



In-Guest High Availability (HA) Configuration in Red Hat OpenStack Cloud using Veritas InfoScale Availability (VCS)

Who should read this paper

Administrators who want to implement in-guest HA architectures for unmanaged application services in the cloud

Last updated: 2017-03-13

Table of Contents

INTRODUCTION	3
CONFIGURATION	3
PREPARING OPENSTACK TO CONFIGURE A VIP FOR HIGH AVAILABILITY	6
CP SERVER CONFIGURATION.....	9
SAMPLE VCS CONFIGURATION FILE (MAIN.CF)	10
TESTED CONFIGURATIONS	12
LIMITATIONS	12
REFERENCES	12

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

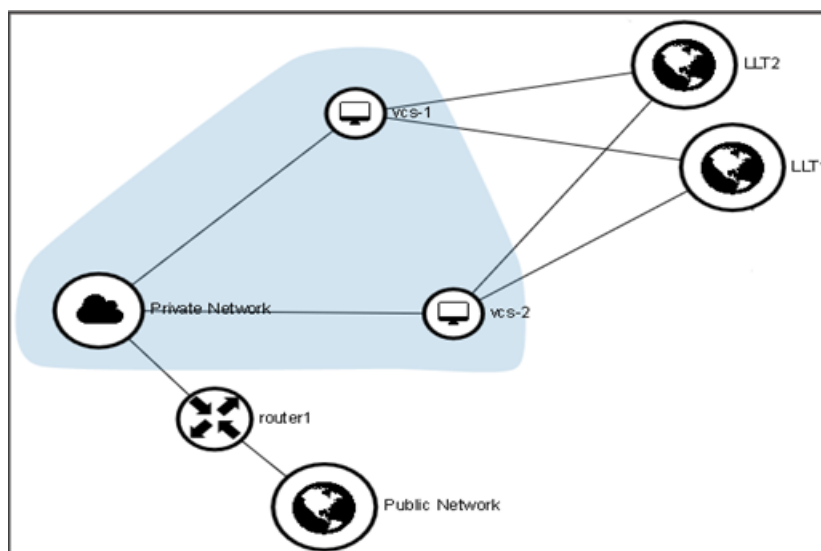
Introduction

OpenStack is based entirely on open source software and is backed by a vibrant global ecosystem of users and vendors. Initiated in 2010, this flexible cloud platform has matured rapidly and is now ready for production cloud deployments in many environments.

In this document, we describe how to prepare Red Hat OpenStack to configure a virtual IP (VIP) for high availability (HA). Veritas InfoScale Availability (formerly Veritas Cluster Server or VCS) is used to provide in-guest HA to the VIP.

Configuration

The following graphic depicts an OpenStack network configuration with two VCS nodes running on a Red Hat OpenStack virtual machine (VM) in the same availability zones:



Note: This sample configuration uses Red Hat OpenStack version 10 deployed with the KVM hypervisor, and Veritas InfoScale Availability 7.2 is configured on an OpenStack VM.

This graphic includes the following elements:

