Veritas InfoScale 7.4.2 Installation, Upgrade, and Configuration Using Ansible - Linux

Last updated: 2020-06-01

Legal Notice

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
2625 Augustine Drive
Santa Clara, CA 95054
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:
infoscaledocs@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

Preparing to use Ansible in InfoScale

This chapter includes the following topics:

Inroduction to Ansible

Ansible is a popular configuration management tool that automates various configuration and deployment operations in your environment. Ansible playbooks are files written in the YAML format, which contains human-readable code. Ansible playbooks can be used to define operations in your environment.

Veritas now provides an Ansible module that can be used to run Ansible playbooks to deploy Veritas InfoScale products and configure features.

Table 1-1 Operations that can be performed by using Ansible

Deployment-related operations	Feature-related operations
■ Installation ■ Licensing ■ Component configuration ■ Starting a product ■ Stopping a product ■ Full upgrade ■ Rolling upgrade ■ Uninstallation	■ Configuring a Cluster File System (CFS) resource ■ Creating a disk group volume ■ Configuring an File System (FS) resource ■ Configuring VVR and CVR ■ Configuration of LLT over UDP ■ Adding node ■ Deleting node ■ Creating a single-node cluster ■ Disk Tagging ■ Tagging based volume creation

Supported platforms

You can use Ansible to deploy and configure Veritas InfoScale on all RHEL distributions supported by InfoScale.

Supported Ansible version

Veritas InfoScale products can be deployed and configured using Ansible version 1.9.2 or later.

Downloading Ansible modules for InfoScale

Refer to the following link to download the Ansible modules, playbook templates, and user guide for using Ansible in Veritas InfoScale.

- https://sort.veritas.com/utility/ansible

Download and save the Ansible modules to the following location on your Ansible server.

```
/usr/share/ansible/plugins/modules/
```

Before you begin

Ensure that the following prerequisites are met in your environment:

- Ansible requires passwordless SSH communication to be established between the Ansible server and nodes to be managed by the Ansible server.

Note: The user can use the pl utility to set up the SSH and RSH connections automatically.

- Ensure that Python 2.6 or later is installed and configured on all machines in the environment. Use a Python Jinja2 template engine of version 8 or later.

Using site_facters module

The site_facters Ansible module is used to collect system-related data from all nodes in a cluster. You must use the site_facters module in your playbooks, while performing all operations in InfoScale.

Chapter 2

Installing using Ansible

This chapter includes the following topics:

Ansible modules for Installing InfoScale

Use the following Ansible modules in your playbooks to perform installation-related operations in the InfoScale environment. Refer to the following table for a list of modules, along with a sample playbook, used for each of the operations:

Table 2-1 Installation-related operations

<table>
<thead>
<tr><th>Operation</th><th>Required modules</th><th>Sample playbook</th></tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>- site_facters
- yum</td>
<td><pre>

- hosts: cpicluster11
  gather_facts: false
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: Install InfoScale Enterprise
    veritas_infoscale:
     module: yum
     repository_name: RepositoryName
     repository_baseurl: RepositoryBaseURL
     gpgcheck: 1
     gpgkey: http://xx.xxx.xxx.xx/
rpms/RPM-GPG-KEY-veritas-infoscale7
     product: 'ENTERPRISE'
     product_version: 7.4.2
     facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"
     state: present
</pre></td>
</tr>
<tr>
<td>Licensing</td>
<td>- site_facters
- licensing</td>
<td><pre>

- hosts: cpicluster11
  gather_facts: false
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: License
    veritas_infoscale:
     module: licensing
     state: present
     product_version: 7.4.2
     license: 'ENTERPRISE'
     facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"
</pre></td>
</tr>
</tbody>
</table>

Table 2-1 Installation-related operations *(continued)*

<table>
<tr><th>Operation</th><th>Required modules</th><th>Sample playbook</th></tr>
<tr><td>Uninstall</td><td>- site_facters
- yum</td><td><pre>

- hosts: cpicluster11
  gather_facts: False
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: Uninstall InfoScale Enterprise
    veritas_infoscale:
     module: yum
     product: 'ENTERPRISE'
     product_version: 7.4.2
     seednode: objstorer820-1-vm16
     state: absent
     facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"
</pre></td></tr>
</table>

List of pre-defined keywords

Refer to the following tables for a list of keywords that are used with each of the installation-related modules.

yum

The yum module is used to install or uninstall InfoScale products using yum. Use the following keywords while referencing the module in your playbook.

Table 2-2 Yum module keywords

Name	Description	Mandatory/optional
repository_name	Name of the yum repository used to install InfoScale rpms. **Example**: `repo-InfoScale742`	Mandatory If the Base URL is provided, the system will create a repository with the specified repository name.

