# InfoScale Availability Agents for OpenStack Installation and Configuration Guide

Linux

7.0

VERITAS™

# Veritas InfoScale™ Availability Agents

Last updated: 2019-02-15

## Legal Notice

Copyright © 2019 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.

This product may contain third-party software for which Veritas is required to provide attribution to the third party (“Third Party Programs”). Some of the Third-Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Refer to the third-party legal notices document accompanying this Veritas product or available at:

https://www.veritas.com/about/legal/license-agreements

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Veritas Technologies LLC and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. VERITAS TECHNOLOGIES LLC SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043

http://www.veritas.com

## Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

| | |
|---|---|
| Worldwide (except Japan) | CustomerCare@veritas.com |
| Japan | CustomerCare_Japan@veritas.com |

## Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

## Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

xyz@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

## Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

# Contents


Chapter 1 Introducing the Veritas InfoScale agents for OpenStack ..... 5

About the InfoScale Availability agents for OpenStack ..... 5
OpenStackIP agent functions ..... 6
OpenStackVol agent functions ..... 9
OpenStackAuth agent functions ..... 10

Chapter 2 Installing, upgrading, and removing the agent for OpenStack ..... 12

Installing the agents in an InfoScale environment ..... 12
Uninstalling the agents in an InfoScale Availability environment ..... 13

Chapter 3 Configuring the agent for OpenStack ..... 15

Before configuring the resources for OpenStack ..... 15
Importing the agent types files in an InfoScale environment ..... 16
OpenStackIP agent attributes ..... 16
OpenStackVol agent attributes ..... 18
OpenStackAuth agent attributes ..... 18

Chapter 4 Configuring the service groups for OpenStack ..... 20

Before configuring the service groups for OpenStack ..... 20
Known issues and limitations ..... 20
Configuring service groups for OpenStack ..... 22

Appendix A Sample configurations ..... 27

About sample configurations for the agents for OpenStack ..... 27
Sample agent type definition for OpenStack ..... 27
Sample service group configurations for OpenStack ..... 28
Sample resource type definitions for OpenStackIP ..... 29
Sample service group configuration for OpenStackAuth ..... 31
Sample cluster configurations using OpenStack agents ..... 32

Chapter 1

# Introducing the Veritas InfoScale agents for OpenStack

This chapter includes the following topics:


- About the InfoScale Availability agents for OpenStack
- OpenStackIP agent functions
- OpenStackVol agent functions
- OpenStackAuth agent functions


## About the InfoScale Availability agents for OpenStack

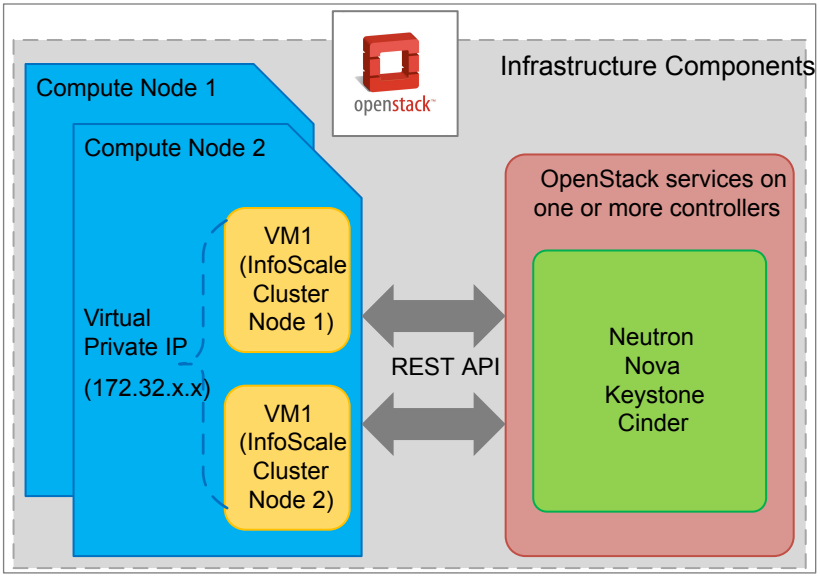
InfoScale Availability lets you configure applications for high availability (HA) in an OpenStack environment. An HA configuration in OpenStack needs to support the underlying network and storage resources as well.

InfoScale Availability provides the following agents to support the configurations for these resources:

- OpenStackIP
- OpenStackVol
- OpenStackAuth

**Note:** The OpenStackAuth agent supports Keystone API v2.0 and v3.

**Figure 1-1** Components used in an InfoScale HA configuration for OpenStack




# OpenStackIP agent functions

The OpenStackIP agent lets you failover an IP between compute instances by adding IP addresses to and removing IP addresses from the OpenStack device ports. The following table describes the various functions that this agent performs.

**Table 1-1** Functions of the OpenStackIP agent

| Function | Description |
|---|---|
| Online | The agent performs various tasks as part of this function depending on whether the following attributes are set for an OpenStackIP resource:<br>■ PrivateIP: If this attribute is set, the agent first checks that value of the UseNewPort attribute.<br>  ■ If UseNewPort is set to 1 (default), the agent creates a new Neutron port using the PrivateIP in the same network as the device. It also updates the `allowed_address_pairs` field of the device port with the value of PrivateIP, only if the value is not part of any CIDR entries in the `allowed_address_pairs` field of the device port.<br>  ■ If UseNewPort is set to 0 (zero), the agent appends the value of PrivateIP to the `--fixed_ips` field of the device port. By default, `--fixed_ips` can accommodate only 5 IP addresses. You can change this value by updating `/etc/neutron/neutron.conf` (`max_fixed_ips_per_port = 5`) and restarting the neutron service.<br>■ AllowedAddressPair: If this attribute is set, the agent adds its value to the `--allowed_address_pairs` field of the device port.<br>■ FloatingIP: If this attribute is set, the agent associates its value with PrivateIP. If the BaseIP attribute is specified instead of PrivateIP, the agent associates the value of FloatingIP with BaseIP.<br>■ OverlayIP: If this attribute is set, the agent adds its value to the `--allowed_address_pairs` field of the device port. The agent also adds the static route on the routers provided in the RouteTableNames attribute.<br>**Note:** FloatingIP cannot be associated with OverlayIP. |

