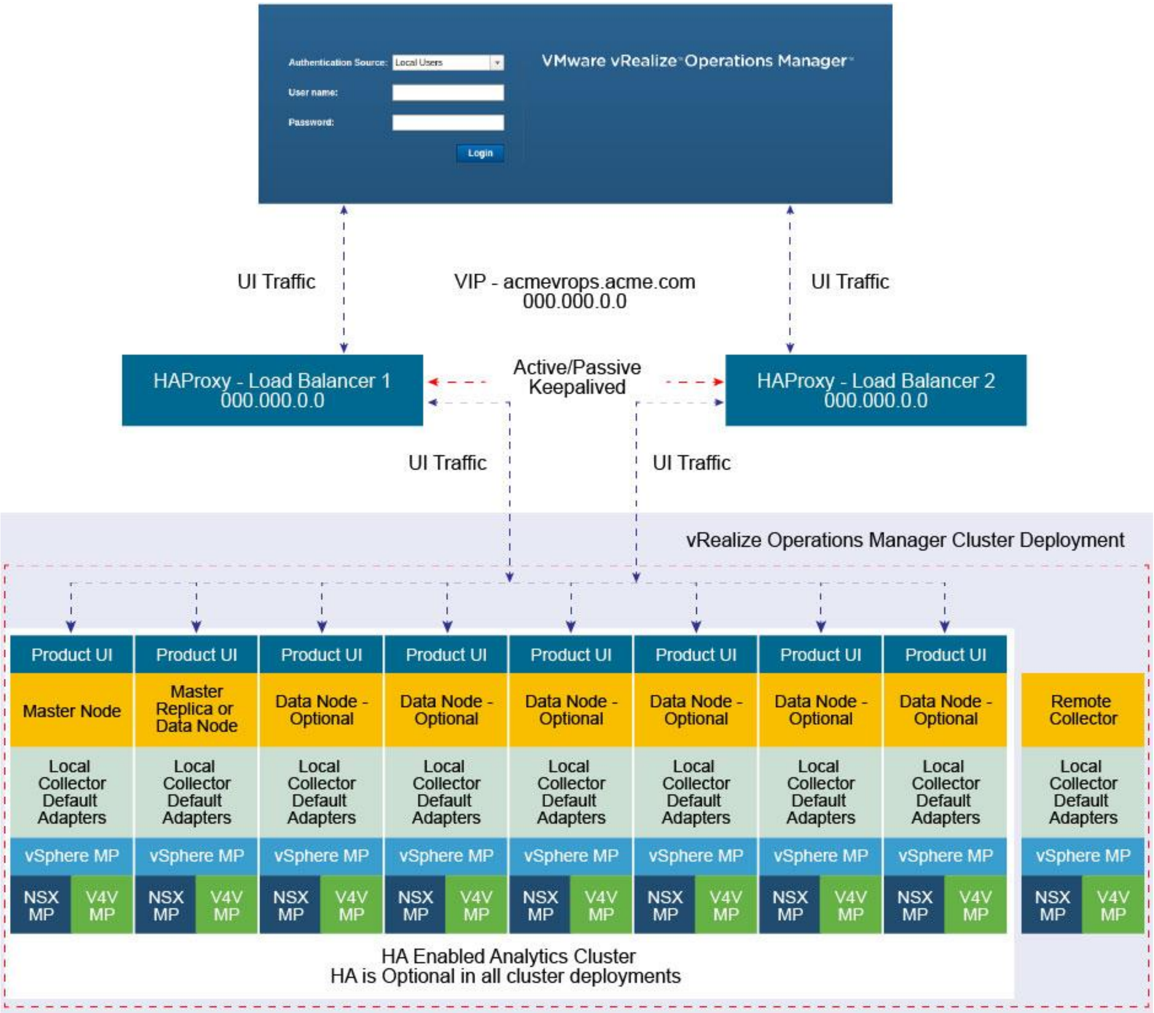

Authentication Source:
Local Users
User name:
Password:
Login
VMware vRealize Operations Manager
UI Traffic
VIP - acmevrops.acme.com
000.000.0.0
UI Traffic
HAProxy - Load Balancer 1
000.000.0.0
Active/Passive
Keepalived
HAProxy - Load Balancer 2
000.000.0.0
UI Traffic
UI Traffic
vRealize Operations Manager Cluster Deployment
Product UI
Master Node
Master Replica or Data Node
Data Node - Optional
Remote Collector
Local Collector Default Adapters
vSphere MP
NSX MP
V4V MP
HA Enabled Analytics Cluster
HA is Optional in all cluster deployments

FIGURE 3. VREALIZE OPERATIONS MANAGER 8-NODES CLUSTER USING HAPROXY WITH KEEPALIVED

Configure HAProxy with Keepalived

1. Clone the HAProxy VM or install a new VM with the same configuration as the first deployed HAProxy.
2. Change Hostname and IP Address
3. Create VIP and point to main DNS record for vRealize Operations Manager cluster. For example: acmevrops6.acme.com / 192.168.1.5)

   You will now have 2x HAProxy load balancers running. For example: LB1/192.168.1.6 and LB2/192.168.1.7.
4. Verify HAProxy configuration is located on both the load balancers. You should be able to access either one and access vRealize Operations Manager cluster successfully.

   When both the HAProxies are confirmed working and contain identical configurations, you should configure the Keepalived to ensure that you have availability between the two load balancers.
5. SSH to LB1 which we will consider is the MASTER election.

   ```
   yum install keepalived
   ```
6. You should configure the kernel to use a VIP to bind to vi /etc/sysctl.conf. Add the following line to the file

```
net.ipv4.ip_nonlocal_bind=1
```

7. In order for the kernel to pick up the new changes without rebooting, run the following command:

   ```
   sysctl -p
   ```

8. Delete the file:

   ```
   /etc/keepalived/keepalived.conf
   ```

9. Create a new file:

   ```
   /etc/keepalived/keepalived.conf
   ```

10. In the new keepalived.conf file add the following

    ```
    Master Node

    global_defs {

      router_id haproxy2 # The hostname of this host.

    }

    vrrp_script haproxy {

      script "killall -0 haproxy"

      interval 2

      weight 2

    }

    vrrp_instance 50 {

      virtual_router_id 50

      advert_int 1

      priority 50

      state MASTER

      interface eth0

      virtual_ipaddress {

         Virtual_IPaddress dev eth0  # The virtual IP address that will be shared betwee
    MASTER and BACKUP

      }

      track_script {

          haproxy

      }

    }
    ```

11. Verify that above the Router_ID is the HOSTNAME of the local load balancer that you are setting up.

12. Verify that you have set up the correct network device, check if you are using eth0.
13. Verify that above the Virtual_IPaddress is the VIP address, and not the local IP address of the LB1 node.
14. Set the priority in increments of 50. In this example, the node has the highest priority, so it is set to 100. Verify that the node is set as the master node.
15. Save the configuration file and restart the services.
16. You must enable the Keepalived service:

- For CentOS 7.x: systemctl enable keepalived
- For CentOS 6.x: chkconfig keepalived on

17. Run the commands:

```
service keepalived restart

service haproxy restart
```

18. To display if the node has the active load balancer IP, run:

```
ip a | grep eth0
```

19. If the system you are on displays the primary IP address of the load balancer, then this is the active system processing traffic. Verify that only one system displays the primary IP address of the load balancer.
20. If the address is present on both the machines, the configuration is incorrect and both the machines might not be able to communicate with each other.
21. To configure the second LB2 Keepalived service perform the same steps as above and configure Keepalived service on LB2.
22. In the new keepalived.conf file add the following for the slave node:

```
global_defs {
  router_id haproxy4 # The hostname of this host !
}
vrrp_script haproxy {
  script "killall -0 haproxy"
  interval 2
  weight 2
}
vrrp_instance 50 {
  virtual_router_id 50
  advert_int 1
  priority 50
  state BACKUP
  interface eth0
  virtual_ipaddress {
     Virtual_IPaddress dev eth0 # The virtual IP address that will be shared betwee
MASTER and BACKUP.

  }

  track_script {

    haproxy

  }

}
```

23. Verify that the Router_ID is the HOSTNAME of the local load balancer that you are setting up.
24. Verify that above the Virtual_IPaddress is the VIP address and not the local IP address of the LB1 node.
25. Set the priority in increments of 50. In this example, the node has the highest priority, so it is set to 100. Verify that the node is set as the backup.
26. Save the configuration file and restart the services.
27. You must enable the Keepalived service:

- For CentOS 7.x: systemctl enable keepalived
- For CentOS 6.x: chkconfig keepalived on

28. Run the commands:

```
service keepalived restart

service haproxy restart
```

29. To display if the node has the active load balancer IP, run:

```
ip a | grep eth0
```

30. If the system you are on displays the primary IP address of the load balancer, then this is the active system processing traffic

F5 Big IP Installation & Configuration

The F5 Big IP load balancer configuration is similar to the HAProxy configuration. The F5 uses the SSL pass-through in the same manner as the HAProxy configuration. The F5 configuration has been tested in a one-arm configuration, which means that the load balancer is on the same network as the pool members. Multi-arm configuration has not been tested.