Table 2-2 Yum module keywords *(continued)*

Name	Description	Mandatory/optional
repository_baseurl	Specifies the URL to the directory where the repodata of a repository is located. **Example**: `http://`*`xx.xxx.xxx.xx`*`/rpms/`	Optional This keyword is not required if you are using a yum repository already configured on the system, and are providing the repository name.
gpgcheck	Specify whether to check the integrity of the yum packages by using the gpgkey provided with the InfoScale installation media. This is a boolean variable and must be specified using 0 or 1. By default the value is set to 0.	Optional

gpgkey	Specifies the location of the gpgkey (typically located in the rpms directory of the installation media).
	Example:
	`http://`*`xx.xxx.xxx.xx`*`/rpms/`
	`RPM-GPG-KEY-veritas-infoscale7`
		Mandatory if gpgcheck is 1, otherwise optional

product	Specifies the name of the product you want to install. **Examples**: `ENTERPRISE`, `AVAILABILITY`, `STORAGE`, or `FOUNDATION`.	Mandatory
product_version	Specifies the version of the product that you want to install or upgrade. **Example**: 7.4.2	Mandatory
state	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional

licensing

The licensing module is used to apply licenses in InfoScale. Use the following keywords while referencing the module in your playbook.

Table 2-3 Licensing module keywords

Name	Description	Mandatory/optional
state	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional
product_version	Specifies the version of the product that you want to install or upgrade. **Example**: 7.4.2	Mandatory
license	Specifies the path to the slf license file to be registered on the system. Ensure that the license file is accessible. The license file must be available on each node. If you are performing a keyless installation or upgrade you can simply enter the product name. **Examples**: ■ `/license_key/Unix/perpetual/`*`xxxxxxxxxxxxxxxxx`*`.slf` ■ `ENTERPRISE` ■ `AVAILABILITY`	Mandatory

Configuring InfoScale product components

This chapter includes the following topics:

- Ansible modules for configuring InfoScale
- List of pre-defined keywords

Ansible modules for configuring InfoScale

Use the following Ansible modules in your playbooks to configure Veritas InfoScale product components. Refer to the following table for a list of modules , along with a sample playbook, used for each of the configuration-related operations:

Table 3-1 Component configuration-related operations

Operation	Required modules	Sample playbook
Component configuration	■ site_facters ■ *<component_name>* _config Example: sfcfsha_config sfha_config	

Table 3-1 Component configuration-related operations *(continued)*

Operation	Required modules	Sample playbook
		Configuring sfcfsha

```
---
- hosts: cpicluster11
  gather_facts: false
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name:  Configure Enterprise
    veritas_infoscale:
     module: sfcfsha_config
     cluster_name: clust_cpi9
     systems: [objstorer820-1-vm16,objstorer820-1-vm17]
     cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61
     component: sfcfsha
     product_version: '7.4.2'
     license: 'ENTERPRISE'
     method: ethernet
     enable_lgf: 1
     seednode: dl380g10-09-vm7
     state: present
     private_link: eth1,eth2
     low_priority_link: eth0
     vcs_clusterid: 23838
     facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"
```

Configuring sfha

```
---
- hosts: cpicluster11
  gather_facts: true
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name:  Configure Enterprise
    veritas_infoscale:
```

Table 3-1 Component configuration-related operations *(continued)*

Operation	Required modules	Sample playbook

		`module: sfha_config`
		`cluster_name: clust_cpi9`
		`systems: [objstorer820-1-vm16,objstorer820-1-vm17]`
		`cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61`
		`component: sfha`
		`product_version: '7.4.2'`
		`license: 'ENTERPRISE'`
		`method: ethernet`
		`enable_lgf: 1`
		`seednode: dl380g10-09-vm7`
		`state: present`
		`private_link: eth1,eth2`
		`low_priority_link: eth0`
		`vcs_clusterid: 23838`
		`facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"`
		Configuring vcs

Table 3-1 Component configuration-related operations *(continued)*

Operation	Required modules	Sample playbook

```
---
- hosts: cpicluster11
  gather_facts: true
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name:  Configure Enterprise
    veritas_infoscale:
     module: vcs_config
     cluster_name: clust_cpi9
     systems: [dl380g10-12-kvm-04,dl380g10-12-kvm-03]
     cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61
     component: vcs
     product_version: '7.4.2'
     license: 'ENTERPRISE'
     method: ethernet
     enable_lgf: 1
     seednode: dl380g10-09-vm7
     state: present
     private_link: eth1,eth2
     low_priority_link: eth0
     vcs_clusterid: 23838
```

Configuring sf

Table 3-1 Component configuration-related operations *(continued)*

<table>
<tr><th>Operation</th><th>Required modules</th><th>Sample playbook</th></tr>
<tr><td></td><td></td><td><pre>

- hosts: cpicluster11
  gather_facts: true
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: Configure Enterprise
    veritas_infoscale:
     module: sf_config
     component: sf
     product_version: '7.4.2'
     license: 'ENTERPRISE'
     seednode: dl380g10-09-vm7
     state: present
</pre></td></tr>
<tr><td>Starting a product</td><td>- site_facters
- process</td><td><pre>

- hosts: cpicluster11
  gather_facts: false
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: Start Product
    veritas_infoscale:
     module: process
     component: sfcfsha
     product: enterprise
     product_version: '7.4.2'
     seednode: objstorer820-1-vm17
     state: present
     facters: "{{ groups['all'] |map('extract',
     hostvars, ['facts','infoscale_facts'])| select()|list }}"
</pre></td></tr>
</table>

Table 3-1 Component configuration-related operations *(continued)*

Operation	Required modules	Sample playbook

Stopping a product	■ site_facters
	■ process
		`---`
		`- hosts: cpicluster11`
		`  gather_facts: false`
		`  any_errors_fatal: true`
		`  tasks:`
		`  - name: Facters`
		`    veritas_infoscale:`
		`     module: site_facters`
		`    register: facts`
		`  - name: Start Product`
		`    veritas_infoscale:`
		`     module: process`
		`     component: sfcfsha`
		`     product: enterprise`
		`     product_version: '7.4.2'`
		`     seednode: objstorer820-1-vm17`
		`     state: absent`
		`     facters: "{{ groups['all'] |map('extract',`
		`     hostvars, ['facts','infoscale_facts'])| select()|list }}"`

List of pre-defined keywords

Refer to the following tables for a list of the keywords that are used with each of the configuration-related modules.

sfcfsha_config

The sfcfsha_config module is used configure the component in InfoScale . Use the following keywords while referencing the module in your playbook.