**Table 1-1** Functions of the OpenStackIP agent *(continued)*

| Function | Description |
|---|---|
| Offline and Clean | The agent performs various tasks as part of these functions depending on whether the following attributes are set for an OpenStackIP resource:<br>■ PrivateIP: If this attribute is set, the agent first checks that value of the UseNewPort attribute.<br>  ■ If UseNewPort is set to 1 (default), the agent deletes the Neutron port that is mapped to PrivateIP in the same network as the device. It also removes the PrivateIP entry from the `allowed_address_pairs` field if it is present.<br>  ■ If UseNewPort is set to 0 (zero), the agent removes the PrivateIP entry from the `--fixed_ips` field of the device port.<br>■ AllowedAddressPair: If this attribute is set, the agent removes its value from the `--allowed_address_pairs` field of the device port.<br>■ FloatingIP: If this attribute is set, the agent dissociates its value from PrivateIP. During the offline operation, if the BaseIP attribute is specified instead of PrivateIP, the agent dissociates the value of FloatingIP from BaseIP.<br>■ OverlayIP: If this attribute is set, the agent removes its value from the `--allowed_address_pairs` field of the device port. During the offline operation, the agent also deletes the static route from the routers provided in the RouteTableNames attribute. |

**Table 1-1** Functions of the OpenStackIP agent *(continued)*

| Function | Description |
|---|---|
| Monitor | The agent performs various tasks as part of these functions depending on whether the following attributes are set for an OpenStackIP resource:<br>■ PrivateIP: If this attribute is set, the agent first checks that value of the UseNewPort attribute.<br>  ■ If UseNewPort is set to 1 (default), the agent checks for the port with the PrivateIP and also whether the PrivateIP is part of any CIDR/IP in the `--allowed_address_pair` field of the device port. If any of these checks fail, the agent reports that the resource is offline.<br>  ■ If UseNewPort is set to 0 (zero), the agent checks whether the PrivateIP entry exists in the `--fixed_ips` field of the device port. If the value does not exist, the agent reports that the resource is offline.<br>■ AllowedAddressPair: If this attribute is set, the agent checks whether the Is AllowedAddressPair entry exists in the `--allowed_address_pair` field of the device port. If the value does not exist, the agent reports that the resource is offline.<br>■ FloatingIP: If this attribute is set, the agent checks whether its value is associated with either PrivateIP or BaseIP. If the values are not associated, the agent reports that the resource is offline.<br>■ OverlayIP: If this attribute is set, the agent checks whether its entry exists in the `--allowed_address_pairs` field of the device port and whether the static route entry from the routers exists in the RouteTableNames attribute. If any of these checks fail, the agent reports that the resource is offline. |

# OpenStackVol agent functions

The OpenStackVol agent attaches Cinder volumes to, or detaches Cinder volumes from, the OpenStack compute instance on which an application is configured for HA. It moves the configured Cinder volumes between the cluster nodes for failover and failback.

The agent supports the following configurations:

- Heterogeneous volume managers
  For example; Veritas Volume Manager (VxVM) and Logical Volume Manager (LVM)
- Volumes from multiple block storage drivers
  For example: LVM, NFS, and arrays

The following table describes the various functions that this agent performs.

**Table 1-2** Functions of the OpenStackVol agent

| Function | Description |
|---|---|
| Online | ■ Detaches the specified volumes if they are already attached to an instance that is part of the service group system list.<br>■ Attaches the specified volumes to the instance and waits until volumes go into ATTACHED state. |
| Offline and Clean | Detaches the specified volumes from the instance and waits until the volumes go into DETACHED state. |
| Monitor | ■ Reports the resource as ONLINE if all the specified volumes are attached to the instance.<br>■ Reports the resource as OFFLINE if any of the specified volumes is not attached to the instance.<br>**Note:** The OpenStackVol agent does not support Multi-Attach or bootable volumes. |

# OpenStackAuth agent functions

The OpenStackAuth agent generates and validates the tokens and endpoint URLs that the InfoScale agents need to communicate with the OpenStack components.

**Table 1-3** Functions of the OpenStackAuth agent

| Function | Description |
|---|---|
| Online | Creates an empty authorization file, `/var/VRTSvcs/lock/.OpenStackAuth`. |
| Offline and Clean | Removes the authorization file, `/var/VRTSvcs/lock/.OpenStackAuth` from the system. |

**Table 1-3** Functions of the OpenStackAuth agent *(continued)*

| Function | Description |
|---|---|
| Monitor | Checks for the presence of the authorization file on the system, and takes action accordingly:<br>■ If the file is present, checks whether the existing token in the file is valid:<br>  ■ If the token is valid, takes no action.<br>  ■ If the token is not valid, generates a new token and updates the file with the new token,Neutron, Cinder, and Nova public end point.<br>■ If the file is not present, marks the resource as offline.<br>**Note:** If the number of monitor cycles mentioned in LevelTwoMonitorFreq have passed, the agent generates a new token regardless of whether the existing token is valid or not. Then, it updates the authorization file with the new token, Neutron, Cinder and Nova public end point. |

Chapter 2

# Installing, upgrading, and removing the agent for OpenStack

This chapter includes the following topics:


- Installing the agents in an InfoScale environment
- Uninstalling the agents in an InfoScale Availability environment


## Installing the agents in an InfoScale environment

Install the agents for OpenStack on each node in the cluster.

