

InfoScale™ Installation, Upgrade, and Configuration Using Ansible - Linux

Last updated: 2026-01-27

Legal Notice

Copyright ©2026 Arctera US LLC. All rights reserved.

Arctera and the Arctera Logo are trademarks or registered trademarks of Arctera US 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 Arctera 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 Arctera product or available at:

<https://www.arctera.io/license-agreements>

The product described in this document is distributed under licenses restricting its use, copying, distribution, and de-compilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Arctera US 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. ARCTERA US 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 Arctera 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.

Arctera US LLC | www.arctera.io

Technical Support

Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within the company to answer your questions in a timely fashion.

Our support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about our support offerings, you can visit our website at the following URL:

www.arctera.io/support

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

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 here:

<https://sort.veritas.com/arctera/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:

productdocs@arctera.io

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

<https://vox.veritas.com/category/arctera-discussions/discussions/infoscale>

Services and Operations Readiness Tools (SORT)

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

Section 1	InfoScale Ansible playbooks	6
Chapter 1	Preparing to use Ansible for InfoScale	7
	Introduction to Ansible	7
	Downloading Ansible modules for InfoScale	9
	Before you begin	10
	Using site_factors module	10
	Ansible modules for downloading release matrix	11
	List of pre-defined keywords	12
Chapter 2	Ansible roles for InfoScale	14
	InfoScale get host info	14
	Start InfoScale services	15
	Stop InfoScale services	17
	Start InfoScale services after a full upgrade	18
	Stop InfoScale services before a full upgrade	20
Chapter 3	Installing InfoScale using Ansible	22
	Ansible modules for installing InfoScale	22
	List of pre-defined keywords	30
Chapter 4	Configuring InfoScale components using Ansible	34
	Ansible modules for configuring InfoScale	34
	List of pre-defined keywords	49
Chapter 5	Upgrading InfoScale using Ansible	54
	Ansible modules for upgrading InfoScale	54
	List of pre-defined keywords	69

Chapter 6	Configuring InfoScale features using Ansible	80
	Ansible modules for configuring features in InfoScale	80
	List of pre-defined keywords	131
Section 2	InfoScale Ansible end to end workflow / solution	166
Chapter 7	VVR solution	167
	Configuring end to end InfoScale VVR deployment using Ansible playbooks	167
	Ansible playbook for configuring end to end VVR deployment in InfoScale	169
	List of pre-defined keywords	170
	Input file for VVR solution (vars_file - main.json)	171
	List of pre-defined keywords	176
	Out file for VVR solution	180
Chapter 8	Solution templates	182
	Customizing InfoScale deployment using solution templates	182

InfoScale Ansible playbooks

- [Chapter 1. Preparing to use Ansible for InfoScale](#)
- [Chapter 2. Ansible roles for InfoScale](#)
- [Chapter 3. Installing InfoScale using Ansible](#)
- [Chapter 4. Configuring InfoScale components using Ansible](#)
- [Chapter 5. Upgrading InfoScale using Ansible](#)
- [Chapter 6. Configuring InfoScale features using Ansible](#)

Preparing to use Ansible for InfoScale

This chapter includes the following topics:

- [Introduction to Ansible](#)
- [Downloading Ansible modules for InfoScale](#)
- [Before you begin](#)
- [Using site_factors module](#)
- [Ansible modules for downloading release matrix](#)
- [List of pre-defined keywords](#)

Introduction 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 contain human-readable code, and can be used to define operations in your environment.

Arctera provides an Ansible module that can be used to run Ansible playbooks to deploy Arctera InfoScale products and configure features. The Ansible module for Linux is SonarQube compliant.

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

Deployment-related operations	Feature-related operations
<ul style="list-style-type: none"> ■ Installing a product ■ Licensing ■ Configuring components ■ Starting a product ■ Stopping a product ■ Performing a full upgrade ■ Performing a rolling upgrade ■ Uninstalling a product 	<ul style="list-style-type: none"> ■ Configuring a Cluster File System (CFS) resource ■ Creating a disk group volume ■ Configuring an File System (VxFS) resource ■ Configuring Volume Replicator (VVR) and cluster volume replication (CVR) ■ Configuring LLT over UDP ■ Configuring I/O fencing ■ Adding a node to a cluster ■ Deleting a node from a cluster ■ Creating a single-node cluster ■ Tagging disks ■ Tagging-based volume creation ■ Enabling or disabling EO-compliant logging ■ Setting tunables ■ Setting routes and rules for subnets across AZs on AWS ■ Adding disks to or removing disks from a disk group ■ Changing replication state ■ Unconfiguring replication

Supported platforms

You can use Ansible to deploy and configure Arctera InfoScale on all Red Hat Enterprise Linux (RHEL) distributions supported by InfoScale.

You can also use Ansible to deploy and configure InfoScale on SUSE Linux Enterprise Server (SLES) with limited playbooks—install, configuration, patch upgrade, full upgrade, rolling upgrade, start services, and stop services.

A `changelog.md` file is available for each Ansible module release, which lists the versions qualified with that specific release of the module. Before you deploy InfoScale using Ansible, refer to the `changelog.md` file at:

<https://sort.veritas.com/public/infoscale/ansible/linux/logs/changelog.md>.

For the specific RHEL and SLES versions supported with an InfoScale release, refer to the *InfoScale Software Compatibility List - Linux*.

Supported cloud platforms

The InfoScale Ansible module supports the Azure and AWS cloud platforms. You can perform InfoScale installation, upgrade, and configuration operations on these cloud platforms using Ansible playbooks.

Table 1-2 Ansible playbooks for InfoScale installation, upgrade, and configuration on Azure and AWS

Playbooks for Azure	Playbooks for AWS
<ul style="list-style-type: none"> ■ Installation ■ Licensing ■ Component configuration ■ Starting a product ■ Stopping a product ■ Full upgrade ■ Rolling upgrade ■ Uninstallation ■ Configuring a Cluster File System resource ■ Configuring an File System resource ■ Resize a File System ■ Creating a disk group volume ■ Configuring GCO ■ Configuring VVR and CVR ■ Configuring CPS based fencing ■ Adding node ■ Deleting node 	<ul style="list-style-type: none"> ■ Installation ■ Licensing ■ Component configuration ■ Starting a product ■ Stopping a product ■ Full upgrade ■ Rolling upgrade ■ Uninstallation ■ Configuring a Cluster File System resource ■ Configuring an File System resource ■ Resize a File System ■ Creating a disk group volume ■ Configuring CPS based fencing ■ Patch upgrade ■ Deleting node ■ Adding node ■ Configuring GCO ■ Configuring VVR and CVR

Supported Ansible version

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

Downloading Ansible modules for InfoScale

To download the Ansible modules, playbook templates, and documentation for InfoScale, visit the following link:

<https://sort.veritas.com/utility/ansible>

From Infoscale Ansible v2.4.0 onwards, the `veritas_infoscalle.py` file is shipped as an Ansible collection.

To install the Ansible collection—if the Ansible version is 2.9.10 or higher, use the following command:

```
# ansible-galaxy collection install
veritas-infoscalle-<release_version>.tar.gz
```

To upgrade the local Ansible collection, use the following command:

```
# ansible-galaxy collection install  
veritas-infoscale-<release_version>.tar.gz --force
```

The InfoScale Ansible collection is installed at the following path:

```
/root/.ansible/collections/ansible_collections/veritas/infoscale/
```

If the Ansible version is not compatible for collections—lower than 2.9.10, you must untar the tar file and navigate to the following file path:

```
veritas/infoscale/plugins/modules/veritas_infoscale.py inside the tar  
file and move the veritas_infoscale.py to  
/usr/share/ansible/plugins/modules/ to use it as a module.
```

Before you begin

Ensure that the following prerequisites are met in your environment:

- Passwordless SSH communication is established between the Ansible server and the nodes that are to be managed by the Ansible server. The `p1` utility can be used to set up the SSH and RSH connections automatically.
- Ensure that Python 2.6 or later is installed and configured on all the systems in the environment. Use a Python Jinja2 template engine of version 8 or later.

Using `site_factors` module

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

Table 1-3 Release matrix-related operations

Name	Description	Mandatory/optional
release_matrix_path	<p>Specifies the location where release matrix json file would exist.</p> <p>There are two options to select release matrix location:</p> <p>Method 1: sort : Picks up release matrix data from SORT website.</p> <p>Method 2: Custom_url/custom_path: Can be valid URL or accessible path from host machine.</p> <ul style="list-style-type: none"> ■ Custom URL: The following URL must be accessible: Note: rhelX_86_64-ga.json would not be required: <URL>/rhelX_86_64-ga.json ■ Custom path: The path must contain the following files: kernel_padv.json rhelX_x86_64-ga.json rhelX_x86_64-patch.json rhelX_x86_64-hotfix.json <p>If the release matrix failed to update from SORT or custom URL/custom path, then the local matrix would be utilized. To check which release matrix data is being used you can refer to the logs generated by that playbook</p> <p>Example:</p> <pre>release_matrix_path : "sort" release_matrix_path : "http://<URL>" release_matrix_path : "/root/matrix/"</pre>	Optional

Ansible modules for downloading release matrix

Use the following Ansible modules in your playbooks to download the release matrix json files. Refer to the following table for the module, along with a sample playbook, used for download release matrix operations:

Table 1-4 Download release matrix operation

Operation	Required modules	Sample playbook
Download release matrix	download_release_matrix	<p>Sample playbook - YAML</p> <pre> --- - hosts: "{{ vtas_hosts }}" gather_facts: false any_errors_fatal: true tasks: - name: Download Release matrix from SORT veritas_infoscale: module: download_release_matrix local_directory_path: "{{ vtas_local_directory_path }}" </pre> <hr/> <p>Sample playbook - JSON</p> <pre> --- [{ "hosts": "{{ vtas_hosts }}", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Download Release matrix from SORT", "veritas_infoscale": { "module": "download_release_matrix", "local_directory_path": "{{ vtas_local_directory_path }}" } }] }] </pre>

List of pre-defined keywords

Refer to the following tables for a list of the keywords that are used with each of the release matrix path related modules.

The `download_release_matrix` module is used to download the release matrix json files. Use the following keywords while referencing the module in your playbook.

Table 1-5 Download release matrix module keywords

Name	Description	Mandatory/optional
local_directory_path	Specifies the location where you can download the release matrix. This location must exist before running the playbook. Example: <code>/root/matrix</code> Note: Please use absolute path	Mandatory

Ansible roles for InfoScale

This chapter includes the following topics:

- [InfoScale get host info](#)
- [Start InfoScale services](#)
- [Stop InfoScale services](#)
- [Start InfoScale services after a full upgrade](#)
- [Stop InfoScale services before a full upgrade](#)

InfoScale get host info

This documentation explains how Ansible role can be used for collecting the required information from the nodes.

The goals of this guide is to collect host information of the nodes which is later used for other InfoScale operations.

Pre-requisites

- Ansible control node is properly set to run Ansible playbooks.
- The role is installed or copied as per your environment

Role information

The role can be used like this:

```
``yaml
---
- name: My playbook for InfoScale tasks
  hosts: cluster_nodes
```

```
roles:  
  - get_host_info
```

This role should be used when a user wants to collect information about the nodes.

Variables

`release_matrix_path` (optional): If the user has downloaded the latest release matrix, this variable can be used to provide the path of the release matrix.

Using variables in Ansible

There are several ways of modifying the default variables. You can check the [Ansible documentation](#) for more information.

Command line

This role can be included in the playbook as per the user's environment workflow or requirements. The playbook can be run as below:

From the command line, it can be done with the `-e EXTRA_VARS` or `--extra-vars EXTRA_VARS` flag:

```
``cli  
ansible-playbook -i hosts playbooks/myinfoscale.yml -e "release_matrix_path="
```

Role usage

This role can be used in the playbooks in all the ways Ansible supports. The following is the example of how to import this role.

```
``yaml  
  
- name: Get host info  
  import_role:  
    name: get_host_info
```

Start InfoScale services

This documentation explains how Ansible role can be used for starting InfoScale services on the cluster nodes.

The goals of this guide is to start services on the cluster nodes of InfoScale.

Pre-requisites

- Ansible control node is properly set to run Ansible playbooks.
- The role is installed or copied as per your environment

Role information

The role can be used like this:

```
```yaml

- name: My playbook for InfoScale tasks
 hosts: cluster_nodes

 roles:
 - start_processes
```
```

This role should be used when the user wants to start the services on the cluster nodes of InfoScale.

Variables

- `vtas_component`: Name of the component configured on the InfoScale nodes: (Such as SF, SFHA, SFCFSHA, and more.)
- `vtas_product`: Name of the product installed on the InfoScale nodes. (FOUNDATION, AVAILABILITY, STORAGE, ENTERPRISE)
- `vtas_product_version`: Product version installed on the InfoScale nodes. (Such as 8.0, 7.4.2, and more.)

Using variables in Ansible

There are several ways of modifying the default variables. You can check the [Ansible documentation](#) for more information.

Dependencies

This role internally has dependencies on the following roles which are defined in `meta/main.yml`:

- role: `get_host_info`
- role: `set_infoscale_facts`

The `get_host_info` and `set_infoscale_facts` roles compliment the `start_processes` role to have recent information from the cluster node before starting the services.

Command line

This role can be included in the playbook as per the user's environment workflow or requirements. The playbook can be run as below:

From the command line, it can be done with the `-e EXTRA_VARS` or `--extra-vars EXTRA_VARS` flag:

```
``cli
ansible-playbook -i hosts playbooks/myinfoscale.yml -e "vtas_component='sfha'
vtas_product='ENTERPRISE'" -e "vtas_product_version='sfha'"
```

Role usage

This role can be used in the playbooks in all the ways Ansible supports. The following is the example of how to import this role.

```
``yaml
- name: Starting processes
  import_role:
    name: start_processes
```

Stop InfoScale services

This documentation explains how Ansible role can be used for stopping InfoScale services on the cluster nodes.

The goals of this guide is to stop services on the cluster nodes of InfoScale.

Pre-requisites

- Ansible control node is properly set to run Ansible playbooks.
- The role is installed or copied as per your environment

Role information

The role can be used like this:

```
``yaml
---
- name: My playbook for InfoScale tasks
  hosts: cluster_nodes

  roles:
    - stop_processes
```

This role should be used when the user wants to stop the services on the cluster nodes of InfoScale.

Variables

- `vtas_component`: Name of the component configured on the InfoScale nodes: (Such as SF, SFHA, SFCFSHA, and more.)

- `vtas_product`: Name of the product installed on the InfoScale nodes. (FOUNDATION, AVAILABILITY, STORAGE, ENTERPRISE)
- `vtas_product_version`: Product version installed on the InfoScale nodes. (Such as 8.0, 7.4.2, and more.)

Using variables in Ansible

There are several ways of modifying the default variables. You can check the [Ansible documentation](#) for more information.

Command line

This role can be included in the playbook as per the user's environment workflow or requirements. The playbook can be run as below:

From the command line, it can be done with the `-e EXTRA_VARS` or `--extra-vars EXTRA_VARS` flag:

```
``cli
ansible-playbook -i hosts playbooks/myinfoscale.yml -e "vtas_component='sfha'
vtas_product='ENTERPRISE'" -e "vtas_product_version='sfha'"
```

Role usage

This role can be used in the playbooks in all the ways Ansible supports. The following is the example of how to import this role.

```
``yaml
- name: Stopping processes
  import_role:
    name: stop_processes
```

Start InfoScale services after a full upgrade

This documentation explains how Ansible role can be used for starting InfoScale services on the cluster nodes after an upgrade.

The goals of this guide is to start services on the cluster nodes of InfoScale after an upgrade.

Pre-requisites

- Ansible control node is properly set to run Ansible playbooks.
- The role is installed or copied as per your environment

Role information

The role can be used like this:

```

  ``yaml
  ---
  - name: My playbook for InfoScale tasks
    hosts: cluster_nodes

    roles:
      - upgradestart_processes
  ...
  
```

This role should be used when the user wants to start the services on the cluster nodes of InfoScale.

Variables

- **vtas_component**: Name of the component configured on the InfoScale nodes: (Such as SF, SFHA, SFCFSHA, and more.)
- **vtas_product**: Name of the product installed on the InfoScale nodes. (FOUNDATION, AVAILABILITY, STORAGE, ENTERPRISE)
- **vtas_product_version**: Product version installed on the InfoScale nodes. (Such as 8.0, 7.4.2, and more.)

Using variables in Ansible

There are several ways of modifying the default variables. You can check the [Ansible documentation](#) for more information.

