

# Cluster Server Agent for Network Appliance SnapMirror Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

8.0.2

# Veritas InfoScale™ Availability Agents

Last updated: 2023-06-30

## Legal Notice

Copyright © 2023 Veritas Technologies LLC. All rights reserved.

Veritas and the Veritas Logo are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

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

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

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

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

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

Veritas Technologies LLC  
2625 Augustine Drive  
Santa Clara, CA 95054

<http://www.veritas.com>

## Technical Support

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

<https://www.veritas.com/support>

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

<https://my.veritas.com>

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

Worldwide (except Japan)

[CustomerCare@veritas.com](mailto:CustomerCare@veritas.com)

Japan

[CustomerCare\\_Japan@veritas.com](mailto:CustomerCare_Japan@veritas.com)

## Documentation

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

<https://sort.veritas.com/documents>

## Documentation feedback

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

[infoscaledocs@veritas.com](mailto:infoscaledocs@veritas.com)

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

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

## Veritas Services and Operations Readiness Tools (SORT)

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

[https://sort.veritas.com/data/support/SORT\\_Data\\_Sheet.pdf](https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf)

# Contents

<b>Chapter 1</b>	<b>Introducing the agent for Network Appliance SnapMirror .....</b>	<b>6</b>
	About the agent for Network Appliance SnapMirror .....	6
	Supported software .....	7
	Supported hardware for Network Appliance SnapMirror .....	7
	Typical Network Appliance SnapMirror setup in a VCS cluster .....	7
	Network Appliance SnapMirror agent functions .....	8
	About the Network Appliance SnapMirror agent's online function .....	11
<b>Chapter 2</b>	<b>Installing and removing the agent for Network Appliance SnapMirror .....</b>	<b>12</b>
	Before you install the agent for Network Appliance SnapMirror .....	12
	NetApp agent installation issue (dependency on the perl(XML::Parser) module) .....	12
	Installing the agent for Network Appliance SnapMirror .....	13
	Upgrading the agent for Network Appliance SnapMirror .....	13
	Removing the agent for Network Appliance SnapMirror .....	16
<b>Chapter 3</b>	<b>Configuring the agent for Network Appliance SnapMirror .....</b>	<b>17</b>
	Configuration concepts for Network Appliance SnapMirror agent .....	17
	Resource type definition for the NetAppSnapMirror agent .....	17
	Attribute definitions for the NetAppSnapMirror agent .....	18
	Sample configuration for the Network Appliance SnapMirror agent .....	21
	Before you configure the agent for Network Appliance SnapMirror .....	23
	About cluster heartbeats .....	24
	Configuring the agent for Network Appliance SnapMirror .....	24
	Configuring the agent manually in a global cluster .....	24
	Configuring the agent to compute RPO .....	25
	Configuring the agent for SnapMirror replication over multiple paths .....	26

	Configuring multiple paths in SnapMirror .....	26
	The agent behavior when SnapMirror over multiple path is configured .....	27
<b>Chapter 4</b>	<b>Managing and testing clustering support for Network Appliance SnapMirror .....</b>	<b>29</b>
	Performing a disaster test .....	29
	Failure scenarios for Network Appliance SnapMirror .....	30
	Site disaster .....	30
	All host or all application failure .....	30
<b>Chapter 5</b>	<b>Setting up a fire drill .....</b>	<b>31</b>
	About fire drills .....	31
	About the NetAppSnapMirrorSnap agent .....	32
	NetAppSnapMirrorSnap agent functions .....	32
	Resource type definition for the NetAppSnapMirrorSnap agent .....	33
	Attribute definitions for the NetAppSnapMirrorSnap agent .....	33
	Before you configure the fire drill service group .....	35
	Configuring the fire drill service group .....	35
	Creating the fire drill service group using Cluster Manager (Java Console) .....	35
	Creating the fire drill service group using the Fire Drill SetUp Wizard .....	37
	Verifying a successful fire drill .....	38
	Sample configuration for a fire drill service group .....	38

# Introducing the agent for Network Appliance SnapMirror

This chapter includes the following topics:

- [About the agent for Network Appliance SnapMirror](#)
- [Supported software](#)
- [Supported hardware for Network Appliance SnapMirror](#)
- [Typical Network Appliance SnapMirror setup in a VCS cluster](#)
- [Network Appliance SnapMirror agent functions](#)

## About the agent for Network Appliance SnapMirror

The Cluster Server (VCS) agent for Network Appliance (NetApp) SnapMirror provides application failover support and recovery in environments that use SnapMirror to replicate data between NetApp filers.