Table 3-2 Sfcfsha_config module keywords

Name	Description	Mandatory/optional
cluster_name:	Define a name for the cluster that you want to deploy. **Example**: `Cluster1`	Mandatory
cluster_uuid:	Define a unique alphanumeric ID to assign to the cluster you want to deploy. **Example**: `c7c2d65e-057f-11e8-a32c-c094107f3b61`	Mandatory

Table 3-2 Sfcfsha_config module keywords *(continued)*

Name	Description	Mandatory/optional
component:	Specify which components you want to configure in your product. Note that the product license acquired must support the required components. **Example**: `SF`, `VCS`, or `SFCFSHA`	Mandatory
method:	Specify the communication protocol that you want to deploy in the cluster. **Example**: `ethernet`, or `udp`	Mandatory
enable_lgf:	Specify either **0** or **1**. **0** creates a single-node cluster on each system provided in the playbook run, without enabling LLT, GAB, or I/O fencing. **1** (default value) creates a multi node cluster of all the servers in the playbook run.	Optional
seednode:	Select any node from the cluster that will be used to run commands related to the operations of that cluster. Ensure that you enter the host name as provided in the in the `/etc/ansible/hosts` file.	Mandatory
state:	Specifies what state the configuration should be in after the task is completed. The value for this keyword can be either present or absent.	Mandatory
private_link:	Lists the name of the NICs that the heartbeat link uses on each of the nodes in the cluster.	Mandatory
low_priority_link:	Lists the name of the NICs that the low priority heartbeat link uses on each of the nodes in the cluster.	Optional

Table 3-2 Sfcfsha_config module keywords *(continued)*

<table>
<tr><th>Name</th><th>Description</th><th>Mandatory/optional</th></tr>
<tr><td>udp_link</td><td>This attribute is only required if you want to configure LLT over a UDP link (method must be set to <code>udp</code>). Specify information for configuring a UDP link. List the name, IP address, and port number of the NICs that the UDP link uses on each of the nodes in the cluster.
<b>Example</b>:<pre>
udp_link:
    - system: 'dl380g10-12-kvm-24'
      heartbeat_link:
          - nic: eth1
            ip: 192.xxx.x.x/xx
            port: 50000
          - nic: eth2
            ip: 192.xxx.x.x/xx
            port: 50001
    - system: 'dl380g10-12-kvm-20’
      heartbeat_link:
          - nic: eth1
            ip: 192.xxx.x.x/xx
            port: 50000
          - nic: eth2
            ip: 192.xxx.x.x/xx
            port: 50001
</pre></td><td>Optional</td></tr>
<tr><td>vcs_clusterid:</td><td>Define a unique number to be assigned to the cluster.</td><td>Mandatory</td></tr>
</table>

process

The process module is used to start and stop component processes. Use the following keywords while referencing the module in your playbook.

Table 3-3 Process module keywords

Name	Description	Mandatory/optional
component:	Specify which components you want to configure in your product. Note that the product license acquired must support the required components. **Example**: `SF`, `VCS`, or `SFCHA`	Mandatory
product:	Specifies the name of the product you want to install. **Examples**: `ENTERPRISE`, `AVAILABILITY`, `STORAGE`, or `FOUNDATION`.	Mandatory

Table 3-3 Process module keywords *(continued)*

Name	Description	Mandatory/optional
product_version:	Specifies the version of the product that you want to install or upgrade. **Example**: `http://`*`xx.xxx.xxx.xx`*`/rpms/`	Mandatory
state:	Specifies what state the process should be in after the task is completed. The value for this keyword can be either **present** or **absent**. If you do not give any value for this parameter, by default, the state of the package is set to **present**, and the package will be installed.	Optional

Chapter 4

Upgrading InfoScale using Ansible

This chapter includes the following topics:

Ansible modules for upgrading InfoScale

Use the following Ansible modules in your playbooks to perform upgrade-related operations. Refer to the following table for a list of modules, along with a sample playbook, used for each of the operations:

Table 4-1 Upgrade-related operations

Operation	Required modules	Sample playbook

Full upgrade	■ site_facters
	■ upgrade
		`---`
		`- hosts: cpicluster7`
		`  gather_facts: false`
		`  any_errors_fatal: true`
		`  tasks:`
		`  - name: Facters`
		`    veritas_infoscale:`
		`     module: site_facters`
		`    register: facts`
		`  - name: Upgrade Infoscale to 7.4.2`
		`    veritas_infoscale:`
		`     module: upgrade`
		`     repository_name: `*`RepositoryName`*
		`     product_version: 7.4.2`
		`     repository_baseurl: `*`RepositoryBaseURL`*
		`     gpgcheck: 1`
		`     gpgkey: http://`*`xx.xxx.xxx.xx`*`/sde2/7.4.2/dvd1-redhatlinux/`
		`rhel7_x86_64/rpms/RPM-GPG-KEY-veritas-infoscale7`
		`     license: /license_key/Unix/perpetual/`*`xxxxxxxxxxxx`*`.slf`
		`     seednode: dl380g10-09-vm7`
		`     edgeserver_hostname: telemetry.veritas.com`
		`     edgeserver_port: 443`
		`     state: present`
		`     facters: "{{ groups['all'] |map('extract',`
		`hostvars, ['facts','infoscale_facts'])| select()|list }}"`