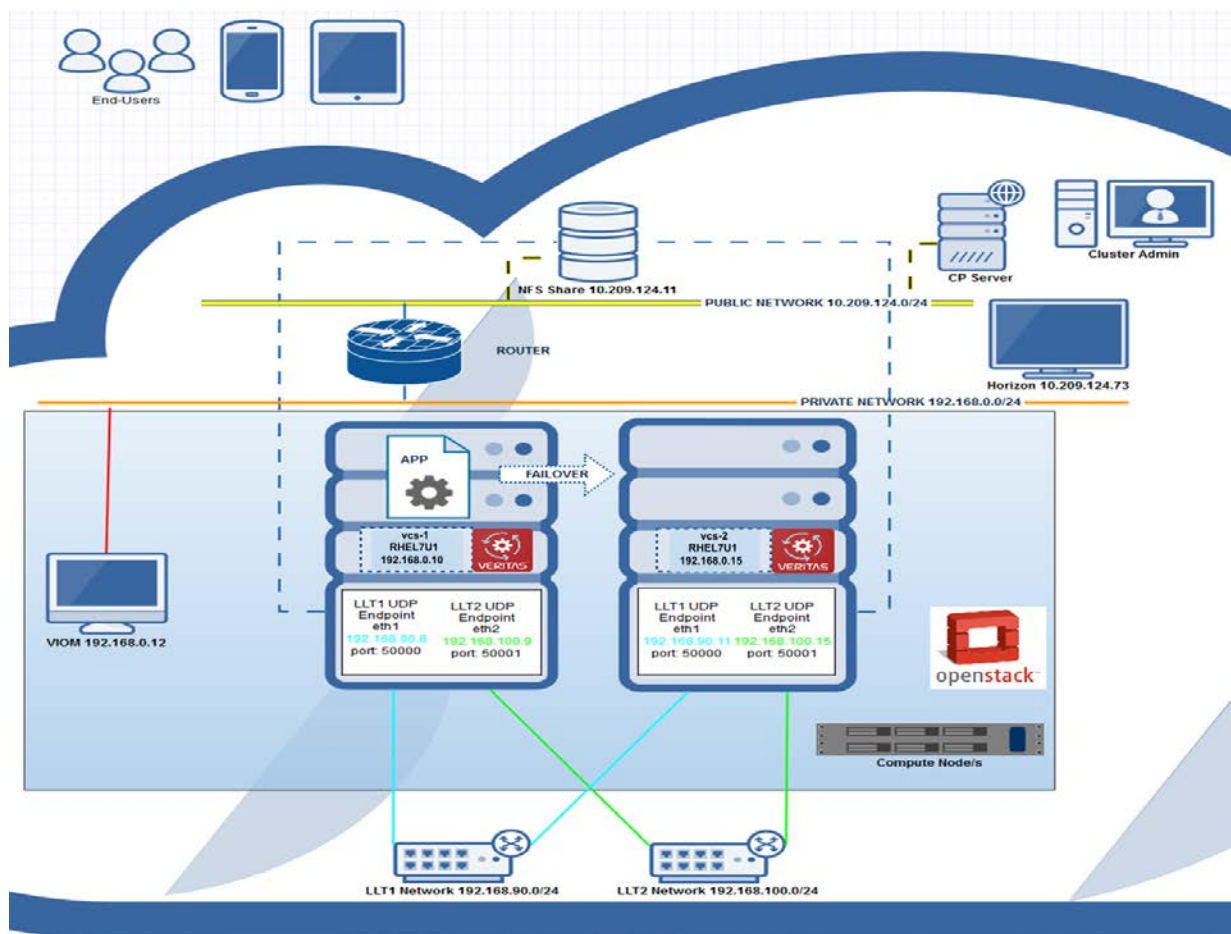
Element	Description
Public network	A public network contains a floating IP address, which is a service provided by Neutron. It does not use any DHCP service or an IP is not set statically within the guest. The operating system of the guest does not know that it was assigned a floating IP address. The L3 agent of Neutron is responsible for the delivery of packets to the interface with the assigned floating address. Instances that have a floating IP address assigned can be accessed from the public network by using the floating IP.

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

Element	Description
Private network	A private network contains a private IP address, which the DHCP server assigns to the network-interface of an instance. The address is visible from within the instance by using a command like “ip a”. The address is typically part of a private network and is used for communication between instances.
Router 1	The router that connects the private network to the public network to further access any external networks.
vcs-1 and vcs-2	These VMs are created in the OpenStack environment within the same availability zone, and they would form the VCS cluster nodes.
LLT1 and LLT2	These networks are used for low latency transport (LLT) communication between the VCS cluster nodes. Note: These networks must have static IP address. Otherwise, a network partition may occur and the cluster may get into a jeopardy situation.

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

The following graphic depicts an application configured for in-guest HA in Red Hat OpenStack using VCS:



- The OpenStack instances host Apache web server and an application, the binaries of which are placed on an NFS location, all of which are managed by VCS. Both the instances exist in a private subnet and are connected to the public network via a router for internet access.
- This sample configuration uses LLT over UDP. However, if you are using a single availability zone and subnet, you can also use LLT over Ethernet.
- Each OpenStack VM has two network interfaces, eth1 and eth2. Both the instances, vcs-1 and vcs-2, has subnet 192.168.90.0/24 on eth1 and subnet 192.168.100.0/24 on eth2.
- The LLT links are configured to use both these subnets as a part of the intra-cluster communication.

