Veritas Access Installation Guide

Linux

7.4.1

VERITAS

Veritas Access Installation Guide

Last updated: 2018-12-07

Document version: 7.4.1 Rev 3

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 document version appears on page 2 of each guide. 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:

doc.feedback@veritas.com

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

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

Veritas Services and Operations Readiness Tools (SORT)

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

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

Contents

Chapter 1

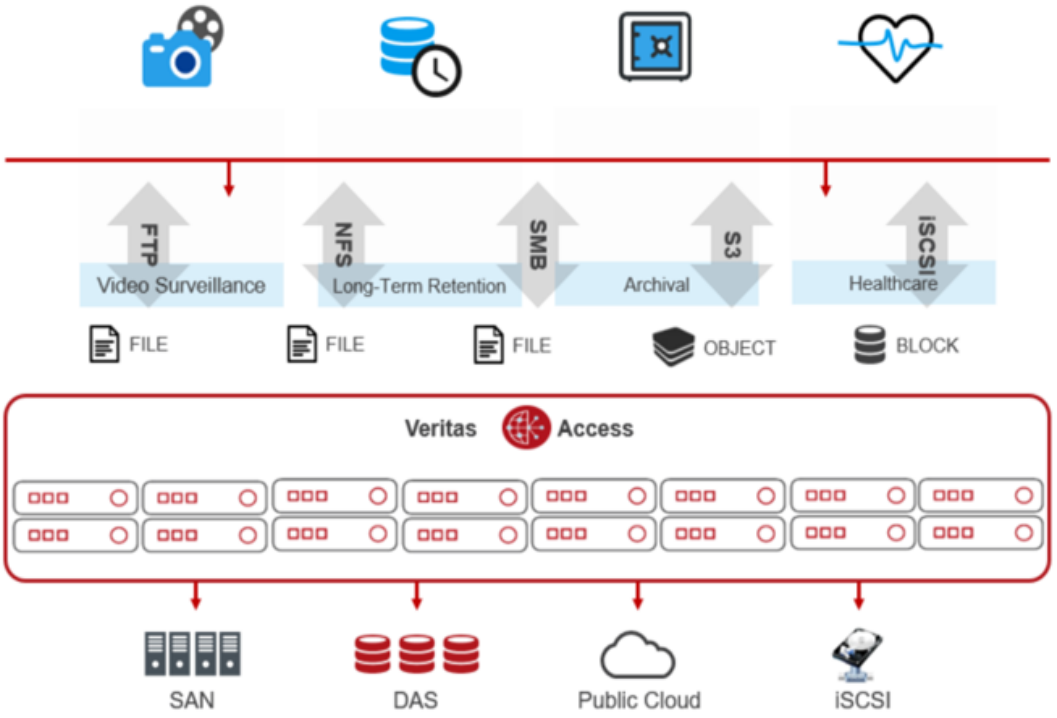
Introducing Veritas Access

This chapter includes the following topics:

About Veritas Access

Veritas Access is a software-defined, scale-out network-attached storage (NAS) solution for unstructured data that works on commodity hardware. Veritas Access provides resiliency, multi-protocol access, and data movement to and from the public or the private cloud based on policies.

Figure 1-1 Veritas Access architecture



You can use Veritas Access in any of the following ways.

Table 1-1 Interfaces for using Veritas Access

Interface	Description
GUI	Centralized dashboard and quick actions with operations for managing your storage. See the GUI and the online Help for more information.
RESTful APIs	Enables automation using scripts, which run storage administration commands against the Veritas Access cluster. See the *Veritas Access RESTful API Guide* for more information.
Command-line interface (CLI or CLISH)	Single point of administration for the entire cluster. See the manual pages for more information.

Table 1-2 describes the features of Veritas Access.

Table 1-2 Veritas Access key features

Feature	Description
Multi-protocol access	Veritas Access supports the following protocols: ■ Amazon S3 ■ CIFS ■ FTP ■ iSCSI target ■ NFS ■ Oracle Direct NFS ■ SMB 3 ■ NFS with S3
WORM storage for Enterprise Vault Archiving	Veritas Access can be configured as WORM primary storage for archival by Enterprise Vault. Veritas Access is certified as a CIFS primary WORM storage for Enterprise Vault 12.1. For more information, see the *Veritas Access Enterprise Vault Solutions Guide*.
WORM support over NFS	Veritas Access supports WORM over NFS.
Creation of Partition Secure Notification (PSN) file for Enterprise Vault Archiving	A Partition Secure Notification (PSN) file is created at a source partition after the successful backup of the partition at the remote site. For more information, see the *Veritas Access Enterprise Vault Solutions Guide*.
Managing application I/O workloads using maximum IOPS settings	The MAXIOPS limit determines the maximum number of I/Os processed per second collectively by the storage underlying the file system.
Flexible Storage Sharing (FSS)	Enables cluster-wide network sharing of local storage.

Table 1-2 Veritas Access key features *(continued)*

Feature	Description
Scale-out file system	The following functionality is provided for a scale-out file system: ■ File system that manages a single namespace spanning over both on-premises storage as well as cloud storage, which provides better fault tolerance for large data sets. ■ Highly available NFS and S3 shares. You use scale-out file systems if you want to store a large capacity of data in a single namespace (3 PB is the maximum file system size). ■ Creation of CIFS shares. ■ File sharing for a scale-out file system using FTP.
Cloud as a tier for a scale-out file system	Veritas Access supports adding a cloud service as a storage tier for a scale-out file system. You can move data between the tiers based on file name patterns and when the files were last accessed or modified. Use scheduled policies to move data between the tiers on a regular basis. Veritas Access moves the data from the on-premises tier to Amazon S3, Amazon Glacier, Amazon Web Services (AWS), GovCloud (US), Azure, Google cloud, Alibaba, Veritas Access S3, IBM Cloud Object Storage, and any S3-compatible storage provider based on automated policies. You can also retrieve data archived in Amazon Glacier.
SmartIO	Veritas Access supports both read and writeback caching on solid state drives (SSDs) for applications running on Veritas Access file systems.
SmartTier	Veritas Access's built-in SmartTier feature can reduce the cost of storage by moving data to lower-cost storage. Veritas Access storage tiering also facilitates the moving of data between different drive architectures and on-premises.
Snapshot	Veritas Access supports snapshots for recovering from data corruption. If files, or an entire file system, are deleted or become corrupted, you can replace them from the latest uncorrupted snapshot.
Deduplication	You can run post-process periodic deduplication in a file system, which eliminates duplicate data without any continuous cost.

Table 1-2 Veritas Access key features *(continued)*

Feature	Description
Compression	You can compress files to reduce the space used, while retaining the accessibility of the files and having the compression be transparent to applications. Compressed files look and behave almost exactly like uncompressed files: the compressed files have the same name, and can be read and written as with uncompressed files.
Erasure coding	Erasure coding is configured with the EC log option for the NFS use case.
IP load balancing	With IP load balancing, a single virtual IP is used to act as a load balancer IP, which distributes the incoming requests to the different nodes in the Veritas Access cluster for the services that are run on an active-active cluster.
Veritas Access as an iSCSI target for RHEL 7.x	Veritas Access as an iSCSI target can be configured to serve block storage. iSCSI target as a service is hosted in the active-active mode in the Veritas Access cluster.
Configuring Veritas Access in IPv4 and IPv6 mixed mode	Support for configuring the Veritas Access cluster in an IPv4 environment, or an IPV6 environment, or in a mixed mode environment where you have both IPv4 and IPv6 addresses.
NetBackup integration	Built-in NetBackup client for backing up your file systems to a NetBackup master or media server. Once data is backed up, a storage administrator can delete unwanted data from Veritas Access to free up expensive primary storage for more data.
OpenDedup integration	Integration with OpenDedup for deduplicating your data to on-premises or cloud storage for long-term data retention. See the *Veritas Access NetBackup Solutions Guide* for more information.
OpenStack plug-in	Integration with OpenStack: ■ OpenStack Cinder integration that allows OpenStack instances to use the storage hosted by Veritas Access. ■ OpenStack Manila integration that lets you share Veritas Access file systems with virtual machines on OpenStack Manila.

Table 1-2 Veritas Access key features *(continued)*

Feature	Description
Quotas	Support for setting file system quotas, user quotas, and hard quotas.
Replication	Periodic replication of data over IP networks. See the `episodic(1)` man page for more information. Synchronous replication of data over IP networks See the `continuous(1)` man page for more information.
Support for LDAP, NIS, and AD	Veritas Access uses the Lightweight Directory Access Protocol (LDAP) for user authentication.
Partition Directory	With support for partitioned directories, directory entries are redistributed into various hash directories. These hash directories are not visible in the namespace view of the user or operating system. For every new create, delete, or lookup, this feature performs a lookup for the respective hashed directory and performs the operation in that directory. This leaves the parent directory inode and its other hash directories unobstructed, which vastly improves file system performance. By default, this feature is not enabled. See the `storage_fs`(1) manual page to enable this feature.
Isolated storage pools	Enables you to create an isolated storage pool with a self-contained configuration. An isolated storage pool protects the pool from losing the associated metadata even if all the configuration disks in the main storage pool fail.
Performance and tuning	Workload-based tuning for the following workloads: ■ Media server - Streaming media represents a new wave of rich Internet content. Recent advancements in video creation, compression, caching, streaming, and other content delivery technology have brought audio and video together to the Internet as rich media. You can use Veritas Access to store your rich media, videos, movies, audio, music, and photos. ■ Virtual machine support ■ Other workloads

Chapter 2

Licensing in Veritas Access

This chapter includes the following topics:

About Veritas Access product licensing

In this release, Veritas has introduced the TB-per-core licensing model for Veritas Access. The per-core and per-terabyte licensing model of earlier releases is also supported in this release.

The TB-per-core licensing model is based on both capacity per-core and time period. You can now license Veritas Access as per your requirement for raw capacity. This is managed through the software.

Depending on the capacity to core ratio, three types of capacity-based licenses are available. Each license has an allotted storage capacity in the range of 2001 TB - Unlimited.

- Premium
- Standard
- Basic

The time-based license category includes the following licenses:

- Perpetual: A license with unlimited validity period.
- Subscription: A license that is valid for a subscribed period, and needs to be renewed from time to time. Typically, the subscription can be for a period of 1 year, 2 years, or 3 years, and so on.
- Trialware: A license that is valid for 60 days.

Veritas recommends the tier that is best suited for your needs based on your current system configuration across the clusters. The new metering and recommended tier is based on capacity utilization to core ratio. Capacity utilization is the raw capacity utilized while the core refers to the physical cores present across the cluster. This information is also available in the GUI in the **Recommended Tier**.

Table 2-1 Licensing methods

Tiering model	TB-per-core meter capacity	Capacity tier range	Time-based licensing
Premium	TB to core ratio <= 4 TB/core	2001 TB - Unlimited	Subscription - 1 year, 2 years, and 3 years Perpetual - Unlimited for a product version Trialware- 60 days
Standard	TB to core ratio Between 4 TB/core and 25 TB/core	2001 TB - Unlimited	Subscription - 1 year, 2 years, and 3 years Perpetual - Unlimited for a product version
Basic	TB to core ratio > 25 TB/core	2001 TB - Unlimited	Subscription - 1 year, 2 years, and 3 years Perpetual - Unlimited for a product version

You can download Veritas Access from the Veritas Access External Product page for evaluation.

The trialware has the premium tier licensing model with a storage capacity range of 2001 TB – Unlimited. You can upgrade to any valid per-core license from the trialware. If you have the Veritas Access 7.3.1 product with the per-core or per-terabyte licensing, and you upgrade to Veritas Access 7.4, you can continue to use the 7.3.1 per-core or 7.3.1 per-terabyte license.

Notes:

- You must provide a valid license during the product installation. If you do not provide a valid license, a 60-days trialware license is installed.
- If you exceed the licensed storage capacity, the product usage is not affected. However, Veritas recommends that in such cases, you must procure or renew your license to a higher capacity.
- If you fail to procure or renew your license before the expiry date, a grace period of 60-days is provided without any effect on the product usage.

- If you fail to procure or renew your license after the grace period, the services fails to start after a system restart or when services such as, CIFS, S3, NFS, and FTP are restarted.
- Veritas reserves the right to ensure entitlement and compliance through auditing.
- If you encounter problems while licensing this product, visit the Veritas Licensing Support website.
  https://www.veritas.com/licensing/process

Table 2-2 Functional enforcements of Veritas Access licensing

Enforcement	Action
During Validity	None
During Grace period	Persistent message (in the GUI only)
Post Grace Period	Before you restart the node, you can stop the NFS, CIFS, FTP, and S3 services, but you cannot start the services again (even if you have not restarted the node). After you restart the node, the NFS, CIFS, FTP, and S3 services do not come online on the restarted node.

If you add the Veritas Access license using the GUI:

- When a node is restarted after the license has expired, the NFS, CIFS, FTP, and S3 services are stopped on that node. The status of the service appears online if the service is running anywhere in the cluster, even if it is offline on this node. Check the alerts on each node individually to see if the service is online or offline locally.
- An option to start, stop, and check the status of NFS, CIFS, and S3 services is available. You cannot start, stop, or check the status of the FTP service.
- You can only provide the license file from the local system, the `scp` path is not supported through the GUI.

If you add the Veritas Access license using the CLISH:

- When a node is restarted after the license has expired, the NFS, CIFS, FTP, and S3 services are stopped on that node. You can use the `support services show` command to display the node-wise status of the service.
- An option to start, stop, and check the status of NFS, CIFS, FTP, and S3 services is available.

- You can add the license using the `license add` command. The `license add` command provides support for `scp` path as well.
- The `license list` and `license list details` commands provide details for the license that is installed on each node of the cluster.

Chapter 3

System requirements

This chapter includes the following topics:

Important release information

Review the *Veritas Access Release Notes* for the latest information before you install the product.

The hardware compatibility list contains information about supported hardware and is updated regularly. You can use any commodity hardware that is certified and mentioned in the hardware compatibility list.

For the latest information on supported hardware, see the compatibility list at:

https://sort.veritas.com/documents/doc_details/isa/7.4.1/Linux/CompatibilityLists/

For important updates regarding this release, review the Late-Breaking News TechNote on the Veritas Technical Support website:

https://www.veritas.com/support/en_US/article.100043385

System requirements

Table 3-1 lists the per-node system requirements for running the Veritas Access system software.

Table 3-1 System requirements for Veritas Access

Minimum	Recommended
Each Veritas Access node using a 64-bit Intel-based server architecture that is compatible with Red Hat Enterprise Linux (RHEL) 7 Update 3 and 4, Oracle Linux (OL) 7 Update 4, or AMD64, and Intel EMT. Itanium is not supported.	Two nodes of dual or quad core processors at 2.0 GHz or later for optimal performance.
32 GB error-correcting code (ECC) random-access memory (RAM)	The recommended values depend on the expected workload.
One internal drive with size equal to size of RAM + 60 GB	Dual boot drives each of size RAM + 60 GB or more capacity. In an FSS-based environment, additional internal drives (SSD + HDD) are recommended.
Four 1G Ethernet interfaces (Two ethernet interface are used for public and two for private network.)	Four 10G ethernet interfaces are recommended (Two ethernet interface are used for public and two for private network.).
One Fibre Channel Host Bus Adapters (HBA)	Two Fibre Channel Host Bus Adapters (HBAs) are recommended for high availability (HA) if you are using shared LUNs that need to be mapped over a Fibre Channel protocol. If the environment has only DAS or iSCSI disks, then the HBA requirement is optional.
Internal/external USB DVD-ROM DVD drive	N/A
Redundant power supply	Recommended, but not required.
SmartIO caching feature	A PCI-based SSD card is recommended if you want to use the SmartIO caching feature.
Minimum number of servers required is 1	N/A

Table 3-2 lists the operating system (OS) partition requirements for running the Veritas Access system software.

Table 3-2 Operating system partition requirements for Veritas Access

Partition	Recommended size (Minimum)	Details
/opt	100 GB	Used to store Veritas Access software, logs, and core dumps.
/usr	3 GB	Used to install the dependent OS rpms.
swap	8 GB	Used to swap space when physical memory is full.
/	30 GB	Used for the operating system.

Note: The operating system partition requirements are only for Veritas Access, additional space is required for OS-specific packages, which needs to be allocated as required.

Linux requirements

Each release of Veritas Access has strict operating system (OS) versioning requirements. The minimum operating system requirements are enforced during the Veritas Access installation. A Kickstart file is also available on request for Veritas Access 7.4.1 to assist partners with the operating system installation requirements.

Operating system patches, including security vulnerability patches, can be installed without requiring certification from Veritas. However, operating system Kernel RPMs should not be patched without specific approval from Veritas.

The Veritas Access 7.4.1 release requires and supports the following Red Hat Enterprise Linux (RHEL) or the Oracle Linux (OL) operating system versions:

- Red Hat Enterprise Linux (RHEL)
  - RHEL 7 Update 3 and 4
- Oracle Linux (only in RHEL compatible mode)
  - OL 7 Update 4

The certification of the RHEL OS updates requires a new minor version of Veritas Access. You require an agreement with Veritas to install the RHEL OS updates.

Note: Veritas Access does not support the OS on which Veritas Access runs.

Veritas Access can be installed on computers running the following operating systems:

Requirement	Version	Version
Red Hat Enterprise Linux version	RHEL 7 Update 3	RHEL 7 Update 4
Oracle Linux	OL 7 Update 4	
Kernel version	3.10.0-693.el7	
Required RPMs	See "Required operating system RPMs for RHEL 7.4" on page 26. See "Required operating system RPMs for OL 7.4" on page 23.	

Operating system RPM installation requirements and operating system patching

Before you install Veritas Access you need these OS RPMs. Veritas has categorized the OS RPMs into four groups.

Category 1:

- This set of RPMs are kernel RPMs that are required to be installed with exact predefined RPM versions only.
- The required RPM versions are different for RHEL 7.3 and RHEL 7.4.
- The required RPM versions are different for OL 7.4.
- The RPMs in this category should not be patched without specific approval from Veritas.
- See "Kernel RPMs that are required to be installed with exact predefined RPM versions" on page 22.
- See "OL kernel RPMs that are required to be installed with exact predefined RPM versions" on page 23.

Category 2:

- This set of RPMs include the OS libs and OS packages that must be installed with minimum predefined RPM versions.
- The required RPM versions are different for RHEL 7.3 and RHEL 7.4.

- The required RPM versions are different for OL 7.4.
- The RPMs in this category can be patched using official Red Hat patches.
- An approval or certification from Veritas is not required to patch these RPMs.
- See “Required operating system RPMs for OL 7.4” on page 23.
- See “Required operating system RPMs for RHEL 7.3” on page 25.
- See “Required operating system RPMs for RHEL 7.4” on page 26.

Category 3:

- This set of RPMs are required by Category 2 RPMs as dependencies, their installation is enforced by Red Hat.
- Veritas Access does not require any specific versions of these RPMs to be installed.
- The versions of these RPMs are determined by Red Hat.
- The RPMs in this category can be patched using official Red Hat patches.
- An approval or certification from Veritas Access is not required to patch these RPMs.
- Veritas does not document these RPMs as required RPMs for Veritas Access.

Category 4:

- This set of RPMs are third-party RPMs that are included in the Veritas Access ISO.
- These RPMs are not OS RPMs. It includes Samba, Ganesha, and other third-party products.
- The RPMs in this category should not be patched without specific approval from Veritas.
- Veritas installs these RPMs as they are included in the Veritas Access ISO.

Kernel RPMs that are required to be installed with exact predefined RPM versions

After you install the RHEL OS, install the following RPMs, and then restart the system.

RHEL 7 Update 3 kernel packages:

The following RPMs are included in the DVD image under the **os_rpms** directory and are installed using the CPI installation.

- kernel-debuginfo-3.10.0-514.el7.x86_64.rpm

- kernel-headers-3.10.0-514.el7.x86_64.rpm
- kernel-debuginfo-common-x86_64-3.10.0-514.el7.x86_64.rpm

RHEL 7 Update 4 kernel packages:

The following RPM is included in the DVD image under the **os_rpms** directory and is installed using the CPI installation.

- kernel-headers-3.10.0-693.el7.x86_64.rpm
- kernel-debuginfo-common-x86_64-3.10.0-693.el7.x86_64.rpm
- kernel-headers-3.10.0-693.21.1.el7.x86_64.rpm
- kernel-headers-3.10.0-693.el7.x86_64.rpm

OL kernel RPMs that are required to be installed with exact predefined RPM versions

The OL environment should be the following with Red Hat compatible kernels only:

- OL 7.4

Note: For the OL 7.x OS, the `uek` kernel is not supported.

Example:

```
[root@oel_01 ~]# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.4 (Maipo)
[root@oel_01 ~]# cat /etc/oracle-release
Oracle Linux Server release 7.4
[root@oel_01 ~]# uname -r
3.10.0-693.el7.x86_64
```

Required operating system RPMs for OL 7.4

The RPM version numbers specified in these lists are the minimum required version numbers for these operating system RPMs.

```
Required OS packages for OL 7.4:
PyYAML 3.10-11                          apr-devel 1.4.8-3
apr-util-devel 1.5.2-6                  arptables 0.0.4-8
at 3.1.13-22                            autogen-libopts 5.18-5
avahi-libs 0.6.31-17                    bash 4.2.46-28
binutils 2.25.1-31                      cairo 1.14.8-2
coreutils 8.22-18                       cups-libs 1.6.3-29
ethtool 4.8-1                           fuse 2.9.2-8
```

fuse-devel 2.9.2-8
fuse-libs 2.9.2-8
glibc-common 2.17.196
glibc-devel.x86_64 2.17.196
glibc-headers 2.17.196
glibc-utils 2.17.196
glibc.i686 2.17.196
glibc.x86_64 2.17.196
httpd 2.4.6-67
httpd-devel 2.4.6-67
httpd-manual 2.4.6-67
httpd-tools 2.4.6-67
infiniband-diags 1.6.7-1
initscripts 9.49.39-1
iproute 3.10.0-87
ipvsadm 1.27-7
iscsi-initiator-utils 6.2.0.874-4
jansson 2.10-1
kmod 20-15
krb5-devel 1.15.1-8
krb5-libs 1.15.1-8
krb5-workstation 1.15.1-8
libibumad 13-7
libibverbs-utils 13-7
libjpeg-turbo 1.2.90-5
libpcap 1.5.3-9
libtirpc 0.2.4-0.10
libyaml 0.1.4-11
lshw B.02.18-7
lsof 4.87-4
lsscsi 0.27-6
memcached 1.4.15-10
mlocate 0.26-6
mod_ssl 2.4.6-67
mod_wsgi 3.4-12
net-snmp 5.7.2-28
net-snmp-utils 5.7.2-28
net-tools 2.0-0.22
nfs-utils 1.3.0-0.48
nmap-ncat 6.40-7
nscd 2.17-196
nss-pam-ldapd 0.8.13-8
ntp 4.2.6p5-25
ntpdate 4.2.6p5-25
openldap 2.4.44-5
openldap-clients 2.4.44-5
opensm 3.3.19-1
opensm-libs 3.3.19-1
openssl 1.0.2k-12.el7
openssl-devel 1.0.2k-12.el7
openssl-libs 1.0.2k-12.el7
pango 1.40.4-1
perl 5.16.3
perl-Convert-ASN1 0.26-4
perl-JSON 2.59-2
perl-LDAP 0.56-5
perl-Net-Telnet 3.03-19.el7
perl-XML-Parser 2.41-10
psmisc 22.20-15
python-backports 1.0-8
python-backports-ssl_match_hostname 3.4.0.2-4
python-chardet 2.2.1-1
python-memcached 1.59-1.noarch
python-paramiko 1.7.7.1-3
python-requests 2.6.0-1
python-setuptools 0.9.8-7
python-six 1.9.0-2
python-urllib3 1.10.2-3
rrdtool 1.4.8-9
rsh 0.17-76
sg3_utils 1.37-12
sg3_utils-libs 1.37-12
strace 4.12-4
sysstat 10.1.5-12
targetcli 2.1.fb46-1
telnet 0.17-64
traceroute 2.0.22-2
tzdata-java
unzip 6.0-16
vim-enhanced 7.4.160
vsftpd 3.0.2-22
wireshark 1.10.14-14
yp-tools 2.14-5
ypbind 1.37.1-9
zip 3.0-11