---

**Note:** The agent package for OpenStack includes the OpenStackIP, the OpenStackVol, and the OpenStackAuth agents. The following procedure lets you install all the agents.

---

**To install the agent in an InfoScale environment**

1. Download the agent from the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

   You can download either the complete Agent Pack tar file or an individual agent tar file.

2. Uncompress the file to a temporary location, say `/tmp`.

3 If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

```
cd1/linux/generic/vcs/application/openstack_agent/
```

```
vcs_version/version_agent/rpms
```

If you downloaded the individual agent tar file, navigate to the `rpms` directory.

4 Log in as a superuser.

5 Install the package.

```
# rpm -ihv \ VRTSopenstackag-AgentVersion-GA_GENERIC.noarch.rpm
```

After installing the agent package, you must import the agent type configuration file.

See "Importing the agent types files in an InfoScale environment" on page 16.

# Uninstalling the agents in an InfoScale Availability environment

You must uninstall the agents for OpenStack from a cluster while the cluster is active.

---

**Note:** The agent package for OpenStack includes the OpenStackIP, the OpenStackVol, and the OpenStackAuth agents. The following procedure to remove the agents for OpenStack removes all the agents.

---

**To uninstall the agent in an InfoScale Availability environment**

1 Log in as a superuser on any node in the cluster.

2 Set the cluster configuration mode to read-write by running the following command:

```
# haconf -makerw
```

3 Remove all the OpenStack resources from the cluster by updating the `main.cf` file.

4 Verify that all resources have been removed by running the following commands:

```
# hares -list Type=OpenStackIP
```

```
# hares -list Type=OpenStackVol
```

```
# hares -list Type=OpenStackAuth
```

5 Remove the agent type from the cluster configuration by running the following command:

```
# hatype -delete OpenStackIP

# hatype -delete OpenStackVol

# hatype -delete OpenStackAuth
```

When you remove the type file of an agent from the cluster, the include statement for the agent is removed from the `main.cf` file, but the type file is not removed from the cluster configuration directory. You can manually remove the type file from the cluster configuration directory later.

6 Save the changes that you made so far, and set the cluster configuration mode to read-only by running the following command:

```
# haconf -dump -makero
```

7 Remove the agents for OpenStack from each node in the cluster by using the native software management program of the operating system:

**#** `rpm -e VRTSopenstackag`

Chapter 3

# Configuring the agent for OpenStack

This chapter includes the following topics:


- Before configuring the resources for OpenStack
- Importing the agent types files in an InfoScale environment
- OpenStackIP agent attributes
- OpenStackVol agent attributes
- OpenStackAuth agent attributes


## Before configuring the resources for OpenStack

Before you configure the OpenStack resources, you must:

- Verify that the Cluster Server components are installed and configured on all the nodes in the cluster where you want to configure the service group.
  For more information on installing and configuring the Cluster Server components, refer to the *Veritas InfoScale Installation Guide* and the *Cluster Server Configuration and Upgrade Guide* respectively.
- Verify that the agents for OpenStack are installed on all nodes in the cluster.
  See "Installing the agents in an InfoScale environment" on page 12.
- The operating system-provided `curl` rpm must be present on each cluster node.

# Importing the agent types files in an InfoScale environment

To use the InfoScale agents for OpenStack, you must import the agent types file into the cluster. You can import the agent types file by using the Cluster Manager or the CLI.

**To import the agent types file using the CLI**

1. Log on to any one of the systems in the cluster as the superuser.
2. To import the agent types, run the following command:

   ```
   # sh /etc/VRTSagents/ha/conf/OpenStackAuth/OpenStackTypes.cmd
   ```

3. To verify that the agent types file is successfully imported to the VCS engine, run the following commands:

   ```
   # hatype -display OpenStackIP

   # hatype -display OpenStackVol

   # hatype -display OpenStackAuth
   ```

If the file is successfully imported, you can proceed to create OpenStack resources.

# OpenStackIP agent attributes

The following table describes the attributes that are required to configure an OpenStack IP resource for high availability (HA).

**Table 3-1** Required attributes for the OpenStackIP agent

| Attribute | Description |
|---|---|
| Name: **Device**<br>Type: String<br>Dimension: Scalar | Name of the network device. For example, eth0, eth1.<br>Use the `ifconfig -a` command on the system to list all the network adapters.<br>Default value: (blank)<br>Example: `eth0` |

The following table describes the optional attributes, which you can use to configure an OpenStack IP resource for HA in various scenarios.

**Table 3-2** Optional attributes for the OpenStackIP agent

| Attribute | Description |
|---|---|
| Name: **PrivateIP**<br>Type: String<br>Dimension: Scalar | A virtual IP that needs to be failed over from the device port of one instance to the device port of another instance within the same subnet. As part of online, offline, and clean operations agent updates the `--fixed_ips` field of the Neutron port.<br>Default value: (blank)<br>Example: `10.209.x.x` |
| Name: **FloatingIP**<br>Type: String<br>Dimension: Scalar | Public IP that is designed for the private cloud environment. A floating IP can be associated with the private address (VIP) or the base address of the device.<br>Default value: (blank)<br>Example: `10.209.x.x` |
| Name: **OverlayIP**<br>Type: String<br>Dimension: Scalar | An IP that exists outside the connected networks, which is required to fail over a private IP between the cluster nodes that are spread across subnets. As part of the online, offline, and clean operations, the agent updates the `--allowed_address_pairs` field of the Neutron port and the static route on the connected routers.<br>**Note:** This attribute depends on the RouteTableNames attribute.<br>Default value: (blank)<br>Example: `17.16.8.1/32` |
| Name: **RouteTableNames**<br>Type: String<br>Dimension: Vector | A list of routers that must be updated so that the overlay IP can work across subnets.<br>**Note:** The OverlayIP attribute depends on this attribute.<br>Default value: (blank)<br>Example: `{ router1 }` |
| Name: **AllowedAddressPair**<br>Type: String<br>Dimension: Scalar | IP address or CIDR block address that needs to be allowed on the device port. As part of the online, offline, and clean operations, the agent updates the `--allowed_address_pairs` field of the Neutron port.<br>Default value: (blank)<br>Example: `17.16.10.0/24` |