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

The following graphic depicts a sample `/etc/llttab` file for the OpenStack instance, `vcs-1`:

```
set-node vcs-1
set-cluster 53959
link eth1 udp - udp 50000 - 192.168.90.8 -
link eth2 udp - udp 50001 - 192.168.100.9 -
set-addr 1 eth1 192.168.90.11
set-addr 1 eth2 192.168.100.15
set-bcasthb 0
set-arp 0
```

The following graphic depicts a sample `/etc/llttab` file for the OpenStack instance, `vcs-2`:

```
set-node vcs-2
set-cluster 53959
link eth1 udp - udp 50000 - 192.168.90.11 -
link eth2 udp - udp 50001 - 192.168.100.15 -
set-addr 0 eth1 192.168.90.8
set-addr 0 eth2 192.168.100.9
set-bcasthb 0
set-arp 0
```

- The application runs inside the OpenStack instances, while the application binaries are placed over the NFS location 10.209.124.11, which is outside the OpenStack network. The scope of this configuration includes the mount agent only, which has been tested with NFS v3 and NFS v4 in this sample.
- A coordination point (CP) server is configured outside the OpenStack network, which takes care of fencing in case there is a network partition between the two cluster nodes. If required, you can also configure a CP server inside the OpenStack network.
- The OpenStack instances have Apache server installed and running, to make the Apache service highly available across the cluster nodes. For access to these from outside the OpenStack environment, a virtual private IP is configured, all of which are managed by VCS.

Preparing OpenStack to configure a VIP for high availability

To allow a VIP to communicate across cluster nodes, you need to map the same VIP to multiple ports.

Neutron does not allow such a configuration by default. To work around this limitation and provide HA for a VIP, we use the `allowed-address-pairs` feature that Neutron provides.

Allowed-address-pairs allow you to specify IP address (CIDR) pairs that pass through a port. This enables the use of protocols such as VRRP, which floats an IP address between two instances to enable fast data plane failover.

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

To map the VIP with the instance port

1. Create a Neutron port for VIP with the appropriate network. This example uses a private network named port-vip1.

```
[root@vcsr720-05 ~(keystone_admin)]# neutron net-list
```

id	name	subnets
2a122a5b-8c4f-4298-b89b-aa8a4274f72c	LLT2	d49c01d9-7e55-4b62-8121-895e86c7ebb3 192.168.100.0/24
cb81c278-072b-4b49-9d53-f866197cec3e	LLT1	3abc3017-40ea-437e-b1ea-ec0732229d8b 192.168.90.0/24
863820f9-187e-4f53-9609-8438e07b71dd	Public Network	ab6b3179-3ddb-4738-93ce-c0e26ee4c3ca 10.209.124.0/24
ed6bdf17-0aee-4b24-85c2-88bcb8d88107	Private Network	d83304cb-cc31-49e4-9cff-ba01c7f144e5 192.168.0.0/24

```
[root@vcsr720-05 ~(keystone_admin)]#
```

2. After creating the port IP address 192.168.0.19 can be used as a VIP, which the DHCP network provided:

```
[root@vcsr720-05 ~(keystone_admin)]# neutron port-create --name port-vip1 ed6bdf17-0aee-4b24-85c2-88bcb8d88107
```

Created a new port:

Field	Value
admin_state_up	True
allowed_address_pairs	
binding:vnic_type	normal
device_id	
device_owner	
dns_assignment	{ "hostname": "host-192-168-0-19", "ip_address": "192.168.0.19", "fqdn": "host-192-168-0-19.openstacklocal." }
dns_name	
fixed_ips	{ "subnet_id": "d83304cb-cc31-49e4-9cff-ba01c7f144e5", "ip_address": "192.168.0.19" }
id	9ac4f9c-f933-44a8-a2a6-b23f99c042cb
mac_address	fa:16:3e:3f:a4:93
name	port-vip1
network_id	ed6bdf17-0aee-4b24-85c2-88bcb8d88107
security_groups	d6a685e4-d351-4b4e-b428-535cfcb09c0e
status	DOWN
tenant_id	98327efcb5124849a04d1be87a1f3167

```
[root@vcsr720-05 ~(keystone_admin)]#
```

Alternatively, you can assign a specific IP address to the newly created port using the following command:

```
[root@vcsr720-05 ~(keystone_admin)]# neutron port-create --name port-vip2 --fixed-ip subnet_id=d83304cb-cc31-49e4-9cff-ba01c7f144e5,ip_address=192.168.0.101 ed6bdf17-0aee-4b24-85c2-88bcb8d88107
```