The agent monitors and manages the state of a volume or a qtree for which SnapMirror replication is configured on the filers. The agent ensures that at any given time, only one host has safe exclusive access to the configured filer volume or qtree. It determines the current role of the filer volume or qtree with respect to replication, and promotes a read-only target to a read-write source during a wide-area failover. You can use the agent in global clusters that run VCS.

The agent supports SnapMirror with Data ONTAP (7-mode) as well as Clustered Data ONTAP (cluster-mode). However, it supports the following configurations only with 7-mode and not with cluster-mode:

- Qtree replication
- Multi-pathing
- Fire drill

---

**Note:** Cluster-mode support for this agent is available only on Linux. Also, only data protection relationships are supported with cluster-mode. Other relationships like vault and load-sharing are not supported.

---

See the following Technical Support TechNote for the latest updates or software issues for this agent:

<http://www.veritas.com/docs/000030982>

## Supported software

For information on the software versions that the () agent for SnapMirror supports, see the Veritas Services and Operations Readiness Tools (SORT) site:

<https://sort.veritas.com/agents>

## Supported hardware for Network Appliance SnapMirror

The agent supports the following versions of Network Appliance hardware:

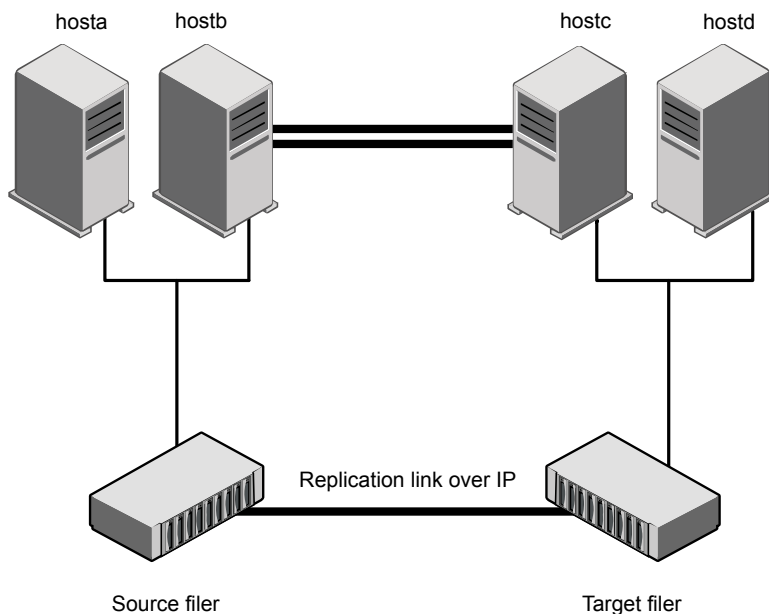
- Network Appliance FAS devices (any model)
- Network Appliance ONTAP 6.4 and later.

The agent does not support NetApp F800 filer series.

## Typical Network Appliance SnapMirror setup in a VCS cluster

[Figure 1-1](#) displays a typical cluster setup in a SnapMirror environment.

**Figure 1-1** Typical clustering setup for the agent



Clustering in a SnapMirror environment typically consists of the following hardware infrastructure:

- One or more source hosts are connected by an ethernet cable to a Network Appliance device. The Network Appliance SnapMirror agent supports NFS as a transport protocol.
- One or more target hosts that are attached to a second filer with similar characteristics as the source hosts. The volume or qtree are SnapMirror targets.
- Network heartbeating between the two data centers to determine their health; this network heartbeating could be LLT or TCP/IP.  
See [“About cluster heartbeats”](#) on page 24.
- In a global cluster environment, you must attach all hosts in the same cluster to the same filer.