**Table 3-2** Optional attributes for the OpenStackIP agent *(continued)*

| Attribute | Description |
|---|---|
| Name: **UseNewPort**<br>Type: Boolean<br>Dimenstion: Scalar | Indicates to the agent that a new neutron port must be created using PrivateIP in the same network as the device.<br>Default value: `1`<br>Example: `0` |

# OpenStackVol agent attributes

The following table describes the attribute that is required to configure an OpenStack volume resource for high availability (HA).

**Table 3-3** Required attribute

| Attribute | Description |
|---|---|
| Name: **VolumeIds**<br>Type: String<br>Dimension: Vector | IDs of the Cinder volumes to be attached to or detached from an OpenStack compute instance.<br>Default: No default value<br>Example: { 5e6d7424-c533-447f-8eab-ea6d477cde52, fa07508f-c22e-4259-b6f4-d9ac67e0a5e4 } |

# OpenStackAuth agent attributes

The OpenStackAuth agent enables communication between the other InfoScale agents for OpenStack and the other OpenStack components or resources. The following table describes the attributes that are required to configure an OpenStackAuth type resource.

**Table 3-4** Required attributes for the OpenStackAuth agent

| Attribute | Description |
|---|---|
| Name: **Tenant**<br>Type: String<br>Dimension: Scalar | Project name to which the Compute instance belongs.<br>Default value: (blank)<br>Example: `Project_1` |

**Table 3-4** Required attributes for the OpenStackAuth agent *(continued)*

| Attribute | Description |
|---|---|
| Name: **Username**<br>Type: String<br>Dimension: Scalar | Authorized or valid user name that is required for generating the token.<br>Default value: (blank)<br>Example: `FirstUser` |
| Name: **Password**<br>Type: String<br>Dimension: Scalar | Encrypted password required for generating the token.<br>Default value: (blank)<br>Example: (encrypted) |
| Name: **Region**<br>Type: String<br>Dimension: Scalar | Region name of the OpenStack deployment.<br>Default value: (blank)<br>Example: `RegionOne` |
| Name: **KeystoneEP**<br>Type: String<br>Dimension: Scalar | Public endpoint URL of the identity service.<br>Default value: (blank)<br>Examples:<br>■ **http://10.x.x.x:5000/v2.0**<br>■ **http://[2620:128:xxxx:xxxx::157]:5000/v3** |

**Note:** The value of the LevelTwoMonitorFreq attribute (default: 30), which is common to all resources, should always be less than the token expiry time. For example, if the token is valid for 45 minutes and the MonitorInterval attribute of OpenStackAuth resource is set to 60 seconds, the LevelTwoMonitorFreq value should be less than 45.

Chapter 4

# Configuring the service groups for OpenStack

This chapter includes the following topics:


- Before configuring the service groups for OpenStack
- Known issues and limitations
- Configuring service groups for OpenStack


## Before configuring the service groups for OpenStack

Before you configure the OpenStack service group, you must:

- Verify that InfoScale Availability or InfoScale Enterprise is installed and configured on all the cluster nodes where you want to configure the service group.
  For more information, refer to the *Veritas InfoScale Installation Guide*.
- Verify that the InfoScale agents for OpenStack are installed and configured identically on all the cluster nodes.
  See "Installing the agents in an InfoScale environment" on page 12.

## Known issues and limitations

### Known issue with OpenStackVol

When TLS/SSL is enabled, you may encounter a security certificate issue for which an error is written to the OpenStackVol agent log.

For example:

```
2018/04/17 05:16:23 VCS ERROR V-16-55049-20013 OpenStackVol:
openstackvol_res1:monitor:Failed to get volume details for
5e7751c7-ecf8-4d3e-8d05-495559e4d094 [500] [Can't verify
SSL peers without knowing which Certificate Authorities to trust

This problem can be fixed by either setting the PERL_LWP_SSL_CA_FILE
environment variable or by installing the Mozilla::CA module.

To disable verification of SSL peers set the PERL_LWP_SSL_VERIFY_HOSTNAME
environment variable to 0.  If you do this you can't be sure that you
communicate with the expected peer.]
```

Perform the following tasks to work around this issue:

1. Stop HAD from any cluster node by using:

   ```
   # hastop -all
   ```

   or

   ```
   # hastop -all - force
   ```

2. Add the following lines to the `custom_vcsenv` file, which is located at `/opt/VRTSvcs/bin/`:

   - On SLES 11 or SLES 12

     ```
     PERL_LWP_SSL_CA_PATH=/etc/ssl/certs/
     export PERL_LWP_SSL_CA_PATH
     ```

   - On RHEL 6 or RHEL 7

     ```
     HTTPS_CA_FILE=/etc/pki/tls/certs/ca-bundle.crt
     export HTTPS_CA_FILE
     ```

   The value of `PERL_LWP_SSL_CA_PATH` or `HTTPS_CA_FILE` may vary for every system.