Dependencies

This role internally has dependencies on the following roles which are defined in meta/main.yml:

- role: get_host_info
- role: set_infoscale_facts

The `get_host_info` and `set_infoscale_facts` roles compliment the `start_processes` role to have recent information from the cluster node before starting the services.

Command line

This role can be included in the playbook as per the user's environment workflow or requirements. The playbook can be run as below:

From the command line, it can be done with the `-e EXTRA_VARS` or `--extra-vars EXTRA_VARS` flag:

```

  ``cli
  ansible-playbook -i hosts playbooks/myinfoscale.yml -e "vtas_component='sfha'
  "vtas_product='ENTERPRISE'" -e "vtas_product_version='sfha'"
  
```

Role usage

This role can be used in the playbooks in all the ways Ansible supports. The following is the example of how to import this role.

```
```yaml
- name: Starting processes
 import_role:
 name: upgradestart_processes
```

## Stop InfoScale services before a full upgrade

This documentation explains how Ansible role can be used for stopping InfoScale services on the cluster nodes after an upgrade.

The goals of this guide is to stop services on the cluster nodes of InfoScale before an upgrade.

### Pre-requisites

- Ansible control node is properly set to run Ansible playbooks.
- The role is installed or copied as per your environment

### Role information

The role can be used like this:

```
```yaml
---
- name: My playbook for InfoScale tasks
  hosts: cluster_nodes

  roles:
    - upgradestop_processes
...`
```

This role should be used when the user wants to stop the services on the cluster nodes of InfoScale.

Variables

- `vtas_component`: Name of the component configured on the InfoScale nodes: (Such as SF, SFHA, SFCFSHA, and more.)
- `vtas_product`: Name of the product installed on the InfoScale nodes. (FOUNDATION, AVAILABILITY, STORAGE, ENTERPRISE)

- `vtas_product_version`: Product version installed on the InfoScale nodes. (Such as 8.0, 7.4.2, and more.)

Using variables in Ansible

There are several ways of modifying the default variables. You can check the [Ansible documentation](#) for more information.

Dependencies

This role internally has dependencies on the following roles which are defined in `meta/main.yml`:

- role: `get_host_info`
- role: `set_infoscale_facts`

The `get_host_info` and `set_infoscale_facts` roles compliment the `start_processes` role to have recent information from the cluster node before starting the services.

Command line

This role can be included in the playbook as per the user's environment workflow or requirements. The playbook can be run as below:

From the command line, it can be done with the `-e EXTRA_VARS` or `--extra-vars EXTRA_VARS` flag:

```
``cli
ansible-playbook -i hosts playbooks/myinfoscale.yml -e "vtas_component='sfha'
vtas_product='ENTERPRISE'" -e "vtas_product_version='sfha'"
```

Role usage

This role can be used in the playbooks in all the ways Ansible supports. The following is the example of how to import this role.

```
``yaml
- name: Stopping processes
  import_role:
    name: upgradestop_processes
```

Installing InfoScale using Ansible

This chapter includes the following topics:

- [Ansible modules for installing InfoScale](#)
- [List of pre-defined keywords](#)

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 3-1 Installation-related operations

Operation	Required modules	Sample playbook
Installation	<ul style="list-style-type: none"> ■ site_factors ■ yum (RHEL) 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: Install InfoScale veritas_infoscale: module: yum repository_name: repo_infoScale742 repository_baseurl: "http://xx.xxx.xx.xx/\ Infoscale/7.4.2/rhel7_x86_64/rpms/" gpgcheck: 1 gpgkey: "http://xx.xxx.xx.xx/\ Infoscale/7.4.2/rhel7_x86_64/rpms/\ RPM-GPG-KEY-veritas-infoscale7" product: ENTERPRISE product_version: '7.4.2' factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" state: present </pre>

Table 3-1 Installation-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cpicluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_facters", "release_matrix_path": "<URL>" }, "register": "facts" }, { "name": "Install InfoScale", "veritas_infoscale": { "module": "yum", "repository_name": "repo_infoScale742", "repository_baseurl": "http://xx.xxx.xx.xx/\ Infoscale/7.4.2/rhel7_x86_64/rpms/", "gpgcheck": 1, "gpgkey": "http://xx.xxx.xx.xx/\ Infoscale/7.4.2/rhel7_x86_64/rpms/\ RPM-GPG-KEY-veritas-infoscale7", "product": "ENTERPRISE", "product_version": "7.4.2", "facters": "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list } }", "state": "present" } }] }]</pre>

Table 3-1 Installation-related operations (continued)

Operation	Required modules	Sample playbook
Installation	site_factors InstallPackages	Sample playbook - YAML <pre> - hosts: cluster gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_factors register: facts - name: Install InfoScale veritas_infoscale: module: InstallPackages repository_name: repo_infoScale742 repository_baseurl: http://xx.xxx.xx.xx/\ Infoscale/7.4.2/rhel7_x86_64/rpms/ gpgcheck: 0 gpgkey: product: ENTERPRISE product_version: 7.4.2 factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" state: present ignore_required_patch: 0 ignore_patchid_check: 0 patch_repo_list: </pre>

Table 3-1 Installation-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cluster", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", "veritas_infoscalse": { "module": "site_factors" }, "register": "facts" }, { "name": "Install InfoScale", "veritas_infoscalse": { "module": "InstallPackages", "repository_name": "repo_infoScale742", "repository_baseurl": "http://xx.xxx.xx.xx/\ Infoscale/7.4.2/rhel7_x86_64/rpms/", "gpgcheck": 0, "gpgkey": null, "product": "ENTERPRISE", "product_version": "7.4.2", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}" "state": "present", "ignore_required_patch": 0, "ignore_patchid_check": 0, "patch_repo_list": } }] }]</pre>

Table 3-1 Installation-related operations (*continued*)

Operation	Required modules	Sample playbook
Licensing	<ul style="list-style-type: none">■ <code>site_factors</code>■ <code>licensing</code>	<p>Sample playbook - YAML</p> <pre>--- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: License veritas_infoscale: module: licensing state: present product_version: 7.4.2 license: 'ENTERPRISE' factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}"</pre>

Table 3-1 Installation-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cpicluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "License", "veritas_infoscale": { "module": "licensing", "state": "present", "product_version": "7.4.2", "license": "ENTERPRISE", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 3-1 Installation-related operations (*continued*)

Operation	Required modules	Sample playbook
Uninstall	<ul style="list-style-type: none">■ <code>site_factors</code>■ <code>yum</code>	<p>Sample playbook - YAML</p> <pre>--- - hosts: cpicluster11 gather_facts: False any_errors_fatal: true tasks: - name: Factors veritas_infoscalse: module: site_factors release_matrix_path: <URL> register: facts - name: Uninstall InfoScale Enterprise veritas_infoscalse: module: yum product: 'ENTERPRISE' product_version: 7.4.2 state: absent factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}"</pre>

Table 3-1 Installation-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cpicluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_facters", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Uninstall InfoScale Enterprise", "veritas_infoscale": { "module": "yum", "product": "ENTERPRISE", "product_version": "7.4.2", "state": "absent", "facters": "{ ansible_play_hosts_all map('extract', hostvars,['facters','infoscale_facters']) select() list }}" } }] }]</pre>

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/InstallPackages

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

Table 3-2 Yum module keywords / InstallPackages module keywords

Name	Description	Mandatory/optional
repository_name	Name of the yum/zypper repository used to install InfoScale rpms. Example: <code>repo-InfoScale742</code>	Mandatory If the Base URL is provided, the system will create a repository with the specified repository name.
repository_baseurl	Specifies the URL to the directory where the repodata of a repository is located. Example: <code>http://xx.xxx.xxx.xx/rpms/</code>	Optional This keyword is not required if you are using a yum/zypper repository already configured on the system, and are providing the repository name.
gpcheck	Specify whether to check the integrity of the yum/zypper 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: <code>http://xx.xxx.xxx.xx/rpms/RPM-GPG-KEY-veritas-infoscale7</code>	Mandatory if gpcheck 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

Table 3-2 Yum module keywords / InstallPackages 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
gendeploy_path	Specifies the absolute path of the gendeploy script from the local node to register InfoScale hosts to the VIOM Management Server. It is supported on InfoScale 8.0.2 and later. Example: gendeploy_path: '/root/script_path.pl'	Optional

licensing

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

Table 3-3 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

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

Name	Description	Mandatory/optional
license	<p>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.</p> <p>Note: Perpetual key is not applicable for InfoScale 8.0.2 onwards.</p> <p>For installation, configuration, or upgrade operations with InfoScale 8.0.2 or later, do not use the license parameter.</p> <p>If you are performing a keyless installation or upgrade you can simply enter the product name.</p> <p>Examples:</p> <ul style="list-style-type: none"> ■ <code>/license_key/Unix/perpetual/ xxxxxxxxxxxxxxxxxxxxx.slf</code> ■ ENTERPRISE ■ AVAILABILITY 	Optional

Configuring InfoScale components using Ansible

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 Arctera 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 4-1 Component configuration-related operations

Operation	Required modules	Sample playbook
Component configuration	<ul style="list-style-type: none"> ■ site_factors ■ <component_name>_config Example: sfcfsha_config sfha_config	

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Configuring sfcfsha</p> <p>Sample playbook - YAML</p> <pre> --- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscail: module: site_facters release_matrix_path: <URL> register: facts - name: Configure Enterprise veritas_infoscail: module: sfcfsha_config cluster_name: clust_cpi9 cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61 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 mtu: 1100 eo_compliant_logging: on facters: "{ ansible_play_hosts_all map('extract',hostvars, ['facts','infoscail_facters'] select() list }}" </pre> <p>Sample playbook - JSON</p> <pre> [{ "hosts": "cpicluster11", "gather_facters": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", </pre>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<pre> "veritas_infoscalse": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Configure Enterprise", "veritas_infoscalse": { "module": "sfcfs_sha_config", "cluster_name": "clust_cpi9", "cluster_uuid": "c7c2d65e-058f-11e8-a32c-c094107f3b61", "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", "mtu": 1100, "eo_compliant_logging": "on", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, 'facts','infoscale_facts') select() list }}" } }] </pre> <p>Configuring sfcfs</p> <p>Sample playbook - YAML</p>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<pre> --- - hosts: cluster-47-48-49 gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: Configure SFCFS veritas_infoscale: module: sfcfsha_config cluster_name: sfcfs_cluster_40_42 cluster_uuid: product_version: '7.4.2' method: ethernet seednode: dl380g10-12-kvm-42 license: STORAGE state: present private_link: eth1,eth2 udp_link: null low_priority_link: eth0 tcp_link: null enable_lgf: 1 eo_compliant_logging: on factors: "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre> <p>Sample playbook - JSON</p>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<pre>[{ "hosts": "cluster-47-48-49", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Configure SFCFS", "veritas_infoscale": { "module": "sfcfsha_config", "cluster_name": "sfcfs_cluster_40_42", "cluster_uuid": null, "product_version": "7.4.2", "method": "ethernet", "seednode": "dl380g10-12-kvm-42", "license": "STORAGE", "state": "present", "private_link": "eth1,eth2", "udp_link": null, "low_priority_link": "eth0", "tcp_link": null, "enable_lgf": 1, "eo_compliant_logging": "on", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre> <p>Configuring sfha Sample playbook - YAML</p>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<pre> --- - hosts: cpicluster11 gather_facts: true any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: Configure Enterprise veritas_infoscale: module: sfha_config cluster_name: clust_cpi9 cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61 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 mtu: 1100 eo_compliant_logging: on factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre> <p>Sample playbook - JSON</p>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<pre>[{ "hosts": "cpicluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_facters", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Configure Enterprise", "veritas_infoscale": { "module": "sfha_config", "cluster_name": "clust_cpi9", "cluster_uuid": "c7c2d65e-058f-11e8-a32c-c094107f3b61", "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", "mtu": 1100, "eo_compliant_logging": "on", "facters": "{ { ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre> <p>Configuring vcs Sample playbook - YAML</p>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<pre> --- - hosts: cpicluster11 gather_facts: true any_errors_fatal: true tasks: - name: Factors veritas_infoscalse: module: site_factors release_matrix_path: <URL> register: facts - name: Configure Enterprise veritas_infoscalse: module: vcs_config cluster_name: clust_cpi9 cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61 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 mtu: 1100 eo_compliant_logging: on factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre> <p>Sample playbook - JSON</p>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<pre>[{ "hosts": "cpicluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Configure Enterprise", "veritas_infoscale": { "module": "vcs_config", "cluster_name": "clust_cpi9", "cluster_uuid": "c7c2d65e-058f-11e8-a32c-c094107f3b61", "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", "mtu": 1100, "eo_compliant_logging": "on", "factors": "{(ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list)}" } }] }]</pre> <p>Configuring sf Sample playbook - YAML</p>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<pre> --- - hosts: cpicluster11 gather_facts: true any_errors_fatal: true tasks: - name: Facters veritas_infoscalse: module: site_facters release_matrix_path: <URL> register: facters - name: Configure Enterprise veritas_infoscalse: module: sf_config product_version: '7.4.2' license: 'ENTERPRISE' seednode: dl380g10-09-vm7 state: present eo_compliant_logging: on facters: "{{ ansible_play_hosts_all map('extract', hostvars, ['facters','infoscalse_facters']) select() list }}" </pre>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cpicluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Configure Enterprise", "veritas_infoscale": { "module": "sf_config", "product_version": "7.4.2", "license": "ENTERPRISE", "seednode": "dl380g10-09-vm7", "state": "present", "eo_compliant_logging": "on", "factors": "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
Starting a product	<ul style="list-style-type: none"> ■ site_factors ■ process 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscail: module: site_factors release_matrix_path: <URL> register: facts - name: Start Product veritas_infoscail: module: process component: sfcfsha product: enterprise product_version: '7.4.2' state: present factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cpicluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Start Product", "veritas_infoscale": { "module": "process", "component": "sfcfsha", "product": "enterprise", "product_version": "7.4.2", "state": "present", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
Stopping a product	<ul style="list-style-type: none"> ■ site_factors ■ process 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: Start Product veritas_infoscale: module: process component: sfcfsha product: enterprise product_version: '7.4.2' state: absent factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre>

Table 4-1 Component configuration-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cpicluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Stop Product", "veritas_infoscale": { "module": "process", "component": "sfcfsha", "product": "enterprise", "product_version": "7.4.2", "state": "absent", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

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 to configure the component in InfoScale. Use the following keywords while referencing the module in your playbook.