Required operating system RPMs for RHEL 7.3

The RPM version numbers specified in this list are the minimum required version numbers for this operating system RPM.

```
Required OS lib rpms for RHEL 7.3:
bc-1.06.95-13.el7.x86_64          coreutils-8.22-18.el7.x86_64
ed-1.9-4.el7.x86_64               findutils-4.5.11-5.el7.x86_64
glibc-2.17-157.el7.x86_64         libgcc-4.8.5-16.el7.x86_64
libstdc++-4.8.5-16.el7.x86_64     libreport-plugin-mailx-2.1.11-38.el7.x86_64
mailx-12.5-16.el7.x86_64          openssl-libs-1.0.2k-12.el7.x86_64
perl-Exporter-5.68-3.el7.noarch   perl-Socket-2.010-4.el7.x86_64
policycoreutils-2.5-8.el7.x86_64  python-2.7.5-48.el7.x86_64
python-libs-2.7.5-48.el7.x86_64   zlib-1.2.7-17.el7.x86_64

Required OS packages for RHEL 7.3:
apr-devel 1.4.8-3                                 apr-util-devel 1.5.2-6
arptables 0.0.4-8                                 at 3.1.13-22
autogen-libopts 5.18-5                            avahi-libs 0.6.31-17
bash 4.2.46-20                                    binutils 2.25.1-22.base
cairo 1.14.8-2                                    coreutils 8.22-18
cups-libs 1.6.3-29                                ethtool 4.8-1
fuse 2.9.2-8                                      fuse-devel 2.9.2-8
fuse-libs 2.9.2-8                                 glibc-common 2.17.196
glibc-devel-2.17-196.el7_4.2                      glibc-headers 2.17.196
glibc-utils 2.17.196                              glibc-2.17-157.el7
infiniband-diags 1.6.7-1                          initscripts 9.49.37-1
iproute 3.10.0-74                                 ipvsadm 1.27-7
iscsi-initiator-utils 6.2.0.874-4                 jansson 2.10-1
kernel-debuginfo 3.10.0-514.el7                   kernel-debuginfo-common-x86_64 3.10.0-514.el7
kernel-headers 3.10.0-514.el7                     kmod 20-9
krb5-devel 1.15.1-8                               krb5-libs 1.15.1-8
krb5-workstation 1.15.1-8                         ksh-20120801-35.el7_4
libibumad 13-7                                    libibverbs-utils 13-7
libjpeg-turbo 1.2.90-5                            libpcap 1.5.3-9
libtirpc 0.2.4-0.10                               libyaml 0.1.4-11
lshw B.02.18-7                                    lsof 4.87-4
lsscsi 0.27-4                                     memcached 1.4.15-10
mlocate 0.26-6                                    net-snmp 5.7.2-28
net-snmp-utils 5.7.2-28                           net-tools 2.0-0.22.20131004git
nfs-utils 1.3.0-0.48                              nmap-ncat 6.40-7
nscd 2.17-196                                     nss-pam-ldapd 0.8.13-8
ntp 4.2.6p5-25                                    ntpdate 4.2.6p5-25
openldap 2.4.44-5                                 openldap-clients 2.4.44-5
```

```
opensm 3.3.19-1                                  opensm-libs 3.3.19-1
openssl 1.0.2k-12.el7                            openssl-devel 1.0.2k-12.el7
openssl-libs 1.0.2k-12.el7                       pango 1.40.4-1
perl 5.16.3-292                                  perl-Convert-ASN1 0.26-4
perl-JSON 2.59-2                                 perl-LDAP 0.56-5
perl-Net-Telnet 3.03-19.el7                      perl-XML-Parser 2.41-10
psmisc 22.20-15                                  python-backports 1.0-8
python-backports-ssl_match_hostname 3.4.0.2-4    python-chardet 2.2.1-1
python-memcached 1.59-1.noarch                   python-paramiko 2.1.1-4
python-requests 2.11.1-1                         python-setuptools 0.9.8-7
python-six 1.9.0-2                               python-urllib 3 1.16-1
PyYAML 3.10-11                                   rdma 7.3_4.7_rc2-6
rrdtool 1.4.8-9                                  rsh 0.17-76
sg3_utils 1.37-12                                sg3_utils-libs 1.37-12
strace 4.12-4                                    sysstat 10.1.5-12
targetcli 2.1.fb46-1                             telnet 0.17-64
traceroute 2.0.22-2                              tzdata-java 2018d-1.el7.noarch
unzip 6.0-16                                     vim-enhanced 7.4.160-2
vsftpd 3.0.2-22                                  wireshark 1.10.14-14
yp-tools 2.14-5                                  ypbind 1.37.1-9
zip 3.0-11
```

Required operating system RPMs for RHEL 7.4

The RPM version numbers specified in this list are the minimum required version numbers for this operating system RPM.

```
Required OS lib rpms for RHEL 7.4:
bc-1.06.95-13.el7.x86_64          coreutils-8.22-18.el7.x86_64
ed-1.9-4.el7.x86_64               findutils-4.5.11-5.el7.x86_64
glibc-2.17-196.el7.x86_64         libacl-2.2.51-12.el7.x86_64
libgcc-4.8.5-16.el7.x86_64        libstdc++-4.8.5-16.el7.x86_64
libreport-plugin-mailx-2.1.11-38.el7.x86_64
mailx-12.5-16.el7.x86_64          openssl-libs-1.0.2k-12.el7.x86_64
perl-Exporter-5.68-3.el7.noarch   perl-Socket-2.010-4.el7.x86_64
policycoreutils-2.5-22.el7.x86_64
python-2.7.5-58.el7.x86_64        python-libs-2.7.5-58.el7.x86_64
zlib-1.2.7-17.el7.i686            zlib-devel-1.2.7-17.el7.x86_64

Required OS packages for RHEL 7.4:
apr-devel 1.4.8-3                       apr-util-devel 1.5.2-6
arptables 0.0.4-8                       at 3.1.13-22
autogen-libopts 5.18-5                  avahi-libs 0.6.31-17
bash 4.2.46-28                          binutils 2.25.1-31.base
```

cairo 1.14.8-2
coreutils 8.22-18
cups-libs 1.6.3-29
ethtool 4.8-1
fuse 2.9.2-8
fuse-devel 2.9.2-8
fuse-libs 2.9.2-8
glibc-common 2.17.196
glibc-devel-2.17-196.el7_4.2.x86_64
glibc-headers 2.17.196
glibc-utils 2.17.196
glibc-2.17-222.el7.i686
glibc-2.17-222.el7.x86_64
infiniband-diags 1.6.7-1
initscripts 9.49.39-1
iproute 3.10.0-87
ipvsadm 1.27-7
iscsi-initiator-utils 6.2.0.874-4
jansson 2.10-1
ksh-20120801-34
kmod 20-15
kernel-headers-3.10.0-693.el7
kernel-debuginfo-common-x86_64-3.10.0-693
kernel-headers-3.10.0-693.21.1.el7
kernel-headers-3.10.0-693.el7
krb5-devel 1.15.1-8
krb5-libs 1.15.1-8
krb5-workstation 1.15.1-8
libibumad 13-7
libibverbs-utils 13-7
libjpeg-turbo 1.2.90-5
libpcap 1.5.3-9
libtirpc 0.2.4-0.10
libyaml 0.1.4-11
lshw B.02.18-7
lsof 4.87-4
lsscsi 0.27-6
memcached 1.4.15-10
mlocate 0.26-6
net-snmp 5.7.2-28
net-snmp-utils 5.7.2-28
net-tools 2.0-0.22
nfs-utils 1.3.0-0.48
nmap-ncat 6.40-7
nscd 2.17-196
nss-pam-ldapd 0.8.13-8
ntp 4.2.6p5-25
ntpdate 4.2.6p5-25
openldap 2.4.44-5
openldap-clients 2.4.44-5
opensm 3.3.19-1
opensm-libs 3.3.19-1
openssl 1.0.2k-12.el7
openssl-devel 1.0.2k-12.el7
openssl-libs 1.0.2k-12.el7
pango 1.40.4-1
perl 5.16.3-292
perl-Convert-ASN1 0.26-4
perl-JSON 2.59-2
perl-LDAP 0.56-5
perl-Net-Telnet 3.03-19.el7
perl-XML-Parser 2.41-10
psmisc 22.20-15
python-backports 1.0-8
python-backports-ssl_match_hostname 3.4.0.2-4
python-chardet 2.2.1-1
python-memcached 1.59-1.noarch
python-paramiko 2.1.1-4.el7.noarch
python-requests-2.11.1-1.el7ost.noarch
python-setuptools 0.9.8-7
python-six 1.9.0-2
python-urllib3 1.16-1.el7ost.noarch
PyYAML 3.10-11
rrdtool 1.4.8-9
rsh 0.17-76
sg3_utils 1.37-12
sg3_utils-libs 1.37-12
strace 4.12-4
sysstat 10.1.5-12
targetcli 2.1.fb46-1
telnet 0.17-64
traceroute 2.0.22-2
tzdata-java 2018d-1.el7.noarch
unzip 6.0-16
vim-enhanced 7.4.160-2.el7.x86_64

```
vsftpd 3.0.2-22                          wireshark 1.10.14-14
yp-tools 2.14-5                          ypbind 1.37.1-9
zip 3.0-11
```

Software requirements for installing Veritas Access in a VMware ESXi environment

Table 3-3 Software requirements for installing Veritas Access in a VMware ESXi environment

Item	Description
Operating system (OS)	RHEL 7.3 and 7.4 OL 7.3 and 7.4
VMware environment	VMware ESXi 5.5, 6.0 (certified versions)
IP address	Nine IPs are required for a two-node cluster with two public NICs: ■ Four IP addresses are used to configure physical IPs. ■ Four IP addresses are used to configure virtual IPs. ■ One IP address is used for the management console. ■ One IP address is used for replication.

Hardware requirements for installing Veritas Access virtual machines

Table 3-4 Hardware requirements for installing Veritas Access virtual machines

Item	Description
CPU	1 CPU – 64 bit, dual, or quad core, 2.0 GHz or later
RAM	■ 32 GB of RAM for physical servers ■ 60 GB (or more) RAM size internally available storage capacity for boot disk
Network interface card (NIC)	Four NIC cards ■ Two NIC cards for public network (minimum) ■ Two NIC cards for private network

Table 3-4 Hardware requirements for installing Veritas Access virtual machines *(continued)*

Item	Description
Fibre Channel HBA	Two-port Fibre Channel HBAs are required if you want to use shared LUNs. If the environment has only DAS disks, then the HBA requirement is optional.

Management Server Web browser support

The following are the supported Web browsers for Veritas Access:

Table 3-5

Browser	Version	Comments
Internet Explorer	■ IE 10 ■ IE 11	JavaScript: Enabled Cookies: Enabled
Firefox	Firefox 4.x and later	JavaScript: Enabled Cookies: Enabled
Google Chrome	Google Chrome 10 and later version	JavaScript: Enabled Cookies: Enabled

Additional considerations for supported Web browsers:

- Your browser must support JavaScript 1.2 or later.
- If you use pop-up blockers (including Yahoo Toolbar or Google Toolbar), either disable them or configure them to accept pop-ups from the Veritas Access node to which you connect.
- For Internet Explorer 8.0 on Windows Server 2003, download and install the hot fix from the following location:
  http://support.microsoft.com/kb/938397/en-gb
- If you are unable to download the gendeploy script using Internet Explorer 9.0, visit the following location to resolve the issue:
  http://support.microsoft.com/kb/2549423
- For Internet Explorer, enable the play animations in web pages option in the multimedia category of Advanced Internet options.
- For Internet Explorer, when popup-blocker is turned on, make sure that the filter Level is set to Medium or lower.

- For Internet Explorer, ensure that the site is included in the list of trusted sites.
- If you cannot add the site to the list of trusted sites, enable the Binary and script Behaviors option in security settings.
- You must install Adobe Flash plug-in version 10, or later.

Supported NetBackup versions

Veritas Access supports NetBackup version 7.7.3 or later.

Supported OpenStack versions

The OpenStack drivers, Cinder and Manila, are supported on the RHEL 7 OS and the OpenStack Kilo, Mitaka, Newton, or Ocata releases.

The Cinder and Manila drivers were tested with the following:

- OpenStack Kilo, Mitaka, Newton, or Ocata versions from the DevStack repository
- OpenStack RDO

Note: The Manila driver works only with kernel NFS. It does not work with NFS-Ganesha.

Supported Oracle versions and host operating systems

Veritas Access supports Oracle using Direct NFS. Veritas Access Direct NFS supports only NFS protocol version 3.

Veritas Access supports Oracle single instance only. OracleRAC is not supported.

The following are the supported Oracle versions for Veritas Access:

- Oracle version 11gR2 (11.2.0.4 or later)
- Oracle 12c (12.1.0.1)

The following are the supported Oracle host operating systems in the order of importance for Veritas Access:

- Linux
- AIX
- Solaris
- HP-UX
- Oracle Linux

Supported IP version 6 Internet standard protocol

Table 3-6 describes the IP version 6 (IPv6) Internet standard protocol.

Table 3-6 IPv6 Internet standard protocol

Description	Example format
Preferred form	ABCD:EF01:2345:6789:ABCD:EF01:2345:6789
Compressed form	FF01::101
Mixed form	0:0:0:0:0:FFFF:129.144.52.38

Network and firewall requirements

Table 3-7 displays the default ports that Veritas Access uses to transfer information.

Table 3-7 Default Veritas Access ports

Port	Protocol or Service	Purpose	Impact if blocked
21	FTP	Port where the FTP server listens for connections. **Note:** Users can configure another port if desired.	FTP features are blocked.
22	SSH	Secure access to the Veritas Access server	Veritas Access is not accessible.
25	SMTP	Sending SMTP messages.	The SMTP messages that are sent from Veritas Access are blocked.
53	DNS queries	Communication with the DNS server	Domain name mapping fails.
111	rpcbind	RPC portmapper services	RPC services fail.

Table 3-7 Default Veritas Access ports *(continued)*

Port	Protocol or Service	Purpose	Impact if blocked
123	NTP	Communication with the NTP server	Server clocks are not synchronized across the cluster. NTP-reliant features (such as DAR) are not available.
139	CIFS	CIFS client to server communication	CIFS clients cannot access the Veritas Access cluster
161	SNMP	Sending SNMP alerts	SNMP alerts cannot be broadcast.
445	CIFS	CIFS client to server communication	CIFS clients cannot access the Veritas Access cluster.
514	syslog	Logging program messages	Syslog messages are not recorded.
756, 757, 755	statd	NFS statd port	NFS v3 protocol cannot function correctly.
2049	NFS	NFS client to server communication	NFS clients cannot access the Veritas Access cluster.
3172, 3173	ServerView	ServerView port	ServerView cannot work.
4001	mountd	NFS mount protocol	NFS clients cannot mount file systems in the Veritas Access cluster.
4045	lockd	Processes the lock requests	File locking services are not available.
5634	HTTPS	Management Server connectivity	Web GUI may not be accessible.

Table 3-7 Default Veritas Access ports *(continued)*

Port	Protocol or Service	Purpose	Impact if blocked
56987	Replication	File synchronization, Veritas Access replication	Veritas Access replication daemon is blocked. Replication cannot work.
8088	REST server	REST client to server communication	REST client cannot access REST API of Veritas Access.
8143	S3	Data port for Veritas Access S3 server	User cannot use the Veritas Access object server.
8144	ObjectAccess service	Administration port for Veritas Access S3 server.	User cannot create access or secret keys for using the ObjectAccess service.
11211	Memcached port	CLISH framework	CLISH cannot function correctly, and cluster configuration may get corrupted.
30000:40000	FTP	FTP passive port	FTP passive mode fails.
14161	HTTPS	Access Veritas Access GUI	User cannot access the Veritas Access GUI.

NetBackup ports

NetBackup uses TCP/IP connections to communicate between one or more TCP/IP ports. Depending on the type of operation and configuration on the environment, different ports are required to enable the connections. NetBackup has different requirements for operations such as backup, restore, and administration.

Table 3-8 shows some of the most-common TCP and UDP ports that Veritas Access NetBackup uses to transfer information. For more information, see the *Veritas NetBackup Security and Encryption Guide*.

Table 3-8 Default NetBackup TCP and UDP ports

Port Range	Protocol
1556	TCP, UDP
13701-13702, 13705-13706	TCP
13711, 13713, 13715-13717, 13719	TCP
13720-13722	TCP, UDP
13723	TCP
13724	TCP, UDP
13782-13783	TCP, UDP
13785	TCP

OpenDedup ports and disabling the iptable rules

This use case is specific to running OpenDedup on Veritas Access. Each time a SDFS volume is created and mounted on Veritas Access, it starts listening on a specific port. Initially, it starts with port 6442 and goes on incrementing +1 for further subsequent volumes.

Table 3-9 OpenDedup ports

Port Range	Protocol or Service	Purpose	Impact if Blocked
Starts from 6442 and increments +1 for subsequent volumes	TCP	Allows communication between Veritas Access and OpenDedup	Veritas Access cannot communicate with OpenDedup

To allow communication to the OpenDedup port running on Veritas Access, disable the iptable rules completely

1. Use the `df` command to show that the SDFS volume is mounted and on which port it is listening.

   The SDFS volume is already mounted as part of the LTR script.

   ```
   [root@ltrclust_02 ~]# df -h | tail -2
   sdfs:/etc/sdfs/pool100-volume-cfg.xml:6442
   11G     0   11G   0% /pool100
   ```

2. Use the `netstat` command to verify that the port is open.

   ```
   [root@ltrclust_02 ~]# netstat -tulpn | grep 6442
   tcp        0      0 :::6442   :::*       LISTEN
   3761/jsvc.exec
   ```

3. Disable the `iptable` rules to allow communication to the OpenDedup port once the volume is mounted and to disallow traffic to this port once the volume is unmounted.

   Use the following commands to disable the `iptable` rules:

   ```
   [root@ltrclust_02 ~]# iptables -F

   [root@ltrclust_02 ~]# /etc/init.d/iptables stop

   [root@ltrclust_02 ~]# iptables -L
   ```

   Use the `iptables -L` command to verify that all the `iptable` rules are disabled.

   The `iptable` rules should be run on all the Veritas Access cluster nodes and on the NetBackup media server if OpenDedup is installed on it.

4. An alternative to disabling the `iptable` rules in Step 3 is to add an `iptable` rule to open the OpenDedup port, so that the existing `iptable` rules are also used.

   Example:

   ```
   [root@ltrclust_02 ~]# iptables -A INPUT -p tcp --dport 6442 -j ACCEPT
   ```

CIFS protocols and firewall ports

For the CIFS service to work properly in an Active Directory (AD) domain environment, the following protocols and firewall ports need be allowed or opened

to enable the CIFS server to communicate smoothly with Active Directory Domain Controllers and Windows/CIFS clients.

Internet Control Message Protocol (ICMP) protocol must be allowed through the firewall from the CIFS server to the domain controllers. Enable "Allow incoming echo request" is required for running the CIFS service.

Table 3-10 lists additional CIFS ports and protocols.

Table 3-10 Additional CIFS ports and protocols

Port	Protocol	Purpose
53	TCP, UDP	DNS
88	TCP, UDP	Kerberos
139	TCP	DFSN, NetBIOS Session Service, NetLog
445	TCP, UDP	SMB, CIFS, SMB2, DFSN, LSARPC, NbtSS, NetLogonR, SamR, SrvSvc
464	TCP, UDP	Kerberos change or set a password
3268	TCP	LDAP GC
4379	TCP	CTDB in CIFS

Table 3-11 lists the ports that are required for LDAP with SSL.

Table 3-11 LDAP with SSL ports

Port	Protocol	Purpose
636	TCP	LDAP SSL
3269	TCP	LDAP GC SSL

Maximum configuration limits

The maximum configuration limits for configuring the Veritas Access system software are as follows:

Table 3-12 Maximum configuration limits

Veritas Access system software	Configuration limit
File system size	5 PB for a non-scale-out file system without cloud tiering support. 3 PB for a scale-out file system with cloud tiering support.
Veritas Access nodes	20
Supported LUNs	The maximum number of disks is theoretically limited to the number that can be attached to the operating system. However, it has only been tested in the thousands.
Supported file systems	500
Tiers within a file system	2 (primary tier and secondary tier)

Chapter 4

Preparing to install Veritas Access

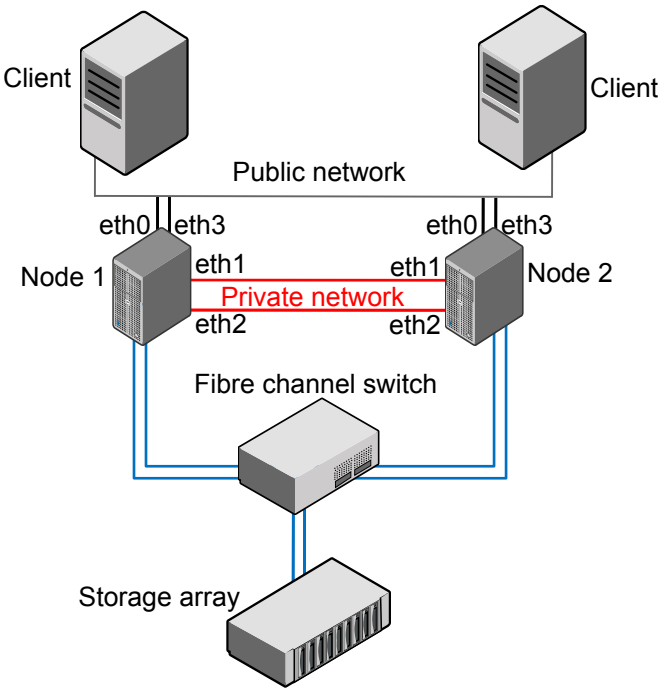
This chapter includes the following topics:

Overview of the installation process

The Veritas Access cluster is a set of connected servers called "nodes." Together these nodes form a unified entity called a cluster.

Figure 4-1 shows an example of an Veritas Access cluster.

Figure 4-1 Sample of Veritas Access cluster overview



Note: The NIC names mentioned in Figure 4-1 are only for examples. You need to determine the actual names of your NICs during the installation.

An overview of the Veritas Access software installation includes the following steps:

- Gather network information from your network administrator.
- Connect your network hardware.
- Install the operating system on each of the nodes.
- Install Veritas Access on the node. If the driver node is one of the nodes of the cluster, you must start the installer from the console of the node. If the driver node is not part of the cluster, the installer can be run from the driver node to install and configure the cluster over an SSH connection.
  From the Veritas Access 7.2 release, the installer can be run from any node of the cluster.
  See "Installing and configuring the Veritas Access software on the cluster" on page 62.
  See "About the driver node" on page 58.

- Run the installation and configuration on the node to configure the entire cluster. Installation times vary depending on your configuration.

Hardware requirements for the nodes

The following table summarizes the hardware requirements for each node.

Table 4-1 Hardware requirements for the nodes

Item	Requirements
Network interface card (NIC)	At least four NICs are required for each node. Two NICs connected to a private network. ■ For a two-node cluster, either cross connect two private NICs on each node or use a switch. ■ If the cluster has more than two nodes, make sure that you have a dedicated switch. This switch can be a public or a private switch with a dedicated VLAN. Make sure that all the private NICs are connected to the switch. Connect two public NICs from each node to the public network. The gateway must be reachable to each public NIC.
IP address	For a two-node cluster, make sure that you have nine IP addresses available. ■ Four IP addresses are used to configure physical IPs. ■ Four IP addresses are used to configure virtual IPs. ■ One IP address is used to configure the Operations Manager console. ■ One IP address is used for replication, which is optional. Make sure that these nine IP addresses are different from the IP addresses that are assigned to the target cluster nodes to install Veritas Access over the Secure Shell (SSH).

Connecting the network hardware

Before you install the Veritas Access software, you must assemble a cluster by configuring all the nodes with the required network hardware, and connecting the Ethernet interfaces to the private and the public networks.

To assemble the cluster, do the following:

- Determine a preferred location for the cluster.

- Make sure that each node has at least two redundant Ethernet interfaces (gigabit Ethernet) to connect to a private network for cluster internal control.
- Make sure that each node has at least two additional Ethernet interfaces (gigabit Ethernet) to connect to the public network. You can use the public Ethernet interfaces from the embedded interfaces on the motherboard or from the add-on (PCI) network adapter interfaces.
- To connect the public NICs, connect one end of the Ethernet cables to the Ethernet interfaces on the back of the nodes. Connect the other end of the Ethernet cables to your corporate network so that they can reach the gateway. At least two public interfaces are required for each node.
- To connect the private NICs, use the first two available NICs when sorted by NIC name. Available NICs are those not connected to the public network or excluded from the node.