Created a new port:

Field	Value
admin_state_up	True
allowed_address_pairs	
binding:vnic_type	normal
device_id	
device_owner	
dns_assignment	{ "hostname": "host-192-168-0-101", "ip_address": "192.168.0.101", "fqdn": "host-192-168-0-101.openstacklocal." }
dns_name	
fixed_ips	{ "subnet_id": "d83304cb-cc31-49e4-9cff-ba01c7f144e5", "ip_address": "192.168.0.101" }
id	a65144f4-3bc5-4e0c-a151-5acea6255ae0
mac_address	fa:16:3e:ee:33:c8
name	port-vip2
network_id	ed6bdf17-0aee-4b24-85c2-88bcb8d88107
security_groups	d6a685e4-d351-4b4e-b428-535cfcb09c0e
status	DOWN
tenant_id	98327efcb5124849a04d1be87a1f3167

```
[root@vcsr720-05 ~(keystone_admin)]#
```

3. Now, map your VIP (192.168.0.19) with the appropriate ports of all the cluster nodes, so that the VIP can fail over and communicate across the cluster using allowed-address-pairs:

```
[root@vcsr720-05 ~(keystone_admin)]# nova list
```

ID	Name	Status	Task State	Power State	Networks
eb21b984-78cf-4d98-bc79-16779924ae4a	vcs-1	ACTIVE	-	Running	LLT1=192.168.90.8; LLT2=192.168.100.9; Private Network=192.168.0.10, 10.209.124.73
a41fcedb-952e-4b1d-b825-5428f7bb69f3	vcs-2	ACTIVE	-	Running	LLT1=192.168.90.11; LLT2=192.168.100.15; Private Network=192.168.0.15, 10.209.124.74

```
[root@vcsr720-05 ~(keystone_admin)]#  
[root@vcsr720-05 ~(keystone_admin)]# neutron port-list | grep 192.168.0.10  
| b29cc9ad-e498-4ce6-a379-28372e891c1 | | fa:16:3e:b9:b9:59 | { "subnet_id": "d83304cb-cc31-49e4-9cff-ba01c7f144e5", "ip_address": "192.168.0.10" } |  
[root@vcsr720-05 ~(keystone_admin)]#  
[root@vcsr720-05 ~(keystone_admin)]# neutron port-list | grep 192.168.0.15  
| 87dda133-34bf-426e-a2b0-f9bcd62f8944 | | fa:16:3e:86:a4:f2 | { "subnet_id": "d83304cb-cc31-49e4-9cff-ba01c7f144e5", "ip_address": "192.168.0.15" } |  
[root@vcsr720-05 ~(keystone_admin)]#
```

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

```
[root@vcsr720-05 ~(keystone_admin)]# neutron port-update b28cc9ad-e408-4ce6-a379-28372e08f1c1 --allowed_address_pairs list=true type=dict ip_address=192.168.0.19
Updated port: b28cc9ad-e408-4ce6-a379-28372e08f1c1
[root@vcsr720-05 ~(keystone_admin)]#
[root@vcsr720-05 ~(keystone_admin)]# neutron port-update 87dda133-34bf-426e-a2b0-f9bcd62f0944 --allowed_address_pairs list=true type=dict ip_address=192.168.0.19
Updated port: 87dda133-34bf-426e-a2b0-f9bcd62f0944
[root@vcsr720-05 ~(keystone_admin)]#
```