Table 4-1 Upgrade-related operations *(continued)*

Operation	Required modules	Sample playbook
Rolling upgrade	■ site_facters ■ ru_phase1 ■ ru_phase2	

Table 4-1 Upgrade-related operations *(continued)*

Operation	Required modules	Sample playbook

```
---
- hosts: cpicluster11
  gather_facts: False
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts

- hosts: cpicluster11
  gather_facts: False
  any_errors_fatal: true
  serial: 1
  tasks:
  - name: Rolling Upgrade(phase1) Infoscale to 7.4.2
    veritas_infoscale:
     module: ru_phase1
     repository_name: RepositoryName
     product_version: 7.4.2
     repository_baseurl: RepositoryBaseURL
     gpgcheck: 1
     gpgkey: http://xx.xxx.xxx.xx/rpms/
RPM-GPG-KEY-veritas-infoscale7
     license: ENTERPRISE
     edgeserver_port: '443'
     edgeserver_hostname: 'telemetry.veritas.com'
     seednode: objstorer820-1-vm16
     state: present
     facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"

- hosts: cpicluster11
  gather_facts: False
  any_errors_fatal: true
  tasks:
  - name: Rolling Upgrade(phase2) Infoscale to 7.4.2 phase2
    veritas_infoscale:
    module: ru_phase2
    repository_name: RepositoryName
    product_version: 7.4.2
    repository_baseurl: RepositoryBaseURL
```

Table 4-1 Upgrade-related operations *(continued)*

<table>
<tr><th>Operation</th><th>Required modules</th><th>Sample playbook</th></tr>
<tr><td></td><td></td><td><pre>
    gpgcheck: 1
    gpgkey: http://xx.xxx.xxx.xx/
Infoscale/7.4.2/rpms/RPM-GPG-KEY-veritas-infoscale7
    license: ENTERPRISE
    seednode: objstorer820-1-vm16
    edgeserver_hostname: telemetry.veritas.com
    edgeserver_port: 443
    state: present
    facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"
</pre></td></tr>
</table>

List of pre-defined keywords

Refer to the following tables for a list of the keywords that are used with each of the upgrade-related modules.

upgrade

The upgrade module is used to perform a full upgrade in infoScale. Use the following keywords while referencing the module in your playbook.

Table 4-2 Upgrade module keywords

Name	Description	Mandatory/optional
product_version:	Specifies the version of the product that you want to install or upgrade. **Example**: `http://xx.xxx.xxx.xx/rpms/`	Mandatory
gpg_check:	Specify whether to check the integrity of the yum packages by using the gpgkey provided with the InfoScale installation media. This is a boolean variable and must be specified using 0 or 1. By default the value is set to 0.	Optional

gpg_key:	Specifies the location of the gpgkey (typically located in the rpms directory of the installation media).
	Example:
	`http://xx.xxx.xxx.xx/rpms/`
	`RPM-GPG-KEY-veritas-infoscale7`
		Mandatory, if gpgcheck is 1

Table 4-2 Upgrade module keywords *(continued)*

Name	Description	Mandatory/optional
state:	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional
license:	Specifies the path to the slf license file to be registered on the system. Ensure that the license file is accessible store on the same server where the installer is saved. If you are performing a keyless installation or upgrade you can simply enter the product name. **Examples**: ■ `/license_key/Unix/perpetual/`*`xxxxxxxxxxxxx`*`.slf` ■ `ENTERPRISE` ■ `AVAILABILITY`	Mandatory
edgeserver_hostname	Use this parameter to enter the server host name to which installation, deployment, and usage data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. The Veritas Cloud Receiver (VCR) is a preconfigured, cloud-based edge server deployed by Veritas. Enter **telemetry.veritas.com** to use the Veritas Cloud Receiver as an edge server for your environment.	Mandatory
edgeserver_port	Use this parameter to enter the port number of the edge server to which data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. Enter **443** if you are using the Veritas Cloud Receiver as an edge server for your environment.	Mandatory

ru_phase1

The ru_phase1 module is used to perform the first phase of a rolling upgrade on all systems sequentially (kernel packages). Use the following keywords while referencing the module in your playbook.