Table 4-2 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	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	Optional
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. Seednode can be set as FQDN, IP Address, and short hostname. Examples: seednode: objstorer820-1-vm17.veritas.com seednode: xx.xxx.xxx.xxx seednode: objstorer820-1-vm17	Mandatory
reconfig	Specifies whether the component configuration should be reconfigured. The value should be 1 to reconfigure component.	Optional

Table 4-2 Sfcfsa_config module keywords (continued)

Name	Description	Mandatory/optional
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
udp_link	<p>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.</p> <p>System can be <code>ip/fqdn/nofqdn</code></p> <p>Example:</p> <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>	Optional
vcs_clusterid:	Define a unique number to be assigned to the cluster.	Optional
mtu	Define MTU value to be used.	Optional
eo_compliant_logging	Specifies whether EO Logging should be enabled or disabled. Value should be "on" to enable and "off" to disable.	Optional

Table 4-2 Sfcfsha_config module keywords (continued)

Name	Description	Mandatory/optional
gendeploy_path	<p>Specifies the absolute path of the gendeploy script from the local node to register InfoScale hosts to the VIOM Management Server.</p> <p>It is supported on InfoScale 8.0.2 and later.</p> <p>Example:</p> <pre>gendeploy_path: '/root/script_path.pl'</pre>	Optional

process

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

Table 4-3 Process module keywords

Name	Description	Mandatory/optional
component:	<p>Specify which components you want to configure in your product. Note that the product license acquired must support the required components.</p> <p>Example: SF, VCS, or SFCHA</p>	Mandatory
product:	<p>Specifies the name of the product you want to install.</p> <p>Examples: ENTERPRISE, AVAILABILITY, STORAGE, or FOUNDATION.</p>	Mandatory
product_version:	<p>Specifies the version of the product that you want to install or upgrade.</p> <p>Example: <code>http://xx.xxx.xxx.xx/rpms/</code></p>	Mandatory
state:	<p>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.</p>	Optional

Table 4-3 Process module keywords (continued)

Name	Description	Mandatory/optional
Seednode:	<p>Select any node from the cluster that will be used to run commands related to the operations of that cluster.</p> <p>Seednode can be set as FQDN, IP Address, and short hostname.</p> <p>Examples:</p> <pre>seednode: objstorer820-1-vm17.veritas.com</pre> <pre>seednode: xx.xxx.xxx.xxx</pre> <pre>seednode: objstorer820-1-vm17</pre>	Optional
gendeploy_path	<p>Specifies the absolute path of the gendeploy script from the local node to register InfoScale hosts to the VIOM Management Server.</p> <p>It is supported on InfoScale 8.0.2 and later.</p> <p>Example:</p> <pre>gendeploy_path: '/root/script_path.pl'</pre>	Optional

Upgrading InfoScale using Ansible

This chapter includes the following topics:

- [Ansible modules for upgrading InfoScale](#)
- [List of pre-defined keywords](#)

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 5-1 Upgrade-related operations

Operation	Required modules	Sample playbook
Full upgrade	<ul style="list-style-type: none"> ■ site_factors ■ upgrade 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cluster-47-48-49 gather_facts: False any_errors_fatal: true tasks: - name: Factors veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: Full Upgrade veritas_infoscale: module: upgrade repository_name: repo_infoScale743 product_version: 7.4.3 repository_baseurl: "http://xx.xxx.xx.xx/\ Infoscale/7.4.3/rhel7_x86_64/rpms/" gpgcheck: 1 gpgkey: "http://xx.xxx.xx.xx/\ Infoscale/7.4.3/rhel7_x86_64/rpms/\ RPM-GPG-KEY-veritas-infoscale7" license: ENTERPRISE seednode: <one of the nodes in cluster> state: present factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cluster-47-48-49", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>" }, "register": "facts" }, { "name": "Full Upgrade", "veritas_infoscale": { "module": "upgrade", "repository_name": "repo_infoScale743", "product_version": "7.4.3", "repository_baseurl": "http://xx.xxx.xx.xx/\ Infoscale/7.4.3/rhel7_x86_64/rpms/", "gpgcheck": 1, "gpgkey": "http://xx.xxx.xx.xx/\ Infoscale/7.4.3/rhel7_x86_64/rpms/\ RPM-GPG-KEY-veritas-infoscale7", "license": "ENTERPRISE", "seednode": "<one of the nodes in cluster>", "state": "present", "factors": "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 5-1 Upgrade-related operations (*continued*)

Operation	Required modules	Sample playbook
Rolling upgrade	<ul style="list-style-type: none">■ site_factors■ ru_phase1■ ru_phase2	

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
		<pre> Sample playbook - YAML --- - hosts: <ansible_host_group> gather_facts: False any_errors_fatal: true tasks: - name: Facters veritas_infoscalse: module: site_factors release_matrix_path: <URL> register: facts - hosts: <ansible_host_group> gather_facts: False any_errors_fatal: true serial: 1 tasks: - name: Rolling Upgrade(phase1) Infoscalse veritas_infoscalse: module: ru_phasel repository_name: repo_infoScale743 product_version: '7.4.3' repository_baseurl: "http://xx.xxx.xx.xx/\ Infoscalse/7.4.3/rhel7_x86_64/rpms/" gpgcheck: 1 gpgkey: "http://xx.xxx.xx.xx/\ Infoscalse/7.4.3/rhel7_x86_64/rpms/\ RPM-GPG-KEY-veritas-infoscalse7" license: ENTERPRISE seednode: <one of the nodes in cluster> state: present facts: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscalse_facts']) select() list }}" - hosts: <anible_host_group> gather_facts: False any_errors_fatal: true tasks: - name: Rolling Upgrade(phase2) Infoscalse veritas_infoscalse: module: ru_phase2 </pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
		<pre> repository_name: repo_infoScale743 product_version: '7.4.3' repository_baseurl: "http://xx.xxx.xx.xx/\ Infoscale/7.4.3/rhel7_x86_64/rpms/" gpgcheck: 1 gpgkey: "http://xx.xxx.xx.xx/\ Infoscale/7.4.3/rhel7_x86_64/rpms/\ RPM-GPG-KEY-veritas-infoscale7" license: ENTERPRISE seednode: <one of the nodes in cluster> state: present factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "<ansible_host_group>", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>" }, "register": "facts" }] }, { "hosts": "<ansible_host_group>", "gather_facts": false, "any_errors_fatal": true, "serial": 1, "tasks": [{ "name": "Rolling Upgrade(phasel) Infoscale", "veritas_infoscale": { "module": "ru_phase1", "repository_name": "repo_infoScale743", "product_version": "7.4.3", "repository_baseurl": "http://xx.xxx.xx.xx/\ Infoscale/7.4.3/rhel7_x86_64/rpms/", "gpgcheck": 1, "gpgkey": "http://xx.xxx.xx.xx/\ Infoscale/7.4.3/rhel7_x86_64/rpms/\ RPM-GPG-KEY-veritas-infoscale7", "license": "ENTERPRISE", "seednode": "<one of the nodes in cluster>", "state": "present", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
		<pre> }] }, { "hosts": "<ansible_host_group>", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Rolling Upgrade(phase2) Infoscale", "veritas_infoscale": { "module": "ru_phase2", "repository_name": "repo_infoScale743", "product_version": "7.4.3", "repository_baseurl": "http://xx.xxx.xx.xx/\ Infoscale/7.4.3/rhel7_x86_64/rpms/", "gpgcheck": 1, "gpgkey": "http://xx.xxx.xx.xx/\ Infoscale/7.4.3/rhel7_x86_64/rpms/\ RPM-GPG-KEY-veritas-infoscale7", "license": "ENTERPRISE", "seednode": "<one of the nodes in cluster>", "state": "present", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}" } }] }] </pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
Patch upgrade (method 1)	<ul style="list-style-type: none"> ■ site_factors ■ patchupgrade 	<p>Sample playbook - YAML</p> <pre> --- - hosts: <ansible_host_group> gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: Install InfoScale veritas_infoscale: module: patchupgrade ignore_patchid_check: 0 patch_repo_list: - "file:///mnt/7.4.2/patch_central/P/\ infoscale/rhel7_x86_64/7.4.2.1200/rpms" factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" state: present </pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "<ansible_host_group>", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>" }, "register": "facts" }, { "name": "Install InfoScale", "veritas_infoscale": { "module": "patchupgrade", "ignore_patchid_check": 0, "patch_repo_list": ["file:///mnt/7.4.2/patch_central/P/\ infoscale/rhel7_x86_64/7.4.2.1200/rpms"], "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" , "state": "present" } }] }]</pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
Patch upgrade (method 2)	<ul style="list-style-type: none"> ■ site_factors ■ yum 	<p>Sample playbook - YAML</p> <pre> --- - hosts: <ansible_host_group> gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: Install InfoScale veritas_infoscale: module: yum repository_name: <repository name> repository_baseurl: <repository base URL> gpgcheck: 1 gpgkey: <gpgkey> product: ENTERPRISE product_version: '7.4.2' ignore_patchid_check: 0 patch_repo_list: - "file:///mnt/7.4.2/patch_central/P/\ infoscale/rhel7_x86_64/7.4.2.1200/rpms" facts: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" state: present </pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "<ansible host group>", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>" }, "register": "facts" }, { "name": "Install InfoScale", "veritas_infoscale": { "module": "yum", "repository_name": "<repository name>", "repository_baseurl": "<repository base URL>", "gpgcheck": 1, "gpgkey": "<gpgkey>", "product": "ENTERPRISE", "product_version": "7.4.2", "ignore_patchid_check": 0, "patch_repo_list": ["file:///mnt/7.4.2/patch_central/P\ infoscale/rhel7_x86_64/7.4.2.1200/rpms"], "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}" "state": "present" } }] }]</pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
DG upgrade	<ul style="list-style-type: none"> ■ site_factors ■ DgUpgrade 	<p>Sample playbook - YAML</p> <pre> - hosts: mycluster gather_facts: False become: yes tasks: - name: Factors veritas.infoscale.veritas_infoscale: module: site_factors register: facts - name: DG Upgrade veritas.infoscale.veritas_infoscale: module: DgUpgrade dg_info: "{{ dg_info }}" seednode: <one of the nodes in cluster> facts: "{{ ansible_play_hosts_all map('extract', hostvars['\facts', 'infoscale_facts']) select() list }}" </pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre data-bbox="552 378 1341 1194"> [{ "hosts": "mycluster", "gather_facts": false, "become": yes, "tasks": [{ "name": "Factors", "veritas_infoscald": { "module": "site_factors" }, "register": "facts" }, { "name": " DG Upgrade", "veritas_infoscald": { "module": " DgUpgrade ", "dg_info": "{{ dg_info }}", "seednode": "{{ seednode }}", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscald_facts']) select() list }}" }, "register": "facts" }] }] </pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
DLV upgrade	<ul style="list-style-type: none"> ■ DlvUpgrade 	<p>Sample playbook - YAML</p> <pre> - hosts: mycluster gather_facts: False become: yes tasks: - name: Facters veritas.infoscale.veritas_infoscale: module: site_facters register: facts - name: DLV Upgrade veritas.infoscale.veritas_infoscale: module: DlvUpgrade fs_info: "{{ fs_info }}" seednode: <one of the nodes in cluster> facters: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre>

Table 5-1 Upgrade-related operations (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre data-bbox="552 378 1231 1225">[{ "hosts": "mycluster", "gather_facts": false, "become": yes, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_facters" }, "register": "facts" }, { "name": " DLV Upgrade", "veritas_infoscale": { "module": " DlvUpgrade ", "fs_info": "{{ fs_info }}", "seednode": "{{ seednode }}", "facters": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" }, "register": "facts" }] }]</pre>

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 5-2 Upgrade module keywords

Name	Description	Mandatory/optional
product_version:	Specifies the version of the product that you want to install or upgrade. Example: <code>http://xx.xxx.xxx.xx/rpms/</code>	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: <code>http://xx.xxx.xxx.xx/rpms/ RPM-GPG-KEY-veritas-infoscale7</code>	Mandatory, if gpgcheck is 1
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. Note: Perpetual key is not applicable for InfoScale 8.0.2 onwards. For installation, configuration, or upgrade operations with InfoScale 8.0.2 or later, do not use the license parameter. If you are performing a keyless installation or upgrade you can simply enter the product name. Examples: <ul style="list-style-type: none"> ■ <code>/license_key/Unix/perpetual/ xxxxxxxxxxxxxx.slf</code> ■ ENTERPRISE ■ AVAILABILITY 	Optional

Table 5-2 Upgrade module keywords (continued)

Name	Description	Mandatory/optional
seednode	<p>Select any node from the cluster that will be used to run commands related to the operations of that cluster.</p> <p>Seednode can be set as FQDN, IP Address, and short hostname. Examples:</p> <pre>seednode: objstorer820-1-vm17.veritas.com</pre> <pre>seednode: xx.xxx.xxx.xxx</pre> <pre>seednode: objstorer820-1-vm17</pre> <p>Ensure that you enter the host name as provided in the <code>/etc/ansible/hosts</code> file.</p>	Mandatory
ignore_patchid_check	<p>Specifies whether patch ID of the patch should be verified with that of the respective InfoScale version. The value of this keyword can be either 0 or 1.</p>	Optional
patch_repo_list	<p>Specifies the list of repositories which provide the patch paths or repository names.</p> <ul style="list-style-type: none"> ■ If patch repositories are created and already exists on the hosts then this list should contain the name of those repositories. ■ If patch repositories are not created then this list should contain patch paths. ■ A combination of both - patch paths and repository names are also accepted in this list. <p>For example:</p> <pre>patch_repo_list:["ansible_patch_repo1", "ansible_patch_repo2", patch_repo_list:["file:///mnt/linux/7.4.2/patch_central/P/ infoscale/rhel7_x86_64/7.4.2.1200/rpms/"] patch_repo_list:["ansible_patch_repo1", "file:///mnt/linux/7.4.2/patch_central/P/ infoscale/rhel7_x86_64/7.4.2.1200/rpms/"]</pre>	Optional

Table 5-2 Upgrade module keywords (continued)

Name	Description	Mandatory/optional
gendeploy_path	<p>Specifies the absolute path of the gendeploy script from the local node to register InfoScale hosts to the VIOM Management Server.</p> <p>It is supported on InfoScale 8.0.2 and later.</p> <p>Example:</p> <pre>gendeploy_path: '/root/script_path.pl'</pre>	Optional

ru_phasel

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 5-3 Ru_phasel module keywords

Name	Description	Mandatory/optional
product_version:	<p>Specifies the version of the product that you want to install or upgrade.</p> <p>Example: <code>http://xx.xxx.xxx.xx/rpms/</code></p>	Mandatory
gpgcheck:	<p>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.</p>	Optional
gpgkey:	<p>Specifies the location of the gpgkey (typically located in the rpms directory of the installation media).</p> <p>Example:</p> <pre>http://xx.xxx.xxx.xx/sde2/7.4.2/ dvd1-redhatlinux/rhel7_x86_64/rpms/ RPM-GPG-KEY-veritas-infoscale7</pre>	Mandatory. if gpgcheck is 1
product:	<p>Specifies the name of the product you want to install.</p> <p>Examples: ENTERPRISE, AVAILABILITY, STORAGE, or FOUNDATION.</p>	Mandatory
state:	<p>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.</p>	Optional

Table 5-3 Ru_phase1 module keywords (continued)

Name	Description	Mandatory/optional
license:	<p>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.</p> <p>Note: Perpetual key is not applicable for InfoScale 8.0.2 onwards.</p> <p>For installation, configuration, or upgrade operations with InfoScale 8.0.2 or later, do not use the license parameter.</p> <p>If you are performing a keyless installation or upgrade you can simply enter the product name.</p> <p>Examples:</p> <ul style="list-style-type: none"> ■ /license_key/Unix/perpetual/ xxxxxxxxxxxxxxxxxxxxx.slf ■ ENTERPRISE ■ AVAILABILITY 	Optional
component:	<p>Specify which components you want to configure in your product. Note that the product license acquired must support the required components.</p> <p>Example: SF, VCS, or SFCHA</p>	Mandatory
start_process	<p>Specifies to start InfoScale processes if value is set to yes. By default this option is set to no.</p>	Optional
seednode:	<p>Select any node from the cluster that will be used to run commands related to the operations of that cluster.</p> <p>Seednode can be set as FQDN, IP Address, and short hostname.</p> <p>Examples:</p> <pre>seednode: objstorer820-1-vm17.veritas.com</pre> <pre>seednode: xx.xxx.xxx.xxx</pre> <pre>seednode: objstorer820-1-vm17</pre> <p>Ensure that you enter the host name as provided in the /etc/ansible/hosts file.</p>	Mandatory
ignore_patchid_check	<p>Specifies whether patch ID of the patch should be verified with that of the respective InfoScale version. The value of this keyword can be either 0 or 1.</p>	Optional

Table 5-3 Ru_phase1 module keywords (continued)

Name	Description	Mandatory/optional
patch_repo_list	<p>Specifies the list of repositories which provide the patch paths or repository names.</p> <ul style="list-style-type: none"> ■ If patch repositories are created and already exists on the hosts then this list should contain the name of those repositories. ■ If patch repositories are not created then this list should contain patch paths. ■ A combination of both - patch paths and repository names are also accepted in this list. <p>For example:</p> <pre>patch_repo_list:["ansible_patch_repo1", "ansible_patch_repo2", patch_repo_list:["file:///mnt/linux/7.4.2/patch_central/P/ infoscale/rhel7_x86_64/7.4.2.1200/rpms/"] patch_repo_list:["ansible_patch_repo1", "file:///mnt/linux/7.4.2/patch_central/P/ infoscale/rhel7_x86_64/7.4.2.1200/rpms/"]</pre>	Optional

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 5-4 ru_phase2 module keyword

Name	Description	Mandatory/optional
product_version:	<p>Specifies the version of the product that you want to install or upgrade.</p> <p>Example: <code>http://xx.xxx.xxx.xx/rpms/</code></p>	Mandatory
gpgcheck:	<p>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.</p>	Optional

Table 5-4 ru_phase2 module keyword (continued)