## Network Appliance SnapMirror agent functions

The Network Appliance SnapMirror agent monitors and manages the state of replicated Network Appliance volume or qtree that are attached to VCS nodes.



Agent functions bring resources online, take them offline, and perform different monitoring actions.

**Table 1-1** Agent functions

Agent function	Description
online	<p>Creates a lock file on the local host to indicate that the resource is online.</p> <p>Depending on whether the filer is source or target and the state of the specified volume or qtree on the filer, the agent performs the online function.</p> <p>See <a href="#">“About the Network Appliance SnapMirror agent’s online function”</a> on page 11.</p>
offline	<p>The agent removes the lock file. The agent does not perform any filer operations because taking the resource offline is not an indication of a pending role swap.</p>
monitor	<p>Verifies that the lock file exists.</p> <ul style="list-style-type: none"><li>■ If the lock file exists, the monitor entry point reports the status of the resource as online.</li><li>■ If the lock file does not exist, the monitor entry point reports the status of the resource as offline.</li></ul>
open	<p>Removes the lock file on the system on which this entry point is called. The agent prevents potential concurrency violation if the group fails over to another node.</p> <p>Note that the agent does not remove the lock file if the agent was started after a <code>hastop -force</code> command.</p>
clean	<p>Performs the same operation as offline.</p>
action/PreSwitch	<p>Ensures that the remote site cluster can come online during a planned failover within a GCO configuration. The VCS engine on the remote cluster invokes the PreSwitch action on all the resources of the remote site during a planned failover using the <code>hagrp -switch</code> command.</p> <p>For this, the PreSwitch attribute must be set to 1. The option <code>-nopre</code> indicates that the VCS engine must switch the servicegroup regardless of the value of the PreSwitch service group attribute.</p> <p>If running the PreSwitch action fails, the failover should not occur. This minimizes the application downtime and data loss.</p>

**Table 1-1** Agent functions (*continued*)

Agent function	Description
action/reestablish	Re-establishes the replication with a destination volume or qtree that is broken off. Invokes a 'snapmirror resync' to resynchronize contents with the source without doing a full transfer. This action also write-disables the destination volume or qtree.
actions/swap	Reverses the replication roles to make the original source the new destination and vice versa. It is intended for use when the original source comes back alive after an outage at the source location and the original destination is broken-off in the meantime to takeover the applications at the destination location. As part of this action, the new destination (i.e. the original source) is made to sync up with the new source.
actions/update	Used to run unscheduled updates.  SnapMirror updates the destination automatically based on the update schedule specified in the /etc/snapmirror.conf. However, to run unscheduled updates, this update action may be invoked on the NetAppSnapMirror resource.
action/GetCurrentRPO	Fetches the current point-in-time RPO from the NetApp array in terms of SnapMirror lag. Invoke this action function on the DR system.  <b>Note:</b> The agent does not store the computed RPO; make a note of the RPO for future reference.

The following functions are not supported in cluster-mode:

- action/PreSwitch

---

**Note:** In case of SnapMirror with cluster-mode, the agent bypasses the actions of this function and triggers the failover.

---

- action/reestablish
- actions/swap
- actions/update
- action/GetCurrentRPO

## About the Network Appliance SnapMirror agent's online function

The agent's online entry point checks the state of the specified volume or qtree on the local filer device. The volume or qtree on the local filer is in the SOURCE state when it is the replication source. The volume or qtree is in the SNAPMIRRORED state when it is the replication target.

---

**Note:** In case of SnapMirror with cluster-mode, if a volume on the local filer is of the read-write type, the agent considers it to be the source for replication.