Table 4-3 Ru_phase1 module keywords

Name	Description	Mandatory/optional
product_version:	Specifies the version of the product that you want to install or upgrade. **Example**: `http://xx.xxx.xxx.xx/rpms/`	Mandatory
gpgcheck:	Specify whether to check the integrity of the yum packages by using the gpgkey provided with the InfoScale installation media. This is a boolean variable and must be specified using 0 or 1. By default the value is set to 0.	Optional
gpgkey:	Specifies the location of the gpgkey (typically located in the rpms directory of the installation media). **Example**: `http://xx.xxx.xxx.xx/sde2/7.4.2/dvd1-redhatlinux/rhel7_x86_64/rpms/RPM-GPG-KEY-veritas-infoscale7`	Mandatory. if gpgcheck is 1
product:	Specifies the name of the product you want to install. **Examples**: `ENTERPRISE`, `AVAILABILITY`, `STORAGE`, or `FOUNDATION`.	Mandatory
state:	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional
license:	Specifies the path to the slf license file to be registered on the system. Ensure that the license file is accessible store on the same server where the installer is saved. If you are performing a keyless installation or upgrade you can simply enter the product name. **Examples**: ■ `/license_key/Unix/perpetual/xxxxxxxxxxxxxxxxxxx.slf` ■ `ENTERPRISE` ■ `AVAILABILITY`	Mandatory
component:	Specify which components you want to configure in your product. Note that the product license acquired must support the required components. **Example**: `SF`, `VCS`, or `SFCHA`	Mandatory

Table 4-3 Ru_phase1 module keywords *(continued)*

Name	Description	Mandatory/optional
start_process	Specifies to start InfoScale processes if value is set to **yes**. By default this option is set to **no**.	Optional
seednode:	Select any node from the cluster that will be used to run commands related to the operations of that cluster. Ensure that you enter the host name as provided in the in the `/etc/ansible/hosts` file.	Mandatory
edgeserver_hostname	Use this parameter to enter the server host name to which installation, deployment, and usage data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. The Veritas Cloud Receiver (VCR) is a preconfigured, cloud-based edge server deployed by Veritas. Enter **telemetry.veritas.com** to use the Veritas Cloud Receiver as an edge server for your environment.	Mandatory
edgeserver_port	Use this parameter to enter the port number of the edge server to which data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. Enter **443** if you are using the Veritas Cloud Receiver as an edge server for your environment.	Mandatory

ru_phase2 module keywords

The yum module is used to perform second phase of rolling upgrade on all systems simultaneously (non-kernel packages). Use the following keywords while referencing the module in your playbook.

Table 4-4

Name	Description	Mandatory/optional
product_version:	Specifies the version of the product that you want to install or upgrade. **Example**: `http://`*`xx.xxx.xxx.xx`*`/rpms/`	Mandatory

Table 4-4 *(continued)*

Name	Description	Mandatory/optional
gpgcheck:	Specify whether to check the integrity of the yum packages by using the gpgkey provided with the InfoScale installation media. This is a boolean variable and must be specified using 0 or 1. By default the value is set to 0.	Optional
gpgkey:	Specifies the location of the gpgkey (typically located in the rpms directory of the installation media). **Example**: `http://`*`xx.xxx.xxx.xx`*`/rpms/RPM-GPG-KEY-veritas-infoscale7`	Mandatory if gpgcheck is 1
product:	Specifies the name of the product you want to install. **Examples**: `ENTERPRISE`, `AVAILABILITY`, `STORAGE`, or `FOUNDATION`.	Mandatory
state:	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional
license:	Specifies the path to the slf license file to be registered on the system. Ensure that the license file is accessible store on the same server where the installer is saved. If you are performing a keyless installation or upgrade you can simply enter the product name. **Examples**: ■ `/license_key/Unix/perpetual/`*`xxxxxxxxxxxxxxxxxxxxxx`*`.slf` ■ `ENTERPRISE` ■ `AVAILABILITY`	Mandatory
component:	Specify which components you want to configure in your product. Note that the product license acquired must support the required components. **Example**: `SF`, `VCS`, or `SFCHA`	Mandatory
start_process	Specifies to start InfoScale processes if value is set to **yes**. By default this option is set to **no**.	Optional

Table 4-4 *(continued)*

Name	Description	Mandatory/optional
seednode:	Select any node from the cluster that will be used to run commands related to the operations of that cluster. Ensure that you enter the host name as provided in the in the `/etc/ansible/hosts` file.	Mandatory
edgeserver_hostname	Use this parameter to enter the server host name to which installation, deployment, and usage data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. The Veritas Cloud Receiver (VCR) is a preconfigured, cloud-based edge server deployed by Veritas. Enter **telemetry.veritas.com** to use the Veritas Cloud Receiver as an edge server for your environment.	Mandatory
edgeserver_port	Use this parameter to enter the port number of the edge server to which data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. Enter **443** if you are using the Veritas Cloud Receiver as an edge server for your environment.	Mandatory

Configuring InfoScale features

This chapter includes the following topics:

Ansible modules for configuring features in InfoScale

Use the following Ansible modules in your playbooks to perform feature configuration-related operations. Refer to the following table for a list of modules, along with a sample playbook, used for each of the operations:

Table 5-1 Feature configuration-related keywords

<table>
<thead>
<tr><th>Operation</th><th>Required modules</th><th>Sample playbook</th></tr>
</thead>
<tbody>
<tr>
<td>Configuring a Cluster File System (CFS)</td>
<td>- site_facters
- cfsresource</td>
<td>

```
---
- hosts: cpicluster11
  gather_facts: false
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: CFS Resource
    veritas_infoscale:
     module: cfsresource
     state: present
     sgname: testsg1
     dgname: testdg5
     volname: testvol1
     mnt: /testvol1
     systems: objstorer820-1-vm16
     facters: "{{ groups['all'] |map('extract', hostvars,
['facts','infoscale_facts'])| select()|list }}"
```