Name	Description	Mandatory/optional
gpgkey:	<p>Specifies the location of the gpgkey (typically located in the rpms directory of the installation media).</p> <p>Example:</p> <pre>http://xx.xxx.xxx.xx/rpms/ RPM-GPG-KEY-veritas-infoscale7</pre>	Mandatory if gpgcheck is 1
product:	<p>Specifies the name of the product you want to install.</p> <p>Examples: ENTERPRISE, AVAILABILITY, STORAGE, or FOUNDATION.</p>	Mandatory
state:	<p>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.</p>	Optional
license:	<p>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.</p> <p>Note: Perpetual key is not applicable for InfoScale 8.0.2 onwards.</p> <p>For installation, configuration, or upgrade operations with InfoScale 8.0.2 or later, do not use the license parameter.</p> <p>If you are performing a keyless installation or upgrade you can simply enter the product name.</p> <p>Examples:</p> <ul style="list-style-type: none"> ■ /license_key/Unix/perpetual/ xxxxxxxxxxxxxxxxxxxxxxxxxxxx.slf ■ ENTERPRISE ■ AVAILABILITY 	Optional
component:	<p>Specify which components you want to configure in your product. Note that the product license acquired must support the required components.</p> <p>Example: SF, VCS, or SFCHA</p>	Mandatory
start_process	<p>Specifies to start InfoScale processes if value is set to yes. By default this option is set to no.</p>	Optional

Table 5-4 ru_phase2 module keyword (continued)

Name	Description	Mandatory/optional
seednode:	<p>Select any node from the cluster that will be used to run commands related to the operations of that cluster.</p> <p>Seednode can be set as FQDN, IP Address, and short hostname Examples:</p> <pre>seednode: objstorer820-1-vm17.veritas.com</pre> <pre>seednode: xx.xxx.xxx.xxx</pre> <pre>seednode: objstorer820-1-vm17</pre> <p>Ensure that you enter the host name as provided in the /etc/ansible/hosts file.</p>	Mandatory
patch_repo_list	<p>Specifies the list of repositories which provide the patch paths or repository names.</p> <ul style="list-style-type: none"> ■ If patch repositories are created and already exists on the hosts then this list should contain the name of those repositories. ■ If patch repositories are not created then this list should contain patch paths. ■ A combination of both - patch paths and repository names are also accepted in this list. <p>For example:</p> <pre>patch_repo_list:["ansible_patch_repo1", "ansible_patch_repo2", patch_repo_list:["file:///mnt/linux/7.4.2/patch_central/P/ infoscale/rhel7_x86_64/7.4.2.1200/rpms/"] patch_repo_list:["ansible_patch_repo1", "file:///mnt/linux/7.4.2/patch_central/P/ infoscale/rhel7_x86_64/7.4.2.1200/rpms/"]</pre>	Optional
gendeploy_path	<p>Specifies the absolute path of the gendeploy script from the local node to register InfoScale hosts to the VIOM Management Server.</p> <p>It is supported on InfoScale 8.0.2 and later.</p> <p>Example:</p> <pre>gendeploy_path: '/root/script_path.pl'</pre>	Optional

patchupgrade module keywords

The patchupgrade module upgrades patch versions for the provided InfoScale version accordingly. A patchupgrade can be performed while installing InfoScale or even post configuration of InfoScale components on the hosts. It upgrades the patches to the highest supported patch versions for the provided InfoScale version.

Table 5-5 patchupgrade module keywords

Names	Description	Mandatory/Optional
ignore_patchid_check	Specifies whether patch ID of the patch should be verified with that of the respective InfoScale version. The value of this keyword can be either 0 or 1.	Optional
patch_repo_list	<p>Specifies the list of repositories which provide the patch paths or repository names.</p> <ul style="list-style-type: none"> ■ If patch repositories are created and already exists on the hosts then this list should contain the name of those repositories. ■ If patch repositories are not created, then this list should contain patch paths. ■ A combination of both – patch paths and repository names are also accepted in this list. <p>Examples</p> <pre>patch_repo_list: ["ansible_patch_repo1", "ansible_patch_repo2",] patch_repo_list: ["file:///mnt/linux/7.4.2/patch_central/P/ infoscale/rhel7_x86_64/7.4.2.1200/rpms/"] patch_repo_list: ["ansible_patch_repo1", "file:///mnt/linux/7.4.2/patch_central/P/ infoscale/rhel7_x86_64/7.4.2.1200/rpms/"]</pre>	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. Default value is 'present'.	Mandatory
factors	<p>Enter the following value to collect system-related data from the cluster servers.</p> <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}"</pre>	Mandatory

Table 5-5 patchupgrade module keywords (continued)

Names	Description	Mandatory/Optional
gendeploy_path	<p>Specifies the absolute path of the gendeploy script from the local node to register InfoScale hosts to the VIOM Management Server.</p> <p>It is supported on InfoScale 8.0.2 and later.</p> <p>Example:</p> <pre>gendeploy_path: '/root/script_path.pl'</pre>	Optional

DgUpgrade model keywords

The DgUpgrade module is used to perform a Disk Group upgrade in InfoScale. Use the following keywords while referencing the module in your playbook.

Table 5-6 DgUpgrade model keywords

Names	Description	Mandatory/Optional
dg_info	<p>Provides disk group information such as name of the disk group and the required upgrade version. If version information is not provided, then it will be upgraded to the latest available version.</p> <p>Example</p> <pre>[[{ dgname: 'dg1', version: '290'}, { dgname: 'dg2' }]]</pre>	Mandatory
seednode	<p>Select any node from the cluster that will be used to run commands related to the operations of that cluster.</p> <p>Seednode can be set as FQDN, IP Address, and short hostname.</p> <p>Examples:</p> <pre>seednode: objstorer820-1-vm17.veritas.com</pre> <pre>seednode: xx.xxx.xxx.xxx</pre> <pre>seednode: objstorer820-1-vm17</pre> <p>Ensure that you enter the host name as provided in the /etc/ansible/hosts file.</p>	Mandatory

DlvUpgrade model keywords

The DlvUpgrade module is used to perform a Disk Layout Version (DLV) upgrade in InfoScale. Use the following keywords while referencing the module in your playbook.

Table 5-7 DgUpgrade model keywords

Names	Description	Mandatory/Optional
fs_info	<p>Provides file system information such as vxfs mountpoint name and the required upgrade version. If version information is not provided, then DLV will be upgraded to the latest available version.</p> <p>Example</p> <pre>fs_info: [{ mountpoint: '/dg1_vol', version: 16}]</pre>	Mandatory
seednode	<p>Select any node from the cluster that will be used to run commands related to the operations of that cluster.</p> <p>Seednode can be set as FQDN, IP Address, and short hostname.</p> <p>Examples:</p> <pre>seednode: objstor820-1-vm17.veritas.com</pre> <pre>seednode: xx.xxx.xxx.xxx</pre> <pre>seednode: objstor820-1-vm17</pre> <p>Ensure that you enter the host name as provided in the /etc/ansible/hosts file.</p>	Mandatory

Configuring InfoScale features using Ansible

This chapter includes the following topics:

- [Ansible modules for configuring features in InfoScale](#)
- [List of pre-defined keywords](#)

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 6-1 Feature configuration-related keywords

Operation	Required modules	Sample playbook
Configuring a Cluster File System (CFS)	<ul style="list-style-type: none"> ■ site_factors ■ cfsresource 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscalse: module: site_factors release_matrix_path: <URL> register: facts - name: CFS Resource veritas_infoscalse: module: cfsresource state: present sname: testsg1 dname: testdg5 volname: testvol1 mnt: /testvol1 seednode: objstorer820-1-vm17 factors: "{{ ansible_play_hosts_all map('extract', hostvars,['facts','infoscalse_facts']) select() list }}" </pre>

Table 6-1 Feature configuration-related keywords (continued)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cpicluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>" }, "register": "facts" }, { "name": "CFS Resource", "veritas_infoscale": { "module": "cfsresource", "state": "present", "sgname": "testsg1", "dgname": "testdg5", "volname": "testvoll", "mnt": "/testvoll", "seednode": "objstor820-1-vm17", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Performing service group operations	<ul style="list-style-type: none"> ■ <code>site_factors</code> ■ <code>servicegroup_operations</code> 	<p>Sample playbook - YAML</p> <pre> - hosts: mycluster gather_facts: false become: yes tasks: - name: Factors veritas_infoscale: module: site_factors register: facts - name: Service Group Operations veritas_infoscale: module: servicegroup_operations service_group_tree: "{{ service_group_tree }}" seednode: "{{ seednode }}" facts: "{{ ansible_play_hosts_all \ map('extract', hostvars, \ ['facts','infoscale_facts']) \ select() list }}" register: result </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre data-bbox="548 409 1327 1256">[{ "hosts": "mycluster", "gather_facts": false, "become": yes, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_factors" }, "register": "facts" }, { "name": " Service Group Operations", "veritas_infoscale": { "module": " servicegroup_operations ", "service_group_tree": "{{ service_group_tree }}", "seednode": "{{ seednode }}", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" }, "register": "facts" }] }]</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Creating a disk group volume	<ul style="list-style-type: none"> ■ site_factors ■ vxvm_dgvolfs 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cluster2 gather_facts: false any_errors_fatal: true tasks: - name: Init disks veritas_infoscalse: module: DiskInit init: initdisks: True disks: ['disk1', 'disk2', 'disk3'] register: disk_info - name: Factors veritas_infoscalse: module: site_factors register: facts - name: Create DG, Volume, FS veritas_infoscalse: module: vxvm_dgvolfs state: present dg_info: - dgname: custappdg dgtype: shared disks: - 10-132-1-56_xen-vd0_1 - 10-132-1-56_xen-vd0_2 - 10-132-0-55_xen-vd0_1 - 10-132-0-55_xen-vd0_2 fss: 1 seednode: ip-10-132-0-55 taginfo: volinfo: - - custappvol - 9.00g - 'layout=mirror-stripe nstripe=2 nmirrors=2' factors: "{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cluster2", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Init disks", "veritas_infoscale": { "module": "DiskInit", "init": { "initdisks": true }, "disks": ["disk1", "disk2", "disk3"] } }, { "name": "Factors", "veritas_infoscale": { "module": "site_factors" }, "register": "facts" }, { "name": "Create DG, Volume, FS", "veritas_infoscale": { "module": "vxvm_dgvols", "state": "present", "dg_info": [{ "dgname": "custappdg", "dgtype": "shared", "disks": ["10-132-1-56_xen-vd0_1", "10-132-1-56_xen-vd0_2", "10-132-0-55_xen-vd0_1",] }] } }] }]</pre>

Table 6-1 Feature configuration-related keywords (continued)

Operation	Required modules	Sample playbook
		<pre> "10-132-0-55_xen-vd0_2"], "fss": 1, "seednode": "ip-10-132-0-55", "taginfo": null, "volinfo": [["custappvol", "9.00g", "layout=mirror-stripe nstripe=2 nmirrors=2"]] }], "factors": "{(ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}" } }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Configuring a File System (FS) resource	<ul style="list-style-type: none"> ■ site_factors ■ fsresource 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cpicluster10 gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: Create FS resource veritas_infoscale: module: fsresource state: present sgnose: testdgsg5 dgname: testdg5 volname: testvol5 mnt: /testvol5 dgresname: dgres5 volresname: volres5 mntresname: mntres5 sgtype: Parallel factors: "{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }"} </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cpicluster10", "gather_facts": false, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_factors" }, "register": "facts" }, { "name": "Create FS resource", "veritas_infoscale": { "module": "fsresource", "state": "present", "sgname": "testdgsg5", "dgname": "testdg5", "volname": "testvol5", "mnt": "/testvol5", "dgresname": "dgres5", "volresname": "volres5", "mntresname": "mntres5", "sgtype": "Parallel", "facts": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Resize FS	Sample playbook - YAML <ul style="list-style-type: none"> ■ <code>site_factors</code> ■ <code>resize_volume</code> 	<pre> --- - hosts: mycluster gather_facts: False any_errors_fatal: true tasks: - name: Factors veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: Resize volume veritas_infoscale: module: resize_volume size: '5084m' operation: 'growto' dgname: '' volname: '' mountpoint: '/mnt1' state: present seednode:<one of the nodes in the cluster> factors: "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre data-bbox="548 378 1300 1341"> [{ "hosts": "mycluster", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Resize volume", "veritas_infoscale": { "module": "resize_volume", "size": "5084m", "operation": "growto", "dgname": "", "volname": "", "mountpoint": "/mnt1", "state": "present", "seednode": "<one of the nodes in the cluster>", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Configuring a Arctera Volume Replicator (VVR) or Cluster Volume Replication (CVR)	<ul style="list-style-type: none"> ■ site_factors ■ vvrresource 	