   If the `custom_vcsenv` file does not exist, create it by using:

   ```
   # cd /opt/VRTSvcs/bin

   # touch custom_vcsenv

   # chmod 750 custom_vcsenv
   ```

3. Start HAD on each cluster node by using:

   ```
   # hastart
   ```

### Limitations with OpenStackIP

If the PrivateIP attribute is specified and the UseNewPort attribute is set to 0 (zero) for the OpenStackIP agent, the following limitations apply:

- The agent appends the value of PrivateIP to the `--fixed_ips` field of the device port. By default, `--fixed_ips` can accommodate only five IP addresses. It already has one address, which is the base address of the device.
  The number of IP addresses per port can be changed in a Neutron configuration. For details, see the value of `max_fixed_ips_per_port` in the `/etc/neutron/neutron.conf` file.
  **Workaround:** If the cluster configuration has more than four OpenStackIP resources for the same device, from the fifth OpenStackIP resource onwards, set the UseNewPort attribute to 1.
- The base IP address of the device in the instance network configuration must be static. If it is set to DHCP, the virtual IP address may be set as the base IP of the device after restart.

### Limitations with OpenStackVol

- Scenario: The compute instance on which the cluster is currently active goes into the PAUSED or the SUSPENDED state
  VCS sets the node to the FAULTED state and then initiates the failover of the service group to another available node. However, it cannot bring the OpenStackVol resource online on the other instance, because it cannot detach the associated volumes from the PAUSED or the SUSPENDED instance.
- Scenario: An LVMVolumeGroup resource is configured and the associated Cinder volumes get detached outside of VCS control
  During the offline operation, VCS fails to export the volume group, and failover of the associated service group fails.

# Configuring service groups for OpenStack

Configuring an OpenStack IP for high availability involves the following tasks:

1. Creating a service group for the OpenStackAuth agent in parallel to the application or the networking service group
2. Creating a service group for the OpenStackIP agent, the OpenStackVol agent, or both

   or

   Adding the OpenStackIP resource, or the OpenStackVol resource, or both to the application or the networking service group

3. Defining the dependency between the OpenStackAuth and the application or the networking service groups

**To create a parallel service group for the OpenStackAuth agent**

**1** Create an OpenStackAuth service group.

```
#hagrp -add OpenStackAuthSvcGrp
```

Here, *OpenStackAuthSvcGrp* represents the OpenStackAuth service group name (for example, **fin_app_os_auth_svc_grp**).

**2** Modify the AutoStartList and the SystemList attributes of the service group to specify the systems on which the service group can be automatically brought online and their order of priority.

```
#hagrp -modify fin_app_os_auth_svc_grp AutoStartList sysA sysB

#hagrp -modify fin_app_os_auth_svc_grp SystemList sysA 0 sysB 1
```

**3** Specify that *OpenStackAuthSvcGrp* is a parallel service group.

```
#hagrp -modify OpenStackAuthSvcGrp Parallel 1
```

**4** Add an authentication resource to the service group.

```
#hares -add OpenStackAuthRes OpenStackAuth OpenStackAuthSvcGrp
```

Here, *OpenStackAuthRes* represents the authentication resource name (for example, **fin_app_os_auth_res**).

**5** Specify the appropriate attribute values for the OpenStackAuth resource. For example:

```
#hares -modify fin_app_os_auth_res Enabled 1

#hares -modify fin_app_os_auth_res KeystoneEP
http://10.209.246.XXX:5000/v2.0

#hares -modify fin_app_os_auth_res Tenant Project1

#hares -modify fin_app_os_auth_res UserName XYZA

#hares -modify fin_app_os_auth_res Password BPHnEPsdeHChDHe

#hares -modify fin_app_os_auth_res Region RegionOne
```

**To add the OpenStackIP resource to the application or the network service group**

1. Add the OpenStackIP resource.

   ```
   # hares -add OpenStackIPRes OpenStackIP AppSvcGrp
   ```

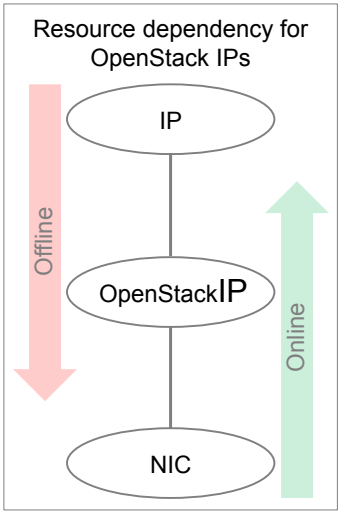
   Here, *OpenStackIPRes* represents the OpenStackIP resource name (for example, **fin_app_os_ip_res**) and *AppSvcGrp* represents the application service group name (for example, an Oracle database).

2. Specify the appropriate attribute values for the OpenStackIP resource. For example:

   ```
   #hares -modify fin_app_os_ip_res Enabled 1

   #hares -modify fin_app_os_ip_res PrivateIP "172.18.0.100"

   #hares -modify fin_app_os_ip_res FloatingIP "10.209.246.122"

   #hares -modify fin_app_os_ip_res Device eth0
   ```

   For more information on creating and modifying the NIC and the IP resource attributes, see the *Cluster Server Bundled Agents Reference Guide*.

3. Create links between the OpenStackIP, IP, and NIC resources.




   ```
   #hares -link OpenStackIPRes NICRes

   #hares -link IPRes OpenStackIPRes
   ```

   Here, *NICRes* represents the NIC device name and *IPRes* represents the IP resource name.