  For example, if your NICs are eth1, eth2, eth3, and eth4, and none of the NICs are connected to the public network or excluded, then use eth1 and eth2 as the private NICs.

  Connect one end of the Ethernet cables to Ethernet interface 1 and 2 on the back of the nodes. For a 2-node cluster, connect the other end of the Ethernet cables to the corresponding Ethernet interfaces on the second node. For a cluster with more than 2 nodes, connect the other end of the Ethernet cables to a dedicated switch or VLAN.
- Ask your network administrator for the IP addresses to use in the Veritas Access installation. The number of IP addresses you need depends on the number of nodes and number of network interface cards in your cluster.

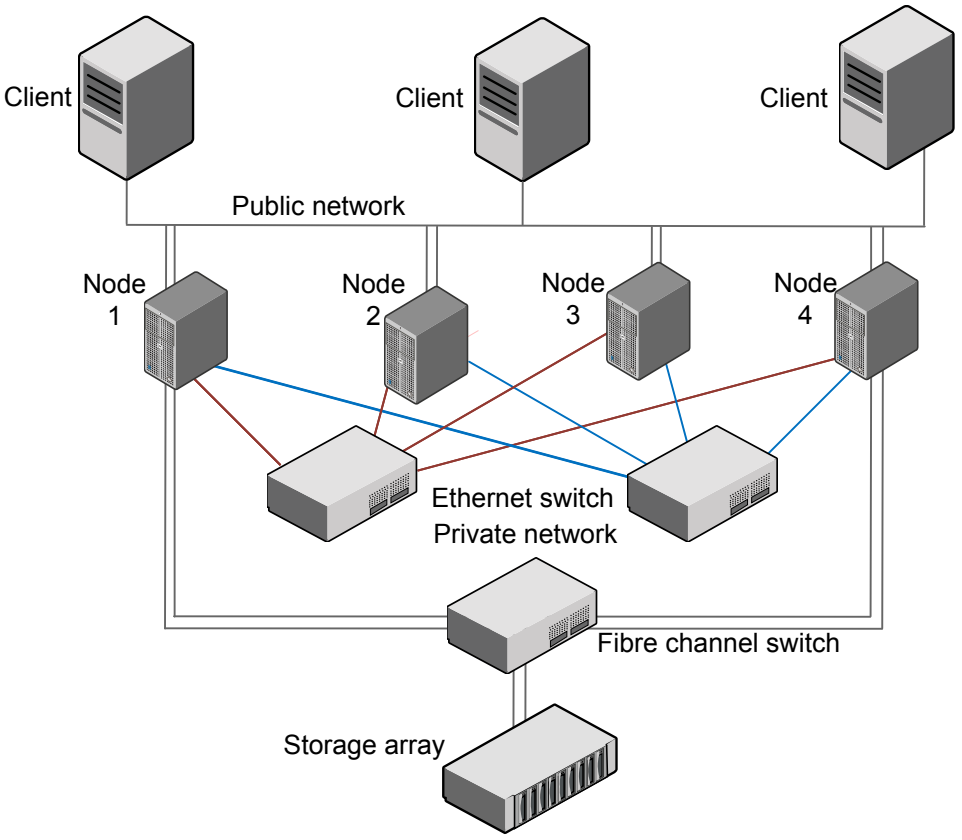
  You need at least one IP address per node per public interface. For virtual IP addresses, you can configure the virtual IP addresses later in the CLISH if you input 0 for the number of virtual IP addresses per NIC during installation time.

  Veritas Access supports both Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6), but they cannot be mixed.

Physical IP address	An IP address that is associated with a specific Ethernet interface address and cannot automatically be failed over.
Virtual IP address (VIP)	An IP address whose association to a specific Ethernet interface (VIP) can be failed over to other interfaces on other nodes by the Veritas Access software.
Console IP address	A dedicated virtual IP address that is used to communicate with the Veritas Access cluster Management Console. This virtual IP address is assigned to the master node. If the master node fails, the Veritas Access software automatically selects a new master node from the cluster and fails the console IP address over to it.

Figure 4-2 shows a diagram of a four-node cluster.

Figure 4-2 Private network configurations: four-node cluster



Note: Two or more Veritas Access private networks cannot be configured on the same IPv4 network.

About obtaining IP addresses

The Veritas Access installation process lets you configure IP addresses for 1 to 20 nodes. The default is two nodes.

Note: You can configure either IPv4 addresses or IPv6 addresses (depending on what you use when installing Veritas Access), but not both. Do not use IP addresses starting with 172.16.X.X either as physical IP addresses or virtual IP addresses since this range of IP addresses are used for the private network.

You need to obtain physical IP addresses, virtual IP addresses, and a netmask for the chosen public network from the network administrator in charge of the facility where the cluster is located. All IP addresses (both physical and virtual) must be part of the same subnet and use the same netmask as the node's access IP.

By design, the installer does not support the use of the localhost (127.0.0.1) IP address during installation

Note: Netmask is used for IPv4 addresses. Prefix is used for IPv6 addresses. Accepted ranges for prefixes are 0-128 (integers) for IPv6 addresses.

The information you obtained from the network administrator is used to configure the following:

- Physical IP addresses
- Virtual IP addresses
- Console IP address
- Replication IP address (optional)
- IP address for the default gateway
- IP address for the Domain Name System (DNS) server
- DNS domain name
- IP address for the Network Time Protocol (NTP) server (optional)
- Virtual IP address for Veritas NetBackup (optional)

About calculating IP address requirements

This section provides an example of how to calculate IP addresses for a two-node cluster. In this example, all the nodes in the cluster have the same hardware configuration. Therefore, the number of network interface cards (NICs) is the same for all the nodes in the cluster.

- Two private NICs and two public NICs should be connected to respective networks.

- One public IP address should be assigned to one of the public interfaces for installation over SSH. None of the private interfaces should have the IP address in the same network segment.
- The public IP address must be made permanent by writing it to the network configuration file `/etc/sysconfig/network-scripts/ifcfg-ethX`.

Table 4-2 Example calculation of required IPs for a standard configuration

Number of IPs	Item
2	Number of nodes in the cluster
4	Number of interfaces on each node
2	Number of the private interfaces that are required for each node

After two private interfaces on each node are selected, all the remaining interfaces act as public interfaces.

To calculate the number of public interfaces per node

- Use the following to calculate the number of public interfaces that are required per node.

```
Total number of interfaces (4)
- Number of private interfaces (2)
= Number of public interfaces

4 - 2 = 2
```

To calculate the physical and the virtual IP addresses for the cluster

1 Use the following to calculate the number of physical IP addresses that are required for the cluster installation.

```
Total number of nodes (2)
x Number of public interfaces per node (2)
= Total number of physical IP addresses

= 2 x 2 = 4
```

2 Use the following to calculate the number of virtual IP addresses that are required for the cluster installation.

```
Total number of nodes (2)
x Number of public interfaces per node (2)
= Total number of virtual IP addresses

= 2 x 2 = 4
```

3 The number of IP addresses required for the Veritas Access Operations Manager is equal to one (1).

To calculate the total number of public IP addresses for the cluster

- Use the following to calculate the number of public IP addresses that are required for the cluster.

```
Total number of physical IP addresses/cluster (4)
+ Total number of virtual IP addresses/cluster (4)
+ Number of IP addresses for the Management Console (1)
= Total number of public IP addresses required for the cluster

= 4 + 4 + 1 = 9
```

To request and specify IP addresses

- Request the Network Administrator for the public IP addresses.

  For example, if the Network Administrator provides you with IP addresses `10.209.105.120` through `10.209.105.128`, you can allocate the resources in the following manner:

  ```
  Start of Physical IP address: 10.209.105.120
  Start of Virtual IP address: 10.209.105.124
  Management Console IP:"10.209.105.128"
  ```

  This entry gives you four physical IP addresses (`10.209.105.120` to `10.209.105.123`), four virtual IP addresses (`10.209.105.124` to `10.209.105.127`), and one IP address for the Operations Manager (`10.209.105.128`).

  `10.209.105.120` and `10.209.105.121` are assigned to pubeth0 and pubeth1 as physical IP addresses on the first node.

  `10.209.105.122` and `10.209.105.123` are assigned to pubeth0 and pubeth1 as physical IP addresses on the second node.

  `10.209.105.124` to `10.209.105.127` are assigned to pubeth0 and pubeth1 as virtual IP addresses on the two nodes.

Reducing the number of IP addresses required at installation time

You can reduce the number of IP addresses required at installation time by not configuring any virtual IP addresses. During the Veritas Access installation, input 0 for the number of virtual IP addresses per NIC.

Virtual IP addresses are not required at installation time. You can configure the virtual IP addresses later using the `Network> ip addr add` command in the CLISH.

See the `network(1)` manual page for more information on adding NICs.

You need at least one IP address per node per public interface at installation time.

Table 4-3 Example configuration of required IP addresses at installation time for a two-node cluster with two public NICs per node

Number of IP addresses	Item
4	Number of physical IP addresses. The four IP addresses include the original physical IP addresses.
1	One IP address for the management console.

About checking the storage configuration

Warning: Do not connect the Fibre Channel HBAs until you finish installing the operating system. If the local disks are bad, connecting the Fibre Channel HBAs prevents the operating system from being installed on the local disks. Because the disk is scanned, it takes longer to install the software on a local disk.

Veritas Access supports Flexible Storage Sharing (FSS), which allows the users to configure and manage direct-attached storage on the Veritas Access appliance.

After you install the operating system, check the storage configuration. If you don't want to use FSS, make sure that each node has the following:

- One or two Fibre Channel Host Bus Adapters (HBAs) for connection to the Storage Area Network (SAN) switch.
  Two Fibre Channel HBAs are recommended, but only one is required. Having only one Fibre Channel HBA enables all the operations of the Fibre Channel (except high availability).
- An internal boot disk. Make sure that one is in place before you install the Veritas Access software.

If you want to use FSS, make sure that each node has attached at least two extra local data disks besides the internal boot disk.

Chapter 5

Deploying virtual machines in VMware ESXi for Veritas Access installation

This chapter includes the following topics:

Setting up networking in VMware ESXi

Before you start, install the ESXi server. You can deploy the first virtual machine on your ESXi host by using the vSphere Client.

To set up a network in VMware ESXi

1. Start the vSphere Client and type the logon details for your host.

   In the **IP address / Hostname** text box, enter the ESXi server IP or host name.

   In the **User name** text box, type **root**.

   In the **Password** text box, type **my_esxi_password**.
2. Set up the networking requirements for Veritas Access.
3. To set up the public network virtual switch:
   - In the **Configuration** tab of the ESXi host, navigate to **Hardware > Networking**.
   - Click **Add Networking** on the top right corner.

- Select the connection type as **Virtual Machine** and click **Next**.
- Select the NIC that is connected to the public network under the **Create a virtual switch** section.
- Enter the appropriate network label for the public virtual switch.
- Verify the summary and click **Finish** to create the public network virtual switch.

Note: If you want to create multiple public network switches, repeat the preceding steps.

---

4 To set up the private network virtual switch:

- In the **Configuration** tab of the ESXi host, navigate to **Hardware > Networking**.
- Click **Add Networking** on the top right corner.
- Select the connection type as **Virtual Machine** and click **Next**.
- Deselect any NIC that is selected by default for creating the virtual switch.
- Enter the appropriate network label for the private virtual switch.
- Verify that the summary shows no-adapters under the physical adapters, and click **Finish** to create the first private network virtual switch.

Note: If you want to create the second private network virtual switch, repeat the preceding steps.

Creating a datastore for the boot disk and LUNs

To create a datastore for the boot disk and LUNs

1 Create a datastore for vmdk files for virtual machines.

2 In the **Configuration** tab of the ESX host, navigate to **Hardware > Storage**.

3 Click **Add Storage** on the top right corner.

4 Select the storage type as **Disk/LUN** and click **Next**.

5 Select the disk that you want to use to create the virtual machine vmdk files.

6 Review the current disk layout and click **Next**.

7 Enter the datastore name of your choice and click **Next**.

8 Select the disk space that you want to dedicate for the datastore. The default option is to use the complete list.

9 Review the details and click **Finish**.

Creating a virtual machine for Veritas Access installation

To create a virtual machine for Veritas Access installation

1 After the networking configuration is complete and the datastore is defined, create the virtual machines.

- Select the **ESXi host IP/hostname** in the top of the tree structure in the leftmost frame.
- From the file menu, select **New Virtual Machine**, which opens a pop-up for creating the virtual machine.
- Select the configuration as **Custom** and click **Next** to decide on the exact configuration of the virtual machine.
- Enter the virtual machine name of your choice and click **Next**.
- Select the datastore that stores the virtual machine `vmdk` file and click **Next**.
- Select the virtual machine version that you want to use and click **Next**. Veritas recommends version 8.
- Select the guest operating system as **Linux** and version as **Red Hat Enterprise Linux 6 or 7 (64-bit)** and click **Next**.

Select the number of CPUs. Veritas recommends eight cores that can be:

- Two virtual sockets and four cores per virtual socket.
- One virtual socket and eight cores per virtual socket.
- Any higher number of cores as per your workload.
  Select the memory configuration. Veritas recommends 32 GB.
- In the network configuration, it is recommended to select the number of NICs as four.
  For NIC1, select the public network virtual switch and validate that the adapter is correct.
  For NIC2, select the public network virtual switch and validate that the adapter is correct.
  For NIC3, select the private network virtual switch 1 and validate that the adapter is correct.

For NIC4, select the private network virtual switch 2 and validate that the adapter is correct.

- Select the SCSI controller as **VMware Paravirtual**.
- In the disk configuration page, select **Create a new virtual disk** and click **Next**.
- Select the boot disk size. Veritas recommends 100 GB.
- Select the disk provisioning type as **Thick Provision Eager Zeroed**.
- Select the datastore as **Specify a data store or data store cluster** and click **Next**.
  After selecting the datastore, click **Next**.
- Select the **Virtual device node** as default (SCSI (0:0) for the boot disk) and click **Next**.
- Review the virtual machine configuration and click **Finish** to create the virtual machine.
  The virtual machine creation task is complete.

2 Select the virtual machine and click **Edit virtual machine settings** to validate the following:

- There should be four network adapters - two for the public network and two for the private network.
- Verify that the memory and CPU configuration is correct.

3 Repeat Step 1 and Step 2 to create the second virtual machine, which is used to form the two-node Veritas Access cluster.

4 Add LUNs/DAS disks to the virtual machines.

To add local DAS disks:

- Select the virtual machine and click **Edit virtual machine** settings.
- Click the **Add** button.
- Select the device type as **Hard Disk** and click **Next**.
- Select **Create a new virtual disk** in the type of disk and click **Next**.
- Select the DAS disk size. Veritas recommends 100 GB.
- Select the disk provisioning type as **Thick Provision Eager Zeroed**.
- Select the datastore as **Specify a data store or data store cluster** and click **Next**.
- Select the **Virtual device node** as SCSI (1:0) for the first SAS disk and click **Next**.

Once all the required DAS disk creation is complete, complete the following:

- Select the SCSI controller 1, which is used for DAS disks.
- Set the SCSI Bus sharing mode as **Virtual.**
  This mode is required so that DAS disks are claimed in VxVM enclosure-based naming (EBN) mode and host name is only prefixed by VxVM when disks are in EBN mode, which distinguishes it from the shared LUNs present in the arrays.

- Click **OK** to create the DAS disk.
  Repeat this step for creating the DAS disk for other Veritas Access nodes.

5 Map the shared disks to the LUNs. Mapping of LUNs from an array is only supported using Raw Device Mapping (RDM) mode.

Mapping shared LUNs to the first virtual machine:

- Select the first virtual machine and click **Edit virtual machine settings**.
- Click the **Add** button.
- Select the device type as **Hard Disk** and click **Next**.
- Select the LUN that you want to map and click **Next**.
- Select the datastore that stores the LUN mapping or select **Store with virtual machine**.
- Select the compatibility mode as **Physical** to access the array LUN hardware directly.
- Select the **Virtual device node** as SCSI (2:0) for the shared disk and click **Next**.
- Review the mapping of the disk and click **Finish** to map the array LUN disk to the virtual machine.
  Repeat this Step for the number of LUNs that you want to map and update the **Virtual device node** to the next free SCSI controller port.

Once all the required LUNs are mapped, complete the following:

- Select the SCSI controller 2, which is used for shared LUNs.
- Set the SCSI Bus sharing mode as **Virtual**.
  This mode is required so that the shared LUNs are claimed in VxVM enclosure-based naming (EBN) mode. This distinguishes it from the shared LUNs present in the arrays.
- Click **OK** to complete the mapping of LUNs in RDM mode.

Mapping shared LUNs to the second virtual machine:

- Select the first virtual machine and click **Edit virtual machine settings**.
- Click the **Add** button.
- Select the device type as **Hard Disk** and click **Next**.
- Select **Use an existing Virtual Disk** in the type of disk and click **Next**.
- Navigate to the corresponding disk path in the datastore where the shared disk was stored when they were mapped to the first virtual machine.
- Select the **Virtual device node** as SCSI (2:0) for the shared disk and click **Next**. Ensure that the sequence of disk mapping is the same as that of the first virtual machine and mapping has been done to the same SCSI controller to achieve a shared disk configuration.
- Review the mapping of the disk and click **Finish** to map the array LUN disk to the virtual machine.
  Repeat this Step for the number of shared LUNs that you have mapped to other virtual machines and update the **Virtual device node** to the next free SCSI controller port.

Once all the required LUNs are mapped, complete the following:

- Select the SCSI controller 2, which is used for the shared LUNs.
- Set the SCSI Bus sharing mode as **Virtual**.
  This mode is required so that the shared LUNs are claimed in VxVM enclosure-based naming (EBN) mode. This distinguishes it from the shared LUNs present in the arrays.
- Click **OK** to complete the mapping of LUNs in RDM mode.
  The networking and storage configuration is complete for the virtual machines.

6 Install the RHEL 7 Update 3 or 4 (64-bit) operating system that is supported by the Veritas Access installer.

See "Installing the operating system on the target Veritas Access cluster" on page 59.

Note: The virtual machine can have DAS disks, shared LUNs, or both of them. For the erasure coded file system, the disks should be DAS only.

Chapter 6

Installing and configuring a cluster

This chapter includes the following topics:

Installation overview

You can install the Veritas Access on a cluster. You can add a minimum of one-node and a maximum of 20 nodes to the cluster. By adding a single node or multiple

nodes to the cluster, you can make the system fault-tolerant and scale it up as required.

Summary of the installation steps

The Veritas Access software installation consists of two main pieces:

- Operating system installation.
  Veritas Access requires Red Hat Enterprise Linux.
  See See "System requirements" on page 18.
- Veritas Access software installation.

Table 6-1 provides a brief summary of the installation steps. The summary includes cross references to where you can find more information about each task.

Table 6-1 Summary of installation steps

Task	Steps	For more information
Task 1: Install the operating system on each node of the cluster.	Steps include: ■ Automatic system discovery of USB devices, hard disk controllers, and so on. ■ Select the installation device. ■ Set the clock and the time zone. ■ System preparation for automated installation. ■ Manual disk partitioning. ■ Minimal Automatic package installation. ■ Install the Red Hat Enterprise Linux kernel update.	See "Installing and configuring the Veritas Access software on the cluster" on page 62.

Table 6-1 Summary of installation steps *(continued)*

Task	Steps	For more information
Task 2: Install the Veritas Access software on the cluster.	Steps include: ■ Install the required Red Hat Enterprise Linux operating system RPMs. If yum is configured, then the installer helps to install the required RPMs during the precheck. ■ Extract the Veritas Access tar file and run the installer. ■ Enter network configuration information (cluster name, IP addresses, DNS information, and so on). ■ Verify installation on the node.	See "Installing and configuring the Veritas Access software on the cluster" on page 62.

Before you install

Before you install the Veritas Access software:

- Make sure that the names of all the public and private interface which are targeted to be part of access configuration during installation should be same across all the Veritas Access cluster nodes on which you want to install Veritas Access.
- If the system has a network interface bond before you install Veritas Access, you need to specify the bond names as bond0, bond1, bond2 and so on, and then you can start the Veritas Access installation.
- Before you install Veritas Access, if the system has VLAN configured network interface on it and if the user want to configure it on the Veritas Access nodes, then the interface name on which VLAN is configured must follow the naming format as follows:
  interface_name.vlan_id
  For example, eth0.100 or ens168.101.
- Make sure that no DHCP servers are running in the private network.
- Disable the USB Ethernet interface in BIOS for all the nodes in the cluster.

- Make sure that there are at least two private connections between the cluster and the public connection. The public connected interface must be reachable to the gateway.
- Connect DAS or SAN storage to the cluster node. If you are configuring the fencing feature to avoid the split-brain condition, make sure that the SAN disks are SCSI3-compliant.
- Assign one public IP address to each node in a cluster. This IP address is used by the installer, and it is reused as one of the public interface physical IP addresses after configuration.
- Make sure that the assigned IP is persistent after a reboot. To make the IP persistent, do the following changes in
  `/etc/sysconfig/network-scripts/ifcfg-XX`
  For example:

```
TYPE=Ethernet
HWADDR=00:50:56:3d:f1:3e
DEVICE=eth2
BOOTPROTO=none
IPADDR=10.200.56.214
NETMASK=255.255.252.0
NM_CONTROLLED=no
ONBOOT=yes
```

Note: Make sure that you have enough public IPs required to install the Veritas Access software.

Installing the operating system on each node of the cluster

Before you install the Veritas Access software, you must install the RHEL OS and kernel version.

To install the RHEL operating system on each node of the cluster

1 **Prerequisites:**

Make sure that your system meets the requirements.

2 Go to the following website and download the required RHEL OS and kernel version, and then install them.

https://access.redhat.com/downloads/

About the driver node

If you do not plan to install Veritas Access from the console of the nodes in the cluster (the local management console of your nodes), you need another server that is not a target node in the Veritas Access cluster to use in the Veritas Access installation. This server is called the driver node.

When you run the Veritas Access installation script, the Veritas Access installer helps set up the SSH connection between the driver node and the target Veritas Access cluster nodes.

The driver node platform can be: RHEL 7, SLES 11 SP2, or SLES 11 SP3.

The following table provides the information about Veritas Access installation support from the cluster node and the driver node with different types of network devices.

Network device type	Driver Node	Cluster Node
Normal network device	Yes	Yes
Create a bond device through installer and add NIC in bond through which installation is started.	Yes	No
Create a bond device on NIC other than the NIC through which installation is started.	Yes	Yes
Create a VLAN through installer on the NIC other than the NIC through which installation is started.	No	Yes
Create a VLAN through installer on NIC through which installation is started.	No	No
Exclude NIC from which installation started.	No	No
Create a bond and a VLAN over bond device on NIC other than the NIC through which installation is started.	No	Yes
Preconfigured bond as public and Installation from other NIC.	Yes	Yes

Network device type	Driver Node	Cluster Node
Create a bond through installer and select it as private connection.	No	No
Create VLAN through installer and select it as private connection.	No	No
Install with public NIC and pre-existing public bond.	Yes	Yes

Installing the operating system on the target Veritas Access cluster

This first task in the installation process is to install the Red Hat Enterprise Linux operating system on each node of the cluster.

To install the operating system

1. Insert the Red Hat Enterprise Linux 7.4 operating system installation DVD, and boot the server from the DVD.

   See "Linux requirements" on page 20.

   You can also use an external USB DVD-ROM.
2. Select **Install Red Hat Enterprise Linux 7.4**.
3. After the system loads, select **English** as a language for installation, and then click **Continue**.

   After the installer displays installation summary, you can customize the installation process.
4. Click **Date & Time**, choose your system location from the provided map, and then click **Done** to apply the configuration.
5. Select **English** language for the Language System Support and the Keyboard language.
6. To select your system software, click **Software Selection** and select a base installation environment from the list.
7. Select **Minimal Install** with **Compatibility Libraries Add-ons**, and then click **Done** to apply these changes to the installation process.
8. Select a disk and perform disk partition manually to configure the system partitions.

   See "System requirements" on page 18.