</td>
</tr>
</tbody>
</table>

Table 5-1 Feature configuration-related keywords *(continued)*

<table>
<tr><th>Operation</th><th>Required modules</th><th>Sample playbook</th></tr>
<tr><td>Creating a disk group volume</td><td><ul><li>site_facters</li><li>vxvm_dgvolfs</li></ul></td><td><pre>

- hosts: cpicluster11
  gather_facts: false
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: Create DG, Volume, FS
    veritas_infoscale:
     module: vxvm_dgvolfs
     state: present
     dg1:
      dgname: testdg5
      dgtype: shared
      fss: 1
      disks: [objstorer820-1-_vmdk0_0]
      volinfo: [[testvol1,750m],[testvol2,800m]]
      seednode: objstorer820-1-vm17
     facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"
</pre></td></tr>
</table>

Table 5-1 Feature configuration-related keywords *(continued)*

<table>
<tr><th>Operation</th><th>Required modules</th><th>Sample playbook</th></tr>
<tr><td>Configuring an File System (FS) resource</td><td>- site_facters
- fsreource</td><td>---
- hosts: cpicluster10
  gather_facts: false
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: Configure File Resource
    veritas_infoscale:
     module: fsresource
     state: present
     sgname : testdgsg5
     systems: [pun685cg7labs4-vm11,pun685cg7labs4-vm12]
     dgresname: dgres5
     volresname: volres5
     mntresname: mntres5
     dgname: testdg5
     volname: testvol5
     mnt: /testvol5
     sgtype: Parallel
     facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"</td></tr>
</table>

Table 5-1 Feature configuration-related keywords *(continued)*

Operation	Required modules	Sample playbook
Configuring a Veritas Volume Replicator (VVR) or Cluster Volume Replication (CVR)	■ site_facters ■ vvrresource	

Table 5-1 Feature configuration-related keywords *(continued)*

Operation	Required modules	Sample playbook

```
---
- hosts: vvrclust1
  gather_facts: False
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: VVR/CVR Resource
    veritas_infoscale:
     module: vvrresource
     aws: null
     seednode: dl380g10-12-kvm-53
     primaryvip: xx.xxx.xxx.xxx
     secondaryvip: xx.xxx.xxx.xxx
     dgname: testdg2
     datavolname: testvol1
     srlvolname: srlvol1
     rvgname: rvg_vvrtest1
     agentinfo:
        primary:
            clusterlist:
                - Cluster1
                - Cluster2
            networkhosts:
                - xx.xxx.xxx.x
            nic: eth0
            netmask: 255.255.252.0
            mountpoint: /mnt_test/primary
        secondary:
            nic: ens256
            netmask: 255.255.252.0
            mountpoint: /mnt_secondary
            clusterlist:
                - Cluster1
                - Cluster2
            networkhosts:
                - xx.xxx.xxx.x
     state: present
     rvg_sg_name : null
     rvg_res_name : null
     mount_sg_name : null
```

Table 5-1 Feature configuration-related keywords *(continued)*

<table>
<tr><th>Operation</th><th>Required modules</th><th>Sample playbook</th></tr>
<tr><td></td><td></td><td>   mount_res_name : null
     datadg_res_name : null
     ip_res_name : null
     nic_res_name : null
     awsip_res_name : null
     rvgprimary_res_name : null
     vvrshare_res_name : null
     logownergrp_sg_name : null
     logownerip_res_name : null
     logowner_res_name : null
     awsip_logowner_res_name : null
     logownernic_res_name : null
     facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"</td></tr>
<tr><td>Adding nodes</td><td>- site_facters
- add_node</td><td>---
- hosts: cpicluster11
  gather_facts: false
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: Add/Del node
    veritas_infoscale:
     module: add_node
     state: present
     clusternode: dl380g10-kvm-12-03
     add_del_nodes: ['dl380g10-kvm-12-56' , 'dl380g10-kvm-12-57']
     cluster_id: 34321
     cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61
     facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"</td></tr>
</table>

Table 5-1 Feature configuration-related keywords *(continued)*

<table>
<tr><th>Operation</th><th>Required modules</th><th>Sample playbook</th></tr>
<tr><td>Deleting nodes</td><td><ul><li>site_facters</li><li>add_node</li></ul></td><td><pre>---
- hosts: cpicluster11
  gather_facts: false
  any_errors_fatal: true
  tasks:
  - name: Facters
    veritas_infoscale:
     module: site_facters
    register: facts
  - name: Add/Del node
    veritas_infoscale:
     module: add_node
     state: absent
     clusternode: dl380g10-kvm-12-03
     add_del_nodes: ['dl380g10-kvm-12-56' , 'dl380g10-kvm-12-57']
     cluster_id: 34321
     cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61
     facters: "{{ groups['all'] |map('extract',
hostvars, ['facts','infoscale_facts'])| select()|list }}"</pre></td></tr>
</table>

List of pre-defined keywords

Refer to the following tables for a list of the keywords that are used with each of the feature configuration-related modules.

cfsresource

The cfsresource module is used to create cluster file systems in InfoScale. Use the following keywords while referencing the module in your playbook.