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - YAML</p> <pre> --- - hosts: <ansible host group> gather_facts: False tasks: - name: Factors veritas_infoscale: module: site_factors release_matrix_path: <URL> register: facts - name: VVR/CVR Resource veritas_infoscale: module: vvrresource seednode: <hostname> primaryvip: xx.xx.xx.xx secondaryvip: xx.xx.xx.xx dgname: testdg1 datavolname: testvol srlvolname: srlvoll rvgname: rvg_vvrtest1 createsecondaryonly: 1 createprimaryonly: 0 start_replication: 0 agentinfo: rvg_sg_name: custom_RVG primary: clusterlist: - mycluster_47 - mycluster_u networkhosts: - xx.xx.xx.xx nic: eth0 netmask: 255.255.240.0 mountpoint_info: - volname: testvol mounname: /new_pri mount_res_name: custom_mount_new requires_res: [''] secondary: nic: ens192 netmask: 255.255.240.0 mountpoint_info: </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre> - volname: testvol mounthname: /new_sec mount_res_name: custom_mount_new requires_res: [''] clusterlist: - mycluster_47 - mycluster_u networkhosts: - xx.xx.xx.xx rvg_res_name : null mount_sg_name : null 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 state: present factors: "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }"} </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "<ansible host group>", "gather_facts": false, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>" }, "register": "facts" }, { "name": "VVR/CVR Resource", "veritas_infoscale": { "module": "vvrresource", "seednode": "<hostname>", "primaryvip": "xx.xx.xx.xx", "secondaryvip": "xx.xx.xx.xx", "dgname": "testdgl", "datavolname": "testvol", "srlvolname": "srlvoll", "rvgname": "rvg_vvrtest1", "createsecondaryonly": 1, "createprimaryonly": 0, "start_replication": 0, "agentinfo": { "rvg_sg_name": "custom_RVG", "primary": { "clusterlist": ["mycluster_47", "mycluster_u"], "networkhosts": ["xx.xx.xx.xx"], "nic": "eth0", "netmask": "255.255.240.0", "mountpoint_info": [{</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre> "volname": "testvol", "mountname": "/new_pri", "mount_res_name": "custom_mount_new", "requires_res": [""] }] }, "secondary": { "nic": "ens192", "netmask": "255.255.240.0", "mountpoint_info": [{ "volname": "testvol", "mountname": "/new_sec", "mount_res_name": "custom_mount_new", "requires_res": [""] }] }, "clusterlist": ["mycluster_47", "mycluster_u"], "networkhosts": ["xx.xx.xx.xx"] }], "rvg_res_name": null, "mount_sg_name": null, "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, </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre> "logownernic_res_name": null }, "state": "present", "factors": "{ { ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list } }" }] }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Adding nodes	<ul style="list-style-type: none"> ■ site_factors ■ add_node 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscalse: module: site_factors release_matrix_path: <URL> register: facts - name: Add/Del node veritas_infoscalse: 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 factors: "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre data-bbox="548 378 1237 1402">[{ "hosts": "cpiccluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Add/Del node", "veritas_infoscale": { "module": "add_node", "state": "present", "clusternode": "dl1380g10-kvm-12-03", "add_del_nodes": ["dl1380g10-kvm-12-56", "dl1380g10-kvm-12-57"], "cluster_id": 34321, "cluster_uuid": "c7c2d65e-058f-11e8-a32c-c094107f3b61", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Deleting nodes	<ul style="list-style-type: none"> ■ site_factors ■ add_node 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscalse: module: site_factors release_matrix_path: <URL> register: facts - name: Add/Del node veritas_infoscalse: 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 factors: "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "cpiccluster11", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "Add/Del node", "veritas_infoscale": { "module": "add_node", "state": "absent", "clusternode": "dl1380g10-kvm-12-03", "add_del_nodes": ["dl1380g10-kvm-12-56", "dl1380g10-kvm-12-57"], "cluster_id": 34321, "cluster_uuid": "c7c2d65e-058f-11e8-a32c-c094107f3b61", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Configure VIP	<ul style="list-style-type: none"> ■ site_factors ■ VipConfig 	<p>Sample playbook - YAML</p> <pre> --- - hosts: <ansible host group> gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscalse: module: site_factors release_matrix_path: <URL> register: facts - name: Configure VIP veritas_infoscalse: module: VipConfig nic: ens3 vip: 'xx.xx.xx.xx' netmask: 'xxx.xxx.xxx.xxx' facters: "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscalse_facts']) select() list }" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "node_vip", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_facters", "release_matrix_path": "<URL>" }, "register": "facts" }, { "name": "Configure VIP", "veritas_infoscale": { "module": "VipConfig", "nic": "ens3", "vip": " xx.xx.xx.xx ", "netmask": "xxx.xxx.xxx.xxx ", "facters": "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Configure Secure Cluster	<ul style="list-style-type: none"> ■ site factors ■ SecureCluster 	<p>Sample playbook - YAML</p> <pre> --- - hosts: <ansible host group> gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscail: module: site_factors register: facts - name: Secure Cluster veritas_infoscail: module: SecureCluster cluster_id: cluster_name: clust_cpi9 seednode: dl380g10-12-kvm-49 state: present fips: 0 read_access: 0 usergroups: facts: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "<ansible host group>", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_facters" }, "register": "facts" }, { "name": "Secure Cluster", "veritas_infoscale": { "module": "SecureCluster", "cluster_id": null, "cluster_name": "clust_cpi9", "seednode": "dl380g10-12-kvm-49", "state": "present", "fips": 0, "read_access": 0, "usergroups": null, "facters": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Configure GCO	<ul style="list-style-type: none"> ■ <code>site_factors</code> ■ <code>gco_config</code> 	<p>Sample playbook - YAML</p> <pre> --- - hosts: <ansible host group> gather_facts: False any_errors_fatal: true tasks: - name: Factors veritas_infoscail: module: site_factors release_matrix_path: <URL> register: facts - name: GCO configuration veritas_infoscail: module: gco_config primary: prefix: '' nic: {'<primary site hostname>' : 'ens192'} vip: xx.xxx.xxx.xx ipver: 4 netmask: xx.xxx.xxx.xx secondary: prefix: '' nic: {'<secondary site hostname>' : 'ens192'} vip: xx.xxx.xxx.xx ipver: 4 netmask: xx.xxx.xxx.xx primary_seednode: <primary site hostname> secondary_seednode: <secondary site hostname> state: present extra_args: null factors: "{ { ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list } }"</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre>[{ "hosts": "<ansible host group>", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_factors", "release_matrix_path": "<URL>", }, "register": "facts" }, { "name": "GCO configuration", "veritas_infoscale": { "module": "gco_config", "primary": { "prefix": "", "nic": { "<primary site hostname>": "ens192" }, "vip": "xx.xxx.xxx.xx", "ipver": 4, "netmask": "xx.xxx.xxx.xx" }, "secondary": { "prefix": "", "nic": { "<secondary site hostname>": "ens192" }, "vip": "xx.xxx.xxx.xx", "ipver": 4, "netmask": "xx.xxx.xxx.xx" }, "primary_seednode": "<primary site hostname>", "secondary_seednode": "<secondary site hostname>", "state": "present", "extra_args": null, "facters": "{ ansible_play_hosts_all</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre> map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Configure REST server	<ul style="list-style-type: none"> ■ site_factors ■ RestServerConfig 	<pre> Configure REST server --- - hosts: <any one ansible host from the cluster on which you want to configure REST server> gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscail: module: site_factors register: facts - name: Configure REST Server veritas_infoscail: module: RestServerConfig REST_server_ip: xx.xx.xx.xx REST_server_netmask: xx.xx.xx.xx REST_server_nic: eth0 REST_server_port: '5637' REST_server_username: ldapuser1 REST_server_ldap_domain: 'example.com' REST_server_ldap_ip: xx.xx.xx.xx REST_server_key: '/certs/server.key' REST_server_cert_file: '/certs/server.crt' REST_server_cacert_file: '/certs/serverca.crt' REST_server_passphrase: rest_config: enabled_ldap reconfig: state: present factors: "{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample Playbook - JSON</p> <pre>[{ "hosts": "<any one ansible host from the cluster on which you want to configure REST server>", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_facters" }, "register": "facts" }, { "name": "Configure REST Server", "veritas_infoscale": { "module": "RestServerConfig", "REST_server_ip": "xx.xx.xx.xx", "REST_server_netmask": "xx.xx.xx.xx", "REST_server_nic": "eth0", "REST_server_port": "5637", "rest_config": "enabled_ldap", "REST_server_username": "ldapuser1" "REST_server_ldap_domain": "example.com" "REST_server_ldap_ip": "xx.xx.xx.xx" "REST_server_key": "/certs/server.key" "REST_server_cert_file": "/certs/server.crt" "REST_server_cacert_file": "/certs/serverca.crt" "REST_server_passphrase": null "reconfig": null, "state": "present", "facters": "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] }]</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Configure CP Server	site factors CPServerConfig	<p>Sample playbook - YAML</p> <pre> --- - hosts: cluster-vip-06 gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscale: module: site_factors register: facts - name: Configure CP Server veritas_infoscale: module: CPServerConfig cps_db_dir: /etc/VRTScps/db cps_name: cps1 cps_network: - vip: ["xx.xx.xx.xx"] netmask: ["xx.xx.xx.xx"] port: 443 systems: - hosts: dl380g9-109vip06 nic: eth0 cps_diskgroup: fendg1 cps_volume: fenvol1 cps_newdg_disk: cps_newvol_volsize: cps_delete_database: 1 cps_delete_config_logs: 1 reconfig: state: present facts: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre data-bbox="548 378 1135 1576"> [{ "hosts": "cluster-vip-06", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_facters" }, "register": "facts" }, { "name": "Configure CP Server", "veritas_infoscale": { "module": "CPServerConfig", "cps_db_dir": "/etc/VRTScps/db", "cps_name": "cps1", "cps_network": [{ "vip": ["xx.xx.xx.xx"], "netmask": ["xx.xx.xx.xx"], "port": 443, "systems": [{ "hosts": "dl380g9-109vip06", "nic": "eth0" }] }] }, "cps_diskgroup": "fendg1", "cps_volume": "fenvol1", "cps_newdg_disk": null, "cps_newvol_volsize": null, "cps_delete_database": 1, }] }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre> "cps_delete_config_logs": 1, "reconfig": null, "state": "present", "factors": "{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" }] }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Enable/Disable EO-compliant logging	site factors Logging	Sample Playbook - YAML <pre> --- - hosts: cluster-44-50 gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscale: module: site_factors register: facts - name: Logging veritas_infoscale: module: Logging factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre> Sample Playbook - JSON [{ "hosts": "cluster-44-50", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_facters" }, "register": "facts" }, { "name": "Logging", "veritas_infoscale": { "module": "Logging", "facters": "{(ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list)}" } }] }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Setting Tunables	site factor SetTunable	

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - YAML</p> <pre> --- - hosts: cluster-vm gather_facts: no tasks: - name: Factors veritas_infoscale: module: site_factors register: facts - name: Set Tunables veritas_infoscale: module: SetTunable llt: - systems: ['dl380g9-127-vm3', 'dl380g9-127-vm4'] tunables: - "set-flow window: 520" - "set-flow highwater: 12000" - systems: ['all'] tunables: - "set-flow window: 521" - "set-flow highwater: 12001" fs: - systems: ['dl380g9-127-vm3'] tunables: - /mnt1: eo_logging_enable: 1 max_retention_time: 0 - global: eo_logging_enable: 0 - systems: ['all'] tunables: - /mnt1: eo_logging_enable: 1 max_retention_time: 0 - global: eo_logging_enable: 0 vm: - systems: ['all'] tunables: volraid_rsrtransmax: 1 voliot_max_open: 32 facters : "{{ ansible_play_hosts_all </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre data-bbox="548 326 1150 383"> map('extract', hostvars, ['facts','infoscale_facts']) select() list }}"</pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre data-bbox="548 378 1099 1576"> [{ "hosts": "cluster-vm", "gather_facts": "no", "tasks": [{ "name": "Facters", "veritas_infoscale": { "module": "site_factors" }, "register": "facts" }, { "name": "Set Tunables", "veritas_infoscale": { "module": "SetTunable", "l1t": [{ "systems": ["dl380g9-127-vm3", "dl380g9-127-vm4"], "tunables": ["set-flow window: 520", "set-flow highwater: 12000"] }, { "systems": ["all"], "tunables": ["set-flow window: 521", "set-flow highwater: 12001"] }] } }, { "fs": [{ "systems": ["dl380g9-127-vm3"] }] }] }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre>], "tunables": [{ "/mnt1": { "eo_logging_enable": 1, "max_retention_time": 0 } }, { "global": { "eo_logging_enable": 0 } }] }, { "systems": ["all"], "tunables": [{ "/mnt1": { "eo_logging_enable": 1, "max_retention_time": 0 } }, { "global": { "eo_logging_enable": 0 } }] }], "vm": [{ "systems": ["all"], "tunables": { "volraid_rsrtransmax": 1, "voliot_max_open": 32 } } </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre> }], "facters": "{ { ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}" }] }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Setting Route and rules for subnets across AZ on AWS	site factors RouteConfig	<p>Sample playbook - YAML</p> <pre> --- - hosts: cluster2 gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscalse: module: site_factors register: facts - name: Configure Routes and Rules veritas_infoscalse: module: RouteConfig udp_link: - system: 'ip-10-0-0-71' heartbeat_link: - nic: eth1 ip: xx.xx.xx.xx /24 - nic: eth2 ip: xx.xx.xx.xx /24 - system: 'ip-10-0-10-168' heartbeat_link: - nic: eth1 ip: xx.xx.xx.xx /24 - nic: eth2 ip: xx.xx.xx.xx/24 facts : "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre data-bbox="548 378 1122 1576"> [{ "hosts": "cluster2", "gather_facts": false, "any_errors_fatal": true, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors" }, "register": "facts" }, { "name": "Configure Routes and Rules", "veritas_infoscale": { "module": "RouteConfig", "udp_link": [{ "system": "ip-10-0-0-71", "heartbeat_link": [{ "nic": "eth1", "ip": " xx.xx.xx.xx/24" }, { "nic": "eth2", "ip": " xx.xx.xx.xx/24" }] }] }, { "system": "ip-10-0-10-168", "heartbeat_link": [{ "nic": "eth1", "ip": " xx.xx.xx.xx/24" }, { "nic": "eth2", "ip": "xx.xx.xx.xx/24" }] }]] }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre> }] }], "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}" } }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Add disks to diskgroup	<ul style="list-style-type: none"> ■ site_factors ■ DiskInit ■ adddisks 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cluster2 gather_facts: false any_errors_fatal: true tasks: - name: Init disks veritas_infoscale: module: DiskInit init: initdisks: True fss: True disks: ['disk1', 'disk2'] register: disk_info - name: Factors veritas_infoscale: module: site_factors register: facts - name: Add disks to Diskgroup veritas_infoscale: module: adddisks state: present dgname: testdg disks: "{{ disk_info['disks'] }}" factors: "{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre data-bbox="548 378 1280 1576"> [{ "hosts": "cluster2", "gather_facts": false, "any_errors_fatal": true, }, "tasks": [{ "name": "Init disks", "veritas_infoscale": { "module": "DiskInit", "init": { "initdisks": true, "fss": true }, "disks": ["disk1", "disk2"] }, "register": "disk_info" }, { "name": "Facters", "veritas_infoscale": { "module": "site_factors" }, "register": "facts" }, { "name": "Add disks to Diskgroup", "veritas_infoscale": { "module": "adddisks", "state": "present", "dgname": "testdg", "disks": "{{ disk_info['disks'] }}", "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" } }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<pre>] }] </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Remove disks from diskgroup	<ul style="list-style-type: none"> ■ site_factors ■ adddisks 	<p>Sample playbook - YAML</p> <pre> --- - hosts: cluster2 gather_facts: false any_errors_fatal: true tasks: - name: Factors veritas_infoscale: module: site_factors register: facts - name: Remove disks from Diskgroup veritas_infoscale: module: adddisks state: absent dgname: testdg disks: ['disk1', 'disk2'] facts: "{ { ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}" </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
		<p>Sample playbook - JSON</p> <pre data-bbox="545 378 1280 1255"> [{ "hosts": "cluster2", "gather_facts": false, "any_errors_fatal": true, }, "tasks": [{ "name": "Factors", "veritas_infoscale": { "module": "site_factors" }, "register": "facts" }, { "name": "Remove disks from Diskgroup", "veritas_infoscale": { "module": "adddisks", "state": "absent", "dgname": "testdg", "disks": ["disk1", "disk2"], "factors": "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" }] } </pre>

Table 6-1 Feature configuration-related keywords (*continued*)

Operation	Required modules	Sample playbook
Change Replication Status	<ul style="list-style-type: none"> ■ site facter ■ change _replication _state 	<p>Sample playbook YAML</p> <pre> --- - hosts: <ansible host group> gather_facts: False become: yes tasks: - name: Change replcation state veritas_infoscale: module: change_replication_state newstate: start rvg_info: [{'dg_name': 'checkdg', 'rvg_list': ['rvg_test1']}] secondaryvip: "{{ vtas_secondaryvip }}" </pre> <hr/> <p>Sample playbook JSON</p> <pre> [{ "hosts": "<ansible host group>", "gather_facts": false, "become": "yes", "tasks": [{ "name": "Change replication state", "veritas_infoscale": { "module": "ChangeReplicationState", "newstate": "start", "rvg_info": [{ "dg_name": "checkdg", "rvg_list": ["rvg_test1"] }] } }, { "secondaryvip": "{{ vtas_secondaryvip }}" }] }] </pre>

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 6-2 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
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
factors:	Enter the following value to collect system-related data from the cluster servers. <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}"</pre>	Mandatory

servicegroup_operations

The servicegroup_operations module is used to create service groups, related resources, resource dependencies and service group dependencies in Infoscale. Use the following keywords while referencing the module in your playbook:

Table 6-3 servicegroup_operations module keywords

Name	Description	Mandatory / Optional
service_group_tree:		Mandatory

Table 6-3 servicegroup_operations module keywords (continued)

Name	Description	Mandatory / Optional
	<p>Specify the json file containing service group tree structure to include service group, related resources, attributes, resource dependency and group dependency information.</p> <p>Sample Input File - servicegroup_operation.json</p> <pre data-bbox="346 513 884 1564"> { "service_group_tree": { "service_groups": { "New_sg1": { "attributes": { "SystemList":{ "infdevlnx071": 1 }, "Parallel":{ "global": 1 } }, "resources": { "new_res1_1": { "attributes": { "Type": { "global": "DiskGroup" }, "Enabled":{ "global": 1 } } }, "new_res1_2": { "attributes": { "Type": { "global": "DiskGroup" }, "Enabled":{ "global": 1 } } }, "new_res1_3": { "attributes": { "Type": { </pre>	

Table 6-3 servicegroup_operations module keywords (continued)

Name	Description	Mandatory / Optional
	<pre> "global": "DiskGroup" }, "Enabled":{ "global": 0 } } }, "resource_dependencies": [{ "parent":"new_res1_1", "child_list": ["new_res1_2","new_res1_3"] }] }, "New_sg2": { "attributes": { "SystemList": { "infdevlnx071": 1 }, "AutoStartList" : { "global": ["infdevlnx071"] }, "Enabled":{ "infdevlnx071": 1, "infdevln0x70": 1 } } }, "resources": { "new_res2_1": { "attributes": { "Type": { "global": "DiskGroup" }, "Enabled": { "global": 1 } } } }, "new_res2_2": { "attributes": { "Type": { </pre>	

Table 6-3 servicegroup_operations module keywords (continued)

Name	Description	Mandatory / Optional
	<pre> "global": "DiskGroup" }, "Enabled":{ "global": 1 } } }, "files6": { "attributes": { "Type": { "global": "FileOnOff" }, "Enabled": { "global": 1 }, "PathName":{ "global": "/tmp/file6" } } } }, "resource_dependencies": [{ "parent":"new_res2_2", "child_list": ["new_res2_1"] }] } }, "servicegroup_dependencies": [{"New_sg2", "New_sg1", "online", "local", "soft"}] } </pre>	

Table 6-3 servicegroup_operations module keywords (continued)

Name	Description	Mandatory / Optional
seednode:	<p>Select any node from the cluster that will be used to run commands related to the service group operations of that cluster.</p> <p>The seednode value can be set to the FQDN, IP address, or short hostname.</p> <p>Examples:</p> <pre>seednode: objstorer820-1-vm17.veritas.com seednode: xx.xxx.xxx.xxx seednode: objstorer820-1-vm17</pre>	Mandatory
factors:	<p>Enter the following value to collect system-related data from the cluster servers.</p> <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}"</pre>	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. Before using vxvm_dgvolfs module, DiskInit module should be executed for disk initialization.