The allowed-address-pairs values are successfully updated for both the instances:

```
[root@vcsr720-05 ~(keystone_admin)]# neutron port-show b28cc9ad-e408-4ce6-a379-28372e08f1c1
-----
| Field | Value |
-----+-----
| admin_state_up | True |
| allowed_address_pairs | {"ip_address": "192.168.0.19", "mac_address": "fa:16:3e:b8:b9:59"} |
| binding:vnic_type | normal |
| device_id | eb21b804-78cf-4d88-bc79-16779824ae4a |
| device_owner | compute:nova |
| dns_assignment | {"hostname": "host-192-168-0-10", "ip_address": "192.168.0.10", "fqdn": "host-192-168-0-10.openstacklocal."} |
| dns_name | |
| extra_dhcp_opts | |
| fixed_ips | {"subnet_id": "d83304cb-cc31-49e4-9cff-ba01c7f144e5", "ip_address": "192.168.0.10"} |
| id | b28cc9ad-e408-4ce6-a379-28372e08f1c1 |
| mac_address | fa:16:3e:b8:b9:59 |
| name | |
| network_id | ed6bdf17-0aee-4b24-85c2-88bcb8d88107 |
| security_groups | fad0ef1e-8c33-46a5-91af-04e8754ca953 |
| status | ACTIVE |
| tenant_id | 98327efcb5124849a04d1be87a1f3167 |
-----
```

```
[root@vcsr720-05 ~(keystone_admin)]# neutron port-show 87dda133-34bf-426e-a2b0-f9bcd62f0944
-----
| Field | Value |
-----+-----
| admin_state_up | True |
| allowed_address_pairs | {"ip_address": "192.168.0.19", "mac_address": "fa:16:3e:86:a4:f2"} |
| binding:vnic_type | normal |
| device_id | a4f1cedb-962e-4b1d-b825-5428f7bb69f3 |
| device_owner | compute:nova |
| dns_assignment | {"hostname": "host-192-168-0-15", "ip_address": "192.168.0.15", "fqdn": "host-192-168-0-15.openstacklocal."} |
| dns_name | |
| extra_dhcp_opts | |
| fixed_ips | {"subnet_id": "d83304cb-cc31-49e4-9cff-ba01c7f144e5", "ip_address": "192.168.0.15"} |
| id | 87dda133-34bf-426e-a2b0-f9bcd62f0944 |
| mac_address | fa:16:3e:86:a4:f2 |
| name | |
| network_id | ed6bdf17-0aee-4b24-85c2-88bcb8d88107 |
| security_groups | d6a685e4-d351-4b4e-b428-535cfcb09c0e |
| status | ACTIVE |
| tenant_id | 98327efcb5124849a04d1be87a1f3167 |
-----
[root@vcsr720-05 ~(keystone_admin)]#
```

After performing the previously listed steps the VIP 192.168.0.19 can be used to communicate across the cluster within network.

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

4. To enable communication outside the OpenStack environment, associate the floating IP to port-vip1 using the dashboard:

Manage Floating IP Associations

IP Address *

IP Address *
10.209.124.80

Select the IP address you wish to associate with the selected instance or port.

Port to be associated *

Select a port

Select a port

None: 192.168.0.19

vcs-1: 192.168.0.10

vcs-2: 192.168.0.15

Cancel Associate

Access & Security

Security Groups Key Pairs Floating IPs API Access

Allocate IP To Project Release Floating IPs

IP Address	Mapped Fixed IP Address	Pool	Status	Actions
10.209.124.75	vcs-3 192.168.0.12	Public Network	Active	Disassociate
10.209.124.74	vcs-2 192.168.0.15	Public Network	Active	Disassociate
10.209.124.80	-	Public Network	Down	Disassociate
10.209.124.76	vcs-4 192.168.0.13	Public Network	Active	Disassociate
10.209.124.52	-	Public Network	Down	Associate
10.209.124.51	-	Public Network	Down	Associate
10.209.124.73	vcs-1 192.168.0.10	Public Network	Active	Disassociate

Displaying 7 items

CP server configuration

The coordination point server (CP server) is a software solution that runs on a remote system or cluster. CP server provides arbitration functionality by allowing the VCS cluster nodes to perform the following tasks:

- Self-register to become a member of an active VCS cluster (registered with CP server) with access to the data drives.
- Check which other nodes are registered as members of this active VCS cluster.
- Self-unregister from this active VCS cluster.
- Forcefully unregister other nodes (preempt) as members of this active VCS cluster.
- If required, set the `loser_exit_delay` parameter in the `/etc/vxfenmode` file according to your cluster setup.

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

In short, the CP server functions as another arbitration mechanism that integrates within the existing I/O fencing module.

You can configure a CP server by invoking the `/opt/VRTS/install/installer -configcps` command.

