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

VERITAS™

Last updated: 2019-02-01

Legal Notice

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

http://www.veritas.com

Technical Support

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

https://www.veritas.com/support

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

https://my.veritas.com

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

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

Documentation

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

https://sort.veritas.com/documents

Documentation feedback

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

infoscaledocs@veritas.com

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

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

Veritas Services and Operations Readiness Tools (SORT)

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

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

Contents

Chapter 1

Preparing to use Ansible in InfoScale

This chapter includes the following topics:

Inroduction to Ansible

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

Veritas now provides Ansible modules that can be used in Ansible playbooks to install or upgrade Veritas InfoScale products, deploy clusters, and configure features such as File System (FS), Cluster File System (CFS), and Disk Group Volume.

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

Deployment-related operations	Feature-related operations
■ Installation ■ Licensing ■ Component configuration ■ Starting a cluster ■ Stopping a cluster ■ Full upgrade ■ Rolling upgrade ■ Uninstallation	■ Configuring a Cluster File System (CFS) ■ Creating a disk group volume ■ Configuring an File System (FS) resource

Supported platforms

You can use Ansible to deploy and configure Veritas InfoScale on all Linux platforms supported by InfoScale.

Supported Ansible version

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

Downloading Ansible modules for InfoScale

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

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

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

- /usr/share/ansible/plugins/modules/

Before you begin

Ensure that the following prerequisites are met in your environment:

- Ansible requires passwordless SSH communication to be established between all servers in the environment.

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

- Ensure that Python 2.x is installed and configured on all machines in the environment.

Using site_facters module

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

Chapter 2

Installing using Ansible

This chapter includes the following topics:

Ansible modules for Installing InfoScale

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

Table 2-1 Installation-related operations

Operation	Required modules	Sample playbook
Installation	■ site_facters ■ yum	```--- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_facters register: facts - name: Install InfoScale Enterprise veritas_infoscale: module: yum repository_name: *RepositoryName* repository_baseurl: *RepositoryBaseURL* gpgcheck: 1 gpgkey: http://xx.xxx.xxx.xx/ rpms/RPM-GPG-KEY-veritas-infoscale7 product: 'ENTERPRISE' product_version: 7.4.1 facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}" state: present```
Licensing	■ site_facters ■ licensing	```--- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_facters register: facts - name: License veritas_infoscale: module: licensing state: present product_version: '7.4.1' license: 'ENTERPRISE' facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"```

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

Operation	Required modules	Sample playbook
Uninstall	■ site_facters ■ yum	```--- - hosts: cpicluster11 gather_facts: False any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_facters register: facts - name: Uninstall InfoScale Enterprise veritas_infoscale: module: yum product: 'ENTERPRISE' product_version: 7.4.1 state: absent facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"```

List of pre-defined keywords

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

yum

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

Table 2-2 Yum module keywords

Name	Description	Mandatory/optional
repository_name:	Name of the yum repository used to install InfoScale rpms. **Example**: `repo-InfoScale741`	Mandatory, if the base URL is not provided (If the Base URL is provided the system will create a repository name, optional for install.)

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

Name	Description	Mandatory/optional
repository_baseurl:	Specifies the URL to the directory where the repodata of a repository is located. **Example**: `http://xx.xxx.xxx.xx/rpms/`	Optional (This keyword is not required if you are using a yum repository already configured on the system, and are providing the repository name.)
gpgcheck:	Specify whether to check the integrity of the yum packages by using the gpgkey provided with the InfoScale installation media. This is a boolean variable and must be specified using 0 or 1. By default the value is set to 0.	Optional
gpgkey:	Specifies the location of the gpgkey (typically located in the rpms directory of the installation media). **Example**: `http://xx.xxx.xxx.xx/rpms/RPM-GPG-KEY-veritas-infoscale7`	Mandatory, if gpgcheck is 1
product:	Specifies the name of the product you want to install. **Examples**: `ENTERPRISE`, `AVAILABILITY`, `STORAGE`, or `FOUNDATION`.	Mandatory
state:	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional

licensing

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

Table 2-3 Licensing module keywords

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

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

Name	Description	Mandatory/optional
product_version:	Specifies the version of the product that you want to install or upgrade. **Example**: `http://xx.xxx.xxx.xx/rpms/`	Mandatory
license:	Specifies the path to the slf license file to be registered on the system. Ensure that the license file is accessible store on the same server where the installer is saved. If you are performing a keyless installation or upgrade you can simply enter the product name. **Examples**: ■ `/license_key/Unix/perpetual/xxxxxxxxxxxxxxxx.slf` ■ `ENTERPRISE` ■ `AVAILABILITY`	Mandatory