9 Click **Network & Hostname** and provide a system host name to set up your network connection.

After you set up the host name, set the **Ethernet** to **On** to bring your network interface up. Click **Configure** and provide your static network settings for your appropriate network connection.

10 After you finish editing the **Ethernet Interface Settings**, click **Done**.

The default installer window is displayed.

11 Verify the installation settings, and then click **Begin Installation** to start the system installation.

12 Click **Root Password**, enter the password, and then click **Done**.

After the installation is finished, the installer displays details of the successful installation.

13 Your Red Hat Enterprise Linux installation is now complete. You can follow the same steps that are shown in this section to install the operating system on other nodes of the cluster.

See the *Red Hat Enterprise Linux documentation* for the detailed procedure.

Installing the Oracle Linux operating system on the target Veritas Access cluster

This first task in the installation process is to install the Oracle Linux (OL) operating system on each node of the cluster.

To install the OL operating system

1 Insert the OL operating system installation DVD, and start the server from the DVD, and press the **Enter** key.

See "Linux requirements" on page 20.

You can also use an external USB DVD-ROM.

2 Press the **Tab** key to move focus to the **Skip** key, then press the **Enter** key to continue.

3 On the **Welcome** screen, click the **Next** option.

4 Select **English** as the language, and then click the **Next** option.

5 Select **U.S. English** as the keyboard setting, and then click the **Next** option.

6 Select the **Basic Storage Devices** option, and then click the **Next** option.

7 Enter a fully qualified host name, then click the **Configure Network** option.

8 Highlight the relevant connection and click the **Edit** option.

9 Select the **Connect automatically** check box. If you are not using DHCP, click on the **IPv4 Settings** tab, set the method to **Manual**, click the **Add** option, and enter the appropriate network details. When you are satisfied with the details, click the **Apply** and **Close** options to return to the host name screen, and then click the **Forward** option.

10 Select the appropriate time zone by clicking on your nearest city on the map. Click the **Next** option.

11 Enter a root password for the server and click on **Next**.

12 Select the partitioning type you require. If you want to amend the default partitioning layout, select the **Review and modify partitioning layout** option. Click the **Next** option.

13 The OEL installer lists the default partitioning scheme for your size disk. Amend them as required and click the **Next** option. Click the **Format** and **Write changes to disk** options.

14 Accept the boot loader settings by clicking the **Next** option.

15 Select the **Minimal Install** and the **Oracle Linux Server** options, and then click the **Next** option.

16 Wait for the installation to complete.

17 Click the **Reboot** option to complete the installation.

18 Veritas Access supports only Red Hat Enterprise Linux compatible kernels. Obtain the Red Hat Enterprise Linux compatible kernels directly after the OEL installation.

Installing Veritas Access on the target cluster nodes

Before you install Veritas Access on the target cluster nodes, you must allocate enough IP addresses for the cluster. You can install up to a 20-node cluster.

Installing the cluster is a one-time activity. It takes about 40 minutes to install a two-node cluster. Installation times may vary depending on your configuration and the number of nodes.

See “About obtaining IP addresses” on page 42.

If you want to install the Veritas Access cluster with IPv6 IP addresses, you need to configure a static IPv6 address on the driver node and on all the nodes of the cluster. You need to make sure that the IPv6 IP auto-assignment is disabled on all

nodes of the cluster. You can then use the IPv6 IPs to install Veritas Access on the cluster nodes.

- If you want to configure the cluster in an IPv6 environment or plan to use the cluster in mixed mode that is, configure both IPv4 and IPv6 IPs, you need to disable the IPv6 IP auto-assignment.
  - To disable the IPv6 IP auto-assignment, add the following entries in the `/etc/sysctl.conf` file for all network interfaces of the node.

    ```
    net.ipv6.conf.<network interface name>.autoconf=0
    net.ipv6.conf.<network interface name>.accept_ra=0
    net.ipv6.conf.<network interface name>.accept_ra_defrtr=0
    ```

To configure a static IPv6 address

1. Modify the `vim /etc/sysconfig/network-scripts/ifcfg-ens161` network interface file by using the following:

   ```
   IPV6INIT="yes"
   IPV6_AUTOCONF="no"
   IPV6ADDR=2001:128:f0a2:900a::121/64
   ```

2. Restart the network service by using the`systemctl restart network` command.

Installing and configuring the Veritas Access software on the cluster

To install and configure the cluster

Note: During the installation, the installer log is located at `/var/tmp`.

---

1. Enter one of the following commands to start the installation.

   ```
   # ./installaccess node1_ip node2_ip
   ```

   Where *node1_ip* and *node2_ip* are the public physical IP addresses that are already assigned to the target cluster nodes to install Veritas Access over SSH.

   These are the current IPs assigned to the nodes for the installation communication.

   The example is used to install two nodes. To install another target node cluster, add *node3_ip* to the command line that is used in this step.

2. The installer checks for the operating system dependencies and automatically installs the required OS RPMs. If the OS RPMs' dependencies are not sorted, the RedHat subscription manager user ID and password are required.

3. The installer installs the Veritas Access RPMs.

4. Choose the licensing method. Answer the licensing questions and follow the prompts.

   ```
   1)  Enter a valid perpetual or subscription license key file
   2)  Register with evaluation mode and complete system licensing later

   How would you like to license the systems? [1-2,q,?] (2)
   ```

5 The installer displays the firewall ports to be opened after the configuration, and asks if you want to open them:

```
Veritas Access needs to open the following ports:
111 Rpcbind (NFS)
11211 Memcached Port
123 NTP Service
139 CIFS Service
14161 GUI
161 SNMP Service
2049 NFS Service
21 FTP Port
22 SSH Service
25 SMTP Port
30000:40000 FTP Passive Port Range
3172,3173 Server View Ports
4001 Mountd (NFS)
4045 NLM (NFS)
4379 CTDB Port
445 CIFS TCP Service
514 Syslog Service
53 DNS Service
5634 VIOM
56987 Replication Service
756,757,755 Statd (NFS)
8088 REST Server
8143 Object Access Gateway
8144 Object Access Admin Gateway
Do you want to proceed? [y,n,q] (y)
```

6 The installer asks the following information to configure the cluster:

```
Enter the cluster name: [q,?]
Do you want to rename the hosts' name like vac_01, vac_02? [y,n,q,b,?] (n)
Enter the public IP starting address or : [b,q,?]
Enter the netmask for the public IP address: [b,q,?] (255.255.255.0)
Enter the number of VIPs per interface: [b,q,?] (0) 1
Enter the virtual IP starting address: [b,q,?]
Enter the default gateway IP address: [b,q,?]
Enter the DNS IP address: [b,q,?] (10.0.2.3)
Enter the DNS domain name: [b,q,?] (community.veritas.com)
Enter the console virtual IP address: [b,q,?]
Do you want to use the separate console port? [y,n,q,b,?] (n)
Enter the private IP starting address: [b,q,?] (172.16.0.3)
```

Note: Cluster names should be DNS-compatible. Cluster name must be at least three characters and no more than 10 characters long. Allowed characters in a cluster name are ‘a-z, 0-9, -‘ lowercase letters, numbers, and hyphens. Any other character is invalid. Also, if a separate console port is chosen, the first public NIC is chosen to work exclusively as a console port.

7 The installer asks if you want to configure the Network Time Protocol (NTP) server.

```
Do you want to configure the Network Time Protocol(NTP) server to
synchronize the system clocks? [y,n,q]  y
Enter the Network Time Protocol server: [q,?]
```

If you enter **y**, you can type in your NTP server. If you enter **n**, the NTP server is not configured.

8 Installer asks to confirm the entered cluster information.

The installer detects the network devices, checks the network device's connectivity with the gateway, and displays information about it.

```
Checking network configuration ..................................... Done
Detecting network devices .......................................... Done
Checking network connection ........................................ Done

Detecting network devices completed successfully.

Common NICs on all systems:

NIC       Type       Properties     Public
=========================================
ens161    Physical   -              N
ens192    Physical   -              Y
ens193    Physical   -              Y
ens224    Physical   -              Y
ens256    Physical   -              N

For the 'Public' field of the NIC:
    Y: means the NIC can connect to the public gateway, and can be selected as
public NIC.
    N: means the NIC cannot connect to the public gateway, and can be selected
as private NIC.
    -: means the NIC was not tested if connect to the public gateway.
    blank: means this NIC is excluded or not selectable.
```

To configure Veritas Access networking, you need to exclude the unused NICs, and to include at least one public NIC, and one private NIC. It is recommended to have two public NICs and two private NICs, and the selected private NICs on all systems should be interconnected.

If you do want to configure NIC bonding or exclude NICs, enter **y**.

If you do not want to configure NIC bonding or exclude NICs, enter **n**.

See “Excluding a NIC” on page 77.

See “Creating a NIC bond” on page 83.

See “Creating a VLAN device ” on page 95.

9 You need to select one of the options from the following for the installation:

- Manually select NIC
- Configure NIC bonding

- Configure VLAN through installer

```
Do you want to manually select NICs, or configure NIC bonding or VLAN tagging? [y,n,q] (n)
Enter n : If you want installer to do auto network configuration for the cluster
Enter y : If you want to select public and private NICs manually, configure NIC bonding
          or to create VLAN using installer.
```

The installer performs the public and private NIC detection tests on each node of the cluster. If physical or virtual IPs that are entered are less than the required IPs for the cluster configuration, the installer asks you to add the required IPs.

10 Verify that the network configuration details such as the new IP addresses, host name, and other details are correct.

11 The installer prompts to verify the network configuration.

Verify that the configuration information such as the new IP addresses, host name, and other details are correct.

```
Configuration checklist:

System        Public NIC Physical IP
============= ========== =============
192.168.10.10 ens192     192.168.10.20
192.168.10.10 ens224     192.168.10.21
192.168.10.10 ens193     192.168.10.22
192.168.10.11 ens192     192.168.10.23
192.168.10.11 ens224     192.168.10.24
192.168.10.11 ens193     192.168.10.25

System        Private NIC
============= ===========
192.168.10.10 ens161
192.168.10.10 ens256
192.168.10.11 ens161
192.168.10.11 ens256

Virtual IP
=========================================================================
192.168.10.30 192.168.10.31 192.168.10.32 192.168.10.33 ...(6 in total)

Console IP
=============
192.168.10.50

Gateway IP   DNS IP       Domain name
============ ============ ===================
192.168.10.1 192.168.10.2 vxindia.veritas.com

Is this information correct? [y,n,q,b,?] (y)
```

12 After you confirm the network configuration details, the installer renames the host name if you have chosen to rename it and assigns the IPs for the systems. The installer also checks the Low Latency Transport (LLT) link status and automatically selects them.

Note: The installer does not check the LLT link status if the InfiniBand NICs are chosen as private NICs.

13 Installer performs the Veritas Access service group configuration.

14 The installer prompts if you want to configure I/O fencing during the installation.

```
Do you want to configure I/O Fencing in enabled mode? [y,n,q,?] (y)
```

If you do not want to configure I/O fencing, enter **n**.

To configure I/O fencing, enter **y**.

See "Configuring I/O fencing" on page 102.

15 The installer automatically restarts the cluster nodes to enables the Kdump function for each node.

16 Check the log file to confirm the installation and configuration. Logs can be found in `/opt/VRTS/install/logs/`.

Note: After the installation, connect to the Veritas Access console using the console IP address you assigned earlier, then log on using the default user name `master` and the default password `master`.

Veritas Access Graphical User Interface

Veritas Access has a Graphical User Interface (GUI) that provides a dashboard for a specific Veritas Access cluster, as well as views for shares, storage infrastructure, reports, and settings. The GUI lets the administrator perform tasks for the cluster and monitor the results. In this release, the GUI is part of Veritas Access.

After you complete I/O fencing configuration successfully, the link to the GUI appears on the screen.

```
Open the https://<console IP>:14161 URL
in your browser to start the Veritas Access GUI application.
```

About managing the NICs, bonds, and VLAN devices

When you enter **y**, the installer allows you to perform the following operations:

```
Do you want to manually select NICs, or configure NIC bonding or VLAN tagging? [y,n,q] (n) y
```

- Select the public NICs.
  See "Selecting the public NICs" on page 71.
- Select the private NICs.
  See "Selecting the private NICs" on page 74.
- Exclude a NIC.
  See "Excluding a NIC" on page 77.
- Include a NIC.
  See "Including a NIC" on page 80.
- Create a new NIC bond and add a NIC to a bond.
  See "Creating a NIC bond" on page 83.
- Remove a bond.
  See "Removing a NIC bond" on page 89.
- Remove a NIC from the bond list.
  See "Removing a NIC from the bond list" on page 92.
- Add a VLAN device on a particular NIC.
  See "Creating a VLAN device " on page 95.
- Remove a VLAN device on a particular NIC.
  See "Removing a VLAN device " on page 98.

Note: The NIC bonding and NIC exclusion configuration options support both a single NIC or bond, and multiple NICs or bonds.

When using the NIC exclusion feature, you can exclude any NIC on the first node. But if you want to exclude any NIC on the other nodes, you can choose to exclude NICs per node.

See "Excluding a NIC" on page 77.

Selecting the public NICs

When you install Veritas Access on a cluster, you may want to configure the specific network devices as a public interface, even though they are not reachable to the gateway and specific network devices as a private interface.

To select the public NICs

1 In the manual selection mode, enter **1** to select public NICs.

```
NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC         Type       Properties       Public
=================================================
ens161      Physical   -                N
ens192      Physical   -                Y
ens193      Physical   -                Y
ens224      Physical   Slave of bond0
ens225      Physical   Slave of bond0
ens256      Physical   -                N
ens257      Physical   -
bond0       Virtual    Bond balance-rr  Y
ens192.100 Virtual    VLAN 100         -

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q] 1
```

2 Select the NIC that you want to choose as public NICs.

```
Choose NICs as public

     1)  ens161
     2)  ens192
     3)  ens193
     4)  ens256
     5)  bond0
     6)  ens192.100
     7)  Unselect previous public NICs
     b)  Back to previous menu

Choose items, separated by spaces: [1-7,b,q] 2 4 5 6
NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC         Type       Properties       Public
=================================================
ens161      Physical   -                N
ens192      Physical   -                Y (Selected)
ens193      Physical   -                Y
ens224      Physical   Slave of bond0
ens225      Physical   Slave of bond0
ens256      Physical   -                Y (Selected)
ens257      Physical   -
bond0       Virtual    Bond balance-rr  Y (Selected)
ens192.100 Virtual     VLAN 100         Y (Selected)

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q]
```

Note: To start the cluster configuration, after the manual public and private NIC selection or configuration, enter **11** to select the `Save and continue` option.

Selecting the private NICs

When you install Veritas Access on a cluster, you may want to configure the specific network devices as a private interface.

To select the private NICs

1 In the manual selection mode, enter **2** to select private NICs.

```
NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC        Type       Properties       Public
=================================================
ens161     Physical   -                 N
ens192     Physical   -                 Y (Selected)
ens193     Physical   -                 Y (Selected)
ens224     Physical   Slave of bond0
ens225     Physical   Slave of bond0
ens256     Physical   -                 N
ens257     Physical   -
bond0      Virtual      Bond balance-rr  Y (Selected)
ens192.100 Virtual   VLAN 100         Y (Selected)

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q]  2
```

2 Select the NIC that you want to choose as private NICs.

```
Choose NICs as private

     1)  ens161
     2)  ens192
     3)  ens193
     4)  ens256
     5)  Unselect previous private NICs
     b)  Back to previous menu

Choose items, separated by spaces: [1-5,b,q] 1 4

NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC         Type       Properties       Public
=====================================================
ens161      Physical   -                N (Selected)
ens192      Physical   -                Y (Selected)
ens193      Physical   -                Y (Selected)
ens224      Physical   Slave of bond0
ens225      Physical   Slave of bond0
ens256      Physical   -                N (Selected)
ens257      Physical   -
bond0       Virtual    Bond balance-rr  Y (Selected)
ens192.100 Virtual    VLAN 100         Y (Selected)

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q]
```

Note: To start the cluster configuration, after the manual public and private NIC selection or configuration, enter **11** to select the `Save and continue` option.

Excluding a NIC

When you install Veritas Access on a cluster, you may want to use some of the NICs for other storage purposes. You can exclude a NIC that you do not want to use for Veritas Access.

Note: The NIC bonding or NIC exclusion configuration options support both a single NIC or bond, and multiple NICs or bonds.

To exclude a NIC

1 In the manual selection mode, enter **3**.

```
NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC         Type       Properties       Public
=================================================
ens161      Physical   -                N
ens192      Physical   -                Y
ens193      Physical   -                Y
ens224      Physical   Slave of bond0
ens225      Physical   Slave of bond0
ens256      Physical   -                N
ens257      Physical   -                Y
bond0       Virtual    Bond balance-rr  Y
ens192.100 Virtual    VLAN 100         -

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q] 3
```

2 Select the NIC that you want to exclude.

```
Choose NICs for exclusion

     1)  ens161
     2)  ens192
     3)  ens193
     4)  ens256
     5)  ens257
     6)  bond0
     7)  ens192.100
     b)  Back to previous menu

Choose items, separated by spaces: [1-7,b,q] 3 5

NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC         Type        Properties        Public
=====================================================
ens161      Physical    -                 N
ens192      Physical    -                 Y
ens193      Physical    -
ens224      Physical    Slave of bond0
ens225      Physical    Slave of bond0
ens256      Physical    -                 N
ens257      Physical    -
bond0       Virtual     Bond balance-rr   Y
ens192.100  Virtual     VLAN 100          -

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue
```

```
Select the NIC option to be configured in this cluster: [1-11,q]
```

Including a NIC

When you install Veritas Access on a cluster, you may want to include one or more NICs that you had previously excluded. You can include the NICs that you want to use for Veritas Access.

To include a NIC

1 In the manual selection mode, enter **4**.

```
Common NICs on all systems:

NIC         Type       Properties       Public
=====================================================
ens161      Physical   -                N
ens192      Physical   -                Y
ens193      Physical   -
ens224      Physical   Slave of bond0
ens225      Physical   Slave of bond0
ens256      Physical   -                N
ens257      Physical   -
bond0       Virtual    Bond balance-rr  Y
ens192.100 Virtual    VLAN 100         -

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q] 4
```

2 Select a NIC that you want to include.

```
Choose NICs for inclusion

     1)  ens193
     2)  ens257
     b)  Back to previous menu

Choose items, separated by spaces: [1-2,b,q]  1


NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC         Type       Properties       Public
=====================================================
ens161      Physical   -                N
ens192      Physical   -                Y
ens193      Physical   -                Y
ens224      Physical   Slave of bond0
ens225      Physical   Slave of bond0
ens256      Physical   -                N
ens257      Physical   -
bond0       Virtual    Bond balance-rr  Y
ens192.100 Virtual    VLAN 100         -

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q]
```

Creating a NIC bond

An administrator can create a bond NIC interface from a given list of public NIC interfaces during Veritas Access installation. This feature allows an administrator to save a number of physical IP addresses that are used for installation and post-installation bond creation.

- You cannot bond InfiniBand NICs because the PCI IDs are identical.

If you do not want to create a bond interface, continue with the installation.

See “About obtaining IP addresses” on page 42.

See “About calculating IP address requirements” on page 43.

To create a bond

1 After you choose manual selection mode, the installer prompts you to enter your selection. Enter **5** to create a new bond.

```
NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC        Type       Properties       Public
================================================
ens161     Physical   -                N
ens192     Physical   -                Y
ens193     Physical   -                Y
ens224     Physical   -                Y
ens225     Physical   -                Y
ens256     Physical   -                N
ens257     Physical   -                Y

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q] 5
```

2 Select a bond mode for the new bond.

```
Configure the mode of the NIC bonding:

     1)  balance-rr
     2)  active-backup
     3)  balance-xor
     4)  broadcast
     5)  802.3ad
     6)  balance-tlb
     7)  balance-alb
     b)  Back to previous menu

Select the bonding mode: [1-7,b,q] 1

bond0 is created. Please add NICs to bond0 to enable it.

Press [Enter] to continue:
```

If you choose **3** or **5**, you need to choose the bond option for the bond mode:

```
1) layer2
2) layer3+4
3) default

Select the bonding option: [1-3,b,q] 1
```

The installer prompts you to select the NIC option that you want to configure for the cluster.

3 Enter **6** to add NICs to bond.

Note: You need to have NIC in bond.

```
NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC        Type       Properties       Public
=================================================
ens161     Physical   -                N
ens192     Physical   -                Y
ens193     Physical   -                Y
ens224     Physical   -                Y
ens225     Physical   -                Y
ens256     Physical   -                N
ens257     Physical   -                Y
bond0      Virtual    Bond balance-rr  -

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q] 6
```

4 Enter **6** to select a NIC that you want to add in a bond.

```
Choose NICs for bonding

     1)  ens161
     2)  ens192
     3)  ens193
     4)  ens224
     5)  ens225
     6)  ens256
     7)  ens257
     b)  Back to previous menu

Choose NICs, separated by spaces: [1-7,b,q,?] 4 5
```

5 Select a bond name for which you want to add the NIC.

```
Choose a bond name to add NICs

     1)  bond0
     b)  Back to previous menu

Choose bonds, separated by spaces: [1-1,b,q] 1

Adding ens224 ens225 to bond0 was successful

Press [Enter] to continue:

NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC        Type       Properties       Public
=====================================================
ens161     Physical   -                N
ens192     Physical   -                Y
ens193     Physical   -                Y
ens224     Physical   Slave of bond0
ens225     Physical   Slave of bond0
ens256     Physical   -                N
ens257     Physical   -                Y
bond0      Virtual    Bond balance-rr  Y

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q]
```

Removing a NIC bond

An administrator can remove a bond.

To remove a NIC bond

1 Enter **7** to remove an existing bond.

```
Common NICs on all systems:

NIC        Type       Properties         Public
=========================================================
ens161     Physical   -                  N
ens192     Physical   -                  Y
ens193     Physical   Slave of bond1
ens224     Physical   Slave of bond0
ens225     Physical   Slave of bond0
ens256     Physical   -                  N
ens257     Physical   -                  Y
bond0      Virtual    Bond balance-rr    Y
bond1      Virtual    Bond active-backup Y

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q] 7
```

2 Select the bond that you want to remove.

```
Choose bonds to be removed

     1)  bond0
     2)  bond1
     b)  Back to previous menu

Choose bonds, separated by spaces: [1-2,b,q] 2

Deleting NIC bonding bond1 succeeded

Press [Enter] to continue:
NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC        Type       Properties       Public
=====================================================
ens161     Physical   -                N
ens192     Physical   -                Y
ens193     Physical   -                Y
ens224     Physical   Slave of bond0
ens225     Physical   Slave of bond0
ens256     Physical   -                N
ens257     Physical   -                Y
bond0      Virtual    Bond balance-rr  Y

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q]
```

Removing a NIC from the bond list

During installation, an administrator can remove a NIC that is already a slave of a bond before the configuration is saved.

To remove a NIC from the bond list

1 During the Veritas Access installation, the installer prompts you to enter your selection. Enter **8** to remove a NIC from the bond list.

Note: The NIC bonding or NIC exclusion configuration options support both a single NIC or bond and multiple NICs or bonds.

```
NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC        Type       Properties         Public
=======================================================
ens161     Physical   -                  N
ens192     Physical   Slave of bond1
ens193     Physical   Slave of bond1
ens224     Physical   Slave of bond0
ens225     Physical   Slave of bond0
ens256     Physical   -                  N
ens257     Physical   -                  Y
bond0      Virtual    Bond balance-rr    Y
bond1      Virtual    Bond active-backup Y

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q] 8
```

2 Select a NIC that you want to remove from the NIC bonding.

```
Choose NICs to be deleted from the NIC bonding

     1)  ens192
     2)  ens193
     3)  ens224
     4)  ens225
     b)  Back to previous menu

Choose NICs, separated by spaces: [1-4,b,q,?] 1

Removing ens192 from bonding was successful

Press [Enter] to continue:
NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC        Type       Properties          Public
=======================================================
ens161     Physical   -                   N
ens192     Physical   -                   Y
ens193     Physical   Slave of bond1
ens224     Physical   Slave of bond0
ens225     Physical   Slave of bond0
ens256     Physical   -                   N
ens257     Physical   -                   Y
bond0      Virtual    Bond balance-rr     Y
bond1      Virtual    Bond active-backup  Y

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue
```

```
Select the NIC option to be configured in this cluster: [1-11,q]
```

About VLAN tagging

When VLANs (Virtual Local Area Network) span multiple switches, VLAN tagging is required. A VLAN is a way to create independent logical networks within a physical network. VLAN tagging is the practice of inserting a VLAN ID into a packet header to identify which VLAN the packet belongs to.