Prerequisites

The following are the prerequisites for a functional F5 configuration in front of a vRealize Operations Manager cluster:

- This document assumes that an F5 device is already deployed in the environment and is configured with network connectivity to the deployed environment where the load balancer instance would be used and run from.
- The F5 can be either physical or virtual and can be deployed in one-arm or multi-arm topologies. Multi-arm configuration has not been tested.
- The Local Traffic Module (LTM) must be configured and licensed as Nominal, Minimum, or Dedicated. You can configure LTM on System > Resource Provisioning page.
- A vRealize Operations Manager cluster has been deployed in the environment and is fully functional and all nodes in the cluster are accepting UI traffic. This cluster might have high availability enabled but it is not a requirement.
- An additional VIP/Virtual Server IP address for vRealize Operations Manager analytics.
- An additional VIP/Virtual Server IP address for EPOps in case you are configuring separate load balancers for analytics and EPOps.

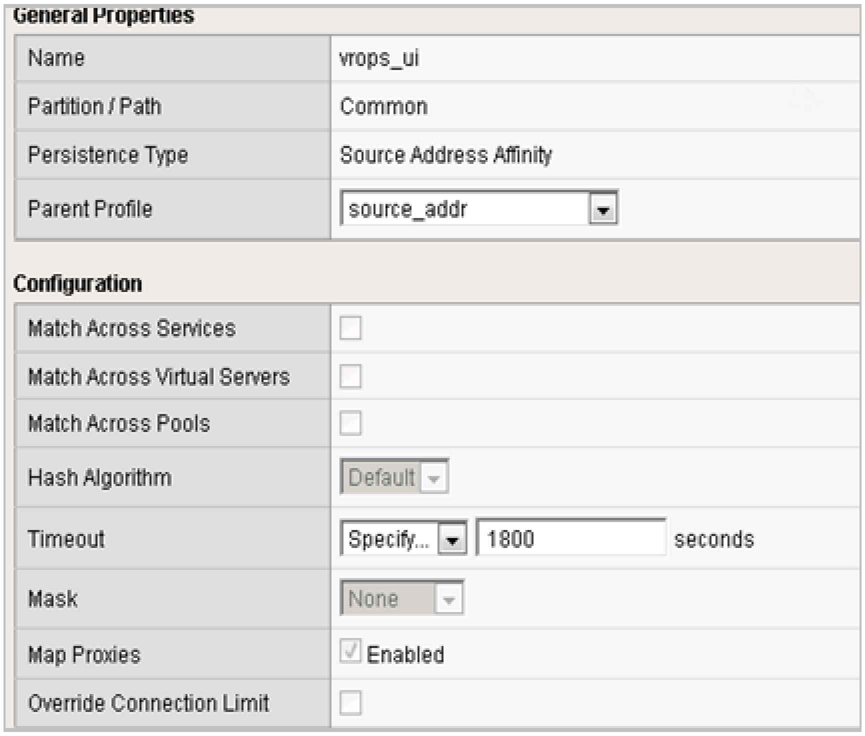
Configure Custom Persistence Profile

There are multiple possible profiles provided out of box in most F5 deployments and creating a custom persistence profile using source addresses affinity. You must create a customer persistence profile by using the following steps:

1. Log in to the F5 and select **Local Traffic** > **Profiles** > **Persistence**.
2. Click **Create**.
3. Enter the name **source_addr_vra** and select **Source Address Affinity** from the drop-down menu.
4. Enable **Custom** mode.
5. Set the **Timeout** to **1800 seconds** (**30 minutes**).
6. Click **Finished**.

NOTE: The timeout of the vRealize Operations Manager user sessions, configured through the Global Settings page is 30 minutes is, consistent with vRealize Operations Manager configuration. If the timeout value is updated for vRealize Operations Manager, it should be updated for F5 too.

Example for vRealize Operations Manager analytics configuration:



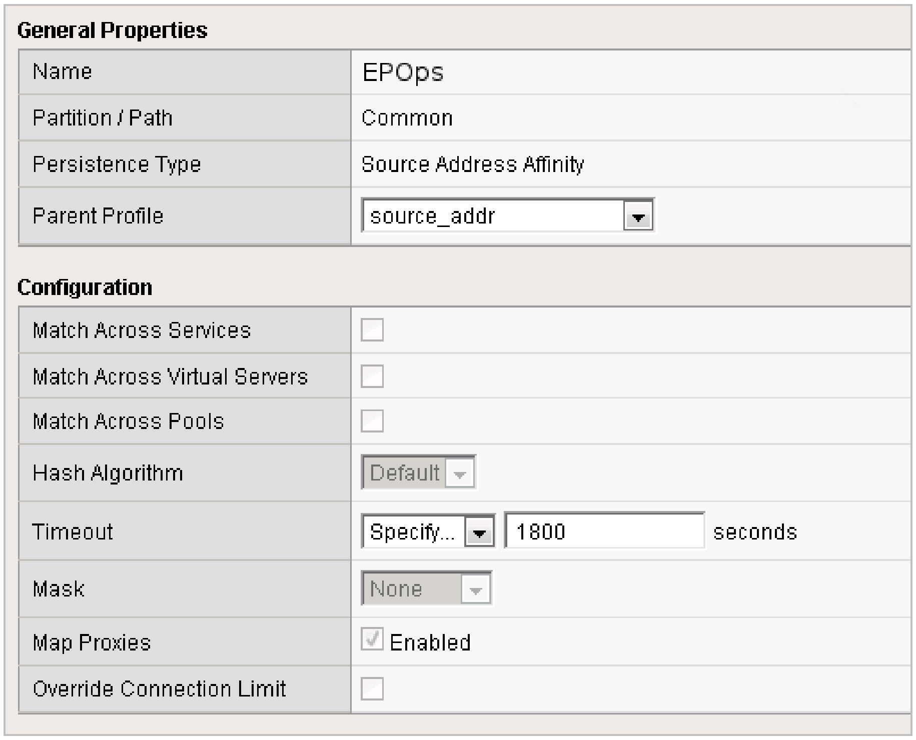
General Properties

Name	vrops_ui
Partition / Path	Common
Persistence Type	Source Address Affinity
Parent Profile	source_addr

Configuration

Match Across Services	□
Match Across Virtual Servers	□
Match Across Pools	□
Hash Algorithm	Default
Timeout	Specify... 1800 seconds
Mask	None
Map Proxies	☑ Enabled
Override Connection Limit	□

Example for EPOps configuration:



General Properties

Name	EPOps
Partition / Path	Common
Persistence Type	Source Address Affinity
Parent Profile	source_addr

Configuration

Match Across Services	□
Match Across Virtual Servers	□
Match Across Pools	□
Hash Algorithm	Default
Timeout	Specify... 1800 seconds
Mask	None
Map Proxies	☑ Enabled
Override Connection Limit	□

Configure Health Monitors

Health monitors are required to ensure the F5 has the proper endpoints on the vRealize Operations Manager node to test to make sure the node is available and functioning for clients to access the node. In this case, create a few Health Monitors to ensure all URLs are checked properly for availability.