Chapter 3

Configuring InfoScale product components

This chapter includes the following topics:

Ansible modules for configuring InfoScale

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

Table 3-1 Component configuration-related operations

Operation	Required modules	Sample playbook
Component configuration	■ site_facters ■ config_component	

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

Operation	Required modules	Sample playbook
		Configuring sfcfsha `---` `- hosts: cpicluster11` `  gather_facts: false` `  any_errors_fatal: true` `  tasks:` `  - name: Facters` `  veritas_infoscale:` `  module: site_facters` `  register: facts` `  - name:  Configure Enterprise` `  veritas_infoscale:` `  module: sfha_config` `  cluster_name: clust_cpi9` `  systems: [objstorer820-1-vm16,objstorer820-1-vm17]` `  cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61` `  component: sfcfsha` `  product_version: '7.4.1'` `  method: ethernet` `  state: present` `  private_link: eth1,eth2` `  low_priority_link: eth0` `  vcs_clusterid: 23838` `  facters: "{{ groups['all'] |map('extract',` `hostvars, ['facts','infoscale_facts'])| select()|list }}"` **Configuring sfha** `---` `- hosts: cpicluster11` `  gather_facts: true` `  any_errors_fatal: true` `  tasks:` `  - name: Facters` `  veritas_infoscale:` `  module: site_facters` `  register: facts` `  - name:  Configure Enterprise` `  veritas_infoscale:` `  module: sfha_config` `  cluster_name: clust_cpi9` `  systems: [objstorer820-1-vm16,objstorer820-1-vm17]`

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

Operation	Required modules	Sample playbook
		cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61 component: sfha product_version: '7.4.1' method: ethernet state: present private_link: eth1,eth2 low_priority_link: eth0 vcs_clusterid: 23838 facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}" **Configuring sf** --- - hosts: cpicluster11 gather_facts: true any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_facters register: facts - name: Configure Enterprise veritas_infoscale: module: sfha_config cluster_name: clust_cpi9 systems: [dl380g10-12-kvm-04,dl380g10-12-kvm-03] cluster_uuid: c7c2d65e-058f-11e8-a32c-c094107f3b61 component: sfha product_version: '7.4.1' method: ethernet state: present private_link: eth1,eth2 low_priority_link: eth0 vcs_clusterid: 23838

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

Operation	Required modules	Sample playbook
Starting a cluster	■ site_facters ■ process	--- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_facters register: facts - name: Start Product veritas_infoscale: module: process component: sfcfsha product: enterprise product_version: '7.4.1' seednode: objstorer820-1-vm17 state: present facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"
Stopping a cluster	■ site_facters ■ process	--- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_facters register: facts - name: Start Product veritas_infoscale: module: process component: sfcfsha product: enterprise product_version: '7.4.1' seednode: objstorer820-1-vm17 state: absent facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"

List of pre-defined keywords

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

sfcfsha_config

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

Table 3-2 Sfcfsha_config module keywords

Name	Description	Mandatory/optional
cluster_name:	Define a name for the cluster that you want to deploy. **Example**: `Cluster1`	Mandatory
cluster_uuid:	Define a unique alphanumeric ID to assign to the cluster you want to deploy. **Example**: `c7c2d65e-057f-11e8-a32c-c094107f3b61`	Mandatory
component:	Specify which components you want to configure in your product. Note that the product license acquired must support the required components. **Example**: `SF`, `VCS`, or `SFCHA`	Mandatory
method:	Specify the communication protocol that you want to deploy in the cluster. **Example**: `ethernet`, `udp`, or `rdma`	Mandatory
state:	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional
private_link:	Lists the name or the address (IPv4 or IPv6) that each heartbeat link uses on each of the nodes in the cluster.	Mandatory
low_priority_link:	Lists the name or the address (IPv4 or IPv6) that each low priority heartbeat link uses on each of the nodes in the cluster.	Optional
vcs_clusterid:	Define a unique number to be assigned to the cluster. If user does not provide a cluster Id it will be automatically assigned by the system.	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 3-3 Process module keywords