By using the VLAN tagging feature, you can:

- Create a VLAN device during installation
- Create a VLAN device on the specified bond interface.

Note: You need to create a bond interface first.

See "Creating a VLAN device " on page 95.
See "Removing a VLAN device " on page 98.

Creating a VLAN device

You can create a VLAN device for a public NIC interface or a public bond.

See "About VLAN tagging" on page 95.

Note: If you need to use VLAN interface as public NIC while configuring the Veritas Access network, it is mandatory to add the NIC on which VLAN is created as public NIC during the Veritas Access installation.

For example, if VLAN is `eth0.100`, you should select `eth0.100` and `eth0` as public NIC during the access network configuration when you install Veritas Access.

To create a VLAN device

1 In the manual selection mode, enter **9** to create a VLAN device.

```
NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC        Type       Properties       Public
=================================================
ens161     Physical   -                N
ens192     Physical   -                Y
ens193     Physical   -                Y
ens224     Physical   Slave of bond0
ens225     Physical   Slave of bond0
ens256     Physical   -                N
ens257     Physical   -                Y
bond0      Virtual    Bond balance-rr  Y

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q] 9
```

2 Select the NICs on which you want to create VLAN devices.

3 Enter the VLAN ID for the device.

```
Choose NICs to create VLAN device on:
     1)  ens161
     2)  ens192
     3)  ens193
     4)  ens256
     5)  ens257
     6)  bond0
     b)  Back to previous menu

Choose VLAN devices, separated by spaces: [1-6,b,q] 2

Enter the VLAN ID for the device (1-4094): [b,q,?] 100


NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC         Type       Properties       Public
=====================================================
ens161      Physical   -                N
ens192      Physical   -                Y
ens193      Physical   -                Y
ens224      Physical   Slave of bond0
ens225      Physical   Slave of bond0
ens256      Physical   -                N
ens257      Physical   -                Y
bond0       Virtual    Bond balance-rr  Y
ens192.100 Virtual    VLAN 100         -

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue
```

```
Select the NIC option to be configured in this cluster: [1-11,q]
```

Removing a VLAN device

You can remove a VLAN device for a public NIC interface or a public bond.

See “About VLAN tagging” on page 95.

To remove a VLAN device

1 In the manual selection mode, enter **10** to remove a VLAN device.

```
Common NICs on all systems:

NIC         Type       Properties       Public
=====================================================
ens161      Physical   -                N
ens192      Physical   -                Y
ens193      Physical   -                Y
ens224      Physical   Slave of bond0
ens225      Physical   Slave of bond0
ens256      Physical   -                N
ens257      Physical   -                Y
bond0       Virtual    Bond balance-rr  Y
ens192.100 Virtual    VLAN 100         -

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q] 10
```

2 Select the VLAN NICs that you want remove.

```
Choose VLAN NICs to be deleted

     1)  ens192.100
     b)  Back to previous menu

Choose VLAN devices, separated by spaces: [1-1,b,q] 1

NIC selection/NIC bonding/NIC VLAN configuration

Common NICs on all systems:

NIC        Type       Properties       Public
=================================================
ens161     Physical   -                N
ens192     Physical   -                Y
ens193     Physical   -                Y
ens224     Physical   Slave of bond0
ens225     Physical   Slave of bond0
ens256     Physical   -                N
ens257     Physical   -                Y
bond0      Virtual    Bond balance-rr  Y

     1)  Select public NICs
     2)  Select private NICs
     3)  Exclude NICs
     4)  Include NICs
     5)  Create a new bond
     6)  Add NICs to a bond
     7)  Remove bonds
     8)  Remove NICs from the bond list
     9)  Create VLAN device
    10)  Delete VLAN device
    11)  Save and continue

Select the NIC option to be configured in this cluster: [1-11,q]
```

Limitations of VLAN tagging

Tthe following are the limitations for using the VLAN tagging:

- Support only for a fresh installation. VLAN tagging is not supported for reconfiguration with the `-updateparameter` option and add node configuration.
- Support only for creating a VLAN device on a bonded NIC.
- Support only for creating one VLAN device at installation time.

Replacing an Ethernet interface card

In some cases, you may need to replace an Ethernet interface card on a node. This section describes the steps to replace the card.

Note: This procedure works for replacing an existing Ethernet interface card. It does not work for adding an Ethernet interface card to the cluster. If the Ethernet interface card you add needs a new device driver, install the new device driver before installing the Ethernet interface card on the node.

To replace an Ethernet interface card

1. Use the `Cluster> shutdown` command to shut down the node.

   For example:

   ```
   Cluster> shutdown access_03
   Stopping Cluster processes on access_03.......done
   Sent shutdown command to access_03
   ```

2. Use the `Cluster> del` command to delete the node from the cluster.

   For example:

   ```
   Cluster> del access_03
   ```

3. Install the replacement Ethernet interface card on the node.
4. Turn on the node.
5. Make sure that the Ethernet interface card is active and online.
6. Use the `Cluster> add` command to add the node back into the cluster.

   For example:

   ```
   Cluster> add 172.16.113.118
   ```

For details on the `Cluster> add` and `Upgrade>`commands that are described in this section, see the relevant man pages.

Configuring I/O fencing

Veritas Access supports two fencing modes:

- Disk-based fencing for a cluster with shared disks
- Majority-based fencing for a cluster with local DAS disks

If you want to use both shared disks (SAN) and local disks, majority-based fencing must be used. Veritas recommends that you do not configure I/O fencing through the installer.

1. During the Veritas Access configuration, after the product is started, the installer asks whether to configure fencing:

   ```
   Do you want to configure I/O Fencing in enabled mode? [y,n,q,?] (y)
   ```

2. Enter **y** to configure fencing.

   You can choose from one of the following fencing modes:

   - If the cluster does not include initialized shared disks, you need to configure the majority-based fencing.

     ```
     1. Majority Based Fencing
     Select the fencing mechanism to be configured:[b](1)
     ```

   - If shared disks are connected and initialized, the disk-based I/O fencing is configured. You are prompted to choose fencing type.

     ```
     1. Majority Based Fencing
     2. Disk Based Fencing
     Select the fencing mechanism to be configured:[b](2)
     ```

     Note: You can choose three available VxVM disks or initialize three disks as VxVM disks to form the fencing disk group. You must choose exactly three disks.

3. The installer stops the product, and applies the fencing configuration before restart.

About configuring Veritas NetBackup

If you use Veritas NetBackup, to comply with the NetBackup End-User License Agreement (EULA), you have to purchase and enter valid license keys on the

external NetBackup master server before you configure NetBackup to work with Veritas Access. For more information on entering the NetBackup license keys on the NetBackup master server, see the *Veritas NetBackup Installation Guide*.

If you use NetBackup, configure the virtual IP address using the `Backup> virtual-ip` command so that it is different from all of the virtual IP addresses, including the console server IP address and the physical IP addresses that are used to install the Veritas Access software.

About enabling kdump during an Veritas Access configuration

During the Veritas Access configuration, the Veritas Access installer tries to enable kdump on your cluster node. To meet the Veritas Access software requirements, the installer modifies the `/etc/kdump.conf` and `/boot/grub/grub.conf` files by using the following options:

- `/boot/grub/grub.conf`
  `crashkernel = 512M-2G:64M, 2G-:256M`

- `/etc/kdump.conf`
  `path /opt/VRTSsnas/core/kernel/`
  `core_collector makedumpfile -c --message-level 1 -d 31`

Reconfiguring the Veritas Access cluster name and network

After you install and configure Veritas Access, you can reconfigure the cluster name and network, if required.

Before you reconfigure the cluster, you have to enable the *support* user for the nodes because the root user access authority is forbidden. The *support* user default password is *veritas*. You can change the password after you log on the first time.

To reconfigure the Veritas Access cluster name and network

1. Log on to the host console using the *support* user name and password.
2. Ensure that all the service groups are offline. Enter the following command:

   `/opt/VRTS/install/installaccess -updateparameter`

3 Enter the private IPs of the systems.

```
172.16.0.3 172.16.0.4
```

Note: Only the private IPs of the systems must be entered. Public IPs should not be used here.

4 Enter the cluster name and network information.

```
Enter the cluster name:
Enter the public IP starting address:
Enter the netmask for the public IP address:
Enter the number of VIPs per interface:
Enter the virtual IP starting address:
Enter the default gateway IP address:
Enter the DNS IP address:
Enter the DNS domain name:
Enter the console virtual IP address:
Do you want to use the separate console port? [y,n,q] (n):
Do you want to configure the Network Time Protocol(NTP) server to
synchronize the system clocks? [y,n,q] (n) y:
Enter the Network Time Protocol server:
```

The installer confirms that the information that you entered is correct. The configuration is completed and the new cluster and IPs are configured on the cluster.

The installer displays the location of the log and summary files. If required, view the files to confirm the configuration status.

Note: The cluster name can contain only alpha characters, numbers, or underscores. The cluster name must start with a letter of the alphabet and can have a length of maximum 15 characters. Also, if a separate console port is chosen, the first public NIC is chosen to work exclusively as a console port.

Note: If your cluster has DAS disks, limit the cluster name to 10 characters.

After formatting the DAS disks, do not change the cluster name.

Configuring a KMS server on the Veritas Access cluster

You can configure a KMS server on the Veritas Access cluster.

To configure a KMS server on the Veritas Access cluster

1. Obtain the KMS server's SSL public key (in base64 format) and its port number. This key is used for communication between the Veritas Access cluster and the KMS server.
2. Generate a self-signed SSL key-pair on the Veritas Access cluster:

   ```
   System> kms certificate generate
   ```

3. Import the KMS server's public key.

   ```
   System> kms certificate import_server_cert
   ```

4. Configure the KMS server. Provide the SSL public key that was obtained in step 1 as input here.

   ```
   System> kms config server <server_ip> <server_port>
   ```

   Where *server_ip* is the KMS server IP.

   server_port is the KMS server port number.

5. KMS admin now sets up a trust certificate using its admin GUI to allow communication between the KMS server and the Veritas Access cluster.

For more information, see the `system_kms` man page.

Chapter 7

Automating Veritas Access installation and configuration using response files

This chapter includes the following topics:

About response files

The installer script generates a response file during any installation, configuration, upgrade, or uninstall procedure. The response file contains the configuration information that you entered during the procedure. When the procedure completes, the installation script displays the location of the response files.

You can use the response file for future installation procedures by invoking an installation script with the `-responsefile` option. The response file passes arguments to the script to automate an installation or uninstallation.

See “Installation script options” on page 151.

Performing a silent Veritas Access installation

You can use the silent installation if you want to install the Veritas Access software on a large number of nodes. You need to prepare a response file. By using this file, the silent installation and configuration of the Veritas Access software can be performed, without any prompts.

Before you perform a silent Veritas Access installation and configuration, you need to manually configure an SSH communication between the nodes.

See "Manually configuring passwordless SSH" on page 153.

You can get the Veritas Access example response file from the root directory of the ISO image.

To use the Veritas Access silent installation feature

- Enter the following command:

```
# ./installaccess -responsefile access.responsefile
```

To generate the `access.response` example file

1. Install and configure the Veritas Access software without any errors.
2. Get the `access.response` example file from the log directory.

To use the `access.response` example file

1. Rename the Veritas Access example response file to `access.responsefile`.
2. Modify the file by changing the cluster name, IP address ranges, and other parameters, as necessary for your configuration.

Installation times may vary depending on your configuration.

See "Installing and configuring the Veritas Access software on the cluster" on page 62.

Response file variables to install and configure Veritas Access

Table 7-1 lists the response file variables that you can define to install and configure Veritas Access.

Table 7-1 Response file variables for installing Veritas Access

Variable	Description
CFG{bondmode}{bond<n>}	Defines the bond modes for BOND. List or scalar: list Optional or required: optional
CFG{bondname}	List of bond names for BOND. List or scalar: list Optional or required: optional
CFG{config_majority_based_fencing}	Enables the majority of the fencing. The value is set to `1`. It cannot be used with I/O fencing variables that are 'fencing_scsi3_disk_policy', 'fencing_newdg_disks', and 'fencing_dgname'. List or scalar: scalar Optional or required: required for majority-based fencing
CFG{fencing_dgname}	Specifies the disk group for I/O fencing. The value is `sfscoorddg`. List or scalar: scalar Optional or required: required for the I/O fencing
CFG{fencing_newdg_disks}	Defines the fencing disks. List or scalar: list Optional or required: required for the I/O fencing
CFG{fencing_option}	Specifies the I/O fencing configuration mode. The value is `2` for the disk-based I/O fencing. List or scalar: scalar Optional or required: required for the I/O fencing

Table 7-1 Response file variables for installing Veritas Access *(continued)*

Variable	Description
CFG{fencing_scsi3_disk_policy}	Specifies the SCSI-3 disk policy to use the I/O fencing. The value is `dmp`. List or scalar: scalar Optional or required: required for the I/O fencing
CFG{fencingenabled}	Defines whether fencing is enabled. The value is `1` if enabled. List or scalar: scalar Optional or required: required for the I/O fencing
CFG{opt}{licensefile}	Specifies the location of the Veritas perpetual or subscription license key file. List or scalar: scalar Optional or required: required
CFG{keys}{"node_ip"}	Specifies the Veritas Access license for each node. List or scalar: scalar Optional or required: required
CFG{newnodes}	Specifies the new access IP for the cluster nodes. The value must be the first public IP address for each node. List or scalar: list Optional or required: required
CFG{opt}{comcleanup}	Cleans up the SSH connection. The installer adds this connection after the configuration. The value is `1`. List or scalar: scalar Optional or required: required
CFG{opt}{confignic}	Performs the NIC configuration with all the network variable values. The value is `1`. List or scalar: scalar Optional or required: required

Table 7-1 Response file variables for installing Veritas Access *(continued)*

Variable	Description
CFG{opt}{configure}	Performs the configuration if the packages are already installed. List or scalar: scalar Optional or required: required
CFG{opt}{install}	Installs Veritas Access RPMs. Configuration can be performed at a later time using the `-configure` option. List or scalar: scalar Optional or required: optional
CFG{opt}{installallpkgs}	Instructs the installer to install all the Veritas Access RPMs based on the variable that has the value set to 1. List or scalar: scalar Optional or required: required
CFG{opt}{noipc}	Disables the connection to SORT for updates check. The value is `0`. List or scalar: scalar Optional or required: required
CFG{opt}{ssh}	Determines whether to use SSH for communication between systems. The value is `1` if enabled. List or scalar: scalar Optional or required: required
CFG{prod}	Defines the product to be installed or uninstalled. List or scalar: scalar Optional or required: required
CFG{publicnetmaskarr}	List of the netmasks that are assigned to public NICs or bonds. List or scalar: list Optional or required: required

Table 7-1 Response file variables for installing Veritas Access *(continued)*

Variable	Description
CFG{publicparr}	List of public IPs that are assigned to public NICs or bonds. List or scalar: list Optional or required: required
CFG{redhat_subscription_username}	Specifies the user name to register with Red Hat subscription management. List or scalar: scalar Optional or required: required, if some of the required OS RPMs are not found on the systems. The user name should be enclosed in single quotes (for example : '1234@abc') if it contains any special character.
CFG{redhat_subscription_password}	Specifies the password to register with Red Hat subscription management. List or scalar: scalar Optional or required: required, if some of the required OS RPMs are not found on the systems. The password should be enclosed in single quotes (for example, '1234@abc') if it contains any special character.
CFG{snas_clustername}	Defines the cluster name of the product. List or scalar: scalar Optional or required: required
CFG{snas_consoleip}	Defines the console IP of the product. List or scalar: scalar Optional or required: required
CFG{snas_defgateway}	Defines the gateway of the product. List or scalar: scalar Optional or required: required

Table 7-1 Response file variables for installing Veritas Access *(continued)*

Variable	Description
CFG{snas_dnsdomainname}	Defines the DNS domain name of the product. List or scalar: scalar Optional or required: required
CFG{snas_dnsip}	Defines the DNS IP of the product. List or scalar: scalar Optional or required: required
CFG{snas_ntpserver}	Defines the NTP server name of the product. List or scalar: scalar Optional or required: required
CFG{snas_nvip}	Defines the number of VIPs on each NIC. List or scalar: scalar Optional or required: required
CFG{snas_pipprefix}	Defines the prefix of public IPs (only in IPV6 environments). List or scalar: scalar Optional or required: required
CFG{snas_pipstart}	Defines the initial IP of the public IPs. List or scalar: scalar Optional or required: required
CFG{snas_pnmaskstart}	Defines the netmask of public IPs (only in IPV4 environments). List or scalar: scalar Optional or required: required
CFG{snas_sepconsoleport}	Defines if use of separate console port. 1 for yes, 0 for no. List or scalar: scalar Optional or required: required

Table 7-1 Response file variables for installing Veritas Access *(continued)*

Variable	Description
CFG{snas_vipprefix}	Defines the prefix of virtual IPs (only in IPV6 environments). List or scalar: scalar Optional or required: required
CFG{snas_vipstart}	Defines the initial IP of the virtual IPs. List or scalar: scalar Optional or required: required
CFG{snas_vnmaskstart}	Defines the netmask of virtual IPs (only in IPV4 environments). List or scalar: scalar Optional or required: required
CFG{systems}	List of systems on which the product is to be installed or to be uninstalled. List or scalar: list Optional or required: required
CFG{vcs_allowcomms}	Indicates whether to start LLT or GAB when the user wants to set up a single node cluster. List or scalar: scalar Optional or required: required
CFG{vcs_clusterid}	Defines the unique cluster ID with a string number. List or scalar: scalar Optional or required: required
CFG{vcs_lltlink<n>}{"new_node_ip"}	Defines the NIC name for the first heartbeat link. List or scalar: scalar Optional or required: required

Table 7-1 Response file variables for installing Veritas Access *(continued)*

Variable	Description
CFG{vcs_userenpw}	Defines the encrypted user password. List or scalar: scalar Optional or required: required
CFG{vcs_username}	Defines the added user name for VCS. List or scalar: scalar Optional or required: required
CFG{vcs_userpriv}	Defines the user privilege. List or scalar: scalar Optional or required: required
CFG{virtualiparr}	List of the virtual IPs that are assigned to public NICs or bonds. List or scalar: list Optional or required: required
CFG{virtualnetmaskarr}	List of the netmasks that are assigned to public NICs or bonds. List or scalar: list Optional or required: required

Sample response file for Veritas Access installation and configuration

The following example shows a response file for installing and configuring Veritas Access.

```
##################################################
our %CFG;
#Installs Product packages.
$CFG{opt}{install}=1;
$CFG{opt}{installallpkgs}=1;
$CFG{opt}{comsetup}=1;
$CFG{opt}{noipc}=1;
$CFG{opt}{ssh}=1;
$CFG{prod}="SNAS73";
```

```
$CFG{opt}{licensefile}="<absolute_path_of_licfile>";

#Performs the configuration if the packages are already installed
$CFG{opt}{configure}=1;

#the PCI IDs of slave NICs
$CFG{bondpool}{bond0}=[ qw(0000:02:09.0 0000:02:07.0) ];
$CFG{bondpool}{bond1}=[ qw(0000:02:04.0 0000:02:08.0) ];

#mode of each bond
$CFG{bondmode}{bond0}=5;
$CFG{bondmode}{bond1}=6;

#names of bond
$CFG{bondname}=[ qw(bond0 bond1) ];

#the PCI IDs of excluded NICs
$CFG{exclusion}=[ qw(0000:02:03.0 0000:02:0a.0) ];

#the PCI IDs of all the bonded NICs
$CFG{publicbond}=[ qw(0000:02:03.0 0000:02:04.0 0000:02:07.0
0000:02:08.0) ];

#public IPs
$CFG{publiciparr}=[ qw(10.200.58.100 10.200.58.101 10.200.58.102
10.200.58.103 10.200.58.104 10.200.58.105 10.200.58.106 10.200.58.107) ];

#netmask for public IPs
$CFG{publicnetmaskarr}=[ qw(192.168.30.10 192.168.30.11 192.168.30.12
192.168.30.13 192.168.30.14 192.168.30.15 192.168.30.16 192.168.30.17) ];

#the user name to register with Red Hat subscription management
$CFG{redhat_subscription_username}="rhel_user";

#the password to register with Red Hat subscription management
$CFG{redhat_subscription_password}="rhel_password";

#clustername of SNAS
$CFG{snas_clustername}="testsnas";

#console IP of SNAS
$CFG{snas_consoleip}="192.168.30.40";
```

```
#default gateway of SNAS
$CFG{snas_defgateway}="192.168.30.1";

#domain name of DNS
$CFG{snas_dnsdomainname}="cdc.veritas.com";

#IP of DNS
$CFG{snas_dnsip}="192.168.30.2";

#NTP server name
$CFG{snas_ntpserver}="ntp.veritas.com";

#number of VIPs on each NIC
$CFG{snas_nvip}=1;

#netmask of public IPs(only ipv4 environment)
$CFG{snas_pnmaskstart}=255.255.255.0;

#the initial IP of public IPs
$CFG{snas_pipstart}="192.168.30.10";

#if use separate console port, 1 for yes, 0 for no
$CFG{snas_sepconsoleport}="0";

#netmask of virutal IPs(only ipv4 environment)
$CFG{snas_vnmaskstart}=255.255.255.0;

#the initial IP of virtual IPs
$CFG{snas_vipstart}="192.168.30.18";

#virtual IPs
$CFG{virtualiparr}=[ qw(192.168.30.18 192.168.30.19 192.168.30.20
 192.168.30.21 192.168.30.22 192.168.30.23 192.168.30.24 192.168.30.25) ];

#netmask for virual IPs
$CFG{virtualnetmaskarr}=[ qw(255.255.255.0 255.255.255.0 255.255.255.0
255.255.255.0 255.255.255.0 255.255.255.0 255.255.255.0 255.255.255.0) ];

#target systems
$CFG{systems}=[ qw(192.168.30.80 192.168.30.81) ];

#indicates whether to start llt/gab when user wants to setup a single
node cluster
```

```
$CFG{vcs_allowcomms}=1;

#define the unique cluser id with a string number
$CFG{vcs_clusterid}=325;

#define the cluster name with a string
$CFG{vcs_clustername}="testsnas";

#define the nic name for the first heartbeat link.
$CFG{vcs_lltlink1}{"192.168.30.10"}="priveth0";
$CFG{vcs_lltlink1}{"192.168.30.13"}="priveth0";
$CFG{vcs_lltlink2}{"192.168.30.10"}="priveth1";
$CFG{vcs_lltlink2}{"192.168.30.13"}="priveth1";

#define the encrypted user password
$CFG{vcs_userenpw}=[ qw(GPQiPKpMQlQQoYQkPN) ];

#define the added username for VCS
$CFG{vcs_username}=[ qw(admin) ];

#define the user privilege
$CFG{vcs_userpriv}=[ qw(Administrators) ];

1;
##################################################
```

Chapter 8

Displaying and adding nodes to a cluster

This chapter includes the following topics:

About the Veritas Access installation states and conditions

Table 8-1 describes the Veritas Access installation states.

Table 8-1 Veritas Access installation states

Installation state	Description
RUNNING	Node is part of the cluster and the Veritas Access processes are running on it.
FAULTED	Node is down and/or the Veritas Access processes are not running on it.

Table 8-1 Veritas Access installation states *(continued)*

Installation state	Description
LEAVING	Node is leaving the cluster gracefully
EXITED	Node has exited the cluster gracefully
UNKNOWN	Exact state of the node cannot be determined

Depending on the cluster condition as described in Table 8-2, output for the `Cluster> show` command changes.

Table 8-2 Cluster conditions and states

Condition	Description
If the node is configured and part of the cluster, but the node is powered off.	State displays as FAULTED, and there is no installation state or network statistics.
If the node is configured and part of the cluster, but the node is physically removed from the cluster.	State displays as FAULTED, and there is no installation state or network statistics.
If the node is configured and part of the cluster, but the node is shutdown using the `Cluster> shutdown` command.	State changes from LEAVING to EXITED.
If the node is configured and part of the cluster, and you use the `Cluster> del` command.	Node is deleted from the cluster, and information about the deleted node is no longer available.