Use the following keywords while referencing the module in your playbook:

Table 6-4 Vxvm_dgvolfs module keywords

Name	Description	Mandatory / Optional
state:	<p>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.</p>	Mandatory

Table 6-4 Vxvm_dgvolfs module keywords (continued)

Name	Description	Mandatory / Optional
fss:	<p>Specifies whether the disk is part of a Flexible Storage Sharing (FSS) environment.</p> <p>The value for this keyword can be either 1 or 0. By default, this value is set to 1.</p> <p>Note: If fss is set to 1, the <code>volinfo</code> list in the sample playbook can contain additional <code>vxassist</code> command options to describe volume information. For example: <code>volinfo: [[testvol1, 750m, 'layout=mirror'], [testvol2, 800m, 'layout=mirror']]</code></p>	Optional
dgtype:	<p>Enter shared if you want to configure a shared type of disk group or leave the keyword empty.</p>	Optional
disks:	<p>Specifies the list of disks that you want to add in the disk group. Use the following format to enter the disk names:</p> <p>Syntax:</p> <pre>[<disk_name_1>, <disk_name_2> ,...<disk_name_n>]</pre> <p><code><disk_name_1>,<disk_name_2>,...<disk_name_n></code> are the names of the disks that you want to add to the disk group.</p> <p>Example:</p> <pre>[disk1,disk2,disk3]</pre> <p>Alternatively, you can provide input as below which will fetch the disk list provided to DiskInit module.</p> <pre>disks: "{{ disk_info['disks'] }}"</pre> <p>Note: If diskgroup already exists then no need to provide list of disks.</p>	Mandatory

Table 6-4 Vxvm_dgvolfs module keywords (continued)

Name	Description	Mandatory / Optional
taginfo:	<p>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.</p> <p>Syntax:</p> <pre>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></pre> <ul style="list-style-type: none"> ■ <i><tag_name_1></i>, <i><tag_name_2></i>, ...<i><tag_name_n></i> are the tag names. ■ <i><tag_value_1></i>, <i><tag_value_2></i>, ...<i><tag_value_n></i> are the tag values that you want to associate with each of the tag names. ■ <i><disk_name_1></i>, <i><disk_name_2></i>, ...<i><disk_name_n></i> are the names of the disks to which you want to apply the tag. <p>Example:</p> <pre>taginfo: - - disk_use - data - - virtio0_3 - virtio0_1 - - disk_size - large - - virtio0_3 - virtio0_2</pre>	Optional

Table 6-4 Vxvm_dgvolfs module keywords (continued)

Name	Description	Mandatory / Optional
volinfo:	<p>Specify the name and size of the volumes that you want to create.</p> <p>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.</p> <p>Also, if you want to unformat the volume, specify “raw” along with volume name, size and tag. If no tag, then empty string needs to be specified.</p> <p>Syntax:</p> <pre>volinfo: - - <volume_name_1> - <volume_size_1> - disktag:<tag_name_1>=<tag_value_1> - raw - - <volume_name_2> - <volume_size_2></pre> <p>Example:</p> <pre>volinfo: - - vol2 - 2g - disktag:disk_use=data - - srlvoll - 5g - '' - raw</pre>	Optional
seednode:	<p>Select any node from the cluster that will be used to run commands related to the operations of that cluster.</p> <p>Ensure that you enter the host name as provided in the in the /etc/ansible/hosts file.</p>	Mandatory
facters:	<p>Enter the following value to collect system-related data from the cluster servers.</p> <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}"</pre>	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 6-5 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
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. <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}"</pre>	Mandatory

resize_volume

The resize volume module is used to resize the VxVM volume in a simplified way. The supported operations are growto, growby, shrinkto and shrinkby which can be performed on a VxVM volume.

Table 6-6 `resize_volume` module keywords

Name	Description	Mandatory / Optional
State	Specifies what state the volume should be after the task is completed. The value for this keyword can be either present or absent.	Mandatory
Seednode	Specifies the node on which the task needs to be performed.	Mandatory
Operation	Specifies the operation to be performed on the volume. The value can be any of these - growto, growby, shrinkto and shrinkby	Mandatory
Size	Specifies the desired size for the operation to be performed. The unit can be k, m or g. Eg. To shrink the volume by 200m, size: 200m	
Mountpoint	Specifies the mountpoint of the volume.	Optional if dgname and volname are provided
Dgname	Specifies the name of the disk group where the volume belongs.	Optional if mountpoint is provided
Volname	Specifies the name of the volume.	Optional if mountpoint is provided
Facters	Enter the following value to collect system-related data from the cluster servers. <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}"</pre>	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 6-7 vvrresource module keywords

Name	Description	Mandatory / Optional
state	<p>Specifies what state the package should be in after the task is completed.</p> <p>Set this attribute to present to configure VVR or CVR.</p> <p>absent to unconfigure VVR or CVR</p>	Mandatory
delete_srlvol	<p>Specifies if srlvol associated with VVR is to be deleted. Value should be 1 to delete srlvol.</p>	Optional
deleteprimaryonly	<p>Specifies if only primary site of VVR is to be unconfigured. Value should be 1 to unconfigure primary site.</p> <p>First primary site needs to be unconfigured (using deleteprimaryonly) before unconfiguring secondary site.</p>	Optional
deletesecondaryonly	<p>Specifies if only secondary site of VVR is to be unconfigured. Value should be 1 to unconfigure secondary site.</p> <p>First primary site needs to be unconfigured (using deleteprimaryonly) before unconfiguring secondary site.</p>	Optional
dgroupName	<p>Name for the disk group where the Replicated Volume Group (RVG) is created.</p>	Mandatory
datavolname	<p>Name of the data volume of the VVR setup.</p>	Mandatory
srlvolname	<p>Name of the Storage Replicator Log (SRL) volume for the VVR setup.</p>	Mandatory
rvgname	<p>Name of the Replicated Volume Group (RVG) to be created</p>	Mandatory
seednode	<p>Specify any node from the cluster that will be used to run commands related to the operations of that cluster. Seednode can be set as FQDN, IP address, and short hostname. It is recommended that the seednode provided should be the master node.</p> <p>Example:</p> <pre>seednode: xx.xxx.xxx.xxx seednode: objstorer820-1-vm17.veritas.com seednode: objstorer820-1-vm17</pre>	Mandatory
primaryvip	<p>Specify a virtual IP for setting up VVR on the primary site.</p>	Mandatory

Table 6-7 vvrresource module keywords (continued)

Name	Description	Mandatory / Optional
secondaryvip	Specify a virtual IP for setting up VVR on the secondary site.	Mandatory
createprimaryonly	Specifies if primary site of VVR is to be created. Value should be 1 to create primary site.	Optional
createsecondaryonly	Specifies if secondary site of VVR is to be created. Value should be 1 to create secondary site.	Optional
start_replication	Specifies if VVR replication is to be started. Value should be 1 to start replication.	Optional

Table 6-7 vvrresource module keywords (continued)

Name	Description	Mandatory / Optional
agentinfo	<p>Specify the information required to create the VVR resource under VCS for site and cluster failover.</p> <p>Skip this attribute if you are not creating a VCS resource for site and cluster failover.</p> <p>Example:</p> <pre>agentinfo: primary: clusterlist: - Cluster1 - Cluster2 networkhosts: - xx.xxx.xxx.x nic: eth0 netmask: 255.255.252.0 secretkey: "xx:xx:xx" mountpoint: /mnt_test/primary secondary: nic: ens256 netmask: 255.255.252.0 secretkey: "xx:xx:xx" mountpoint: /mnt_secondary clusterlist: - Cluster1 - Cluster2 networkhosts: - xx.xxx.xxx.x</pre> <p>Secretkey is Azure specific and mandatory for azure. It specifies the Azure encrypted secret key for Azure cloud platform needed for AzureAuth and AzureIP resource. Provided secret key should be encrypted with respect to the primary and secondary site respectively.</p> <p>To plumb the VVR VIP, nic and netmask details for primary and/or secondary site should be provided.</p>	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

Table 6-7 vvrresource module keywords (continued)

Name	Description	Mandatory / Optional
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. For state: absent, specify its name to delete RVG service group and the resources within it	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. For state: absent, specify its name to delete MOUNT service group and the resources within it	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 6-7 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
azureip_res_name	Specify a name for the AZUREIP resource for VVR. 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
azureauth_res_name	Specify a name for the AZUREAUTH resource for VVR. 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
iptype	Specifies the IP type - privateip or overlayip for AWSIP resource.	Mandatory for AWS only
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. For state: absent, specify its name to delete LOGOWNER service group and the resources within it	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

Table 6-7 vvrresource module keywords (continued)

Name	Description	Mandatory / 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
azureip_logowner_res_name	Specify a name for the AZUREIP LOGOWNER resource. A resource with the Optional user-defined name is created if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
azureauth_logowner_res_name	Specify a name for the AZUREAUTH LOGOWNER resource. A resource with the Optional user-defined name is created if it does not already exist. If this attribute is skipped, a default resource name is used.	Optional
subscriptionid	Specifies the azure subscriptionid for Azure cloud platform needed for AzureAuth and AzureIP resource Example: subscriptionid: "xxx"	Mandatory for Azure
clientid	Specifies the azure clientid for Azure cloud platform needed for AzureAuth and AzureIP resource Example: clientid: "xxx"	Mandatory for Azure
tenantid	Specifies the azure tenantid for Azure cloud platform needed for AzureAuth and AzureIP resource Example: tenantid: "xxx"	Mandatory for Azure

Table 6-7 vvrresource module keywords (continued)

Name	Description	Mandatory / Optional
<p>mountpoint_info</p>	<p>Specify the information about mountpoints.</p> <p>The mountpoints consists of:</p> <ol style="list-style-type: none"> 1. volname: name of the volume 2. mountname: mount location of the volume 3. mount_res_name: resource name of the mount 4. requires_res: list of required VCS resources on which current mount depends upon. <p>Example:</p> <pre>mountpoint_info: - volname: testvoll mountname: /mnt/primary mount_res_name: custom_mount_new requires_res: [''] - volname: testvol2 mountname: /mnt/primary/2 mount_res_name: custom_mount_new2pri requires_res: ['custom_mount_new'] - volname: testvol3 mountname: /mnt/primary/3 mount_res_name: custom_mount_new3pri requires_res: ['custom_mount_new']</pre> <p>In the above example, mount '/mnt/primary/2' and '/mnt/primary/3' depend upon 'custom_mount_new' resource.</p>	<p>Optional</p>
<p>facters</p>	<p>This attribute is used to collect system-related data from the cluster servers. Provide the following value to the attribute:</p> <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}"</pre>	<p>Mandatory</p>

add_node

The add_node module is used to add or remove nodes from a cluster. Run the following commands on the SEEDNODE of the cluster:

- /opt/VRTSvcs/bin/haconf -makerw
- /opt/VRTSvcs/bin/haclus -modify HacliUserLevel COMMANDROOT

■ `/opt/VRTSvcs/bin/haconf -dump -makero`

Use the following keywords while referencing the module in your playbook:

Table 6-8 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 <code>/etc/ansible/hosts</code> 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

VipConfig

The VipConfig module configures the virtual IP (VIP) on the mentioned host in the playbook. Using this module, with one playbook run, one VIP can be configured on the mentioned host.

Use the following keywords while referencing the module in your playbook:

Table 6-9

Name	Description	Mandatory / Optional
nic	Specifies the NIC on which the VIP is to be applied	Mandatory
vip	Specifies the VIP to be configured on the host	Mandatory
netmask	Specifies the netmask of the VIP to be configured	Mandatory

Table 6-9 (continued)

Name	Description	Mandatory / Optional
facters	<p>Enter the following value to collect system-related data from the cluster servers.</p> <pre>"{{ ansible_play_hosts_all map('extract', hostvars,['facts', 'infoscale_facts']) select() list }}"</pre>	Mandatory

SecureCluster

The SecureCluster module provides secure cluster configuration of a Veritas InfoScale cluster in a single playbook run. Run the following commands on the SEEDNODE of the cluster:

- /opt/VRTSvcs/bin/haconf -makerw
- /opt/VRTSvcs/bin/haclus -modify HacliUserLevel COMMANDROOT
- /opt/VRTSvcs/bin/haconf -dump -makero

Use the following keywords while referencing the module in your playbook:

Table 6-10

Name	Description	Mandatory / Optional
cluster_id	Specify the ID of the cluster	Optional
cluster_name	Specify the name of the cluster.	Optional
seednode	<p>Select any node from the cluster that will be used to run commands related to the operations of that cluster.</p> <p>Seednode can be set as FQDN, IP Address, and short hostname. Examples:</p> <pre>seednode: objstorer820-1-vm17.veritas.com seednode: xx.xxx.xxx.xxx seednode: objstorer820-1-vm17</pre>	Mandatory

Table 6-10 (continued)

Name	Description	Mandatory / Optional
state	Specifies what state the nodes should be after the task is completed. The value for this keyword can be either present or absent or present indicates 'Configure Secure Cluster' absent indicates 'Unconfigure Secure Cluster'	Mandatory
reconfig	Specifies if the secure cluster configuration should be reconfigured. Value should be 1 to reconfigure the secure cluster.	Optional
fips	Specifies the mode in which cluster needs to be secured. Fips:0 Secure with fips-disabled mode. Fips:1 Secure with fips-enabled mode.	Mandatory
read_access	Specifies read_access. If 0 then grant read access to every usergroup and if 1 then grant access to either specified usergroups or no groups.	Mandatory
usergroups	Specify the usergroups with read access	Optional
factors	Enter the following value to collect system-related data from the cluster servers. <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}"</pre>	Mandatory

gco_config

The gco_config module configures the Veritas Infoscale Global Cluster Option on primary and secondary site clusters both in one playbook run. Before configuring GCO, it is required both the primary and secondary site clusters are secured.

Use the following keywords while referencing the module in your playbook:

Table 6-11

Name	Description	Mandatory / Optional
primary_seednode	Specifies one of the hosts in the primary site cluster.	Mandatory

Table 6-11 (continued)

Name	Description	Mandatory / Optional
secondary_seednode	Specifies one of the hosts in the secondary site cluster.	Mandatory
state	Specifies what state the cluster should be in after the task is completed. The value for this keyword can be either present or absent.	Mandatory
primary	<p>Specifies the information required to configure GCO on the primary site.</p> <p>Example:</p> <pre>primary: prefix: '' nic: {'<hostname>': 'ens192'} vip: <xx.xxx.xxx.xx> ipver: 4 netmask: '255.255.240.0' secretkey: "xx:xx:xx"</pre> <p>Secretkey is Azure specific. It specifies the Azure encrypted secret key for Azure cloud platform needed for AzureAuth and AzureIP resources. Provided secret key should be encrypted with respect to the primary site. It is mandatory for Azure.</p>	Mandatory
prefix	Specifies IP prefix if ipver is 6.	Optional
nic	<p>Specifies the list of NICs to be used to configure GCO. This keyword accepts a dictionary where the key is hostname and the value is the associated NIC on that host in the cluster. Example:</p> <pre>nic: {'<hostname1>': 'ens192', '<hostname2>': 'ens224'}</pre>	Mandatory
vip	Specifies the static IP to be used for GCO configuration on primary site. This Virtual IP can be in ipv4 or ipv6 format. If it is in ipv6 format then value for 'prefix' is required.	Mandatory
ipver	Specifies the IP version - ipv4 or ipv6. This keyword can have value as 4 or 6.	Mandatory
netmask	Specifies the netmask to be used for GCO configuration.	Mandatory
iptype	Specifies the IP type - privateip or overlayip for AWSIP resource.	Mandatory for AWS only

Table 6-11 (continued)

Name	Description	Mandatory / Optional
secondary	<p>Specifies the information required to configure GCO on the secondary site. For Example,</p> <pre>secondary: prefix: '' nic: {'<hostname>': 'ens192'} vip: <xx.xxx.xxx.xx> ipver: 4 netmask: '255.255.240.0' secretkey: "xx:xx:xx"</pre> <p>Secretkey is Azure specific. It specifies the Azure encrypted secret key for Azure cloud platform needed for AzureAuth and AzureIP resources. Provided secret key should be encrypted with respect to the secondary site. It is mandatory for Azure.</p> <p>The above keywords - prefix, nic, vip, ipver and netmask - have same significance as those for primary site and should be provided here with respect to secondary site.</p>	Mandatory
facters	<p>Enter the following value to collect system-related data from the cluster servers.</p> <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}"</pre>	Mandatory
subscriptionid	<p>Specifies the Azure subscriptionid for Azure cloud platform needed for AzureAuth and AzureIP resource Example: subscriptionid: "xxx"</p>	Mandatory for Azure
clientid	<p>Specifies the Azure clientid for Azure cloud platform needed for AzureAuth and AzureIP resource Example: clientid: "xxx"</p>	Mandatory for Azure
tenantid	<p>Specifies the Azure tenantid for Azure cloud platform needed for AzureAuth and AzureIP resource Example: tenantid: "xxx"</p>	Mandatory for Azure

I/O Fencing

Table 6-12 I/O fencing module keywords

Name	Description	Mandatory / Optional
state:	Specifies what state the nodes should be after the task is completed. The value for this keyword can be either present or absent . where: present indicates 'Configure Fencing' absent indicates 'Delete Fencing configuration'	Mandatory
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 <code>/etc/ansible/hosts</code> file.	Mandatory
vxfenmode	Mode of fencing to be configured	Mandatory the choices are: <ul style="list-style-type: none"> ■ disabled ■ majority ■ cps ■ disks
disks	List of the disks to be used for the fencing. Number of the disks must be 3 or more in odd numbers	Optional Needed only if 'disks' chosen as fencing mode
no_coord_disk	Needs to be set if coordinator disks are not to be used. Allowed values are 0 or 1	Optional Needed only if 'cps' chosen as fencing mode

Table 6-12 I/O fencing module keywords (continued)

Name	Description	Mandatory / Optional
vx fendg	Disk group name for fencing	Optional Needed only if 'disks' chosen as fencing mode For 'cps' based fencing while using disk coord, vx fendg can be utilized if disk group already exists.
cps	List of CP server IP:port	Optional Needed only if 'cps' fencing is chosen
ntp_server	NTP server for time synchronization	Optional
default_https_port	Default HTTPs port for CP servers.	Optional
max_async_sec	Time in seconds for checking maximum async date/time on server	Optional
cpagent	To create coordination point resources groups under VCS. Allowed values are 0 or 1. By default 1 will be set.	Optional

RestServerConfig

The RestServerConfig module configures the REST server on the given host in a secure cluster. Ensure that the cluster is secured before configuring REST server.