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

Chapter 4

Upgrading InfoScale using Ansible

This chapter includes the following topics:

Ansible modules for upgrading InfoScale

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

Table 4-1 Upgrade-related operations

Operation	Required modules	Sample playbook
Full upgrade	■ site_facters ■ upgrade	--- - hosts: cpicluster7 gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_facters register: facts - name: Upgrade Infoscale to 7.4.1 veritas_infoscale: module: upgrade repository_name: *RepositoryName* product_version: 7.4.1 repository_baseurl: *RepositoryBaseURL* gpgcheck: 1 gpgkey: http://*xx.xxx.xxx.xx*/sde2/7.4.1/dvd1-redhatlinux/rhel7_x86_64/rpms/RPM-GPG-KEY-veritas-infoscale7 product: enterprise license: /license_key/Unix/perpetual/*xxxxxxxxxxxx*.slf component: sfcfsha start_process: yes seednode: dl380g10-09-vm7 edgeserver_hostname: telemetry.veritas.com edgeserver_port: 443 state: present facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"

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

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

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

Operation	Required modules	Sample playbook
		`` --- - hosts: cpicluster11 gather_facts: False any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_facters register: facts - hosts: cpicluster11 gather_facts: False any_errors_fatal: true serial: 1 tasks: - name: Rolling Upgrade(phase1) Infoscale to 7.4.1 veritas_infoscale: module: ru_phase1 repository_name: RepositoryName product_version: 7.4.1 repository_baseurl: RepositoryBaseURL gpgcheck: 1 gpgkey: http://xx.xxx.xxx.xx/rpms/ RPM-GPG-KEY-veritas-infoscale7 product: enterprise license: ENTERPRISE component: sfcfsha start_process: 'yes' edgeserver_port: '443' edgeserver_hostname: 'telemetry.veritas.com' seednode: objstorer820-1-vm16 state: present facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}" - hosts: cpicluster11 gather_facts: False any_errors_fatal: true tasks: - name: Rolling Upgrade(phase2) Infoscale to 7.4.1 phase2 veritas_infoscale: module: ru_phase2 ```

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

Operation	Required modules	Sample playbook
		repository_name: *RepositoryName* product_version: 7.4.1 repository_baseurl: *RepositoryBaseURL* gpgcheck: 1 gpgkey: http://*xx.xxx.xxx.xx*/Infoscale/7.4.1/rpms/RPM-GPG-KEY-veritas-infoscale7 product: enterprise license: ENTERPRISE component: sfcfsha start_process: 'yes' seednode: objstorer820-1-vm16 edgeserver_hostname: telemetry.veritas.com edgeserver_port: 443 state: present facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"

List of pre-defined keywords

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

upgrade

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

Table 4-2 Upgrade module keywords

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

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

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

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

Name	Description	Mandatory/optional
edgeserver_hostname	Use this parameter to enter the server host name to which installation, deployment, and usage data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. The Veritas Cloud Receiver (VCR) is a preconfigured, cloud-based edge server deployed by Veritas. Enter **telemetry.veritas.com** to use the Veritas Cloud Receiver as an edge server for your environment.	Mandatory
edgeserver_port	Use this parameter to enter the port number of the edge server to which data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. Enter **443** if you are using the Veritas Cloud Receiver as an edge server for your environment.	Mandatory