Sample VCS configuration file (main.cf)

```
include "OracleASMTypes.cf"
include "types.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SybaseTypes.cf"

cluster openclus (
    SecureClus = 1
    DefaultGuestAccess = 1
    UseFence = SCSI3
)

system vcs-1 (
)

system vcs-2 (
)

group AppGrp (
    SystemList = { openvm4 = 0, openvm5 = 1 }
    ProPCV = 1
)

Application appres (
    StartProgram = "/nfslocation/Application_TestIMFAPP/testapp_start2"
    StopProgram = "/nfslocation/Application_TestIMFAPP/testapp_stop2"
    CleanProgram = "/nfslocation/Application_TestIMFAPP/testapp_clean2"
    MonitorProgram = "/nfslocation/Application_TestIMFAPP/testapp_monitor2"
    PidFiles = { "/nfslocation/Application_TestIMFAPP/testapp.pid",
                 "/nfslocation/Application_TestIMFAPP/testapp2.pid" }
    MonitorProcesses = {
        "/usr/bin/sh /nfslocation/Application_TestIMFAPP/testapp",
        "/usr/bin/sh /nfslocation/Application_TestIMFAPP/testapp2" }
)

Mount nfs-mount (
    MountPoint = "/nfslocation"
    BlockDevice = "10.209.124.133:/nfsshare/"
    FSType = nfs
    MountOpt =
"rw,bg,hard,nointr,tcp,nfsvers=3,timeo=600,rsize=32768,wsiz=32768"
    CreateMntPt = 1
    RecursiveMnt = 1
)
```

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

```
appres requires nfs-mount
```

```
// resource dependency tree
//
//   group AppGrp
//   {
//       Application appres
//       {
//           Mount nfs-mount
//       }
//   }
```

```
group ApacheGroup (
  SystemList = { vcs-1 = 0, vcs-2 = 1 }
)
```

```
Apache apache_res (
  httpdDir = "/usr/sbin"
  HostName = "192.168.0.19"
  ConfigFile = "/etc/httpd/conf/httpd.conf"
  LevelTwoMonitorFreq = 1
)
```

```
IP apache_ipres (
  Device = eth0
  Address = "192.168.0.19"
  NetMask = "255.255.255.0"
)
```

```
NIC apache_nicres (
  Device = eth0
)
```

```
apache_ipres requires apache_nicres
apache_res requires apache_ipres
```

```
group vxfen (
  SystemList = { vcs-1 = 0, vcs-2 = 1 }
  AutoFailOver = 0
  Parallel = 1
)
```

```
CoordPoint coordpoint (
)
```

```
Phantom RES_phantom_vxfen (
```

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

```
)  
  
// resource dependency tree  
//  
//   group vxfen  
//   {  
//       Phantom RES_phantom_vxfen  
//       CoordPoint coordpoint  
//   }
```

Tested configurations

The following configurations have been tested:

OpenStack version	Guest operating system version	InfoScale Availability (VCS) version
RHOSP 8	RHEL 7.1	VCS 6.2.1
RHOSP 8	RHEL 7.1	InfoScale Availability 7.2
RHOSP 8	RHEL 7.2	VCS 6.2.1*
RHOSP 8	RHEL 7.2	InfoScale Availability 7.2
RHOSP 10	RHEL 7.1	VCS 6.2.1
RHOSP 10	RHEL 7.1	InfoScale Availability 7.2
RHOSP 10	RHEL 7.2	VCS 6.2.1*
RHOSP 10	RHEL 7.2	InfoScale Availability 7.2
RHOSP 10	RHEL 7.3	VCS 6.2.1*

* These VCS versions require product patches. For information about the relevant product patches, visit:

<https://sort.veritas.com/productmatrix>

Limitations

The following limitations currently exist with these configurations:

- Any agent that requires a storage component is not supported.
- All the limitations that apply to the allowed-address-pair feature also apply to these configurations.

References

<https://access.redhat.com/documentation/en/red-hat-openstack-platform/10/paged/networking-guide/chapter-18-configure-allowed-address-pairs>

In-guest HA configuration in Red Hat OpenStack cloud using Veritas InfoScale Availability (VCS)

About Veritas Technologies LLC

Veritas Technologies LLC enables organizations to harness the power of their information, with solutions designed to serve the world's largest and most complex heterogeneous environments. Veritas works with 86 percent of Fortune 500 companies today, improving data availability and revealing insights to drive competitive advantage.

For specific country offices and contact numbers, please visit our website.

Veritas World Headquarters
500 East Middlefield Road
Mountain View, CA 94043
+1 (650) 933 1000
www.veritas.com

© 2017 Veritas Technologies LLC. All rights reserved.
Veritas and the Veritas Logo are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.