Use the following keywords while referencing this module in your playbook:

Table 6-13 RestServerConfig module keywords

Name	Description	Mandatory / Optional
state	Specifies what state the cluster should be in after the task is completed. The value for this keyword can be either: present - configure REST server OR absent - unconfigure REST server	Mandatory

Table 6-13 RestServerConfig module keywords (continued)

Name	Description	Mandatory / Optional
REST_server_ip	Specifies the IP of REST server to be configured. This should be a free IP which can be used.	Mandatory
REST_server_netmask	Specifies the netmask to be used to configure REST server.	Mandatory
REST_server_nic	Specifies the NIC to be used for the REST server.	Mandatory
REST_server_port	Specifies the port for the REST server.	Mandatory
rest_config	<p>Specifies which configuration option should be used to configure the REST server. Following are the options:</p> <p>enabled_ldap - To use LDAP server details if it is configured. In this case, REST_server_ldap_domain, REST_server_ldap_ip and REST_server_ldap_username should be specified in the playbook</p> <p>enabled_tpcacert - To use TP CA certificates details. In this case, REST_server_key, REST_server_cert_file and REST_server_cacert_file should be specified in the playbook</p> <p>enabled_tpcacert_ldap - To use TP CA certificates details. In this case, REST_server_key, REST_server_cert_file, REST_server_cacert_file, REST_server_ldap_domain, REST_server_ldap_ip and REST_server_ldap_username should be specified in the playbook</p> <p>default - To configure REST server with Veritas CA certificate, without LDAP</p>	Optional
reconfig	Specifies if the REST server configuration should be reconfigured. The value should be 1 to reconfigure REST server.	Optional
factors	Enter the following value to collect system-related data from the cluster servers. "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}"	Mandatory
REST_server_ldap_domain	Specifies the domain of LDAP server to be used.	Optional
REST_server_ldap_ip	Specifies the IP of LDAP server to be used.	Optional
REST_server_username	Specifies the username of LDAP server to be used.	Optional
REST_server_key	Specifies server key file, required for third party CA certificate option	Optional
REST_server_cert_file	Specifies server certificate file, required for third party CA certificate option	Optional

Table 6-13 RestServerConfig module keywords (continued)

Name	Description	Mandatory / Optional
REST_server_cacert_file	Specifies server CA certificate file, required for third party CA certificate option	Optional
REST_server_passphrase	Specifies server passphrase if available, required for third party CA certificate option	Optional

CPServerConfig

The CPServerConfig module configures the CP Server on the single node VCS cluster.

Use the following keywords while referencing the CPServerConfig module in your playbook:

Table 6-14 CPServerConfig module keywords

Name	Description	Mandatory / Optional
cps_https_vip	Specifies the IP of CP server to be configured. This should be a free IP which can be used.	Mandatory
cps_netmask	Specifies the netmask to be used to configure CP server.	Mandatory
cps_nic_list	Specifies the nic to be used to configure CP server.	Mandatory
cps_https_ports	Specifies the port for the CP server.	Mandatory
cps_delete_database	Specifies whether the database of the CP server should be deleted while removing the configuration. The value should be 1 to delete the database.	Optional
cps_delete_config_logs	Specifies whether the configuration and the log files of the CP server should be deleted while removing the configuration. The value should be 1 to delete the configuration and log files.	Optional
reconfig	Specifies whether the CP Server configuration should be reconfigured. The value should be 1 to reconfigure CP Server.	Optional
state	Specifies what state the cluster should be in after the task is completed. The value for this keyword can be either: present: configure CP Server OR absent: unconfigure CP Server	Mandatory

Table 6-14 CServerConfig module keywords (continued)

Name	Description	Mandatory / Optional
facters	Enter the following value to collect system-related data from the cluster servers. "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}"	Mandatory
vip	Specifies the IP of CP server to be configured. This should be a free IP which can be used.	Mandatory
netmask	Specifies the netmask to be used to configure CP server.	Mandatory
port	Specifies the port for the CP server of specific network.	Mandatory
hosts	Specifies the host for the specific network	Mandatory
nic	Specifies the nic for the specific network and the host.	Mandatory
cps_diskgroup	Specifies the diskgroup for the CP server.	Optional
cps_volume	Specifies the volume for the CP server.	Optional
cps_newdgdisk	Specifies the name of new diskgroup to be created for the CP server configuration.	Optional
cps_newvol_volsize	Specifies the size of new volume to be created for the CP server configuration.	Optional
cps_delete_database	Specifies whether the database of the CP server should be deleted while removing the configuration. The value should be 1 to delete the database.	Optional Optional
cps_delete_config_logs	Specifies whether the config and logs file of the CP server should be deleted during the unconfiguration. The value should be 1 to delete the config and log files.	

EO compliant logging

The EO compliant logging module enables or disables the logging of all components of the cluster.

Use the following keywords while referencing the EO compliant logging module in your playbook:

Table 6-15 EO logging module keywords

Name	Description	Mandatory / Optional
state	Specifies whether EO logging should be enabled or disabled. Value should be "present" to enable and "absent" to disable.	Mandatory
eocompliantlogging_perm	Specifies the custom log file permission to be used. Set this parameter to one of the following values: 0, 1, 2, 3, 10, 11, 12, or 13. If you do not specify a value for this parameter, it is considered to be 0 (zero) by default. For details, refer to the <i>InfoScale Cluster Server Administrator's Guide - Linux</i> or <i>InfoScale Storage Foundation Administrator's Guide - Linux</i> .	Optional
facters	Enter the following value to collect system-related data from the cluster servers. "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}"	Mandatory

Set Tunables

The SetTunable module sets the llt, vxvm and vxfs related tunables for the cluster.

Use the following keywords while referencing the SetTunable module in your playbook:

Table 6-16 Set Tunable module keywords

Name	Description	Mandatory / Optional
llt	<p>Specifies the information required to set LLT related Tunables.</p> <p>Example:</p> <pre> - systems: ['dl380g9-127-vm3'] tunables: - "set-flow window: 520" - "set-flow highwater: 12000" - systems: ['all'] tunables: - "set-flow window: 521" - "set-flow highwater: 12001" </pre> <p>Above keywords: 'systems' is a list of nodes of the cluster or ['all'] if all the nodes of the cluster need to be considered. 'tunables' is list of tunables (string)</p>	Optional
vm	<p>Specifies the information required to set VxVM related Tunables.</p> <p>Example</p> <pre> - systems: ['all'] tunables: volraid_rsrtransmax: 1 voliot_max_open: 32 </pre> <p>'systems' is a list of nodes of the cluster or ['all'] if all the nodes of the cluster need to be considered. 'tunables' is a dict with key as tunable_name and value as tunable_value. Also, clusterwide tunable will be automatically handled.</p>	Optional

Table 6-16 Set Tunable module keywords (continued)

Name	Description	Mandatory / Optional
fs	<p>Specifies the information required to set VxFS related Tunables.</p> <p>Example</p> <pre> - systems: ['dl380g9-127-vm3'] tunables: - /mnt1: eo_logging_enable: 1 max_retention_time: 0 - global: eo_logging_enable: 0 - systems: ['all'] tunables: - /mnt1: eo_logging_enable: 1 max_retention_time: 0 - global: eo_logging_enable: 0 </pre> <p>Above keywords:</p> <p>'systems' is a list of nodes of the cluster or ['all'] if all the nodes of the cluster need to be considered. 'tunables' is a list of mount points and global (tunable will be set to all the mounts) which further are the dicts with key as tunable_name and value as tunable_value.</p>	Optional
facters	<p>Enter the following value to collect system-related data from the cluster servers.</p> <pre> "{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}" </pre>	Mandatory

Set routes and rules for subnets across AZ on AWS

The RouteConfig module sets the routes and rules for subnets across AZ on AWS. On AWS EC2 instance, disable the following nm-cloud setup service which is a pre-requisite for the module:

```

systemctl stop nm-cloud-setup.timer
systemctl stop nm-cloud-setup.service
systemctl disable nm-cloud-setup.timer
                    
```

```
systemctl disable nm-cloud-setup.service
```

Use the following keywords while referencing the RouteConfig module in your playbook:

Table 6-17 RouteConfig module keywords

Name	Description	Mandatory / Optional
system	Specify the system on which routes and rules need to be set. It can be FQDN, IP Address, short hostname. <pre>system: objstorer820-1-vm17.veritas.com system: xx.xxx.xxx.xxx system: objstorer820-1-vm17</pre>	Mandatory
nic	Specify the NIC of the system that will be used as heartbeat link	Mandatory
ip	Specify the private ip of the system	Mandatory
factors	Enter the following value to collect system-related data from the cluster servers. <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts','infoscale_facts']) select() list }}"</pre>	Mandatory

DiskInit

DiskInit module is used for initializing disks present on the node. You can use the following keywords while referencing the DiskInit module in your playbook:

Table 6-18 DiskInit module keywords

Name	Description	Mandatory / Optional
fss	Specifies whether the disk is a part of a Flexible Storage Sharing (FSS) optional environment. The value for this keyword can be either True or False. By default, this value is set to True.	Optional

Table 6-18 DiskInit module keywords (*continued*)

Name	Description	Mandatory / Optional
disks	<p>Enter the following value for initializing disk to fetch the disks list provided in DiskInit module:</p> <p>Syntax: [<code><disk_name_1></code>, <code><disk_name_2></code>, , ...<code><disk_name_n></code>] <code><disk_name_1></code>, <code><disk_name_2></code>, ...<code><disk_name_n></code> are the names of the disks that you want initialise. Example: [<code>disk1</code>, <code>disk2</code>, <code>disk3</code>]</p> <p>If disks parameter is not specified, then all the uninitialized disks present on the system will be initialized using this module</p>	Optional

Add/Remove disks from Diskgroup

The Adddisks module is used to add or remove disks from diskgroup. Before using Adddisks module, DiskInit module should be executed for disk initialization while adding disks to diskgroup. DiskInit module is not needed when removing disks from diskgroup.

Use the following keywords while referencing the Adddisks module in your playbook:

Table 6-19 Adddisks module keywords

Name	Description	Mandatory / Optional
state	<p>Specifies what state the nodes should be in after the task is completed. The value for this keyword can be either present or absent.</p> <p>Use present to add disks to diskgroup and use absent to delete disks from diskgroup.</p>	Mandatory
dgname	Name of the disk group to which disk must be added or removed.	Mandatory

Table 6-19 Adddisks module keywords (continued)

Name	Description	Mandatory / Optional
disks	<p>To add disks:</p> <p>Enter the following value for adding disks in DG to fetch the disks list provided in Disklnit module:</p> <pre>"{{disk_info['disks'] }}"</pre> <p>To remove disks:</p> <p>Specifies the list of disks that you want to remove in the disk group. Use the following format to enter the disk names:</p> <p>Syntax:</p> <pre>[<disk_name_1>, <disk_name_2>, , ...<disk_name_n>] <disk_name_1>, <disk_name_2>, ...<disk_name_n></pre> <p>are the names of the disks that you want to add to the disk group.</p> <p>Example: [disk1, disk2, disk3]</p>	Mandatory
factors	<p>Enter the following value to collect system-related data from the cluster servers.</p> <pre>"{{ ansible_play_hosts_all map('extract', hostvars, ['facts', 'infoscale_facts']) select() list }}"</pre>	Mandatory

Change Replication State

The `change_replication_state` module is used to change replication state of Volume Replicator in InfoScale. It can be start/stop/pause/resume of VVR replication.

Use the following keywords while referencing the module in your playbook:

Table 6-20 Change Replication State module keywords

Name	Description	Mandatory / Optional
newstate	Specifies the required state of VVR replication. It can be start/stop/pause/resume.	Mandatory

Table 6-20 Change Replication State module keywords (continued)

Name	Description	Mandatory / Optional
rvg_info	<p>Specifies the list of dict containing name of dg along with list of rvgs. Syntax:</p> <pre>[{'dg_name': <dg_name>, 'rvg_list': <rvg_list>, 'seednode': <system_name>}, {'dg_name': <dg_name>, 'rvg_list': <rvg_list>}]</pre> <p>Example:</p> <pre>[{'dg_name': 'testdg1', 'rvg_list': ['rvg_vvrtest1', 'rvg_vvrtest2']}, {'dg_name': 'testdg2', 'rvg_list': ['rvg_vvrtest1', 'rvg_vvrtest2'], 'seednode': 'r750xs-32-vm11'}]</pre> <p>Here, seednode is optional parameter which can be any node of the cluster that can be used to run commands related to the vvr operations.</p>	Mandatory
secondaryvip	Specifies the virtual VIP used for setting secondary site.	Mandatory

InfoScale Ansible end to end workflow / solution

- [Chapter 7. VVR solution](#)
- [Chapter 8. Solution templates](#)

VVR solution

This chapter includes the following topics:

- [Configuring end to end InfoScale VVR deployment using Ansible playbooks](#)
- [Ansible playbook for configuring end to end VVR deployment in InfoScale](#)
- [List of pre-defined keywords](#)
- [Input file for VVR solution \(vars_file - main.json\)](#)
- [Out file for VVR solution](#)

Configuring end to end InfoScale VVR deployment using Ansible playbooks

Veritas Volume Replicator (VVR) is data-replication software intended to support an effective disaster recovery plan. VVR enables you to maintain a consistent copy of application data at one or more remote locations.

The Ansible playbook now offers a new one-click approach for configuring end-to-end VVR GCO deployment across two sites using single / multiple RVGs with local or shared storage. This improves user experience while reducing overall deployment operations complexity.

You need to use `infoscale_workflow.yml` and `vvr_solution` role present in `veritas-infoscale` ansible collection. Input needs to be provided in `vars/main.json` file of `vvr_solution` role (Sample input file are present at `vars/main.json` location of `vvr_solution` role).

You should enter all the input details for each cluster, RVGs, GCO configuration details etc in the `vars/main.json` file. Once the input is provided using InfoScale Ansible modules end to end workflow is generated and executed. The Ansible deployment takes care of the following items:

1. Install & configure InfoScale stack
2. Secure cluster configuration
3. GCO configuration
4. Storage discovery
5. Volume & FS creation
6. RVG creation
7. Setting up replication

The end-to-end workflow solution can be used on the physical and virtual machine located on premise.