1. Log in to the F5 and from the main menu select **Local Traffic** > **Monitors**.
2. Click **Create** and provide the required information as shown in the following tables. Leave the default when nothing is specified.

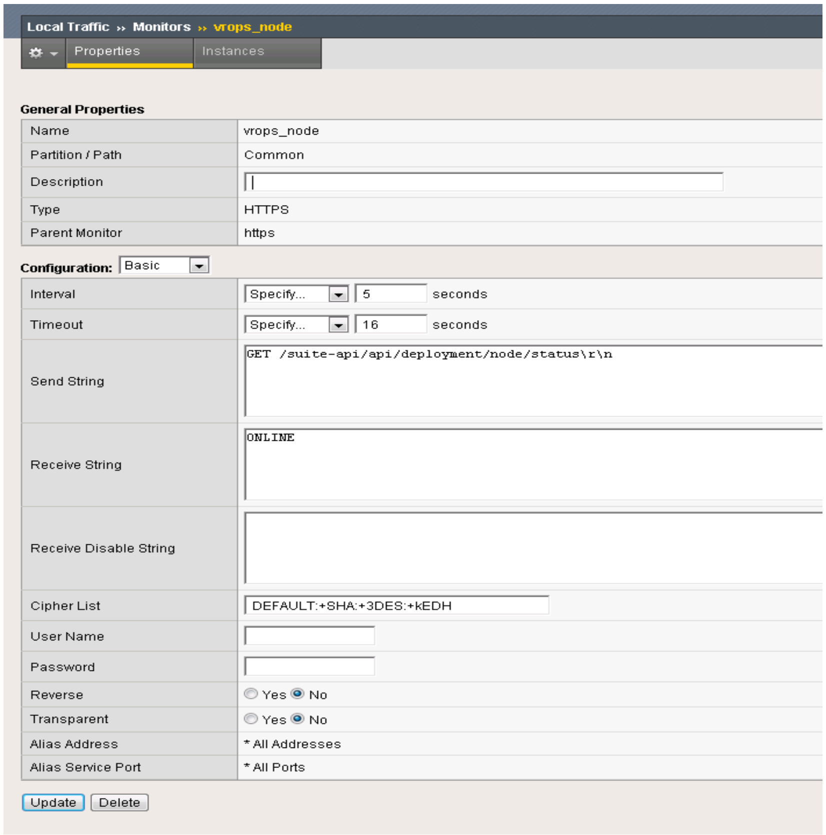
vRealize Operations Manager Analytics configuration:

NAME	TYPE	INTERVAL	TIMEOUT	SEND STRING	RECEIVE STRING	DESCRIPTION
vrops_http	http	60	8	GET HTTP/1.0\r\n\r\n	(2..|3..)	Default HTTP monitor to ensure the HTTP redirect page is accessible
vrops_https1	https	60	8	GET /suite-api/api/deployment/node/status\r\n	ONLINE	Default HTTPS monitor to ensure the HTTPS page is accessible

EPOPS configuration:

NAME	TYPE	INTERVAL	TIMEOUT	SEND STRING	RECEIVE STRING	DESCRIPTION
vrops_epops	https	60	8	GET /epops-webapp/health-check HTTP/1.0\r\n	ONLINE	Heartbeat page used to monitor the epops health

Example:



General Properties

Name	vrops_node
Partition / Path	Common
Description	
Type	HTTPS
Parent Monitor	https

Configuration: Basic

Interval	Specify... 5 seconds
Timeout	Specify... 16 seconds
Send String	GET /suite-api/api/deployment/node/status\r\n
Receive String	ONLINE
Receive Disable String	
Cipher List	DEFAULT:+SHA:+3DES:+kEDH
User Name	
Password	
Reverse	○ Yes ● No
Transparent	○ Yes ● No
Alias Address	* All Addresses
Alias Service Port	* All Ports

Update Delete

Configure Server Pools

Server Pools are used to contain the pools of members or nodes that will be receiving traffic. You will only need to create a single pool for a vRealize Operations Manager cluster with all nodes participating in the UI traffic as members. In most cases you will add each node in the cluster except for the remote collectors.

1. Log in to the F5 load balancer and select **Local Traffic** > **Pools**.
2. Click **Create** and provide the required information. Leave the default when nothing is specified.
3. Enter each pool member as a **New Node** and add it to the **New Members**.
4. Repeat steps 1, 2, and 3 for each row of information in the following table.
5. On the **Members** page, select the **Load Balancing Method** as the **Least Connections (node)** and **Priority Group Activation** as **Disabled**.

   vRealize Operations Manager Analytics configuration:

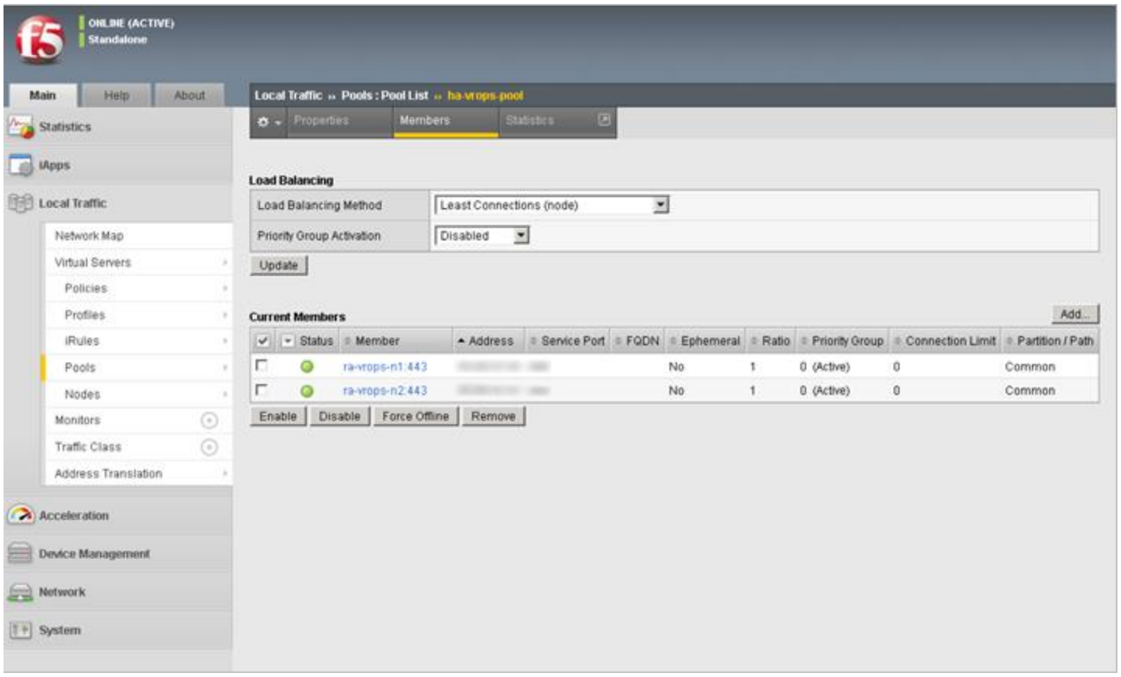
NAME	DESCRIPTION	HEALTH MONITORS	LOAD BALANCING METHOD	NODE NAME
ha-vrops-prod	vRealize Operations Manager Pool	vrops_http vrops_https1	Least Connections	vrops_node1:<ipaddress> vrops_node2:<ipaddress> vrops_node3:<ipaddress>

EPOps configuration:

NAME	DESCRIPTION	HEALTH MONITORS	LOAD BALANCING METHOD	NODE NAME
ha-epops-prod	vRealize Operations Manager Pool	vrops_epops	Least Connections	vrops_node1:<ipaddress> vrops_node2:<ipaddress> vrops_node3:<ipaddress>

NOTE: Ensure that you are using the correct service port: 443 for SSL.