**To add the OpenStackVol resource to the application or the network service group**

1. Add the OpenStackVol resource.

   ```
   # hares -add OpenStackVolRes OpenStackVol AppSvcGrp
   ```

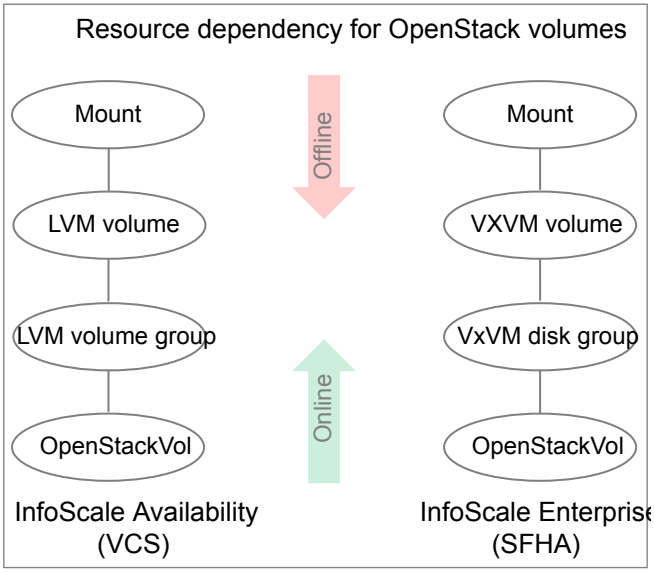
   Here, *OpenStackVolRes* represents the OpenStackVol resource name (for example, **fin_app_os_vol_res**) and *AppSvcGrp* represents the application service group name (for example, an Oracle database).

2. Specify the appropriate attribute values for the OpenStackIP resource. For example:

   ```
   #hares -modify fin_app_os_vol_res Enabled 1
   ```

   ```
   #hares -modify fin_app_os_vol_res VolumeIds
   c5b34a2b-111f-40e2-865a-a4225e2ddbdf
   4d676585-0908-47e6-b964-4fa719ed19b8
   ```

3. Create links between the OpenStackVol, the disk group, and the mount resources.




```
#hares -link diskGroupRes OpenStackVolRes
```

```
#hares -link volRes diskGroupRes
```

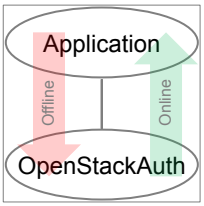
```
#hares -link mountRes volRes
```

Here,

- *diskGroupRes* represents the disk group resource name
- *volRes* represents the volume resource name
- *mountRes* represents the mount resource name

**To create the dependency between the service groups**

- Specify that the OpenStackIP (or application or network) service group depends on the OpenStackAuth service group.




```
# hagrp -link AppSvcGrp OpenStackAuthSvcGrp online local firm
```

Here, *AppSvcGrp* represents the application service group name (for example, an Oracle database) and *OpenStackAuthSvcGrp* represents the OpenStackAuth service group name.

Appendix A

# Sample configurations

This appendix includes the following topics:


- About sample configurations for the agents for OpenStack
- Sample agent type definition for OpenStack
- Sample service group configurations for OpenStack
- Sample resource type definitions for OpenStackIP
- Sample service group configuration for OpenStackAuth
- Sample cluster configurations using OpenStack agents


## About sample configurations for the agents for OpenStack

The sample configurations graphically depict the resource types, resources, and resource dependencies within service groups. Review these dependencies carefully before configuring the agents for OpenStack. For more information about these resource types, see the *Cluster Server Bundled Agents Reference Guide*.

## Sample agent type definition for OpenStack

The types definition file for the OpenStack agents is located at:

```
/etc/VRTSvcs/conf/config/OpenStackTypes.cf
```

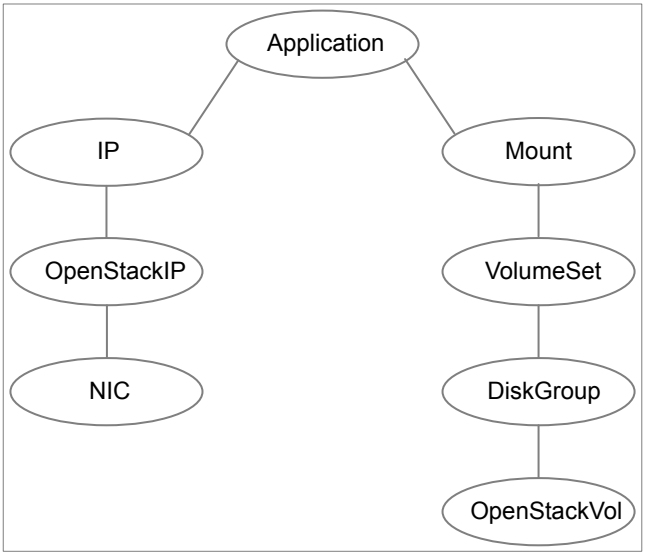
A sample types definition is as follows:

```
type OpenStackAuth (
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/OpenStackAuth"
    static str AgentFile = "/opt/VRTSvcs/bin/Script60Agent"
```

```
    static int LevelTwoMonitorFreq = 30
    static str ArgList[] = { Tenant, Username, Password, KeystoneEP,
     Region }
    str Tenant
    str Username
    str Password
    str KeystoneEP
    str Region
)

type OpenStackIP (
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/OpenStackIP"
    static str AgentFile = "/opt/VRTSvcs/bin/Script60Agent"
    static str ArgList[] = { PrivateIP, FloatingIP, OverlayIP, BaseIP,
     Device, AllowedAddressPair, RouteTableNames, UseNewPort }
    str PrivateIP
    str FloatingIP
    str OverlayIP
    str BaseIP
    str Device
    str AllowedAddressPair
    keylist RouteTableNames
    boolean UseNewPort = 1
)

type OpenStackVol (
     static str AgentDirectory = "/opt/VRTSagents/ha/bin/OpenStackVol"
     static str AgentFile = "/opt/VRTSvcs/bin/Script60Agent"
     static int OnlineRetryLimit = 1
     static str ArgList[] = { VolumeIds }
     str VolumeIds[]
     temp str InstanceId
)
```

# Sample service group configurations for OpenStack

The following graphic depicts a sample application service group with the OpenStack agents:




In a cluster configuration, the OpenStackAuth service group is created in parallel with the application service group. The application service group has an online-local-hard dependency on the OpenStackAuth service group.