Table 5-2 Cfsresource module keywords

Name	Description	Mandatory/Optional
state:	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional
sgname:	Name of the service group where you are creating the cluster file system.	Mandatory

Table 5-2 Cfsresource module keywords *(continued)*

Name	Description	Mandatory/Optional
dgname:	Name of the disk group where you are creating the cluster file system.	Mandatory
volname:	Name of the volume where you are creating the cluster file system.	Mandatory
mnt:	Mount point of the volume where you are creating the cluster file system.	Mandatory
systems:	Specify the list of host names that are part of the cluster. Ensure that you enter the host names as provided in the in the `/etc/ansible/hosts` file. **Example**: `[hostname1,hostname2,hostname3]`	Mandatory
facters:	Enter the following value to collect system-related data from the cluster servers. **`"{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"`**	Mandatory

vxvm_dgvolfs

The vxvm_dgvolfs module is used to create disk group volumes and file systems in InfoScale. You can also use taginfo arguments to tag disks and create volumes based on tag name. Use the following keywords while referencing the module in your playbook.

Table 5-3 Vxvm_dgvolfs module keywords

Name	Description	Mandatory/Optional
state:	Specifies what state the disk group or volume should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Mandatory
fss:	Specifies whether the disk is part of a Flexible Storage Sharing (FSS) environment. The value for this keyword can be either yes or no. By default, this keyword is set to no.	Optional
dgtype:	Enter **shared** if you want to configure a shared type of disk group or leave the keyword empty.	Optional

Table 5-3 Vxvm_dgvolfs module keywords *(continued)*

Name	Description	Mandatory/Optional

disks:	Specifies the list of disks that you want to add in the disk group. Use the following format to enter the disk names:
	Syntax:
	`[<disk_name_1>, <disk_name_2> ,...<disk_name_n>]`
	`<disk_name_1>`, *`<disk_name_2>`*, ...*`<disk_name_n>`* are the names of the disks that you want to add to the disk group.
	Example:
	`[disk1,disk2,disk3]`
		Mandatory

Table 5-3 Vxvm_dgvolfs module keywords *(continued)*

<table>
<tr><th>Name</th><th>Description</th><th>Mandatory/Optional</th></tr>
<tr><td>taginfo:</td><td>

Specify a list of tags. Each tag comprises a name and value. The tags are applied to the disks that are listed below each tag entry.

Syntax:

```
taginfo:
  - - <tag_name_1>
    - <tag_value_1>
    - - <disk_name_1>
      - <disk_name_2>
  - - <tag_name_2>
    - <tag_value_2>
    - - <disk_name_1>
      - <disk_name_3>
  - - <tag_name_3>
    - <tag_value_3>
    - - <disk_name_2>
```

- *<tag_name_1>*, *<tag_name_2>*, ...*<tag_name_n>* are the tag names.
- *<tag_value_1>*, *<tag_value_2>*, ...*<tag_value_n>* are the are the tag values that you want to associate with each of the tag names.
- *<disk_name_1>*, *<disk_name_2>*, ...*<disk_name_n>* are the are the names of the disks to which you want to apply the tag.

Example:

```
taginfo:
  - - disk_use
    - data
    - - virtio0_3
      - virtio0_1
  - - disk_size
    - large
    - - virtio0_3
      - virtio0_2
```

</td><td>Optional</td></tr>
</table>

Table 5-3 Vxvm_dgvolfs module keywords *(continued)*

Name	Description	Mandatory/Optional

volinfo:	Specify the name and size of the volumes that you want to create.
	Additionally, if you want to create volumes based on tags, specify the tag name and tag value along with the volume name and volume size.
	Syntax:
	`volinfo:`
	`  - - <volume_name_1>`
	`    - <volume_size_1>`
	`    - disktag:<tag_name_1>=<tag_value_1>`
	`  - - <volume_name_2>`
	`    - <volume_size_2>`
	Example:
	`volinfo:`
	`  - - vol2`
	`    - 2g`
	`    - disktag:disk_use=data`
	`  - - srlvol1`
	`    - 5g`
		Optional

seednode:	Select any node from the cluster that will be used to run commands related to the operations of that cluster. Ensure that you enter the host name as provided in the in the `/etc/ansible/hosts` file.	Mandatory
facters:	Enter the following value to collect system-related data from the cluster servers. **`"{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"`**	Mandatory

fsresource

The fsresource module is used to create file systems in InfoScale. Use the following keywords while referencing the module in your playbook.

Table 5-4 Fsresource module keywords

Name	Description	Mandatory/Optional
state:	Specifies what state the resource should be in after the task is completed. The value for this keyword can be either **present** or **absent**.	Mandatory

Table 5-4 Fsresource module keywords *(continued)*

Name	Description	Mandatory/Optional
sgname:	Name of the service group where you are creating the file system.	Mandatory
sgtype:	Specify the type of service group. This value can either be Parallel or Failover.	Mandatory
dgname:	Name of the disk group where you are creating the file system.	Mandatory
dgresname:	Name of the corresponding disk group resource.	Mandatory
volname:	Name of the volume where you are creating the cluster file system.	Mandatory
volresname:	Name of the corresponding volume resource.	Mandatory
mnt:	Mount point of the volume where you are creating the file system.	Mandatory
mntresname:	Name of the corresponding mount point resource.	Mandatory
systems:	Specify the list of host names that are part of the cluster. Ensure that you enter the host names as provided in the in the `/etc/ansible/hosts` file. **Example**: `[hostname1,hostname2,hostname3]`	Mandatory
facters:	Enter the following value to collect system-related data from the cluster servers. **`"{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"`**	Mandatory

vvrresource

The vvrresource module is used to setup the Volume Replicator in InfoScale. On the basis of whether provided disk group is shared or non-shared, CVR or VVR resource will be configured. Use the following keywords while referencing the module in your playbook.