Example:



Local Traffic » Pools : Pool List » ha-vrops-pool

Load Balancing

Load Balancing Method	Least Connections (node)
Priority Group Activation	Disabled

Current Members

☑	Status	Member	Address	Service Port	FQDN	Ephemeral	Ratio	Priority Group	Connection Limit	Partition / Path
□	●	ra-vrops-n1:443	[redacted]			No	1	0 (Active)	0	Common
□	●	ra-vrops-n2:443	[redacted]			No	1	0 (Active)	0	Common

Configure Virtual Servers

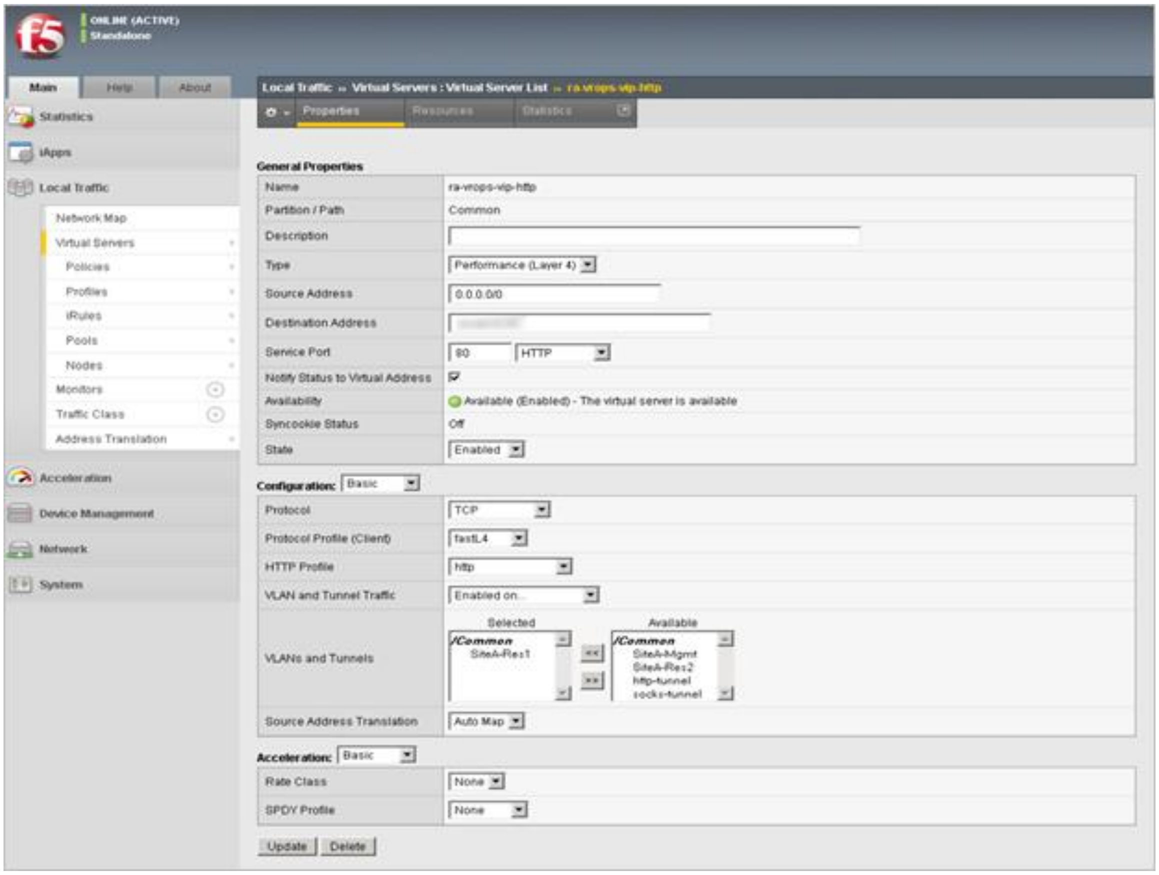
Virtual servers contain the virtual IP address (VIP) for the pools of nodes that will be accessed. In this case, there are two separate VIP's created with the same IP address. One virtual server will be for insecure traffic which will leverage a custom iRule to ensure the traffic gets redirected properly to the HTTPS session. The second virtual server will be used for secure traffic to ensure traffic will be sent directly to the secure HTTPS web page normally.

1. Log in to the F5 load balancer and select **Local Traffic** > **Virtual Servers**.
2. Click **Create** and provide the required information. Leave the default when nothing is specified.
3. When all the settings are configured, click **Update** to create the first virtual server.
4. Repeat the steps to configure the second virtual server by using the settings in the table below.

NAME	TYPE	DESTINATION ADDRESS	SERVICE PORT	SERVICE ADDRESS TRANSLATION	DEFAULT POOL	DEFAULT PERSISTENCE PROFILE	IRULES

ra-vrops-vip1	Performance (Layer 4)	<ipaddress>	80	Auto Map	ha-vrops-prod	ha-vrops-profile	_sys_https_redirect Error: 01070394:3: HTTP_REQUEST event in rule (/Common/_sys_https_redirect) requires an associated HTTP or FASTHTTP profile on the virtual server (/Common/ra-vrops-vip1).
ra-vrops-vip	Performance (Layer 4)	<ipaddress>	443	Auto Map	ha-vrops-prod	ha-vrops-profile	None
epops-vip	Performance (Layer 4)	<ipaddress>	443	Auto Map	ha-epops-prodha-vrops-profile	ha-vrops-profile	None

Example:



Local Traffic » Virtual Servers : Virtual Server List » ra-vrops-vip-http

General Properties

Name	ra-vrops-vip-http
Partition / Path	Common
Description	
Type	Performance (Layer 4)
Source Address	0.0.0.0/0
Destination Address	[redacted]
Service Port	80 HTTP
Notify Status to Virtual Address	☑
Availability	Available (Enabled) - The virtual server is available
Syncookie Status	Off
State	Enabled

Configuration: Basic

Protocol	TCP
Protocol Profile (Client)	fastL4
HTTP Profile	http
VLAN and Tunnel Traffic	Enabled on...
VLANs and Tunnels	Selected: /Common SiteA-Res1 Available: /Common SiteA-Mgmt SiteA-Res2 http-tunnel socks-tunnel
Source Address Translation	Auto Map

Acceleration: Basic

Rate Class	None
SPDY Profile	None

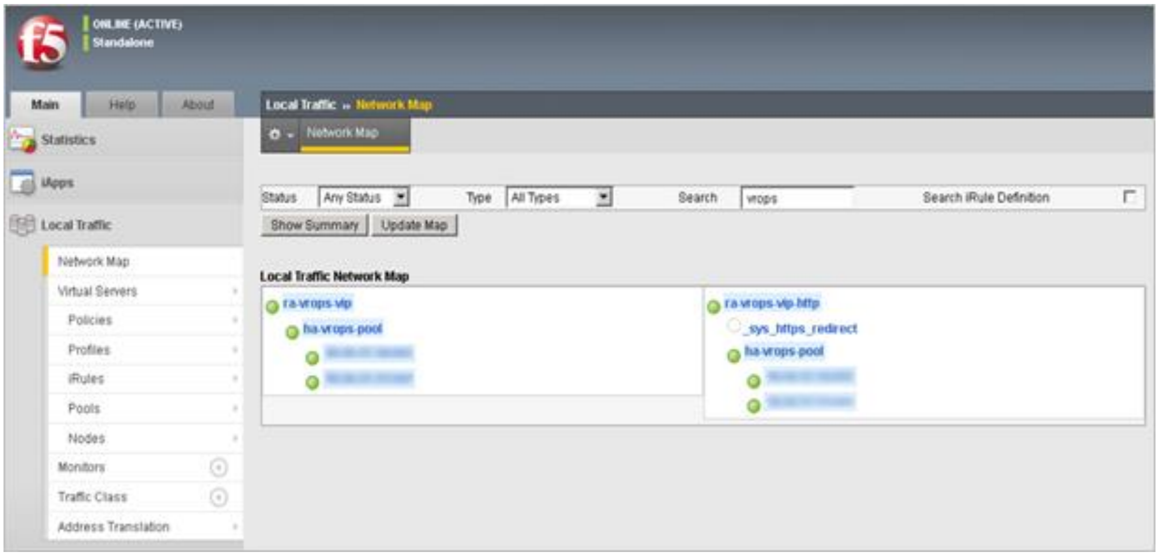
Verify Component and Pool Status

After you complete configuration for health monitors, server pools, and virtual servers, you should verify the status of the configured environment and filter to the specific deployment that was just configured to get an overall view of the nodes, pools, and virtual servers.