# Sample resource type definitions for OpenStackIP

You can use the following samples for reference to configure an OpenStackIP resource for each of the corresponding HA scenarios.

## Scenario 1: Failover of a floating IP

FloatingIP associated with BaseIP of the device:

```
OpenStackIP OpenStackIP_Res (
    FloatingIP = "10.209.x.x"
    BaseIP @sys1= "192.168.x.x"
    BaseIP @sys2 = "192.168.x.x"
    BaseIP @sys3 = "10.x.0.12"
    BaseIP @sys4 = "10.x.0.8"
    Device = eth0
)
```

## Scenario 2: Failover of a private IP to another instance or node within the same subnet

PrivateIP without FloatingIP, and UseNewPort is set to the default value:

```
OpenStackIP OpenStackIP_Res (
    PrivateIP = "192.168.x.x"
    Device = eth0
)
```

PrivateIP associated with FloatingIP, and UseNewPort is set to the default value:

```
OpenStackIP OpenStackIP_Res (
    PrivateIP = "192.168.x.x"
    FloatingIP = "10.209.x.x"
    Device = eth0
)
```

Without creating a new Neutron port for PrivateIP (VIP):

```
OpenStackIP OpenStackIP_Res (
    PrivateIP = "192.168.x.x"
    FloatingIP = "10.209.x.x"
    Device = eth0
    UseNewPort = 0
)
```

PrivateIP using an IPv6 address, and UseNewPort is set to the default value:

```
OpenStackIP OpenstackIP_res (
    PrivateIP = "2009:db2::6"
    Device = eth2
)
```

## Scenario 3: If the configuration requires more than five PrivateIPs or VIPs

Allowed CIDR on the device port:

```
OpenStackIP OpenStackIP_Res (
    AllowedAddressPair = "172.168.0.0/24"
    Device = eth0
)
```

## Scenario 4: Failover of an overlay IP to a guest VM across the subnet or network

Associate OverlayIP with BaseIP of the device:

```
OpenStackIP OpenStackIP_Res (
    OverlayIP = "10.100.x.x/32"
    BaseIP @sys1 = "192.168.x.x"
```

```
    BaseIP @sys2 = "192.168.x.x"
    BaseIP @sys3 = "10.212.x.x"
    BaseIP @sys4 = "10.212.x.x"
    Device = eth0
    RouteTableNames = { router1 }
)
```

# Sample service group configuration for OpenStackAuth

You can use the following sample for reference to configure an OpenStackAuth service group where the endpoint is an IPv4 address.

```
group OpenStackAuthGrp (
    SystemList = { sys1= 0, sys2 = 1, sys3 = 2, sys4 = 3 }
    Parallel = 1
    AutoStartList = { sys1, sys2, sys3, sys4 }
)

OpenStackAuth authres (
    Tenant = Project_1
    Username = FirstUser
    Password = gumSjuJmhMimJ
    KeystoneEP = "http://10.x.x.x:5000/v2.0"
    Region = RegionOne
    LevelTwoMonitorFreq = 40
)
```

You can use the following sample for reference to configure an OpenStackAuth service group where the endpoint is an IPv6 address.

```
group OpenStackAuthGrp (
    SystemList = { sys1= 0, sys2 = 1, sys3 = 2, sys4 = 3 }
    Parallel = 1
    AutoStartList = { sys1, sys2, sys3, sys4 }
)

OpenStackAuth authres (
    Tenant = Project_1
    Username = FirstUser
    Password = gumSjuJmhMimJ
    KeystoneEP = "http://[2620:128:f0a2:900e::157]:5000/v2.0"
    Region = RegionOne
```

```
    LevelTwoMonitorFreq = 40
)
```

# Sample cluster configurations using OpenStack agents

Here are some sample `main.cf` files that you can refer to when you configure an application for failover in an OpenStack environment:

## Sample configuration with private IP

The following sample configuration depicts a cluster where all the nodes are located in the same subnet. It supports the failover of a private IP from one node to another.

```
include "OracleASMTypes.cf"
include "types.cf"
include "CRSResource.cf"
include "CSSD.cf"
include "Db2udbTypes.cf"
include "MultiPrivNIC.cf"
include "OracleTypes.cf"
include "OpenStackTypes.cf"
include "PrivNIC.cf"
include "SybaseTypes.cf"

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

system sys1(
        )

system sys2 (
        )



group db2_grp (
        SystemList = { sys1= 0, sys2 = 1 }
        AutoStartList = { sys1, sys2 }
        )
```

```
        IP db2_ipRes (
                Device = eth0
                Address = "192.168.0.x"
                NetMask = "255.255.255.0"
                )


        NIC db2_nicRes (
                Device = eth0
                )

        OpenStackIP db2_openstackipRes (
                PrivateIP = "192.168.0.x"
                Device = eth0
                )


        requires group OpenStackAuthGrp online local firm
        db2_ipRes requires db2_openstackipRes
        db2_openstackipRes requires db2_nicRes


group OpenStackAuthGrp (
        SystemList = { sys1= 0, sys2 = 1}
        Parallel = 1
        AutoStartList = { sys1, sys2 }
        )

        OpenStackAuth authres (
                Tenant = Project_1
                Username = Firstuser
                Password = gumSjuJmhMimJ
                KeystoneEP = "http://10.209.x.x:5000/v2.0"
                Region = RegionOne
                )
```