ru_phase1

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

Table 4-3 Ru_phase1 module keywords

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

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

Name	Description	Mandatory/optional
gpgkey:	Specifies the location of the gpgkey (typically located in the rpms directory of the installation media). **Example**: `http://xx.xxx.xxx.xx/sde2/7.4.1/dvd1-redhatlinux/rhel7_x86_64/rpms/RPM-GPG-KEY-veritas-infoscale7`	Mandatory. if gpgcheck is 1
product:	Specifies the name of the product you want to install. **Examples**: `ENTERPRISE`, `AVAILABILITY`, `STORAGE`, or `FOUNDATION`.	Mandatory
state:	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional
license:	Specifies the path to the slf license file to be registered on the system. Ensure that the license file is accessible store on the same server where the installer is saved. If you are performing a keyless installation or upgrade you can simply enter the product name. **Examples**: ■ `/license_key/Unix/perpetual/xxxxxxxxxxxxxxxxxx.slf` ■ `ENTERPRISE` ■ `AVAILABILITY`	Mandatory
component:	Specify which components you want to configure in your product. Note that the product license acquired must support the required components. **Example**: `SF`, `VCS`, or `SFCHA`	Mandatory
start_process	Specifies to start InfoScale processes if value is set to **yes**. By default this option is set to **no**.	Optional
seednode:	Select any node from the cluster that will be used to run commands related to the operations of that cluster. Ensure that you enter the host name as provided in the in the `/etc/ansible/hosts` file.	Mandatory

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

Name	Description	Mandatory/optional
edgeserver_hostname	Use this parameter to enter the server host name to which installation, deployment, and usage data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. The Veritas Cloud Receiver (VCR) is a preconfigured, cloud-based edge server deployed by Veritas. Enter **telemetry.veritas.com** to use the Veritas Cloud Receiver as an edge server for your environment.	Mandatory
edgeserver_port	Use this parameter to enter the port number of the edge server to which data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. Enter **443** if you are using the Veritas Cloud Receiver as an edge server for your environment.	Mandatory

ru_phase2 module keywords

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

Table 4-4

Name	Description	Mandatory/optional
product_version:	Specifies the version of the product that you want to install or upgrade. **Example**: `http://xx.xxx.xxx.xx/rpms/`	Mandatory
gpgcheck:	Specify whether to check the integrity of the yum packages by using the gpgkey provided with the InfoScale installation media. This is a boolean variable and must be specified using 0 or 1. By default the value is set to 0.	Optional
gpgkey:	Specifies the location of the gpgkey (typically located in the rpms directory of the installation media). **Example**: `http://xx.xxx.xxx.xx/rpms/RPM-GPG-KEY-veritas-infoscale7`	Mandatory if gpgcheck is 1

Table 4-4 *(continued)*

Name	Description	Mandatory/optional
product:	Specifies the name of the product you want to install. **Examples**: `ENTERPRISE`, `AVAILABILITY`, `STORAGE`, or `FOUNDATION`.	Mandatory
state:	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional
license:	Specifies the path to the slf license file to be registered on the system. Ensure that the license file is accessible store on the same server where the installer is saved. If you are performing a keyless installation or upgrade you can simply enter the product name. **Examples**: ■ `/license_key/Unix/perpetual/xxxxxxxxxxxxxxxxxxxxx.slf` ■ `ENTERPRISE` ■ `AVAILABILITY`	Mandatory
component:	Specify which components you want to configure in your product. Note that the product license acquired must support the required components. **Example**: `SF`, `VCS`, or `SFCHA`	Mandatory
start_process	Specifies to start InfoScale processes if value is set to **yes**. By default this option is set to **no**.	Optional
seednode:	Select any node from the cluster that will be used to run commands related to the operations of that cluster. Ensure that you enter the host name as provided in the in the `/etc/ansible/hosts` file.	Mandatory

Table 4-4 *(continued)*

Name	Description	Mandatory/optional
edgeserver_hostname	Use this parameter to enter the server host name to which installation, deployment, and usage data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. The Veritas Cloud Receiver (VCR) is a preconfigured, cloud-based edge server deployed by Veritas. Enter **telemetry.veritas.com** to use the Veritas Cloud Receiver as an edge server for your environment.	Mandatory
edgeserver_port	Use this parameter to enter the port number of the edge server to which data is sent as part of the Product Improvement Program. The Product Improvement Program allows the product installer to collect installation, deployment, and usage data and submit it anonymously to Veritas. The collected information helps identify how customers deploy and use the product. Enter **443** if you are using the Veritas Cloud Receiver as an edge server for your environment.	Mandatory