1. To check the network map for an overall view of the server pools, select **LTM > Network Map**.
2. Filter the **Network Map** by using the search box to enter the name of the virtual server name used in the configuration.
3. Each status indicator represents the status of the node, the pool, and virtual server or assigned VIP.

Example:

In the following example, you can see both the ra-vrops-vip and the ra-vrops-http VIP are functioning normally. When one of the nodes fail the indicator will turn red and the indicator for the pool will turn yellow to represent a failure in the pool.



Status: Any Status Type: All Types Search: vrops Search iRule Definition □

Local Traffic Network Map

ra-vrops-vip
ha-vrops-pool
[redacted]
[redacted]

ra-vrops-vip-http
_sys_https_redirect
ha-vrops-pool
[redacted]
[redacted]

NSX 6.2.0 Installation & Configuration

The NSX virtual networking solution includes the capability of deploying an Edge gateway as a load balancer. Currently, the NSX load balancer has basic load balancing functionality and it should not be considered a full-fledged load balancer with advanced configuration like F5.

NOTE: Use NSX version 6.1.3 and higher for all deployments as many issues with the load balancers have been resolved in this release.

Prerequisites

The following are the prerequisites for a functional NSX load balancer in front of a vRealize Operations Manager cluster:

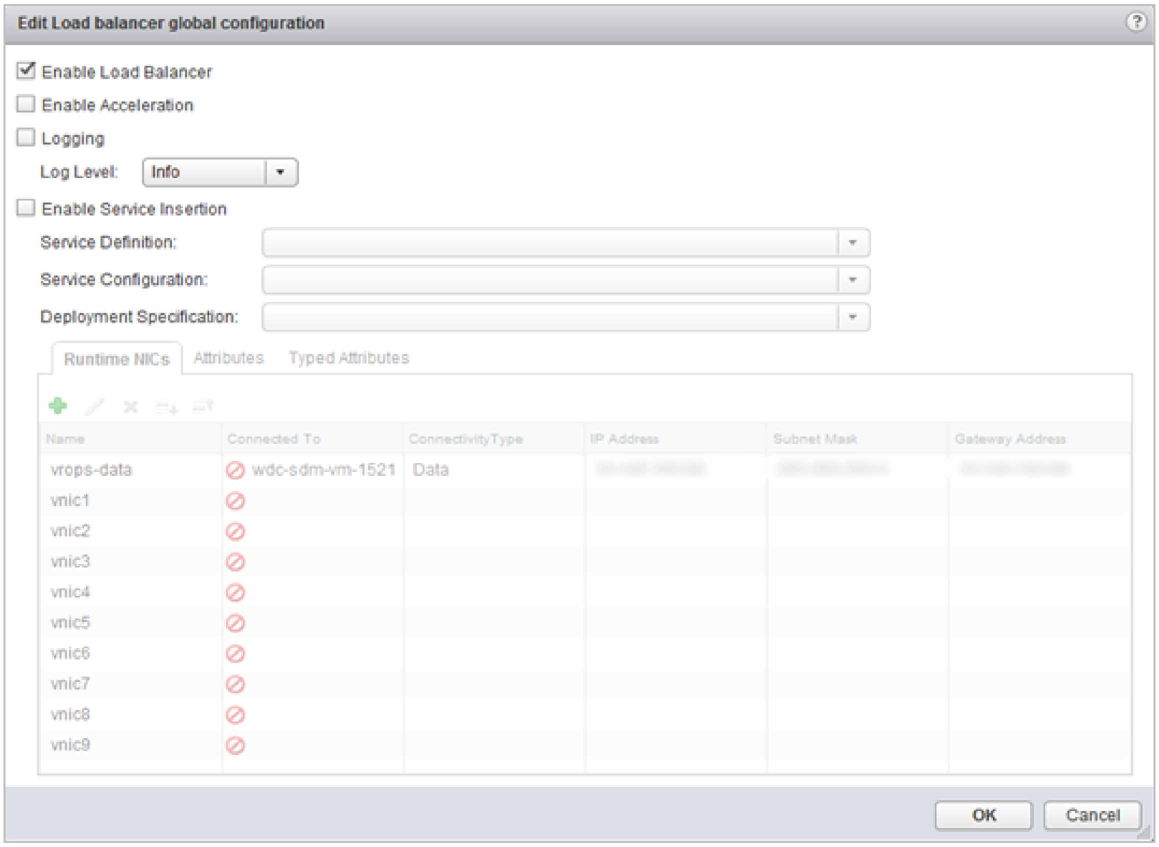
- This document assumes that NSX deployment is already deployed in the environment and is fully functional.
- The NSX deployment is of version 6.1.3 or higher.
- NSX Edge is deployed and has access to the network on which vRealize Operations Manager cluster is deployed.
- Edge can be enabled for high availability, however it is not a requirement
- Currently, there are 2 types of modes the load balancer can use: Accelerated and Non-Accelerated. Accelerated mode uses L4 and LVS and non-accelerated mode uses L7 which is essentially HAProxy.
- Do not configure the load balancer in the accelerated mode.

Install and Configure Edge for Load Balancing

You can specify global load balancer configuration parameters and configure the NSX Edge for load balancing by enabling the load balancer service.

1. Log in to the vSphere Web Client.
2. Click **Networking & Security** and then click **NSX Edges**.
3. Double-click an NSX Edge.
4. Click **Manage** and then click the **Load Balancer** tab.
5. Click **Edit** and select **Enable Load Balancer**.
6. Click **OK** to save changes and enable the service on the Edge.

Example from NSX 6.2.0:



Edit Load balancer global configuration

☑ Enable Load Balancer

□ Enable Acceleration

□ Logging

Log Level: Info

□ Enable Service Insertion

Service Definition:

Service Configuration:

Deployment Specification:

Name	Connected To	ConnectivityType	IP Address	Subnet Mask	Gateway Address
vrops-data	⊘ wdc-sdm-vm-1521	Data	[redacted]	[redacted]	[redacted]
vnic1	⊘				
vnic2	⊘				
vnic3	⊘				
vnic4	⊘				
vnic5	⊘				
vnic6	⊘				
vnic7	⊘				
vnic8	⊘				
vnic9	⊘				

Configure Application Profiles

You must create an application profile to define the behavior of a particular type of network traffic. After configuring a profile, you should associate the profile with a virtual server. The virtual server then processes traffic according to the values specified in the profile. Using profiles enhances your control over managing network traffic, and makes traffic-management tasks easier and more efficient.