Displaying the nodes in the cluster

You can display all the nodes in the cluster, their states, CPU load, and network load during the past 15 minutes.

If you use the `Cluster> show currentload` option, you can display the CPU and network loads collected from now to the next five seconds.

To display a list of nodes in the cluster

1 To display a list of nodes that are part of a cluster, and the systems that are available to add to the cluster, enter the following:

```
Cluster> show
```

Command output includes the following information. See examples below.

Node	Displays the node name if the node has already been added to the cluster. Displays the IP address of the node if it is still in the process of being added to the cluster.
	Example:
	`node_01`
	or
	`192.168.30.10`

State	Displays the state of the node or the installation state of the system along with an IP address of the system if it is installed. See "About the Veritas Access installation states and conditions" on page 118.
CPU	Indicates the CPU load.
pubethX	Indicates the network load for the Public Interface X.
bondX	Indicates the network load for bond NIC X.

2 For nodes already in the cluster, the following is displayed:

```
Node      State     CPU(15 min)  pubeth0(15 min)     pubeth1(15 min)
                        %       rx(MB/s) tx(MB/s)   rx(MB/s) tx(MB/s)
------    -------   ----------- -------- --------   -------- --------
snas_01   RUNNING     1.35        0.00     0.00       0.00     0.00
snas_02   RUNNING     1.96        0.00     0.00       0.00     0.00
```

3 For the nodes that are being added to the cluster, for the nodes that are being deleted from the cluster, and for the nodes that is getting upgraded, the following progress is displayed:

```
Nodes in Transition

Node/IP          Operation       State      Description
------           ---------       -----      -----------
192.168.30.11    Add node        FAILED     Installing packages
snas_03          Delete node     ONGOING    Removing node
snas_01,snas_02  Rolling upgrade ONGOING    Rolling upgrade phase 2
```

Note: The `add node` and `delete node` operations cannot be performed at the same time.

4 To display the CPU and network loads collected from now to the next five seconds, enter the following:

```
Cluster> show currentload
```

Example output:

```
Node       State    CPU(5 sec)   pubeth0(5 sec)    pubeth1(5 sec)
                        %       rx(MB/s) tx(MB/s) rx(MB/s) tx(MB/s)
----       -----    ----------  -------- -------- -------- --------
snas_01    RUNNING    0.26        0.01     0.00      0.01     0.00
snas_02    RUNNING    0.87        0.01     0.00      0.01     0.00
snas_03    RUNNING    10.78       27.83    12.54     0.01     0.00
```

Statistics for network interfaces are shown for each public interface available on the cluster nodes.

Before adding new nodes in the cluster

After you have installed the operating system, you can install and configure a multiple node Veritas Access cluster at one time. If you want to add additional nodes to the cluster after that, you need to complete the following procedures:

- Install the appropriate operating system software on the additional nodes.
  See "Installing the operating system on each node of the cluster" on page 57.
- Disable SELinux on the new node.

- You do not need to install the Veritas Access software on the additional node before you add the node. The Veritas Access software is installed when you add the nodes. If the Veritas Access software is already installed, it is uninstalled and the product (same version as the cluster) is installed after that. The reason to uninstall and then install the product is to make sure that the new node is installed with exactly the same version, and patch level (if any) as the other cluster nodes. The packages are stored in the cluster nodes so the product image is not needed during the addition of the new node.
- Verify that the existing cluster has sufficient physical IP addresses for the new nodes. You can add additional IP addresses with the CLISH command: .

  ```
  Network> ip addr add command
  ```

  For example:

```
Network> ip addr add 192.168.30.107 255.255.252.0 physical
ACCESS ip addr SUCCESS V-288-1031 ip addr add successful.

Network> ip addr show
IP             Netmask/Prefix  Device  Node       Type     Status
--             --------------  ------  ----       ----     ------
192.168.30.10 255.255.252.0   pubeth0 snas_01    Physical
192.168.30.11 255.255.252.0   pubeth1 snas_01    Physical
192.168.30.12 255.255.252.0   pubeth0 snas_02    Physical
192.168.30.13 255.255.252.0   pubeth1 snas_02    Physical
192.168.30.14 255.255.252.0           ( unused ) Physical
192.168.30.15 255.255.252.0           ( unused ) Physical
192.168.30.16 255.255.252.0   pubeth0 snas_01    Virtual  ONLINE (Con IP)
192.168.30.17 255.255.252.0   pubeth1 snas_01    Virtual  ONLINE
192.168.30.18 255.255.252.0   pubeth1 snas_01    Virtual  ONLINE
192.168.30.19 255.255.252.0   pubeth1 snas_01    Virtual
```

  In the example, the unused IP addresses 192.168.30.14 and 192.168.30.15 can be used by the new node as physical IP addresses.

  Note: The network configuration on the new nodes should be the same as that of the cluster nodes, that is, NICs should have same names and connectivity.

  Bonds and vLANs are created automatically to match the cluster configuration if they do not exist already.

- Add the node to your existing cluster.
  See “Adding a node to the cluster” on page 123.

Adding a node to the cluster

You must install the operating system on the nodes before you add nodes to a cluster.

If you use disk-based fencing, the coordinator disks must be visible on the newly added node as a prerequisite for I/O fencing to be configured successfully. Without the coordinator disks, I/O fencing does not load properly and the node cannot obtain the cluster membership.

If you use majority-based fencing, the newly added node does not need to have shared disks.

If you want to add a new node and want to exclude some unique PCI IDs, add the unique PCI IDs to the `/opt/VRTSsnas/conf/net_exclusion_dev.conf` file on each cluster node manually. For example:

```
[root@bob_01 ~]# cat /opt/VRTSsnas/conf/net_exclusion_dev.conf
0000:42:00.0 0000:42:00.1
```

Note: Writeback cache is supported for two-node clusters only, so adding nodes to a two-node cluster changes the caching to read-only.

Newly added nodes should have the same configuration of InfiniBand NICs.

If your cluster has configured the FSS pool and the pool's node group does not have a node, the newly added node is added into the FSS node group. The installer adds the new node's local data disks into the FSS pool.

To add the new node to the cluster

1. Log on to Veritas Access using the `master` or the `system-admin` account.
2. In CLISH, enter the `Cluster` command to enter the `Cluster>` mode.
3. To add the new nodes to the cluster, enter the following:

   ```
   Cluster> add node1ip, node2ip.....
   ```

   Where *node1ip,node2ip,....* are the IP address list of the additional nodes for the SSH connection.

   Note:

   - The node IPs are preserved and additional required are assigned from (unused) pool of physical IPs.

- The physical IPs of new nodes are usable IPs found from the configured public IP starting addresses.
- The virtual IPs are re-balanced to the new node but additional virtual IPs are not assigned.
  Go to step 6 to add new virtual IP addresses to the cluster after you add a node.
- The IPs that are accessible to the new nodes should be given.
- The accessible IPs of the new nodes should be in the public network, they should be able to ping the public network's gateway successfully.

For example:

```
Cluster> add 192.168.30.10
```

Note: You cannot add nodes to a two-node cluster when the writeback caching is enabled. Before you add a node, change the cache mode to read and then try again.

---

4 If a cache exists on the original cluster, the installer prompts you to choose the SSD disks to create cache on the new node when CFS is mounted.

```
1) emc_clariion1_242
2) emc_clariion1_243
b) Back to previous menu
Choose disks separate by spaces to create cache on 192.168.30.11
[1-2,b,q] 1
Create cache on snas_02 .....................Done
```

5 If the cluster nodes have created an FSS pool, and there are more than two local data disks on the new node, the installer asks you to select the disks to add into the FSS pool. Make sure that you select at least two disks for a striped volume layout. The total selected disk size should be no less than the FSS pool's capacity size.

```
Following storage pools need to add disk from the new node:
     1)  fsspool1
     2)  fsspool2
     3)  Skip this step

Choose a pool to add disks [1-3,q] 1
     1)  emc_clariion0_1570 (5.000 GB)
     2)  installres_03_sdc (5.000 GB)
     3)  installres_03_sde (5.000 GB)
     4)  sdd (5.000 GB)
     b)  Back to previous menu


Choose at least 2 local disks with minimum capacity of 10 GB [1-4,b,q] 2 4
Format disk installres_03_sdc,sdd ............................... Done

The disk name changed to installres_03_sdc,installres_03_sdd
    Add disk installres_03_sdc,installres_03_sdd to storage pool fsspool1  Done
```

6 If required, add the virtual IP addresses to the cluster. When you add a node, it does not add new virtual IP addresses or service groups to the cluster.

To add additional virtual IP addresses, use the following command in the Network mode:

```
Network> ip addr add ipaddr virtual
```

For example:

```
Network> ip addr add 192.168.30.14 255.255.252.0 virtual

ACCESS ip addr SUCCESS V-288-1031 ip addr add successful.
```

If a problem occurs when you add a node to a cluster (for example, if the node is temporarily disconnected from the network), do the following to fix the problem:

To recover the node:

- Turn off the node.
- Use the `Cluster> del nodename` command to delete the node from the cluster.

- Turn on the node.
- Use the `Cluster> add nodeip` command to add the node to the cluster.

Adding a node in mixed mode environment

To add a node in mixed mode

1 **Prerequisites:**

Add IPv4 and IPv6 IPs that are equal to the number of public Interfaces.

Use the same type of IP (that is, IPv4 or IPv6) that you have used at the time of the Veritas Access installation.

Make sure that the IPv6 IP auto-assignment is disabled on the new node.

2 Do one of the following:

- If you have used IPv4 address at the time of the Veritas Access installation, run the following command:

```
cluster add <IPV4 IP>
```

- If you have used IPv6 address at the time of the Veritas Access installation, run the following command:

```
cluster add <IPV6 IP>
```

Deleting a node from the cluster

This command deletes a node from the cluster. Use the node name that is displayed in the `Cluster> show` command.

Note: This command is not supported in a single-node cluster.

If the deleted node was in the RUNNING state prior to deletion, after you reboot the node, that node is assigned to the original IP address that can be used to add the node back to the cluster. The original IP address of the node is the IP address that the node used before it was added into the cluster.

If your cluster has configured a FSS pool, you cannot use the installer to delete nodes that would result in a single node in the node group of the FSS pool.

Deleting a node from a two-node cluster that has writeback caching enabled changes the caching to read-only. Writeback caching is only supported for two nodes.

The IP address that was used by the node before it was deleted from the cluster is still accessible until you perform a restart operation.

After the node is deleted from the cluster, when you perform a reboot operation, the old IP configuration is restored. Therefore, make sure to remove the used IPs from Veritas Access for the deleted node or vice versa.

To delete a node from the cluster

1. To show the current state of all nodes in the cluster, enter the following:

   ```
   Cluster> show
   ```

2. To delete a node from a cluster, enter the following:

   ```
   Cluster> del nodename
   ```

   where *nodename* is the node name that appeared in the listing from the `Cluster> show` command. You cannot specify a node by its IP address.

   For example:

   ```
   Cluster> del snas_01
   ```

3. After a node is deleted from the cluster, the physical IP addresses that it used are marked as unused physical IP addresses. The IP addresses are available for use if you add new nodes. The virtual IP addresses used by a node that have been deleted are not removed. Deleting a node moves the virtual IP addresses on the deleted node to the remaining nodes in the cluster.

   For example:

```
Network> ip addr show
IP             Netmask/Prefix  Device     Node            Type     Status
--             --------------  ------     ----            ----     ------
192.168.30.10 255.255.252.0   pubeth0    source_30a_01   Physical
192.168.30.11 255.255.252.0   pubeth1    source_30a_01   Physical
192.168.30.12 255.255.252.0              ( unused )      Physical
192.168.30.13 255.255.252.0              ( unused )      Physical
192.168.30.14 255.255.252.0   pubeth0    source_30a_01   Virtual  ONLINE (Con IP)
192.168.30.15 255.255.252.0   pubeth0    source_30a_01   Virtual  ONLINE
192.168.30.16 255.255.252.0   pubeth0    source_30a_01   Virtual  ONLINE
192.168.30.17 255.255.252.0   pubeth1    source_30a_01   Virtual  ONLINE
192.168.30.18 255.255.252.0   pubeth1    source_30a_01   Virtual  ONLINE
```

If the physical or virtual IP addresses are not going to be used, they can be removed using the following command:

```
Network> ip addr del ipaddr
```

For example:

```
Network> ip addr del 192.168.30.18
ACCESS ip addr SUCCESS V-288-1031 ip addr del successful.
```

Note: If you have configured NIC bonding for the cluster, you also need to delete the configuration of the deleted node on the switch.

Shutting down the cluster nodes

You can shut down a single node or all of the nodes in the cluster. Use the node name that is displayed in the `Cluster> show` command.

To shut down a node or all the nodes in a cluster

1 To shut down a node, enter the following:

```
Cluster> shutdown nodename
```

nodename indicates the name of the node you want to shut down. You cannot specify a node by its IP address.

For example:

```
Cluster> shutdown snas_04
Stopping Cluster processes on snas_04
Sent shutdown command to snas_04. SSH sessions to
snas_04 may terminate.
```

2 To shut down all of the nodes in the cluster, enter the following:

```
Cluster> shutdown all
```

Use `all` as the *nodename* to shut down all of the nodes in the cluster.

For example:

```
Cluster> shutdown all
Stopping Cluster processes on all
SSH sessions to all nodes may terminate.
Sent shutdown command to snas_02
Sent shutdown command to snas_03
Sent shutdown command to snas_04
Sent shutdown command to snas_01
```

Chapter 9

Upgrading Veritas Access and operating system

This chapter includes the following topics:

Upgrading the operating system and Veritas Access

Veritas Access supports the following upgrade paths for upgrades on RHEL.

Table 9-1 Supported upgrade paths for upgrades on RHEL

From product version	From operating system versions	To operating system versions	To product version
7.3.0.1	RHEL 7 Update 3	RHEL 7 Update 4	7.4.1
7.3.1	RHEL 7 Update 3	RHEL 7 Update 4	7.4.1

Upgrading the operating system and Veritas Access includes the following steps:

- Pre-upgrade steps only for the LTR-configured Veritas Access cluster.
- Export the Veritas Access configurations by using the script provided by Veritas Access
- Copy the configuration file
- Install RHEL 7.3 or 7.4
- Install Veritas Access 7.4.1
- Import the Veritas Access configurations

- Verify the imported Veritas Access configurations
- Post-upgrade steps only for the LTR-configured Veritas Access cluster

Pre-upgrade steps only for the LTR-configured Veritas Access cluster

---

Note: These steps are required when OpenDedup volumes are provisioned on the Veritas Access cluster.

---

1. Ensure that the backup or restore jobs from NetBackup are stopped.
2. If the upgrade is from 7.3.0.1, copy the `upgrade_scripts/odd_config_export_va7301.py` script from the ISO to the management console node.

   If the upgrade is from 7.3.1, copy the `upgrade_scripts/odd_config_export_va731.py` script from the ISO to the management console node.
3. Execute the respective script to export the OpenDedup configuration:

   For 7.3.0.1: `python odd_config_export_va7301.py [filename]`

   For 7.3.1: `python odd_config_export_va731.py [filename]`

   Note: If no file name is provided, the default config file name `odd_config.exp` is used.

To export the Veritas Access configurations

1. **Prerequisites:**

   Install the RHEL 7.3 version.

   Verify that the Veritas Access version 7.3.0.1 or 7.3.1 is installed.

   Make sure that you have stopped all I/Os and services related to Veritas Access by using the CLISH, such as CIFS, NFS, FTP, and so on.

   Stop all services by using the `hastop -all` command.
2. From the ISO, copy the `upgrade_scripts/config_export` directory on the cluster node on which the management console service group is online.
3. From the directory, run the following command on the shell (terminal) by using the `root` login to export the Veritas Access configurations:

   ```
   /bin/bash -f export_lib.sh export local <filename>
   ```

To verify the Veritas Access configuration export

- Run the following command on CLISH to see the list of available configurations:

  ```
  system config list
  ```

  The configuration files can be found in:

  ```
  /opt/VRTSnas/conf/backup
  ```

  Note: You need to store these configuration files on a node that is out of the cluster nodes to avoid any damage to the file.

To install RHEL 7.4

1 **Prerequisites:**

Make sure that you stop all the running modules on CLISH and no I/O is running.

Run the `network ip addr show` command and `cluster show` command on CLISH before you install RHEL 7.4. Make a note of these IP addresses and cluster node names. Make sure to use the same IP addresses and cluster name while installing the Veritas Access cluster after RHEL 7.4 is installed.

Examples:

```
upgrade> network ip addr show

IP              Netmask/Prefix  Device     Node            Type     Status
--              --------------  ------     ----            ----     ------
192.168.10.151 255.255.255.0   pubeth0    upgrade_01      Physical
192.168.10.158 255.255.255.0   pubeth1    upgrade_01      Physical
192.168.10.152 255.255.255.0   pubeth0    upgrade_02      Physical
192.168.10.159 255.255.255.0   pubeth1    upgrade_02      Physical
192.168.10.174 255.255.255.0   pubeth0    upgrade_01      Virtual  ONLINE (Con IP)
192.168.10.160 255.255.255.0   pubeth0    upgrade_01      Virtual  ONLINE
192.168.10.161 255.255.255.0   pubeth1    upgrade_01      Virtual  ONLINE

upgrade> cluster show

Node            State      CPU(15 min)    pubeth0(15 min)     pubeth1(15 min)
                                 %        rx(MB/s) tx(MB/s)   rx(MB/s) tx(MB/s)
----            -----      -----------    -------- --------   -------- --------
upgrade_01      RUNNING          11.52       0.67     0.06       0.60     0.00
upgrade_02      RUNNING           4.19       0.61     0.05       0.60     0.00
```

Note: In this example, the cluster name is `upgrade` and the cluster node names are `upgrade_01` and `upgrade_02`.

2 Restart all the nodes of the cluster.

3 Install RHEL 7.4 on the desired nodes.

See “Installing the operating system on the target Veritas Access cluster” on page 59.

Note: It is recommended to select the same disk or disks for the installation on which RHEL 7.3 was installed. Make sure that you do not select any other disk, because those disks may be part of a pool, and may result in data loss.

To install Veritas Access 7.4.1

- After a restart when the nodes are up, start the Veritas Access 7.4.1 installation by using the CPI.

Note: Make sure to use the same IP addresses and cluster name that were used for the Veritas Access installation on RHEL 7.3.

See “Installing Veritas Access on the target cluster nodes” on page 61.

To verify the Veritas Access installation

1 By using the console IP, check whether the CLISH is accessible.

2 Run the following command in CLISH to see whether the disks are accessible:

```
storage disk list
```

Note: If the disks are not visible in the CLISH output, run the `storage scanbus force` command in CLISH.

3 Run the following command in CLISH to see whether the pools are accessible:

```
storage pool list
```

Note: If the pools are not visible in the CLISH output, run the `storage scanbus force` command in CLISH.

4 Run the following command in CLISH to see whether the file systems are accessible:

```
storage fs list
```

Note: If the file systems are not visible in the CLISH output, run the `storage scanbus force` command in CLISH.

5 Make sure that the file systems are online. If the file systems are not online, you need to run the following command in CLISH to bring them online:

```
storage fs online <fs name>
```

To import the Veritas Access configuration

1 **Prerequisites:**

Make sure that the file systems are online. If the file systems are not online, you need to run the following command in CLISH to bring them online:

```
storage fs online <fs name>
```

Note: Make sure that the cluster uses the same IP addresses and cluster name that were used for the Veritas Access installation on RHEL 7.3.

If VIP addresses are not added during installation, which were used for Veritas Access on RHEL 7.3, add them from CLISH after Veritas Access is installed on RHEL 7.4, and then import the configuration.

2 Copy the exported configuration files to the cluster nodes in the following location:

```
/opt/VRTSnas/conf/backup/
```

3 Run the following command in CLISH to see the available exported configuration:

```
system config list
```

4 Log in to CLISH and import the module configuration by using the following command:

```
system config import local <config-filename> <module-to-import>
```

The following modules can be imported:

```
upgrade> system config import local

system config import local <file_name> [config-type]
 -- Import the configuration which is stored locally

file_name     : configuration file name
config-type   : input type of configuration to import (network/admin/all/report/
system/support/cluster_specific/all_except_cluster_specific/nfs/cifs/ftp/backup/
replication/storage_schedules/storage_quota/storage_fs_alert/storage_fs_policy/
compress_schedules/defrag_schedules/storage_dedup/smartio/target/object_access/
loadbalance/opendedup) [all]

upgrade> system config import local
```

Note: The module names are auto-suggested in CLISH.

Post-upgrade steps only for the LTR-configured Veritas Access cluster

Note: These steps are required in addition to the above steps when the OpenDedup volumes are provisioned on the Veritas Access cluster.

1 Enable or start the required authentication services (AD, LDAP, or NIS) that are used by the ObjectAccess service.

Note: If the upgrade is from Veritas Access 7.3.0.1, set the pool for ObjectAccess, and enable the ObjectAccess as follows.

```
Cluster1> objectaccess set pools pool1
ACCESS ObjectAccess INFO V-493-10-0 Set pools successful. Please make
sure the storage is provisioned as per the requirements of the layout.
Cluster1> objectaccess server enable
100% [********************] Enabling ObjectAccess server.
ACCESS ObjectAccess SUCCESS V-493-10-4 ObjectAccess server enabled.
```

2 Start the ObjectAccess service by using the following command:

```
cluster2> objectaccess server start
ACCESS ObjectAccess SUCCESS V-493-10-4 ObjectAccess started successfully.
```

3 Import the OpenDedup configuration by using the following command.

```
cluster2> system config import remote <file location> opendedup
```

Note: You can import the OpenDedup configuration that you have exported by using the steps provided in the section *Pre-upgrade steps only for the LTR-configured Veritas Access cluster*.

4 Offline all the OpenDedup volumes by using the following command:

```
cluster2> opendedup volume offline <vol-name>
```

5 Update all the OpenDedup `config.xml` files as follows:

`"/etc/sdfs/<vol-name>-volume-cfg.xml`

by adding following parameter to the `<extended-config>` tag:

`dist-layout="false"`

Note: This parameter should not be used for the existing OpenDedup volumes because they may have existing data with the default layout. If you use the existing OpenDedup volumes, it may result in data corruption.

---

6 To bring all the OpenDedup volumes Online, use the following command:

```
cluster2> opendedup volume online <vol-name>
```

Chapter 10

Upgrading Veritas Access using a rolling upgrade

This chapter includes the following topics:

About the rolling upgrades

This release of Veritas Access supports rolling upgrades from the Veritas Access 7.3.0.1 and later versions. Rolling upgrade is supported on RHEL 7.3 and 7.4.

A rolling upgrade minimizes the service and application downtime for highly available clusters by limiting the upgrade time to the amount of time that it takes to perform a service group failover. Nodes with different product versions can be run in one cluster.

The rolling upgrade has two main phases. The installer upgrades kernel RPMs in phase 1 and VCS agent RPMs in phase 2. Upgrade should be done on each node individually one by one. You need to perform upgrade first on an each slave node and thereafter on the master node. The upgrade process stops all services and resources on the node, which is being upgraded. All services (including the VIP groups) fail over to the one of the other nodes from the cluster. During the failover process, the clients that are connected to the VIP groups of nodes are intermittently interrupted. For those clients that do not time-out, the service is resumed after the VIP groups become online on the node that is being upgraded.

While the upgrade process is running on the first node, other nodes of the cluster continue to serve the clients. After the first node has been upgraded, it restarts the

services and resources on the first-stage node. After the first node comes up, the upgrade process stops the services and resources on the next slave node and so on. All services and resources are online and serve clients. Meanwhile, the rolling upgrade starts the upgrade process on the remaining nodes. After the upgrade is complete on the remaining nodes, the cluster recovers and services are balanced across the cluster.

Workflow for the rolling upgrade

A rolling upgrade has two main phases where the installer upgrades the kernel RPMs in Phase 1 and VCS agent-related non-kernel RPMs in Phase 2.

1. Disable the I/O fencing before you start the rolling upgrade.
2. The upgrade process is performed on each node one after another.
3. In phase 1, the upgrade process is performed on the slave nodes first. The upgrade process stops all services on the node and the group of services are failed over to an another node in the cluster.
4. During the failover process, the clients that are connected to the VIP groups of the nodes are intermittently interrupted. For those clients that do not time out, the service is resumed after the VIP groups become online on one of the nodes.
5. During phase 1, the installer upgrades the kernel RPMs on the node and the other nodes continue to serve the clients.
6. After the phase 1 for the first slave node is complete, upgrade is started for the second slave node and so on. After master node of the slave node is upgraded, all the service groups from the master node are failed over to some other node.
7. After phase 1 for the first node is successful, you need to check if recovery task is also complete before starting upgrade phase 1 for the next node.