Chapter 5

Configuring InfoScale features

This chapter includes the following topics:

Ansible modules for configuring features in InfoScale

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

Table 5-1 Feature configuration-related keywords

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

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

Operation	Required modules	Sample playbook
Creating a disk group volume	■ site_facters ■ vxvm_dgvolfs	``` --- - hosts: cpicluster11 gather_facts: false any_errors_fatal: true tasks: - name: Facters veritas_infoscale: module: site_facters register: facts - name: Create DG, Volume, FS veritas_infoscale: module: vxvm_dgvolfs state: present dg1: dgname: testdg5 dgtype: shared fss: 1 disks: [objstorer820-1-_vmdk0_0] volinfo: [[testvol1,750m],[testvol2,800m]] seednode: objstorer820-1-vm17 facters: "{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}" ```

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

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

List of pre-defined keywords

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

cfsresource

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

Table 5-2 Cfsresource module keywords

Name	Description	Mandatory/Optional
state:	Specifies what state the package should be after the task is completed. The value for this keyword can be either present or absent. If you do not give any value for this parameter, by default, the state of the package is set to present, and the package will be installed.	Optional
sgname:	Name of the service group where you are creating the cluster file system.	Mandatory
dgname:	Name of the service group where you are creating the cluster file system.	Mandatory
volname:	Name of the volume where you are creating the cluster file system.	Mandatory
mnt:	Mount point of the volume where you are creating the cluster file system.	Mandatory
systems:	Specify the list of host names that are part of the cluster. Ensure that you enter the host names as provided in the in the `/etc/ansible/hosts` file. **Example**: `[hostname1,hostname2,hostname3]`	Mandatory
facters:	Enter the following value to collect system-related data from the cluster servers. **`"{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"`**	Mandatory

vxvm_dgvolfs

The vxvm_dgvolfs module is used to create disk group volumes and file systems in InfoScale. Use the following keywords while referencing the module in your playbook.

Table 5-3 Vxvm_dgvolfs module keywords

Name	Description	Mandatory/Optional
state:	Specifies what state the 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

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

Name	Description	Mandatory/Optional
fss:	Specifies whether the disk is part of a Flexible Storage Sharing (FSS) environment. The value for this keyword can be either yes or no. By default, this keyword is set to no.	Optional
dgtype:	Enter **shared** if you want to configure a shared type of disk group or leave the keyword empty.	Optional
disks:	Specifies the list of disks that you want to add in the disk group. Use the following format to enter the disk names: **Syntax**: `[<disk_name_1>,` `<disk_name_2>` `,...<disk_name_n>]` *<disk_name_1>*, *<disk_name_2>*, ...*<disk_name_n>* are the names of the disks that you want to add to the disk group. **Example**: `[disk1,disk2,disk3]`	Mandatory
volinfo:	**Syntax**: `[[<volume_name_1>,<volume_size_1>],` `[<volume_name_2>,<volume_size_n>],` `...[<volume_name_n>,<volume_size_n>]]` ■ *<volume_name_1>*, *<volume_name_2>*, ...*<volume_name_n>* are the names of the volumes. ■ *<volume_size_1>*, *<volume_size_2>*, ...*<volume_size_n>* are the sizes of the volumes. **Example**: `[[testvol1,200m],[testvol2,300m]]`	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 `/etc/ansible/hosts` file.	Mandatory

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

Name	Description	Mandatory/Optional
facters:	Enter the following value to collect system-related data from the cluster servers. `"{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"`	Mandatory

fsresource

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

Table 5-4 Fsresource module keywords

Name	Description	Mandatory/Optional
state:	Specifies what state the 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 file system.	Mandatory
sgtype:	Specify the type of service group. This value can either be Parallel or Failover.	Mandatory
dgname:	Name of the service 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

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

Name	Description	Mandatory/Optional
facters:	Enter the following value to collect system-related data from the cluster servers. `"{{ groups['all'] |map('extract', hostvars, ['facts','infoscale_facts'])| select()|list }}"`	Mandatory