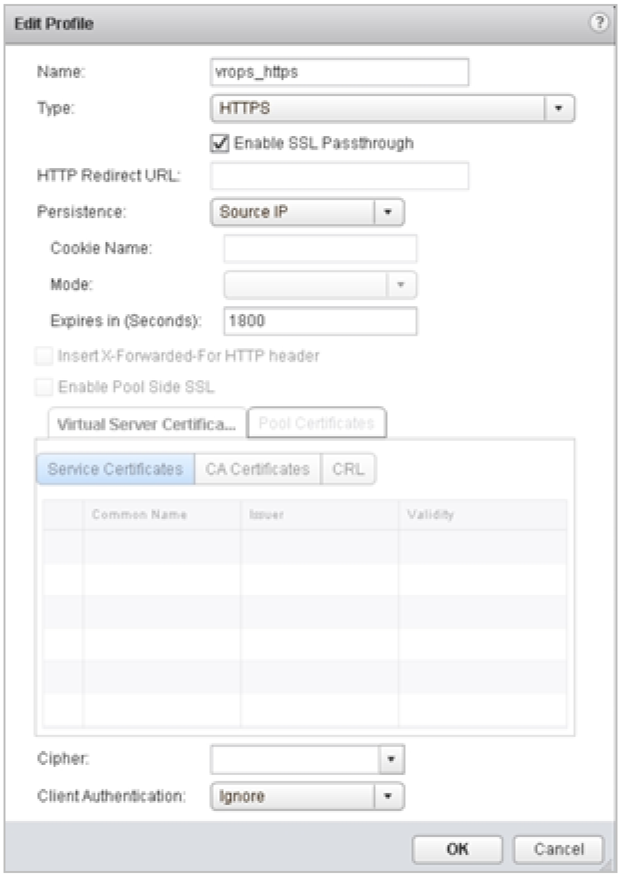
1. Log in to the vSphere Web Client.
2. Click **Networking & Security** and then click **NSX Edges**.
3. Double-click an NSX Edge.
4. Click **Manage** and then click the **Load Balancer** tab.
5. In the left navigation panel, click Application Profiles.
6. Click the Add (⊕) icon.
7. Enter a name for the profile and select the traffic type for which you are creating the profile. For example: vrops_https.
8. Select the **Type**: HTTPS
9. Select **Enable SSL Passthrough**.
10. Select **Persistence** as **Source IP**.
11. Enter **1800** for **Expires in (seconds)**.

12. Select **Ignore** for **Client Authentication**.

13. Click **OK** to save the Profile

NOTE: When the encrypted traffic is balanced, the load balancer cannot differentiate between the traffic for vRealize Operations Manager analytics and EPOps. If you plan to use two load balancers, one for vRealize Operations Manager analytics and one for EPOps, you could use the same profile as both the profiles are identical. If you create two different profiles, only the name of the profiles is different, but the configurations for both the profiles are identical.

Example:



Edit Profile

Name: vrops_https

Type: HTTPS

☑ Enable SSL Passthrough

HTTP Redirect URL:

Persistence: Source IP

Cookie Name:

Mode:

Expires in (Seconds): 1800

□ Insert X-Forwarded-For HTTP header

□ Enable Pool Side SSL

Cipher:

Client Authentication: Ignore

Add Service Monitoring

Configuring service monitoring is similar to creating health checks on other platforms. In NSX 6.1, there is a limitation on how many health checks can be performed against a single node. Currently, you can only have a single health check run against a node to ensure availability.

When you associate a service monitor with a pool, the pool members are monitored according to the service monitor parameters. To configure a Service Monitor perform the following steps:

1. Log in to the vSphere Web Client

2. Click **Networking & Security** and then click **NSX Edges**.

3. Double-click an NSX Edge.

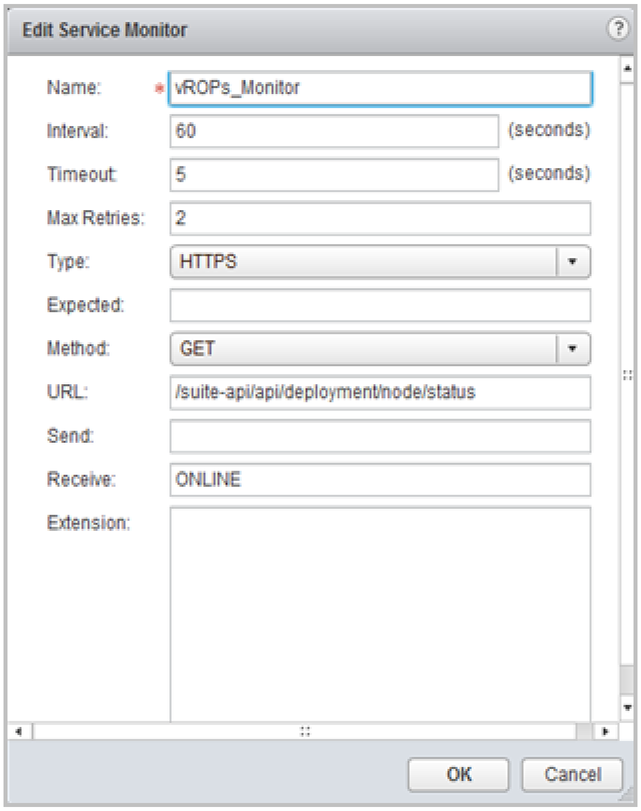
4. Click **Manage** and then click the **Load Balancer** tab.

5. In the left navigation panel, click **Service Monitoring**.

6. Click the Add (⊕) icon.
7. Enter a name for the service monitor. For example: vROps_Monitor
8. Enter an **Interval** at which a server is to be pinged.
9. Enter a **Timeout** in seconds, maximum time within which a response from the server must be received.
10. Enter the number of times the server must be pinged before it is declared down.
11. Select the **Method** in which you want to send the health check request to the server. For example: GET.
12. Insert the health check URL as shown in the following table.
13. Enter the **Receive** data string needed for a successful health check response. For example: ONLINE.
14. Click **OK** to save the new Service Monitor.

NAME	INTERVAL	TIMEOUT	RETRIES	TYPE	METHOD	URL	RECEIVE:
vROps_Monitor	60	5	2	HTTPS	GET	/suite-api/api/deployment/node/status\r\n	ONLINE (upper case)
EPPOS_Monitor	60	5	2	HTTPS	GET	GET\ /epops-webapp/health-check\ HTTP/1.0\r\n	ONLINE (upper case)

Example:



Edit Service Monitor

Name: * vROPs_Monitor

Interval: 60 (seconds)

Timeout: 5 (seconds)

Max Retries: 2

Type: HTTPS

Expected:

Method: GET

URL: /suite-api/api/deployment/node/status

Send:

Receive: ONLINE

Extension:

Add Pools

You can add a server pool to manage and share backend servers, flexibly and efficiently. A pool manages load balancer distribution methods and has a service monitor attached to it for health check parameters.

1. Log in to the vSphere Web Client.
2. Click **Networking & Security** and then click **NSX Edges**.
3. Double-click an NSX Edge.
4. Click **Manage** and then click the **Load Balancer** tab.
5. In the left navigation panel, click **Pools**.
6. Enter a name for the load balancer pool. For example: vROps_Pool.
7. (Optional) Enter a description.
8. Select an **Algorithm** from the drop-down list. For example: LEASTCONN.
9. Select the **Monitors** from the drop-down list. For example: vROps_Monitor.
10. Click the Add (⊕) icon to add your member servers and the required information:

    a. Name

    b. IP Address

    c. Weight: 1

    d. Monitor Port: 443

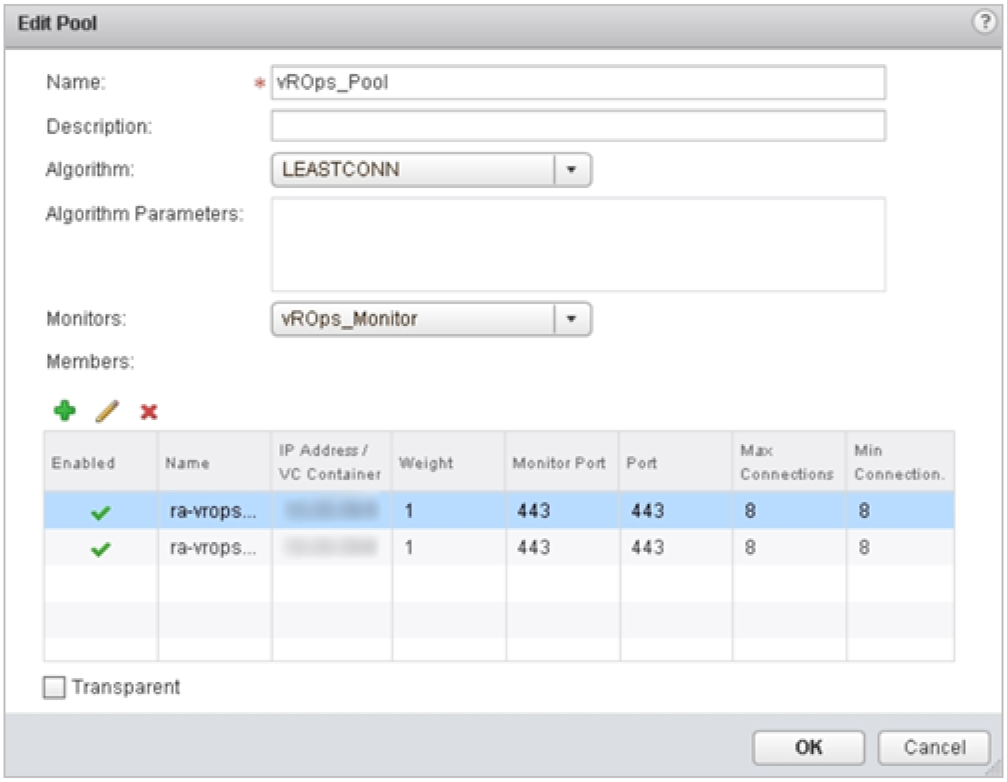
    e. Port: 443

    f. Max Connections: 8

    g. Min Connections: 8

POOL NAME	ALGORITHM	MONITORS	MEMBER NAME	IP ADDRESS/ VCENTER CONTAINER	WEIGHT	PORT	MONITOR PORT	MAX CONNS	MIN CONNS
vROps_Pool	LEASTCONN	vROps_Monitor	vROps_Node1	x.x.x.x	1	443	443	8	8
EPOPS_Pool	LEASTCON	EPPOS_Monitor	EOPOS_NODE1	x.x.x.x	1	443	443	8	8

Example:



Name: * vROps_Pool

Description:

Algorithm: LEASTCONN

Algorithm Parameters:

Monitors: vROps_Monitor

Members:

Enabled	Name	IP Address / VC Container	Weight	Monitor Port	Port	Max Connections	Min Connection.
✓	ra-vrops...	[redacted]	1	443	443	8	8
✓	ra-vrops...	[redacted]	1	443	443	8	8

□ Transparent

Add Virtual Servers

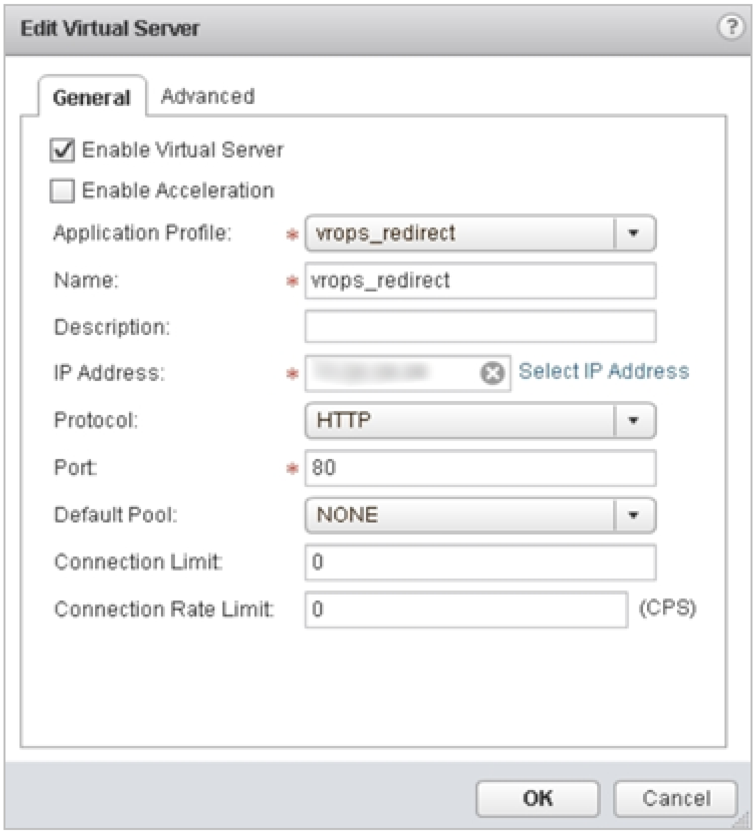
You can add an NSX Edge internal or uplink interface as a virtual server.

1. Log in to the vSphere Web Client.
2. Click **Networking & Security** and then click **NSX Edges**.
3. Double-click an NSX Edge.
4. Click **Manage** and then click the **Load Balancer** tab.
5. In the left navigation panel, click **Virtual Servers**.
6. Click the Add (➕) icon.
7. Enter a name for the virtual server. For example: vROps_Virtual_Server
8. Select **Enable Virtual Server**.
9. Select the **Application Profile** name from the drop-down list. For example: Exp: vrops_https
10. Enter a **Name** for the virtual server.
11. (Optional) Enter a description.
12. Enter the IP Address to be used for the VIP.
13. From the drop-down list for **Protocol**, select **HTTPS**.
14. Enter the **Port** value as 443.
15. From the drop-down list for **Default Pool**, select the default pool that you have configured. For example: vROps_Pool

16. For **Connection Limit** and **Connection Rate Limit**, leave the default as 0.

NOTE: If you are using separate load balancers for vRealize Operations Manager and EPOps, the above steps need to be repeated for EPOps virtual server. Use different names for EPOps profile and respective pool. For example: epops_http and EPOPS_Pool.

Example:



☑ Enable Virtual Server

☐ Enable Acceleration

Application Profile: * vrops_redirect

Name: * vrops_redirect

Description:

IP Address: * [redacted]

Protocol: HTTP

Port: * 80

Default Pool: NONE

Connection Limit: 0

Connection Rate Limit: 0 (CPS)

Configure Auto Redirect from HTTP to HTTPS

When using the NSX load balancer in front of the vRealize Operations Manager cluster you may want the URL to automatically redirect to the HTTPS login page. If you do not configure this the user will need to insert the https field in front of the URL/IP Address. Similar setting is also required in a HAProxy configuration to ensure the redirect works properly. You must configure application profiles and virtual servers for HTTPS redirect.

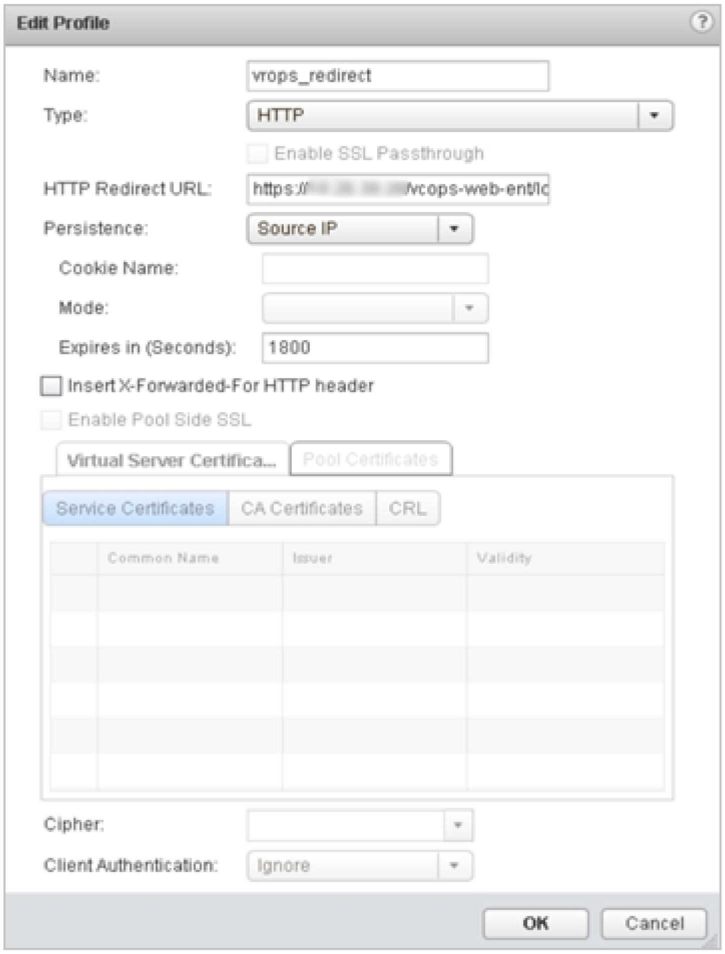
NOTE: Ensure that you are using the HTTPS URLs in a correct manner.