Ansible playbook for configuring end to end VVR deployment in InfoScale

Table 7-1 Infoscale_workflow

Operations	Required roles	Sample Playbook
Generating solutions (set of playbooks) and execute	vvr_solution	Sample playbook - YAML Infoscale_workflow.yml <pre> --- - hosts: localhost gather_facts: false any_errors_fatal: true roles: - role: vvr_solution execute: 1 vars_file: /root/.ansible/collections/ ansible_collections/veritas/infoscale/roles/ vvr_solution/vars/main.json base_working_dir: /root/.ansible/collections/ ansible_collections/veritas/infoscale/roles/ vvr_solution/files playbooks_dir: /root/.ansible/collections/ ansible_collections/veritas/infoscale/playbooks </pre>

Table 7-1 Infoscale_workflow (continued)

Operations	Required roles	Sample Playbook
		<p>Sample playbook - JSON format</p> <pre>[{ "hosts": "localhost", "gather_facts": false, "any_errors_fatal": true, "roles": [{ "role": "vvr_solution", "execute": 1, "vars_file": "/root/.ansible/collections/ ansible_collections/veritas/infoscale/roles/ vvr_solution/vars/main.json", "base_working_dir": "/root/.ansible/collections/ ansible_collections/veritas/infoscale/roles/ vvr_solution/files", "playbooks_dir": "/root/.ansible/collections ansible_collections/veritas/infoscale/playbook" }] }]</pre>

List of pre-defined keywords

Refer to the following tables for a list of the keywords that are used in the `infoscale_workflow.yml` playbook.

Table 7-2 `infoscale_workflow.yml` playbook.

Name	Description	Mandatory/Optional
<code>vars_file</code>	<p>Specify the <code>vars</code> file location where all the input parameters provided</p> <p>Sample file location: <code>/root/.ansible/collections/ansible_collections/veritas/infoscale/roles/vvr_solution/vars/main.json</code></p> <p>Note: Please use absolute path</p>	Mandatory

Table 7-2 infoscale_workflow.yml playbook. (continued)

Name	Description	Mandatory/Optional
base_working_dir	<p>Specifies the location where you can dump the workflow solution playbooks. This location must exist before running the playbook.</p> <p>Sample directory location:/root/.ansible/collections/ansible_collections/veritas/infoscale/roles/vvr_solution/files</p> <p>Note: Please use absolute path</p>	Mandatory
playbooks_dir	<p>Specifies the location where all the Infoscale playbooks are present.</p> <p>Default playbooks location:/root/.ansible/collections/ansible_collections/veritas/infoscale/playbooks</p> <p>Note: Please use absolute path</p>	Mandatory
execute	<p>Specify either 0 or 1.</p> <p>0 Generates the solution workflow.</p> <p>1 (default value) Generates and executes the solution workflow.</p>	Optional

Input file for VVR solution (vars_file - main.json)

Sample 1 - main.json

```
# This is a sample for LLT using Ethernet and shared disk group
{
  "repository_name" : "repo_infoScale80",
  "repository_baseurl" : "http://xx.xxx.xxx.xx/Infoscale/8.0
                        /rhel8_x86_64/rpms",
  "clusters" : {
    "cluster1" : {
      "cluster_nodes" : ["objstorer820-1-vm17.veritas.com",
                        "objstorer820-1-vm18.veritas.com"],
      "public_link" : "eth0",
      "private_links" : "eth1, eth2",
      "product_version" : "8.0",
      "udp_link" : "",
      "product_component" : "SFCFSHA",
      "site_gco_ip" : "xx.xx.xx.xx",
      "site_gco_netmask" : "255.255.240.0",
      "cluster_gateway" : "xx.xx.xx.xx"
    },
  },
}
```

```
"cluster2"      : {
  "cluster_nodes" : ["objstorer820-1-vm19.veritas.com",
                    "objstorer820-1-vm20.veritas.com"],
  "public_link"   : "eth0",
  "private_links" : "eth1, eth2",
  "udp_link"      : "",
  "product_version" : "8.0",
  "product_component" : "SFCSHA",
  "site_gco_ip"    : "xx.xx.xx.xx",
  "site_gco_netmask" : "255.255.240.0",
  "cluster_gateway" : "xx.xx.xx.xx"
}

},

"diskgroup" : {
  "dg1" : {
    "clusters_list" : ["cluster1", "cluster2"],
    "storage_type" : "shared",
    "diskgroup_type" : "shared",
    "rvg_details" : {
      "rvg1" : {
        "volume_size" : ["5G", "10G"],
        "volume_layout" : ["stripe", "concat"],
        "volume_mount_point" : ["/mount_path1", "/mount_path2"],
        "srl_size" : ["3G", "concat"],
        "nic_name" : ["eth0", "eth0"],
        "rvg_ips" : ["xx.xx.xx.xx", "xx.xx.xx.xx"],
        "rvg_ips_netmask" : ["255.255.240.0", "255.255.240.0"]
      },
      "rvg2" : {
        "volume_size" : ["10G"],
        "volume_layout" : ["concat"],
        "volume_mount_point" : ["/mount_path3"],
        "srl_size" : ["4G", "concat"],
        "nic_name" : ["eth0", "eth0"],
        "rvg_ips" : ["xx.xx.xx.xx", "xx.xx.xx.xx"],
        "rvg_ips_netmask" : ["255.255.240.0", "255.255.240.0"]
      }
    }
  }
},

"dg2" : {
  "clusters_list" : ["cluster1", "cluster2"],
```



```
        ["eth2", "xx.xx.xx.xx/24", "50001"]
    ],
    "objstorere820-1-vm18.veritas.com" : [
        ["eth1", "xx.xx.xx.xx/24", "50000"],
        ["eth2", "xx.xx.xx.xx/24", "50001"]
    ]
},

"product_component" : "SFCSHA",
"site_gco_ip" : "xx.xx.xx.xx",
"site_gco_netmask" : "255.255.240.0",
"cluster_gateway" : "xx.xx.xx.xx"
},

"cluster2" : {
    "cluster_nodes" : ["objstorere820-1-vm19.veritas.com",
        "objstorere820-1-vm20.veritas.com"],
    "public_link" : "eth0",
    "private_links" : "eth1, eth2",
    "udp_link" : {
        "objstorere820-1-vm19.veritas.com" : [
            ["eth1", "xx.xx.xx.xx/24", "50000"],
            ["eth2", "xx.xx.xx.xx/24", "50001"]
        ],
        "objstorere820-1-vm20.veritas.com" : [
            ["eth1", "xx.xx.xx.xx/24", "50000"],
            ["eth2", "xx.xx.xx.xx/24", "50001"]
        ]
    }
},

"product_version" : "8.0",
"product_component" : "SFCSHA",
"site_gco_ip" : "xx.xx.xx.xx",
"site_gco_netmask" : "255.255.240.0",
"cluster_gateway" : "xx.xx.xx.xx"
}

},

"diskgroup" : {
    "dg1" : {
```

```
"clusters_list" : ["cluster1", "cluster2"],
"storage_type" : "fss",
"diskgroup_type" : "fss",
"rvg_details" : {
  "rvg1" : {
    "volume_size" : ["5G","10G"],
    "volume_layout" : ["stripe","concat"],
    "volume_mount_point" : ["/mount_path1","/mount_path2"],
    "srl_size" : ["3G","concat"],
    "nic_name" : ["eth0","eth0"],
    "rvg_ips" : ["xx.xx.xx.xx","xx.xx.xx.xx"],
    "rvg_ips_netmask" : ["255.255.240.0","255.255.240.0"]
  },

  "rvg2" : {
    "volume_size" : ["10G"],
    "volume_layout" : ["concat"],
    "volume_mount_point" : ["/mount_path3"],
    "srl_size" : ["4G","concat"],
    "nic_name" : ["eth0","eth0"],
    "rvg_ips" : ["xx.xx.xx.xx","xx.xx.xx.xx"],
    "rvg_ips_netmask" : ["255.255.240.0","255.255.240.0"]
  }
}

},

"dg2" : {
  "clusters_list" : ["cluster1", "cluster2"],
  "storage_type" : "fss",
  "diskgroup_type" : "fss",
  "rvg_details" : {
    "rvg1" : {
      "volume_size" : ["5G","10G"],
      "volume_layout" : ["stripe ncols=2","concat"],
      "volume_mount_point" : ["/mount_path1","/mount_path2"],
      "srl_size" : ["3G","concat"],
      "nic_name" : ["eth0","eth0"],
      "rvg_ips" : ["xx.xx.xx.xx","xx.xx.xx.xx"],
      "rvg_ips_netmask" : ["255.255.240.0","255.255.240.0"]
    }
  }
}
```

```

    },

    "rvg2" : {
        "volume_size"    : ["10G"],
        "volume_layout"  : ["concat"],
        "volume_mount_point" : ["/mount_path3"],
        "srl_size"       : ["4G","concat"],
        "nic_name"       : ["eth0","eth0"],
        "rvg_ips"        : ["xx.xx.xx.xx","xx.xx.xx.xx"],
        "rvg_ips_netmask" : ["255.255.240.0","255.255.240.0"]
    }

}

}

}

}
    
```

List of pre-defined keywords

Refer to the following table for a list of the keywords that are used in the vars/main.json variable file of vvr_solution role.

Table 7-3

Name	Description	Mandatory/Optional
repository_name	Name of the yum repository used to install InfoScale rpms. Example: repo-InfoScale80	Mandatory
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.

Table 7-3 (continued)

Name	Description	Mandatory/Optional
clusters	<p>Specifies Clusters related information. For every cluster, list the nodes, nics, product version, ips, netmask, gateway etc details. Example:</p> <pre> "clusters" : { "cluster1" : { "cluster_nodes" : ["objstorer820-1-vm17.veritas.com", "objstorer820-1-vm18.veritas.com"], "public_link" : "eth0", "private_links" : "eth1, eth2", "product_version" : "8.0", "udp_link" : "", "product_component" : "SFCEFSHA", "site_gco_ip" : "xx.xx.xx.xx", "site_gco_netmask" : "255.255.240.0", "cluster_gateway" : "xx.xx.xx.xx" } } </pre>	
cluster1	Specify name for the cluster that you want to deploy. Example: Cluster1	Optional
cluster_nodes	Specify the list of nodes on you want to deploy InfoScale and part of the same cluster	Mandatory
public_link	Lists the name of the NICs that the low priority heartbeat link uses on each of the nodes in the cluster.	Optional
private_links	Lists the name of the NICs that the heartbeat link uses on each of the nodes in the cluster.	Mandatory
product_version	Specifies the version of the product that you want to install or upgrade. Example: http://xx.xxx.xxx.xx/rpms/	Mandatory

Table 7-3 (continued)

Name	Description	Mandatory/Optional
udp_link	<p>This attribute is only required if you want to configure LLT over a UDP link (method must be set to udp). 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. Example:</p> <pre data-bbox="319 465 1041 755"> "udp_link" : { "objstorer820-1-vm17.veritas.com" : [["eth1", "xx.xx.xx.xx/24", "50000"], ["eth2", "xx.xx.xx.xx/24", "50001"]], "objstorer820-1-vm18.veritas.com" : [["eth1", "xx.xx.xx.xx/24", "50000"], ["eth2", "xx.xx.xx.xx/24", "50001"]] }, </pre> <p>Input is List = [nic1, ip, port]</p>	Optional
product_component	<p>Specify which components you want to configure in your product. Note that the product license acquired must support the required components. Example: SFCFSHA</p>	Optional
site_gco_ip	<p>Specifies the static IP to be used for GCO configuration on site. This Virtual IP can be in ipv4 or ipv6 format. If it is in ipv6 format then value for "prefix" is required.</p>	Mandatory
site_gco_netmask	<p>Specifies the netmask to be used for GCO configuration.</p>	Mandatory
cluster_gateway	<p>Specify the networkhosts information required to create the VVR resource under VCS for site and cluster failover.</p>	

Table 7-3 (continued)

Name	Description	Mandatory/Optional
diskgroup	<p>Specifies Diskgroups related information. For every Diskgroup, list the clusters, storage_type, diskgroup_type, and rvg details. Example:</p> <pre> "diskgroup" : { "dg1" : { "clusters_list" : ["cluster1", "cluster2"], "storage_type" : "shared", "diskgroup_type" : "shared", "rvg_details" : { "rvg1" : { "volume_size" : ["5G","10G"], "volume_layout" : ["stripe","concat"], "volume_mount_point" : ["/mount_path1","/mount_path2"], "srl_size" : ["3G","concat"], "nic_name" : ["eth0","eth0"], "rvg_ips" : ["xx.xx.xx.xx","xx.xx.xx.xx"], "rvg_ips_netmask" : ["255.255.240.0","255.255.240.0"] }, "rvg2" : { "volume_size" : ["10G"], "volume_layout" : ["concat"], "volume_mount_point" : ["/mount_path3"], "srl_size" : ["4G","concat"], "nic_name" : ["eth0","eth0"], "rvg_ips" : ["xx.xx.xx.xx","xx.xx.xx.xx"], "rvg_ips_netmask" : ["255.255.240.0","255.255.240.0"] } } }, }</pre>	Mandatory
clusters_list	Specify a list containing the name of clusters having a GCO setup for cluster failover.	Mandatory
storage_type	Specify the storage type for disks (shared, FSS) which needs to be considered for dg-vol creation.	Mandatory
diskgroup_type	Specify the diskgroup type - shared or fss	Mandatory
rvg_details	Specifies RVG related information. For every RVG, list the volumes size, layout, mountpoint, srl_size, rvg_ip, nic and so on.	Mandatory

Table 7-3 (continued)

Name	Description	Mandatory/Optional
volume_size	Specify a list of volumes size which you want to create as a part of give rvg.	Mandatory
volume_layout	Specify a list of volumes layout which you want to create as a part of give rvg.	Mandatory
volume_mount_point	Specify the path where the volume needs to be mounted.	Mandatory
srl_size	Size for the Storage Replicator Log (SRL) volume for the VVR setup for rvg.	Mandatory
nic_name	Specify the nic information required to create the VVR resource under VCS for site and cluster failover.	Mandatory
rvg_ips	Specify a list of virtual IP for setting up VVR on the primary and secondary site.	Mandatory
rvg_ips_netmask	Specify a List of netmask to be used for rvg ips provided for primary and secondary sites.	Mandatory

Out file for VVR solution

Use the following command to generate and execute the workflow solutions:

```
# ansible-playbook infoscale_workflow.yaml
```

Under `base_working_dir` location, folder along with solution playbooks, inventory and `workflow.yaml` is created and it executes the playbooks in a sequence as mentioned below:

- `install_cluster1.yml`
- `install_cluster2.yml`
- `configure_cluster1.yml`
- `configure_cluster2.yml`
- `secure_cluster1.yml`
- `secure_cluster2.yml`
- `gco_config.yml`
- `create_objects_dg1.yml`
- `create_vip_cluster1_dg1_rvg1.yml`
- `create_vip_cluster2_dg1_rvg1.yml`

- create_vip_cluster1_dg1_rvg2.yml
- create_vip_cluster2_dg1_rvg2.yml
- create_vvrres_dg1_rvg1.yml
- create_vvrres_dg1_rvg2.yml

If the workflow generate option (execute: 0) is used then you need to execute the workflow using this command:

```
# ansible-playbook - i inventory workflow.yaml
```

Solution templates

This chapter includes the following topics:

- [Customizing InfoScale deployment using solution templates](#)

Customizing InfoScale deployment using solution templates

Solution templates are high-level Ansible playbooks that include modular calls to other playbooks or roles for specific InfoScale tasks. These templates give users the flexibility to add more Ansible components—such as playbooks, roles, or tasks—between steps to match their existing InfoScale workflows and environment.

Table 8-1 Infoscale_workflow

Operations	Required modules	Playbook
Full upgrade	<ul style="list-style-type: none"> ■ site_factors ■ upgrade 	<pre> Playbook - YAML full_upgrade.yml --- ## - import_playbook: <mycustomplaybook>.yaml ## tasks: ## - name: My custom task ## module_name: ## needed_args: ## - name: Start processes ## import_role: ## name: upgradestart_processes ## vars: ## arguments: "{{ arguments }}" - import_playbook: fullupgrade_precheck.yml - import_playbook: fullupgrade_preupgrade.yml - import_playbook: fullupgrade_stop_services.yml - import_playbook: fullupgrade_upgrade_rpm.yml - import_playbook: fullupgrade_start_services.yml </pre>

As per the solution template concept, the full upgrade playbook is broken into smaller steps as shown above. The user can add tasks like reboot machine and run their own roles, or call another playbooks in between. You can take reference of the commented part in the playbook to know more about adding more tasks, roles, and playbook. These can be added anywhere in the playbook.

There is no changes in the variables from the full upgrade scenario. To know more about passing variables, See "[Upgrading InfoScale using Ansible](#)" on page 54.