## Sample configuration with floating IPs

The following sample configuration depicts a cluster where all the nodes are located in the same subnet. It supports the failover of multiple private IPs or associating some private IPs with floating IPs.

```
include "OracleASMTypes.cf"
include "types.cf"
include "CRSResource.cf"
include "CSSD.cf"
include "Db2udbTypes.cf"
include "MultiPrivNIC.cf"
include "OracleTypes.cf"
include "OpenStackTypes.cf"
include "PrivNIC.cf"
include "SybaseTypes.cf"

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

system sys1(
        )

system sys2 (
        )



group db2_grp (
        SystemList = { sys1= 0, sys2 = 1 }
        AutoStartList = { sys1, sys2 }
        )


        IP db2_ipRes (
                Device = eth0
                Address = "192.168.x.10"
                NetMask = "255.255.255.0"
                )

        IP db2_ipRes_2 (
                Device = eth0
                Address = "192.168.x.11"
                NetMask = "255.255.255.0"
                )
```

```
        NIC db2_nicRes (
                Device = eth0
                )

        OpenStackIP db2_openstackipRes (
                PrivateIP = "192.168.x.10"
                FloatingIP = "10.16.x.x"
                Device = eth0
                AllowedAddressPair = "192.168.x.0/24"
                UseNewPort = 0
                )


        requires group OpenStackAuthGrp online local firm
        db2_ipRes requires db2_openstackipRes
        db2_ipRes_2 requires db2_openstackipRes
        db2_openstackipRes requires db2_nicRes


group OpenStackAuthGrp (
        SystemList = { sys1= 0, sys2 = 1}
        Parallel = 1
        AutoStartList = { sys1, sys2 }
        )

        OpenStackAuth authres (
                Tenant = Project_1
                Username = Firstuser
                Password = gumSjuJmhMimJ
                KeystoneEP = "http://10.209.x.x:5000/v2.0"
                Region = RegionOne
                )
```

## Sample configuration with Cinder volume

The following sample configuration depicts a cluster where the OpenStackVol agent is used to manage a Cinder volume on which an Oracle database data is hosted.

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

```
include "OpenStackTypes.cf"
include "SybaseTypes.cf"

cluster cluster_cinder (
 SecureClus = 1
 )

system instance-1 (
 )

system instance-2 (
 )

group OpenStackAuthGrp (
 SystemList = { instance-1 = 0, instance-2 = 1 }
 Parallel = 1
 AutoStartList = { instance-1, instance-2 }
 )

 OpenStackAuth authres (
  Tenant = Firstproject
  Username = Firstuser
  Password = JXPvMXmPKpLPm
  KeystoneEP = "http://10.209.82.70:5000/v2.0 "
  Region = RegionOne
  LevelTwoMonitorFreq = 30
  )



 // resource dependency tree
 //
 // group OpenStackAuthGrp
 // {
 // OpenStackAuth authres
 // }


group OracleDBGrp (
 SystemList = { instance-1 = 0, instance-2 = 1 }
 AutoStartList = { instance-1, instance-2 }
 )
```

```
IP ip_res (
 Device = eth0
 Address = "192.169.1.100"
 NetMask = "255.255.252.0"
 )

LVMLogicalVolume lvmvol_res (
 LogicalVolume = testlv
 VolumeGroup = testvg
 )

LVMVolumeGroup lvmvg_res (
 VolumeGroup = testvg
 )

Mount mount_res (
 MountPoint = "/LVMNT"
 BlockDevice = "/dev/mapper/testvg-testlv"
 FSType = ext4
 FsckOpt = "-y"
 )

NIC nic_res (
 Device = eth0
 )

Netlsnr netlsnr_res (
 Owner = oracle
 Home = "/oracle/app/dbbase/dbhome"
 TnsAdmin = "/oracle/app/dbbase/dbhome/network/admin"
 )

OpenStackIP openstackip_res (
 PrivateIP = "192.169.1.100"
 FloatingIP = "10.209.80.243"
 Device = eth0
 )

OpenStackVol openstackvol_res (
 VolumeIds = { 5e6d7424-c533-447f-8eab-ea6d477cde52,
   fa07508f-c22e-4259-b6f4-d9ac67e0a5e4 }
 )
```

```
Oracle oracledb_res (
 Sid = addb
 Owner = oracle
 Home = "/oracle/app/dbbase/dbhome"
 )

requires group OpenStackAuthGrp online local firm
ip_res requires openstackip_res
lvmvg_res requires openstackvol_res
lvmvol_res requires lvmvg_res
mount_res requires lvmvol_res
netlsnr_res requires ip_res
netlsnr_res requires oracledb_res
openstackip_res requires nic_res
oracledb_res requires mount_res


// resource dependency tree
//
// group OracleDBGrp
// {
// Netlsnr netlsnr_res
//     {
//     IP ip_res
//         {
//         OpenStackIP openstackip_res
//             {
//             NIC nic_res
//             }
//         }
//     Oracle oracledb_res
//         {
//         Mount mount_res
//             {
//             LVMLogicalVolume lvmvol_res
//                 {
//                 LVMVolumeGroup lvmvg_res
//                     {
//                     OpenStackVol openstackvol_res
//                     }
//                 }
//             }
//         }
```

```
//       }
// }
```