Configure Application Profile for HTTPS Redirect

1. Log in to the vSphere Web Client.
2. Click **Networking & Security** and then click **NSX Edges**.
3. Double-click an NSX Edge.
4. Click **Manage** and then click the **Load Balancer** tab.
5. In the left navigation panel, click **Application Profiles**.
6. Click the Add (✚) icon.

7. Enter a name for the Application Profile. For example: vROps_Redirect
8. From the drop-down list for **Type**, select **HTTP**.
9. For **HTTP Redirect URL**, enter https://<ip_address_of_vip>/vcops-web-ent/login.action.
10. From the drop-down list for **Persistence**, select **Source IP**.
11. Enter **1800** for **Expires in (seconds)**.
12. Click **OK** to save.

Example:



Edit Profile

Name: vrops_redirect

Type: HTTP

□ Enable SSL Passthrough

HTTP Redirect URL: https://[redacted]/vcops-web-ent/lc

Persistence: Source IP

Cookie Name:

Mode:

Expires in (Seconds): 1800

□ Insert X-Forwarded-For HTTP header

□ Enable Pool Side SSL

	Common Name	Issuer	Validity

Cipher:

Client Authentication: Ignore

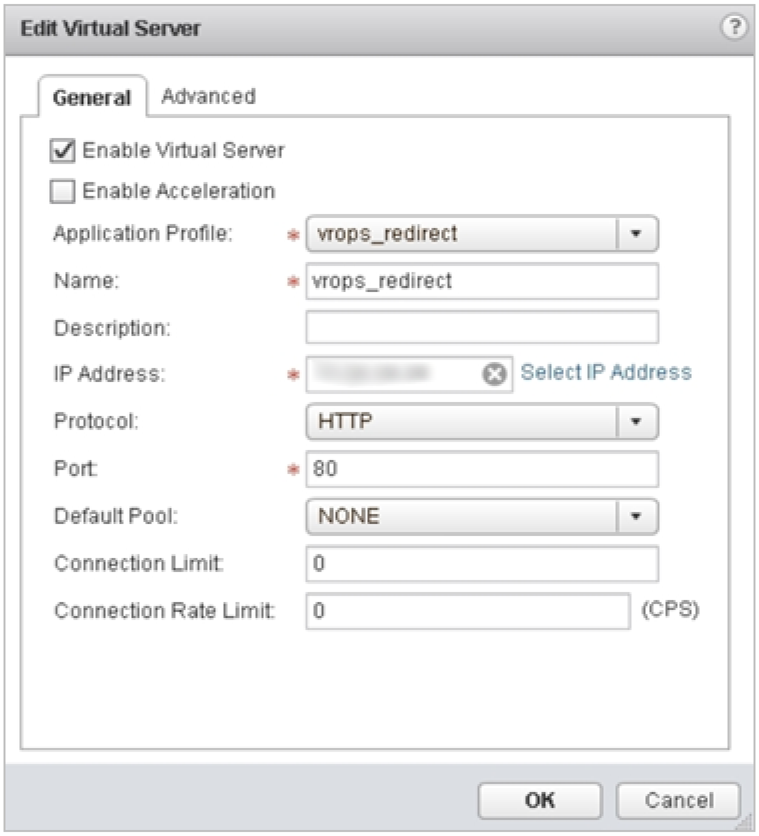
Configure the Virtual Server for HTTPS Redirect

You can configure the virtual server for HTTPS redirect.

1. Log in to the vSphere Web Client.
2. Click **Networking & Security** and then click **NSX Edges**.
3. Double-click an NSX Edge.
4. Click **Manage** and then click the **Load Balancer** tab.
5. In the left navigation panel, click **Virtual Servers**.

6. Click the Add (✚) icon.
7. Select **Enable Virtual Server**.
8. Select an **Application Profile** from the drop-down list that you have created. For example: vrops_redirect
9. Enter a **Name** for the virtual server.
10. (Optional) Enter a **Description**.
11. Enter IP Address for the VIP.
12. From the drop-down list for **Protocol**, select **HTTPS**.
13. Enter the **Port** value as 80.
14. From the drop-down list for **Default Pool**, select **None**.
15. For **Connection Limit** and **Connection Rate Limit**, leave the default as 0.

Example:



Edit Virtual Server

General | Advanced

☑ Enable Virtual Server

☐ Enable Acceleration

Application Profile: * vrops_redirect

Name: * vrops_redirect

Description:

IP Address: * [redacted] Select IP Address

Protocol: HTTP

Port: * 80

Default Pool: NONE

Connection Limit: 0

Connection Rate Limit: 0 (CPS)

Verify Component and Pool Status

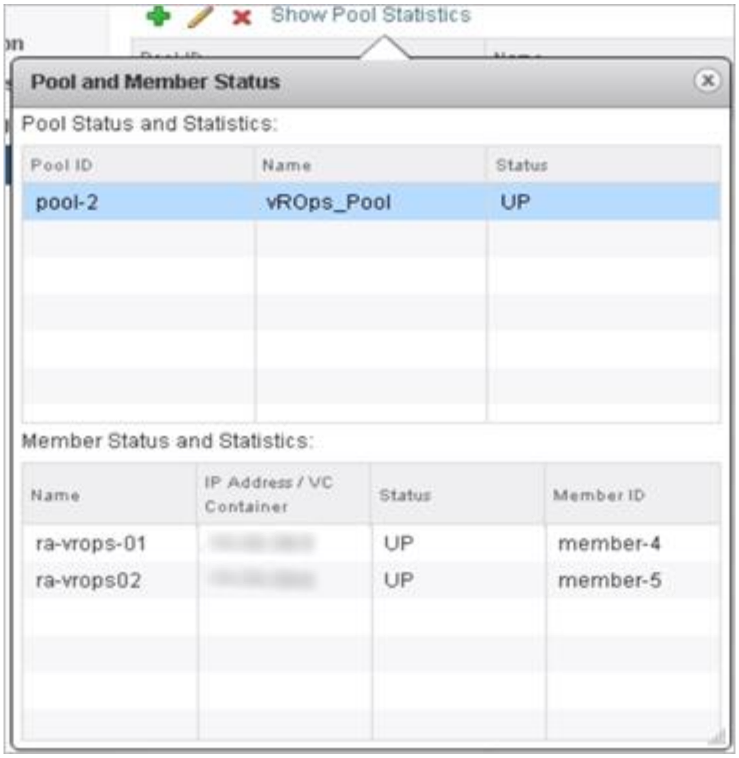
You can verify the status of the components running on the load balancer and you can check the status of the pools from inside the UI of the vSphere Web Client.

1. Log in to the vSphere Web Client.
2. Click **Networking & Security** and then click **NSX Edges**.
3. Double-click an NSX Edge.

4. Click **Manage** and then click the **Load Balancer** tab.
5. In the left navigation panel, click **Pools**.
6. Select the pool you want to verify. For example: vROps_Pool.
7. Click **Show Pool Statistics**. A **Pool and Member Status** pop-up window appears.
8. Select a pool ID. For example: vROps_Pool.

   The member ID and status of the selected pool are displayed. The status can be **UP** or **DOWN**.

Example:



Pool and Member Status

Pool Status and Statistics:

Pool ID	Name	Status
pool-2	vROps_Pool	UP

Member Status and Statistics:

Name	IP Address / VC Container	Status	Member ID
ra-vrops-01	[redacted]	UP	member-4
ra-vrops02	[redacted]	UP	member-5

Last page. Created using a section break. Do not delete this page. Text can appear on this page or it can be left blank.