Table 5-5 vvrresource module keywords

Name	Description	Mandatory/Optional
state	Specifies what state the package should be in after the task is completed. Set this attribute to **present** to configure VVR or CVR.	Mandatory
aws	Specify **1** or **0**. 1: If setup is taking place on an AWS instance. 0: If setup is taking place on a non-AWS instance.	Mandatory

Table 5-5 vvrresource module keywords *(continued)*

<table>
<tr><th>Name</th><th>Description</th><th>Mandatory/Optional</th></tr>
<tr><td>dgname</td><td>Name for the disk group where the Replicated Volume Group (RVG) is created.</td><td>Mandatory</td></tr>
<tr><td>datavolname</td><td>Name of the data volume of the VVR setup.</td><td>Mandatory</td></tr>
<tr><td>srlvolname</td><td>Name of the Storage Replicator Log (SRL) volume for the VVR setup.</td><td>Mandatory</td></tr>
<tr><td>rvgname</td><td>Name of the Replicated Volume Group (RVG) to be created</td><td>Mandatory</td></tr>
<tr><td>seednode</td><td>Specify any node from the cluster that will be used to run commands related to the operations of that cluster. Ensure that you enter the host name as provided in the in the <code>hosts</code> file.</td><td>Mandatory</td></tr>
<tr><td>primaryvip</td><td>Specify a virtual IP for setting up VVR on the primary site.</td><td>Mandatory</td></tr>
<tr><td>secondaryvip</td><td>Specify a virtual IP for setting up VVR on the secondary site.</td><td>Mandatory</td></tr>
<tr><td>agentinfo</td><td>Specify the information required to create the VVR resource under VCS for site and cluster failover.
Skip this attribute if you are not creating a VCS resource for site and cluster failover.
<b>Example</b>:<pre>
agentinfo:
      primary:
        clusterlist:
          - Cluster1
          - Cluster2
        networkhosts:
          - xx.xxx.xxx.x
        nic: eth0
        netmask: 255.255.252.0
        mountpoint: /mnt_test/primary
      secondary:
        nic: ens256
        netmask: 255.255.252.0
        mountpoint: /mnt_secondary
        clusterlist:
          - Cluster1
          - Cluster2
        networkhosts:
          - xx.xxx.xxx.x
</pre></td><td>Optional</td></tr>
</table>

Table 5-5 vvrresource module keywords *(continued)*

Name	Description	Mandatory/Optional
clusterlist	Specify a list containing the name of clusters having a GCO setup for cluster failover.	Mandatory
networkhosts	Specify an IP address for a NIC resource. Configure the NetworkHosts attribute to ensure that the NIC resource is always online.	Mandatory
mountpoint	Specify the path where the volume is mounted.	Mandatory
primary	Specify information about the resources in the primary site. The NICs, MACs, and VIPs are configured to the primary site.	Mandatory
secondary	Specify information about the resources in the primary site. The NICs, MACs, and VIPs are configured to the secondary site.	Mandatory
rvg_sg_name	Specify a name for the RVG service group. A service group with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default service group name is used.	Optional
rvg_res_name	Specify a name for the RVG resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
mount_sg_name	Specify a name for the MOUNT service group. A service group with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default service group name is used.	Optional
mount_res_name	Specify a name for the MOUNT resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
datadg_res_name	Specify a name for the DATADG resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
ip_res_name	Specify a name for the IP resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
nic_res_name	Specify a name for the NIC resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional

Table 5-5 vvrresource module keywords *(continued)*

Name	Description	Mandatory/Optional
awsip_res_name	Specify a name for the AWSIP resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
rvgprimary_res_name	Specify a name for the RVG PRIMARY resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
vvrshare_res_name	Specify a name for the VVRSHARE resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
logownergrp_sg_name	Specify a name for the LOGOWNER service group. A service group with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default service group name is used.	Optional
logownerip_res_name	Specify a name for the LOGOWNERIP resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
logowner_res_name	Specify a name for the LOGOWNER resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
awsip_logowner _res_name	Specify a name for the AWSIP LOGOWNER resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
logownernic_res_name	Specify a name for the LOGOWNER NIC resource. A resource with the user-defined name is created, if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
facters	This attribute is used to collect system-related data from the cluster servers. Provide the following value to the attribute: `"{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"`	Mandatory

add_node

The add_node module is used to add or remove nodes from a cluster. Use the following keywords while referencing the module in your playbook.

Table 5-6 Add_node module keywords

Name	Description	Mandatory/Optional
state:	Specifies what state the nodes should be in after the task is completed. The value for this keyword can be either **present** or **absent**. Use **present** to add nodes and use **absent** to delete nodes.	Mandatory
cluster_node:	Specify any node from the cluster. Ensure that you enter the host name as provided in the in the `/etc/ansible/hosts` file.	Mandatory
add_del_nodes:	List of the nodes that you want to add or delete. **Example**: `['hostname1', 'hostname2']`	Mandatory
cluster_id:	Specify the ID of the cluster.	Mandatory
cluster_uuid:	Specify the UUID of the cluster.	Mandatory