   Note: You need to verify that the upgraded node is not out of the cluster by running the `vxclustadm nidmap`. If it shows that the node is out of cluster, wait for the node to join the existing cluster.

8. During Phase 2 of the rolling upgrade, all remaining RPMs are upgraded on all the nodes of the cluster simultaneously. VCS and VCS-agent packages are upgraded. The kernel drivers are upgraded to the new protocol version. Applications stay online during Phase 2. The `High Availability` daemon (HAD) stops and starts again.

See "Performing a rolling upgrade using the installer" on page 141.

See “Supported rolling upgrade paths for upgrades on RHEL and Oracle Linux” on page 141.

Supported rolling upgrade paths for upgrades on RHEL and Oracle Linux

Table 10-1 Supported upgrade paths for upgrades on RHEL and Oracle Linux

From product version	Operating system versions	To product version
7.3.0.1	RHEL 7 Update 3 and 4	7.4.1
7.3.1	RHEL 7 Update 3 and 4 OL 7 Update 4	7.4.1

Note: See the "Known issues" section of the *Veritas Access Release Notes* before starting a rolling upgrade for other product versions not shown in this table.

See “Performing a rolling upgrade using the installer” on page 141.

See “About the rolling upgrades” on page 139.

Performing a rolling upgrade using the installer

Before you start a rolling upgrade, make sure that the Cluster Server (VCS) is running on all the nodes of the cluster.

You need to stop all activities for all the VxVM volumes that are not under the VCS control. For example, stop any applications such as databases that can access the volumes, and unmount any file systems that have been created on the volumes. Then stop all the volumes.

Unmount all the VxFS file systems that are not under VCS control.

To perform a rolling upgrade

1. In case of the LTR-configured Veritas Access cluster, make sure that the backup or restore jobs from NetBackup are stopped.
2. Phase 1 of a rolling upgrade begins on the second subcluster. Complete the preparatory steps on the second subcluster.

   Unmount all VxFS file systems not under VCS control:

   ```
   # umount mount_point
   ```

3. Complete the updates to the OS, if required.

   Make sure that the existing version of Veritas Access supports the OS updates that you apply. If the existing version of Veritas Access does not support the OS update, first upgrade Veritas Access to a version that supports the OS update.

   For more information, see the RHEL OS documentation.

   Switch the applications to the remaining subcluster and upgrade the OS of the first subcluster.

   The nodes are restarted after the OS updates are completed.
4. If a cache area is online, you must take the cache area offline before you upgrade the VxVM RPMs. Use the following command to take the cache area offline:

   ```
   # sfcache offline cachename
   ```

5. Disable I/O fencing before you perform the rolling upgrade by using the `storage fencing off` command.
6. Log on as a root user and mount the Veritas Access 7.4.1 installation media.
7. From root, start the installer.

   ```
   # ./installaccess -rolling_upgrade
   ```

8. The installer checks system communications, release compatibility, version information, and lists the cluster name, ID, and cluster nodes. The installer asks for permission to proceed with the rolling upgrade.

   ```
   Would you like to perform rolling upgrade on the cluster? [y,n,q] (y)
   ```

   Type **y** to continue.

9 Phase 1 of the rolling upgrade begins. Phase 1 must be performed on one node at a time. The installer asks for the system name.

```
Enter the system names separated by spaces on which you want to perform r
Enter the name or IP address of one of the slave node on which you want t
```

10 The installer performs further prechecks on the nodes in the cluster and may present warnings. You can type **y** to continue or quit the installer and address the precheck's warnings.

11 If the boot disk is encapsulated and mirrored, you can create a backup boot disk.

If you choose to create a backup boot disk, type **y**. Provide a backup name for the boot disk group or accept the default name. The installer then creates a backup copy of the boot disk group.

12 After the installer detects the online service groups, the installer prompts the user to do one of the following:

- Manually switch service groups
- Use the CPI to automatically switch service groups

The downtime is the time that it takes for the failover of the service group.

Note: Veritas recommends that you manually switch the service groups. Automatic switching of service groups does not resolve dependency issues.

---

13 The installer prompts you to stop the applicable processes. Type **y** to continue.

The installer evacuates all service groups to the node or nodes that are not upgraded at this time. The installer stops parallel service groups on the nodes that are to be upgraded.

The installer stops all the related processes, uninstalls the old kernel RPMs, and installs the new RPMs.

14 The installer performs the upgrade configuration and starts the processes. If the boot disk is encapsulated before the upgrade, the installer prompts you to restart the node after performing the upgrade configuration.

15 Complete the preparatory steps on the nodes that you have not yet upgraded.

Unmount all the VxFS file systems that are not under VCS control on all the nodes.

```
# umount mount_point
```

16 If the OS updates are not required, skip this step.

Go to step 4.

Else, complete updates to the OS on the nodes that you have not yet upgraded. For the instructions, see the RHEL OS documentation.

Repeat steps 4 to 14 for each node.

17 Phase 1 of the rolling upgrade is complete for the first node. You can start with the upgrade of phase 1 for the next slave node. Installer again asks for the system name.

Before you start the upgrade of phase 1 for the next node, you need to check if the recovery task is in-progress. You need to wait for a few minutes for the recovery task to start.

On the master node, enter the following command:

```
# vxtask list

Check if following keywords are present:
ECREBUILD/ATCOPY/ATCPY/PLXATT/VXRECOVER/RESYNC/RECOV
```

If any recovery task is in progress, wait for the task to complete, and then start the upgrade of phase 1 for the next node.

18 After the upgrade of phase 1 is done on the node, make sure that the node is not out of the cluster.

Enter the `# vxclustadm nidmap` command.

If the upgraded node is out of the cluster, wait for the node to join the cluster before you start the upgrade of phase 1 for the next node.

19 Set up all cache areas as offline on the remaining node or nodes:

```
# sfcache offline cachename
```

The installer asks for a node name on which the upgrade is to be performed.

20 Type the system names on which you want to perform the rolling upgrade.

```
Enter the system names separated by spaces on which you want to perform r
```

21 Type the cluster node name.

```
Type cluster node name or q to quit.
```

The installer repeats step 9 through step 14.

For clusters with a larger number of nodes, this process may repeat several times. Service groups come down and are brought up to accommodate the upgrade.

22 When phase 1 of the rolling upgrade completes, mount all the VxFS file systems that are not under VCS control manually. Begin phase 2 of the upgrade. Phase 2 of the upgrade includes downtime for the VCS engine (HAD), which does not include application downtime. Type **y** to continue. Phase 2 of the rolling upgrade begins here.

23 The installer determines the remaining RPMs to upgrade. Type **y** to continue.

24 The installer stops Cluster Server (VCS) processes but the applications continue to run. Type **y** to continue.

The installer performs a prestop, uninstalls the old RPMs, and installs the new RPMs. It performs post-installation tasks and the configuration for the upgrade.

25 If you have a network connection to the Internet, the installer checks for updates.

If any updates are discovered, you can apply them now.

26 Verify the cluster's status:

```
# hastatus -sum
```

27 Post-upgrade steps only for the LTR-configured Veritas Access cluster:

Take offline all the OpenDedup volumes by using the following command:

```
cluster2> opendedup volume offline <vol-name>
```

Update all the OpenDedup `config.xml` files as follows:

```
/etc/sdfs/<vol-name>-volume-cfg.xml
```

by adding the following parameter to the `extended-config` tag:

```
dist-layout= "false"
```

Note: This parameter should not be used for the existing OpenDedup volumes because they might have existing data with the default layout. If you use the existing OpenDedup volumes, it might result in data corruption.

Bring online all the OpenDedup volumes by using the following command:

```
cluster2> opendedup volume online <vol-name>
```

See "Supported rolling upgrade paths for upgrades on RHEL and Oracle Linux" on page 141.

See "About the rolling upgrades" on page 139.

Chapter 11

Uninstalling Veritas Access

This chapter includes the following topics:

Before you uninstall Veritas Access

Perform the following steps before uninstalling Veritas Access:

- Before you remove Veritas Access from any node (but not in all the nodes) in a cluster, make sure the node has already been deleted from the running cluster. You can use the `Cluster> show` command to view the cluster node state, and use the `Cluster> delete` command to delete a running node from the Veritas Access cluster.
  See the relevant man pages for more information on the `Cluster> show` and `Cluster> delete` commands.
- Stop all the applications that access the file system over NFS, CIFS, or FTP.
- Destroy all the replication jobs from the cluster.
  Use the `Replication> job show` command to list all the replication jobs on the cluster.

```
Replication> job show
Job Name Role Job Type Encryption Debug Schedule
======== ====== ======== ========== ===== ========
job1 SOURCE DATA OFF ON sch1
State CKPT Count Exclunit Source repunit Target repunit(s)
======== ========== ======== ============== =================
```

```
ENABLED 1 -- scr1 trg1
Link name(s)
============
link1
```

Use the `Replication> job destroy` command to destroy the replication jobs.

```
Replication> job destroy job1
ACCESS replication SUCCESS V-288-0 Removing bandwidth limit on the
link: link1
ACCESS replication SUCCESS V-288-0 Job 'job1' disabled successfully.
ACCESS replication SUCCESS V-288-0 Job 'job1' deleted successfully.
```

- Stop the NFS, CIFS, FTP, GUI, and the replication service on the cluster using the appropriate CLISH command.

```
CLISH> cifs server stop
Stopping CIFS Server.....Success.
CLISH>
CLISH> nfs server stop
Success.
CLISH>
CLISH> ftp server stop
Success.
CLISH>
CLISH.Support> gui server stop
GUI service is OFFLINE.
CLISH>
CLISH> replication service stop
ACCESS replication SUCCESS V-288-0 Replication service stopped
CLISH>
```

- Run the following command to stop the Automated Monitoring Framework (AMF) service:

```
# /etc/init.d/amf stop
Stopping AMF...
AMF: Module unloaded
```

- Run the following command and wait for a couple of minutes:

```
# /opt/VRTS/bin/hastop -all
```

- Run the following command and verify that you only see Port a and Port b:

```
# gabconfig -a
GAB Port Memberships
===================================
Port a gen 7f2d0a membership 01
Port b gen 7f2d09 membership 01
```

Uninstalling Veritas Access using the installer

You can perform an uninstallation of Veritas Access. The Veritas Access uninstall program lets you uninstall Veritas Access without requiring a reinstallation of the operating system. You can also use the uninstall program in cases where there was an incomplete installation of Veritas Access.

Before you use the uninstall program to uninstall Veritas Access on all the nodes in the cluster at the same time, make sure that communication exists between the nodes. By default, Veritas Access cluster nodes can communicate with each other using ssh.

If the nodes cannot communicate with each other, then you must run the uninstall program on each node in the cluster. The uninstall program removes all Veritas Access RPMs.

Removing Veritas Access 7.4.1 RPMs

The uninstall program stops the Veritas Access processes that are currently running during the uninstallation process.

To uninstall Veritas Access 7.4.1 RPMs

1. Log in as the support user from the node where you want to uninstall Veritas Access.
2. Start the uninstall program.

   ```
   # cd /opt/VRTS/install

   # ./uninstallaccess
   ```

   The program specifies the directory where the logs are created. The program displays a copyright notice and a description of the cluster.

3 Enter the IP addresses of the nodes from which you want to uninstall Veritas Access.

The program performs node verification checks and asks to stop all running Veritas Access processes.

4 Enter **y** to stop all the Veritas Access processes.

The program stops the Veritas Access processes and uninstalls the software.

The uninstall program does the following tasks:

- Verifies the communication between nodes.
- Checks the installations on each node to determine the RPMs to be uninstalled.
- Unloads kernel modules and removes the RPMs.

Review the output as the uninstaller stops processes.

You can make a note of the location of the summary, response, and log files that the uninstaller creates after removing all the RPMs.

Running uninstall from the Veritas Access 7.4.1 disc

You may need to use the uninstall program on the Veritas Access 7.4.1 disc in one of the following cases:

- You need to uninstall Veritas Access after an incomplete installation.
- The uninstall program is not available in `/opt/VRTS/install`.

If you mounted the installation media to `/mnt`, access the uninstall program by changing the directory.

```
cd /mnt/

./uninstallaccess
```

Appendix A

Installation reference

This appendix includes the following topics:

Installation script options

Table A-1 lists the available command-line options for the Veritas Access installation script. For an initial install or upgrade, options are not usually required.

Table A-1 Available command-line options

Command Line Option	Function
-configure	Configures an unconfigured product after it is installed.
-install	Installs the product on systems.
-precheck	Performs checks to confirm that systems have met the products installation requirements before installing the product.
-license	Registers or updates product licenses on the specified systems.
-licensefile	Specifies the location of the Veritas perpetual or subscription license key file.
-requirements	Displays the required operating system version, required patches, file system space, and other system requirements to install the product.

Table A-1 Available command-line options *(continued)*

Command Line Option	Function
-responsefile *response_file*	Performs automated installations or uninstallations using information stored in a file rather than prompting for the information. *response_file* is the full path of the file that contains the configuration definitions.
-rolling_upgrade	Performs a rolling upgrade. Using this option, the installer detects the rolling upgrade status on cluster systems automatically without the need to specify rolling upgrade phase 1 or phase 2 explicitly.
-prestop_script *prestop_script*	Executes the customized script provided by user on each host before stop processes during the upgrade procedure.
-poststart_script *poststart_script*	Executes the customized script provided by user on each host after start processes during the upgrade procedure.
-uninstall	Uninstalls the product from systems.
-updateparameter	Updates the network parameter for a running cluster.

Appendix B

Configuring the secure shell for communications

This appendix includes the following topics:

Manually configuring passwordless SSH

You can use the SSH to log into and execute commands on a remote system. SSH enables encrypted communications and an authentication process between two untrusted hosts over an insecure network.

In this procedure, you first create a DSA key pair. From the key pair, you append the public key from the source system to the `authorized_keys` file on the target systems.

To create the DSA key pair

1 On the source system (sys1), log in as **root** user, and navigate to the root directory.

```
sys1 # cd /root
```

2 To generate a DSA key pair on the source system, type the following command:

```
sys1 # ssh-keygen -t dsa
```

System output similar to the following is displayed:

```
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa):
```

3 Press **Enter** to accept the default location of `/root/.ssh/id_dsa`.

4 When the program asks you to enter the pass phrase, press the **Enter** key twice.

```
Enter passphrase (empty for no passphrase):
```

Do not enter a pass phrase. Press **Enter**.

```
Enter same passphrase again:
```

Press **Enter** again.

5 Output similar to the following lines appears.

```
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
1f:00:e0:c2:9b:4e:29:b4:0b:6e:08:f8:50:de:48:d2 root@sys1
```

To append the public key from the source system to the `authorized_keys` file on the target system using secure file transfer

1. From the source system (sys1), move the public key to a temporary file on the target system (sys2).

   Use the secure file transfer program.

   In this example, the file name `id_dsa.pub` in the root directory is the name for the temporary file for the public key.

   Use the following command for secure file transfer:

   ```
   sys1 # sftp sys2
   ```

   If the secure file transfer is set up for the first time on this system, output similar to the following lines is displayed:

   ```
   Connecting to sys2 ...
   The authenticity of host 'sys2 (10.182.00.00)'
   can't be established. DSA key fingerprint is
   fb:6f:9f:61:91:9d:44:6b:87:86:ef:68:a6:fd:88:7d.
   Are you sure you want to continue connecting (yes/no)?
   ```

2. Type `yes`.

   Output similar to the following is displayed:

   ```
   Warning: Permanently added 'sys2,10.182.00.00'
   (DSA) to the list of known hosts.
   root@sys2 password:
   ```

3. Enter the root password of sys2.

4. At the `sftp` prompt, type the following command:

   ```
   sftp> put /root/.ssh/id_dsa.pub
   ```

   The following output is displayed:

   ```
   Uploading /root/.ssh/id_dsa.pub to /root/id_dsa.pub
   ```

5. To quit the SFTP session, type the following command:

   ```
   sftp> quit
   ```

6 Add the `id_dsa.pub` keys to the `authorized_keys` file on the target system. To begin the SSH session on the target system (sys2 in this example), type the following command on sys1:

```
sys1 # ssh sys2
```

Enter the root password of sys2 at the prompt:

```
password:
```

Type the following commands on sys2:

```
sys2 # cat /root/id_dsa.pub >> /root/.ssh/authorized_keys
sys2 # rm  /root/id_dsa.pub
```

7 Run the following commands on the source installation system. If your SSH session has expired or terminated, you can also run these commands to renew the session. These commands fetch the private key into the shell environment and make the key globally available to the root user.

```
sys1 # exec /usr/bin/ssh-agent $SHELL
sys1 # ssh-add

Identity added: /root/.ssh/id_dsa
```

This shell-specific step is valid only while the shell is active. You must execute the procedure again if you close the SSH during the session.

To verify that you can connect to a target system

1 On the source system (sys1), enter the following command:

```
sys1 # ssh -l root sys2 uname -a
```

where sys2 is the name of the target system.

2 The command should execute from the source system (sys1) to the target system (sys2) without the system requesting a pass phrase or password.

3 Repeat this procedure for each target system.

Setting up the SSH and the RSH connections

You can use the `pwdutil.pl` utility to set up the SSH and the RSH connections automatically. This utility can be located at `/opt/VRTS/repository/ga/images/SSNAS/7.4.0.0/scripts/pwdutil.pl`.

```
# ./pwdutil.pl -h
Usage:

Command syntax with simple format:

    pwdutil.pl check|configure|unconfigure ssh|rsh <hostname|IP addr>
    [<user>] [<password>] [<port>]

Command syntax with advanced format:

    pwdutil.pl [--action|-a 'check|configure|unconfigure']
               [--type|-t 'ssh|rsh']
               [--user|-u  '<user>']
               [--password|-p '<password>']
               [--port|-P '<port>']
               [--hostfile|-f '<hostfile>']
               [--keyfile|-k '<keyfile>']
               [-debug|-d]
               <host_URI>

    pwdutil.pl -h | -?
```

Table B-1 Options with pwdutil.pl utility

Option	Usage
--action|-a 'check|configure|unconfigure'	Specifies the action type. The default value is 'check'.
--type|-t 'ssh|rsh'	Specifies the connection type. The default value is 'SSH'.
--user|-u '<user>'	Specifies the user ID. The default value is the local user ID.
--password|-p '<password>'	Specifies the user password. The default value is the user ID.
--port|-P '<port>'	Specifies the port number for the SSH connection. The default value is 22.
--keyfile|-k '<keyfile>'	Specifies the private key file.
--hostfile|-f '<hostfile>'	Specifies the file which lists the hosts.
-debug	Prints the debug information.

Table B-1 Options with pwdutil.pl utility *(continued)*

Option	Usage
-h|-?	Prints the help messages.
<*host_URI*>	Can be in the following formats: *hostname* *user:password@hostname* *user:password@hostname:* *port*

You can check, configure, and unconfigure SSH or RSH using the `pwdutil.pl` utility. For example:

- To check SSH connection for only one host:

  ```
  pwdutil.pl check ssh hostname
  ```

- To configure SSH for only one host:

  ```
  pwdutil.pl configure ssh hostname user password
  ```

- To unconfigure RSH for only one host:

  ```
  pwdutil.pl unconfigure rsh hostname
  ```

- To configure SSH for multiple hosts with the same user ID and password:

  ```
  pwdutil.pl -a configure -t ssh -u user -p password hostname1
  hostname2 hostname3
  ```

- To configure SSH or RSH for different hosts with a different user ID and password:

  ```
  pwdutil.pl -a configure -t ssh user1:password1@hostname1
  user2:password2@hostname2
  ```

- To check or configure SSH or RSH for multiple hosts with one configuration file:

  ```
  pwdutil.pl -a configure -t ssh --hostfile /tmp/sshrsh_hostfile
  ```

- To keep the host configuration file safe, you can use the 3rd-party utility to encrypt and decrypt the host file with password.

For example:

```
### run openssl to encrypt the host file in base64 format
# openssl aes-256-cbc -a -salt -in /hostfile -out /hostfile.enc
enter aes-256-cbc encryption password: <password>
Verifying - enter aes-256-cbc encryption password: <password>

### remove the original plain text file
# rm /hostfile

### run openssl to decrypt the encrypted host file
# pwdutil.pl -a configure -t ssh `openssl aes-256-cbc -d -a
-in /hostfile.enc`
enter aes-256-cbc decryption password: <password>
```

- To use the ssh authentication keys that are not under the default $HOME/.ssh directory, you can use --keyfile option to specify the ssh keys. For example:

```
### create a directory to host the key pairs:
# mkdir /keystore

### generate private and public key pair under the directory:
# ssh-keygen -t rsa -f /keystore/id_rsa

### setup ssh connection with the new generated key pair under
the directory:
# pwdutil.pl -a configure -t ssh --keyfile /keystore/id_rsa
user:password@hostname
```

You can see the contents of the configuration file by using the following command:

```
# cat /tmp/sshrsh_hostfile
user1:password1@hostname1
user2:password2@hostname2
user3:password3@hostname3
user4:password4@hostname4

# all default: check ssh connection with local user
hostname5
The following exit values are returned:

0     Successful completion.
1     Command syntax error.
2     Ssh or rsh binaries do not exist.
```

```
3     Ssh or rsh service is down on the remote machine.
4     Ssh or rsh command execution is denied due to password is required.
5     Invalid password is provided.
255   Other unknown error.
```

Appendix C

Manual deployment of Veritas Access

This appendix includes the following topics:

Deploying Veritas Access manually on a two-node cluster in a non-SSH environment

This section describes the manual steps for deploying a two-node Veritas Access cluster when SSH communication is disabled.

Pre-requisites

- You need to have a two-node cluster.
- Supported operating system version is: RHEL 7.4
- Verify that the Veritas Access image is present in your local system at the `/access_build_dir/rhel7_x86_64/` location.
- The cluster is named as *clus* and the cluster nodes are named as *clus_01* and *clus_02*. Cluster names should be unique for all nodes.
- You need to stop the SSH service on all the nodes.
- Verify that the public NICs are pubeth0, pubeth1, and private NICs are priveth0 and priveth1. NIC names should be consistent across all the nodes. Public NIC names and private NIC names should be the same across all the nodes.

- Use `172.16.0.3` as the private IP address for *clus_01* and `172.16.0.4` as the private IP address for *clus_02*.

To deploy Veritas Access manually on a two-node cluster

1. Copy the Veritas Access image on all the nodes of the desired cluster.
2. Stop the `SSH` daemon on all the nodes.

   ```
   # systemctl stop sshd
   ```

3. Verify if the following RPMs are installed. If not, install the RPMs from the RHEL repository.

   ```
   bash-4.2.46-28.el7.x86_64
   lsscsi-0.27-6.el7.x86_64
   initscripts-9.49.39-1.el7.x86_64
   iproute-3.10.0-87.el7.x86_64
   kmod-20-15.el7.x86_64
   coreutils-8.22-18.el7.x86_64
   binutils-2.25.1-31.base.el7.x86_64
   python-requests-2.6.0-1.el7_1.noarch
   python-urllib3-1.10.2-3.el7.noarch
   ```

4. Install the required operating system RPMs.
   - Create a `repo` file.

     ```
     cat /etc/yum.repos.d/os.repo
       [veritas-access-os-rpms]
       name=Veritas Access OS RPMS
       baseurl=file:///access_build_dir/rhel7_x86_64/os_rpms/
       enabled=1
       gpgcheck=0
     ```

   - Run the following command:

     ```
     # yum updateinfo
     ```

   - Run the following command:

     ```
     # cd /access_build_dir/rhel7_x86_64/os_rpms/
     ```

   - Before running the following command, make sure that there is no RHEL subscription in the system. The `yum repolist` should point to `veritas-access-os-rpms` only.

```
# /usr/bin/yum -y install --setopt=protected_multilib=false
perl-5.16.3-292.el7.x86_64.rpm nmap-ncat-6.40-7.el7.x86_64.rpm
perl-LDAP-0.56-5.el7.noarch.rpm perl-Convert-ASN1-0.26-4.el7.noarch.rpm
net-snmp-5.7.2-28.el7_4.1.x86_64.rpm
net-snmp-utils-5.7.2-28.el7_4.1.x86_64.rpm
openldap-2.4.44-5.el7.x86_64.rpm nss-pam-ldapd-0.8.13-8.el7.x86_64.rpm
rrdtool-1.4.8-9.el7.x86_64.rpm wireshark-1.10.14-14.el7.x86_64.rpm
vsftpd-3.0.2-22.el7.x86_64.rpm openssl-1.0.2k-12.el7.x86_64.rpm
openssl-devel-1.0.2k-12.el7.x86_64.rpm
iscsi-initiator-utils-6.2.0.874-4.el7.x86_64.rpm
libpcap-1.5.3-9.el7.x86_64.rpm libtirpc-0.2.4-0.10.el7.x86_64.rpm
nfs-utils-1.3.0-0.48.el7_4.2.x86_64.rpm
kernel-debuginfo-common-x86_64-3.10.0-693.el7.x86_64.rpm
kernel-debuginfo-3.10.0-693.el7.x86_64.rpm
kernel-headers-3.10.0-693.el7.x86_64.rpm
krb5-devel-1.15.1-8.el7.x86_64.rpm
krb5-libs-1.15.1-8.el7.x86_64.rpm
krb5-workstation-1.15.1-8.el7.x86_64.rpm
perl-JSON-2.59-2.el7.noarch.rpm telnet-0.17-64.el7.x86_64.rpm
apr-devel-1.4.8-3.el7_4.1.x86_64.rpm
apr-util-devel-1.5.2-6.el7.x86_64.rpm
glibc-common-2.17-196.el7_4.2.x86_64.rpm
glibc-headers-2.17-196.el7_4.2.x86_64.rpm
glibc-2.17-196.el7_4.2.x86_64.rpm glibc-2.17-196.el7_4.2.i686.rpm
glibc-devel-2.17-196.el7_4.2.x86_64.rpm
glibc-utils-2.17-196.el7_4.2.x86_64.rpm
nscd-2.17-196.el7_4.2.x86_64.rpm sysstat-10.1.5-12.el7.x86_64.rpm
libibverbs-utils-13-7.el7.x86_64.rpm libibumad-13-7.el7.x86_64.rpm
opensm-3.3.19-1.el7.x86_64.rpm opensm-libs-3.3.19-1.el7.x86_64.rpm
infiniband-diags-1.6.7-1.el7.x86_64.rpm
sg3_utils-libs-1.37-12.el7.x86_64.rpm sg3_utils-1.37-12.el7.x86_64.rpm
libyaml-0.1.4-11.el7_0.x86_64.rpm
memcached-1.4.15-10.el7_3.1.x86_64.rpm
python-memcached-1.59-1.noarch.rpm
python-paramiko-2.1.1-4.el7.noarch.rpm
python-backports-1.0-8.el7.x86_64.rpm
python-backports-ssl_match_hostname-3.4.0.2-4.el7.noarch.rpm
python-chardet-2.2.1-1.el7_1.noarch.rpm
python-six-1.9.0-2.el7.noarch.rpm
python-setuptools-0.9.8-7.el7.noarch.rpm
python-ipaddress-1.0.16-2.el7.noarch.rpm
targetcli-2.1.fb46-1.el7.noarch.rpm
```

```
fuse-2.9.2-8.el7.x86_64.rpm fuse-devel-2.9.2-8.el7.x86_64.rpm
fuse-libs-2.9.2-8.el7.x86_64.rpm PyYAML-3.10-11.el7.x86_64.rpm
arptables-0.0.4-8.el7.x86_64.rpm ipvsadm-1.27-7.el7.x86_64.rpm
ntpdate-4.2.6p5-25.el7_3.2.x86_64.rpm ntp-4.2.6p5-25.el7_3.2.x86_64.rpm
autogen-libopts-5.18-5.el7.x86_64.rpm ethtool-4.8-1.el7.x86_64.rpm
net-tools-2.0-0.22.20131004git.el7.x86_64.rpm
cups-libs-1.6.3-29.el7.x86_64.rpm avahi-libs-0.6.31-17.el7.x86_64.rpm
psmisc-22.20-15.el7.x86_64.rpm strace-4.12-4.el7.x86_64.rpm
vim-enhanced-7.4.160-2.el7.x86_64.rpm at-3.1.13-22.el7_4.2.x86_64.rpm
rsh-0.17-76.el7_1.1.x86_64.rpm unzip-6.0-16.el7.x86_64.rpm
zip-3.0-11.el7.x86_64.rpm bzip2-1.0.6-13.el7.x86_64.rpm
mlocate-0.26-6.el7.x86_64.rpm lshw-B.02.18-7.el7.x86_64.rpm
jansson-2.10-1.el7.x86_64.rpm ypbind-1.37.1-9.el7.x86_64.rpm
yp-tools-2.14-5.el7.x86_64.rpm perl-Net-Telnet-3.03-19.el7.noarch.rpm
tzdata-java-2018d-1.el7.noarch.rpm
perl-XML-Parser-2.41-10.el7.x86_64.rpm
lsof-4.87-4.el7.x86_64.rpm cairo-1.14.8-2.el7.x86_64.rpm
pango-1.40.4-1.el7.x86_64.rpm libjpeg-turbo-1.2.90-5.el7.x86_64.rpm
sos-3.4-13.el7_4.noarch.rpm traceroute-2.0.22-2.el7.x86_64.rpm
openldap-clients-2.4.44-5.el7.x86_64.rpm
```

5 Install the third-party RPMs:

```
# cd /access_build_dir/rhel7_x86_64/ third_party _rpms/
# /bin/rpm -U -v --oldpackage --nodeps --replacefiles --replacepkgs
ctdb-4.6.6-1.el7.x86_64.rpm
perl-Template-Toolkit-2.24-5.el7.x86_64.rpm
perl-Template-Extract-0.41-1.noarch.rpm
perl-AppConfig-1.66-20.el7.noarch.rpm
perl-File-HomeDir-1.00-4.el7.noarch.rpm
samba-common-4.6.6-1.el7.x86_64.rpm
samba-common-libs-4.6.6-1.el7.x86_64.rpm
samba-client-4.6.6-1.el7.x86_64.rpm
samba-client-libs-4.6.6-1.el7.x86_64.rpm
samba-4.6.6-1.el7.x86_64.rpm
samba-winbind-4.6.6-1.el7.x86_64.rpm
samba-winbind-clients-4.6.6-1.el7.x86_64.rpm
samba-winbind-krb5-locator-4.6.6-1.el7.x86_64.rpm
libsmbclient-4.6.6-1.el7.x86_64.rpm
samba-krb5-printing-4.6.6-1.el7.x86_64.rpm
samba-libs-4.6.6-1.el7.x86_64.rpm
libwbclient-4.6.6-1.el7.x86_64.rpm
samba-winbind-modules-4.6.6-1.el7.x86_64.rpm
libnet-1.1.6-7.el7.x86_64.rpm lmdb-libs-0.9.13-2.el7.x86_64.rpm
nfs-ganesha-2.2.0-0.el7.x86_64.rpm
nfs-ganesha-vxfs-2.2.0-0.el7.x86_64.rpm gevent-1.0.2-1.x86_64.rpm
python-msgpack-0.4.6-1.el7ost.x86_64.rpm
python-flask-0.10.1-4.el7.noarch.rpm
python-itsdangerous-0.23-2.el7.noarch.rpm
libevent-libs-2.0.22-1.el7.x86_64.rpm
python-werkzeug-0.9.1-2.el7.noarch.rpm
python-jinja2-2.7.2-2.el7.noarch.rpm sdfs-7.4.0.0-1.x86_64.rpm
psutil-4.3.0-1.x86_64.rpm
python-crontab-2.2.4-1.noarch.rpm libuv-1.9.1-1.el7.x86_64.rpm
```

In this command, you can update the RPM version based on the RPMs in the `/access_build_dir/rhel7_x86_64/third_party _rpms/` directory.

6 Install the Veritas Access RPMs.

- Run the following commands:

```
# cd /access_build_dir/rhel7_x86_64/rpms/repodata/
# cat access73.repo > /etc/yum.repos.d/access73.repo
```

- Update the *baseurl* and *gpgkey* entry in the `/etc/yum.repos.d/access73.repo` for yum repository directory.
  - `baseurl=file:///access_build_dir/rhel7_x86_64/rpms/`
  - `gpgkey=file:///access_build_dir/rhel7_x86_64/rpms/RPM-GPG-KEY-veritas-access7`
- Run the following commands to refresh the yum repository.
  - `# yum repolist`
  - `# yum grouplist`
- Run the following command.

  `# yum -y groupinstall ACCESS73`
- Run the following command.

  `# /opt/VRTS/install/bin/add_install_scripts`

7 Install the Veritas NetBackup client software.

```
# cd /access_build_dir/rhel7_x86_64
# /opt/VRTSnas/install/image_install/netbackup/install_netbackup.pl
/access_build_dir/rhel7_x86_64/netbackup
```

8 Create soft links for Veritas Access. Run the following command.

```
# /opt/VRTSnas/pysnas/install/install_tasks.py
all_rpms_installed parallel
```

9 License the product.

- Register the permanent VLIC key.

  `# /opt/VRTSvlic/bin/vxlicinstupgrade -k <Key>`
- Verify that the VLIC key is installed properly:

  `# /opt/VRTSvlic/bin/vxlicrep`
- Register the SLIC key file:

  `# /opt/VRTSslic/bin/vxlicinstupgrade -k $keyfile`

- Verify that the SLIC key is installed properly:

  ```
  # /opt/VRTSslic/bin/vxlicrep
  ```

10 Take a backup of the following files:

- `/etc/sysconfig/network`
- `/etc/sysconfig/network-scripts/ifcfg-*`
- `/etc/resolv.conf`

11 Configure the private NIC:

```
# cd /etc/sysconfig/network-scripts/
```

- Configure the first private NIC.
  - Run the following command.

    ```
    # ip link set down priveth0
    ```

  - Update the `ifcfg-priveth0` file with the following:

    ```
    DEVICE=priveth0
    NAME=priveth0
    BOOTPROTO=none
    TYPE=Ethernet
    ONBOOT=yes
    ```

  - Add entries in the `ifcfg-priveth0` file.

    ```
    HWADDR=<MAC address>
    IPADDR= 172.16.0.3  (use IPADDR= 172.16.0.4 for second node)
    NETMASK=<netmask>
    NM_CONTROLLED=no
    ```

    For example:

    ```
    HWADDR=00:0c:29:0c:8d:69
    IPADDR=172.16.0.3
    NETMASK=255.255.248.0
    NM_CONTROLLED=no
    ```

  - Run the following command.

    ```
    # ip link set up priveth0
    ```

- Configure the second private NIC.
  You can configure the second private NIC in the same way. Instead of priveth0, use priveth1 for the second node. You do not need to provide IPADDR for priveth1.

12 Configure the public NIC.

```
# cd /etc/sysconfig/network-scripts/
```

- Configure the second public NIC, pubeth1 (in which the host IP is not already configured).
  - Run the following command:

    ```
    # ip link set down pubeth1
    ```

  - Update the `ifcfg-pubeth1` file with the following:

    ```
    DEVICE=pubeth1
    NAME=pubeth1
    TYPE=Ethernet
    BOOTPROTO=none
    ONBOOT=yes
    ```

  - Add entries in the `ifcfg-pubeth1` file.

    ```
    HWADDR=<MAC address>
    IPADDR=<pubeth1_pub_ip>
    NETMASK=<netmask>
    NM_CONTROLLED=no
    ```

  - Run the following command.

    ```
    # ip link set up pubeth1
    ```

- Configure the first public NIC, pubeth0.
  - As the first public NIC goes down, make sure that you access the system directly from its console.
  - Run the following command:

    ```
    # ip link set down pubeth0
    ```

  - Update the `ifcfg-pubeth0` file with the following:

```
DEVICE=pubeth0
NAME=pubeth0
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
```

- Add entries in the `ifcfg-pubeth0` file.

  ```
  HWADDR=<MAC address>
  IPADDR=<pubeth0_pub_ip>
  NETMASK=<netmask>
  NM_CONTROLLED=no
  ```

- Run the following command.

  ```
  # ip link set up pubeth0
  ```

- Verify if pubeth1 is down. If yes, then bring it online.

  ```
  # ip link set up pubeth1
  ```

- Verify the changes.

  ```
  # ip a
  ```

- Run the following command.

  ```
  # service network restart
  ```

  SSH to the above-mentioned IP should work if you start the sshd service.

13 Configure the DNS.

Update the `/etc/resolv.conf` file by adding the following entries:

```
nameserver <DNS>
domain <master node name>
```

For example:

```
nameserver 10.182.128.134
domain clus_01
```

14 Configure the gateway.

Update the `/etc/sysconfig/network` file.

```
GATEWAY=$gateway
NOZEROCONF=yes
```

15 Update the `configfileTemplate` file.

- Enter the following command:

  ```
  # cd /access_build_dir/rhel7_x86_64/manual_install/network
  ```

- Update the `configfileTemplate` file with the current system details:
  - Use *master* as the mode for the master node and *slave* as the mode for the other nodes.
  - The configuration utility script uses this template file to create configuration files.
  - Provide the same name (current host name) in *old_hostname* and *new_hostname*.

16 Generate the network configuration files.

- The configuration utility script named `configNetworkHelper.pl` creates the required configuration files.

  ```
  # cd /access_build_dir/rhel7_x86_64/manual_install/network
  # chmod +x configNetworkHelper.pl
  ```

- Run the configuration utility script.

  ```
  # ./configNetworkHelper.pl -f configfileTemplate
  ```

- ```
  # cat /opt/VRTSnas/scripts/net/network_options.conf >
  /opt/VRTSnas/conf/network_options.conf
  ```

- ```
  # sed -i -e '$a\' /opt/VRTSnas/conf/net_console_ip.conf
  ```

- Update the `/etc/hosts` file.

  ```
  # echo "172.16.0.3  <master hostname>" >> /etc/hosts
  # echo "172.16.0.4  <slave node name>" >> /etc/hosts
  ```

  For example:

```
# echo "172.16.0.3  clus_01" >> /etc/hosts
# echo "172.16.0.4  clus_02" >> /etc/hosts
```

17 Create the S3 configuration file.

```
# cat /opt/VRTSnas/conf/ssnas.yml
ObjectAccess:
  config: {admin_port: 8144, s3_port: 8143, server_enable: 'no',
  ssl: 'no'}
  defaults:
    fs_blksize: '8192'
    fs_encrypt: 'off'
    fs_nmirrors: '2'
    fs_options: ''
    fs_pdirenable: 'yes'
    fs_protection: disk
    fs_sharing: 'no'
    fs_size: 20G
    fs_type: mirrored
    poollist: []
  filesystems: {}
  groups: {}
  pools: {}
```

18 Set up the Storage Foundation cluster.

- `# cd /access_build_dir/rhel7_x86_64/manual_install/network/SetupClusterScripts`
- `# mkdir -p /opt/VRTSperl/lib/site_perl/UXRT72/CPIR/Module/veritas/`
- `# cp sfcfsha_ctrl.sh /opt/VRTSperl/lib/site_perl/UXRT72/CPIR/Module/veritas/sfcfsha_ctrl.sh`
- `# cp module_script.pl /tmp/`
- `# chmod +x /tmp/module_script.pl`
- Update the cluster name, system name, and NIC name in the following command and execute it:

  ```
  # /tmp/module_script.pl veritas::sfcfsha_config '{"cluster_name" =>
  "<Provide cluster name here>","component" => "sfcfsha","state" =>
  ```

```
"present","vcs_users" => "admin:password:Administrators,user1:
passwd1:Operators","vcs_clusterid" => 14865,"cluster_uuid" =>
"1391a-443ab-2b34c","method" => "ethernet","systems" =>
"<Provide hostnames separated by comma>","private_link" =>
"<provide private nic name separated by comma>"}'
```

For example, if the cluster name is *clus* and the host names are *clus_01* and *clus_02*.

```
/tmp/module_script.pl veritas::sfcfsha_config '
{"cluster_name" => "clus","component" => "sfcfsha",
"state" => "present","vcs_users" =>
"admin:password:Administrators,user1:passwd1:Operators",
"vcs_clusterid" => 14865,"cluster_uuid" => "1391a-443ab-2b34c",
"method" => "ethernet","systems" => "clus_01,clus_02",
"private_link" => "priveth0,priveth1"}'
```

- Update and configure the following files:
  - `# rpm -q --queryformat '%{VERSION}|%{BUILDTIME:date}|% {INSTALLTIME:date}|% {VERSION}\n' VRTSnas > /opt/VRTSnas/conf/version.conf`
  - `# echo NORMAL > /opt/VRTSnas/conf/cluster_type`
  - `# echo 'path /opt/VRTSsnas/core/kernel/' >> /etc/kdump.conf`
  - `# sed -i '/^core_collector\b/d;' /etc/kdump.conf`
  - `# echo 'core_collector makedumpfile -c --message-level 1 -d 31' >> /etc/kdump.conf`

19 Start the Veritas Access product processes.

- Provide the current host name in the following command and execute it.

```
# /tmp/module_script.pl veritas::process '{"state" => "present",
"seednode" => "<provide current hostname here>","component"
 => "sfcfsha"}'
```

For example, if the host name is *clus_01*:

```
# /tmp/module_script.pl veritas::process '{"state" =>
"present","seednode" => "clus_01","component" => "sfcfsha"}'
```

If you are running it on *clus_02*, then you have to provide “seednode" => "clus_02”.

- Run the following command.

```
# /opt/VRTSnas/pysnas/install/install_tasks.py
all_services_running serial
```

20 Create the CVM group.

If the `/etc/vx/reconfig.d/state.d/install-db` file exists, then execute the following command.

```
# mv /etc/vx/reconfig.d/state.d/install-db
/etc/vx/reconfig.d/state.d/install-db.a
```

If CVM is not configured already, run the following command on the master node.

```
# /opt/VRTS/bin/cfscluster config -t 200 -s
```

21 Enable hacli.

Verify in the `/etc/VRTSvcs/conf/config/main.cf` file. If `HacliUserLevel = COMMANDROOT` exists, then move to step 22 , else follow the below steps to enable hacli in your system.

```
# /opt/VRTS/bin/hastop -local
```

Update the `/etc/VRTSvcs/conf/config/main.cf` file.

If it does not exist, add the following line:

```
HacliUserLevel = COMMANDROOT  in cluster <cluster name> ( ) loop
```

For example:

```
cluster clus (
   UserNames = { admin = aHIaHChEIdIIgQIcHF, user1 = aHIaHChEIdIIgFEb }
   Administrators = { admin }
   Operators = { user1 }
   HacliUserLevel = COMMANDROOT
# /opt/VRTS/bin/hastart
```

Verify that hacli service is running.

```
# /opt/VRTS/bin/hacli -cmd "ls /" -sys clus_01
```

22 Verify that the HAD daemon is running.

```
# /opt/VRTS/bin/hastatus -sum
```

23 Configure Veritas Access on the second node by following steps 1 to 22 .

24 Verify that the system is configured correctly.

- Verify that LLT is configured correctly.

  ```
  # lltconfig -a list
  ```

  For example:

  ```
  [root@clus_02 SetupClusterScripts]# lltconfig -a list
  Link 0 (priveth0):
    Node    0 clus_01    :    00:0C:29:0C:8D:69
    Node    1 clus_02    :    00:0C:29:F0:CC:B6  permanent

  Link 1 (priveth1):
   Node    0 clus_01     :    00:0C:29:0C:8D:5F
   Node    1 clus_02     :    00:0C:29:F0:CC:AC  permanent
  ```

- Verify that GAB is configured properly.

  ```
  # gabconfig -a
  ```

  For example:

  ```
  [root@clus_01 network]# gabconfig -a
  GAB           Port    Memberships
  ============  ======  ================
  Port a gen    43b804  membership 01
  Port b gen    43b807  membership 01
  Port h gen    43b821  membership 01
  ```

- Verify the LLT state.

  ```
  # lltstat -nvv
  ```

  For example:

  ```
  [root@clus_01 network]# lltstat -nvv
  LLT node information:
        Node         State    Link    Status  Address
     * 0 clus_01     OPEN
                              priveth0   UP    00:0C:29:0C:8D:69
  ```

```
                          priveth1   UP    00:0C:29:0C:8D:5F
      1 clus_02    OPEN
                          priveth0   UP      00:0C:29:F0:CC:B6
                          priveth1   UP      00:0C:29:F0:CC:AC
      2          CONNWAIT
                        priveth0   DOWN
                        priveth1   DOWN
```

- The `vxconfigd` daemon should be online on both nodes.

  ```
  # ps -ef | grep vxconfigd
  ```

  For example:

  ```
  # ps -ef | grep vxconfigd
  root   13393 1  0 01:33 ?  00:00:00 vxconfigd -k -m disable -x syslog
  ```

25 Run the Veritas Access post-start actions.

- Make sure that `HAD` is running on all the nodes.

  ```
  # /opt/VRTS/bin/hastatus
  ```

- On all the nodes, create a `communication.conf` file to enable hacli instead of ssh.

  ```
  vim /opt/VRTSnas/conf/communication.conf
  {
   "WorkingVersion": "1",
   "Version": "1",
   "CommunicationType": "HACLI"
  }
  ```

- Run the installer to install Veritas Access. Run the following command only on the master node.

  ```
  # /opt/VRTSnas/install/image_install/installer -m master
  ```

26 Run the join operation on the slave node.

```
# /opt/VRTSnas/install/image_install/installer -m join
```

27 Run the following command on both the nodes.

```
# echo "<first private nic name>" >
/opt/VRTSnas/conf/net_priv_dev.conf
```

For example:

```
# echo "priveth0" > /opt/VRTSnas/conf/net_priv_dev.conf
```

28 Enable NFS resources. Run the following commands on the master node.

```
# /opt/VRTS/bin/haconf -makerw
# /opt/VRTS/bin/hares -modify ssnas_nfs Enabled 1
# /opt/VRTS/bin/haconf -dump -makero
```

You can now use the two-node Veritas Access cluster.

Enabling internal sudo user communication in Veritas Access

By default, Veritas Access uses SSH communication between the nodes for the root user. If you want to use sudo user-based communication, you can set the internal communication to use the sudo user communication after you have installed Veritas Access successfully.

You can follow the following steps to set up the sudo user communication.

- Phase 1: Create a Veritas Access user on each of the nodes of the Veritas Access cluster.
- Phase 2: Set up a passwordless communication between the root and a Veritas Access user on each node
- Phase 3: Select the communication type as SUDO_SSH

Phase 1: Create a Veritas Access user on each of the nodes of the Veritas Access cluster

1 Create the `access_user` and set the password.

For example:

```
[root@access1_01 ~]# useradd access_user
[root@access1_01 ~]# passwd access_user
Changing password for user access_user.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

2 Add the `access_user` to the `sudoers` file.

For example:

```
[root@access1_01 ~]# echo "access_user ALL=(ALL) NOPASSWD: ALL"
>> /etc/sudoers
```

Complete Phase 1 on all the nodes of the cluster.

Phase 2: Set up a passwordless communication between the root and a Veritas Access user on each node

1 Generate a `rsa` key for the **root** user if it is not present.

For example:

```
[root@access1_01 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:hRIBljcpsmGMCtfUUjyVGOfe957OXyiXcRyiYBprmZk root@access1_01
The key's randomart image is:
+---[RSA 2048]----+
| o o+*=*o.       |
|o *.= O+..       |
|oo + +.+oo.  . . |
|. .    oXo. . ...|
|       ES ... . o|
|      .    . . = |
|             . = .|
|              = ..|
|              .=..|
+----[SHA256]-----+
```

2 Copy the `rsakey.pub` of the **root** user to the `access_user` for each of the nodes in the cluster.

For example:

```
[root@access1_01 ~]# ssh-copy-id access_user@access1_01
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new
key(s),to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed --
if you are prompted now it is to install the new keys
access_user@access1_01's password:

Number of key(s) added: 1
```

Phase 3: Select the communication type as SUDO_SSH

- Create the `/opt/VRTSnas/conf/communication.conf` file.

```
[root@access1_01 ~]# cat /opt/VRTSnas/conf/communication.conf
{
    "WorkingVersion": "1",
    "Version": "1",
    "CommunicationType": "SUDO_SSH"
}
```

Index

V