---

The agent performs the following online operations:

- If the state of the volume or qtree on the local filer device is source, the agent creates a lock file to indicate that the resource can come online
- If the filer is a target, the agent attempts to reverse the direction of replication. To reverse the replication, the agent promotes the local filer to a source and the original source to a target by performing the following operations:
  - Updates the destination volume or qtree
  - Quiesces' the replication
  - Breaks off the SnapMirror connection
  - Removes the SnapMirror schedule from the target filer and creates a new SnapMirror schedule on local filer
  - Resynchronizes the local filer with the new target filer
- If the original source is down, the agent attempts a takeover to enable local write access by performing the following operations:
  - Quiesces' the replication
  - Breaks off the SnapMirror connection
- If the original source comes online, you must resynchronize data manually
- If read-write access is enabled successfully, the online entry point touches a lock file

# Installing and removing the agent for Network Appliance SnapMirror

This chapter includes the following topics:

- [Before you install the agent for Network Appliance SnapMirror](#)
- [Installing the agent for Network Appliance SnapMirror](#)
- [Upgrading the agent for Network Appliance SnapMirror](#)
- [Removing the agent for Network Appliance SnapMirror](#)

## Before you install the agent for Network Appliance SnapMirror

Before you install the VCS agent for Network Appliance SnapMirror, ensure that you install and configure VCS on all nodes in the cluster.

Set up replication and the required hardware infrastructure.

See [“Typical Network Appliance SnapMirror setup in a VCS cluster”](#) on page 7.

## NetApp agent installation issue (dependency on the perl(XML::Parser) module)

The NetApp SnapMirror agent depends on the perl(XML::Parser) module. This dependency is inherited from NaServer, which is an SDK interface for SnapMirror. If the perl(XML::Parser) module is not installed on your operating system, then the

agent installation will fail. Make sure that you install this module before installing the SnapMirror agent. For example, on the RHEL platform, the install will fail with the following error:

```
# rpm -ivh VRTSvcsna-7.0.0.0-Linux_GENERIC.noarch.rpm
```

Error for failed dependencies:

```
perl(XML::Parser) is needed by VRTSvcsna-7.0.0.0-Linux_GENERIC.noarch
```

## Installing the agent for Network Appliance SnapMirror

You must install the Network Appliance SnapMirror agent on each node in the cluster. In global cluster environments, install the agent on each node in each cluster.

These instructions assume that you have already installed VCS.

### To install the agent in a VCS environment

- 1 Download the Agent Pack from the Veritas Services and Operations Readiness Tools (SORT) site: <https://sort.veritas.com/agents>.
- 2 Log in as a superuser.
- 3 Install the package.

```
AIX      # installp -ac -d VRTSvcsna.rte.bff VRTSvcsna.rte
HP-UX    # swinstall -s 'pwd' VRTSvcsna
(IA/PA)
Linux    # rpm -ihv
          VRTSvcsna-AgentVersion-Linux_GENERIC.noarch.rpm
Solaris  # pkgadd -d . VRTSvcsna
```

## Upgrading the agent for Network Appliance SnapMirror

You must upgrade the agent on each node in the cluster.

## To upgrade the agent software

### 1 Save the VCS configuration.

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

### 2 Stop the agent if it is running.

- If the NetAppSnapMirror agent is configured and running:

```
# haagent -stop NetAppSnapMirror -force -sys system
```

- If the NetAppSnapMirrorSnap agent is configured and running:

```
# haagent -stop NetAppSnapMirrorSnap -force -sys system
```

### 3 Verify the status of the agent and ensure that it is not running.

- For NetAppSnapMirror agent:

```
# haagent -display NetAppSnapMirror | grep Running
```

The command output resembles the following:

```
NetAppSnapMirror Running No
```

- For NetAppSnapMirrorSnap agent:

```
# haagent -display NetAppSnapMirrorSnap | grep Running
```

The command output resembles the following:

```
NetAppSnapMirrorSnap Running No
```

### 4 Upgrade the agent.

- For Linux, run the following command:

```
# rpm -Uvh VRTSvcsna-AgentVersion-Linux_GENERIC.noarch.rpm
```

- For AIX, HP-UX (IA/PA), and Solaris, perform the following steps:

- Remove the previous version of the agent from the node.  
See [“Removing the agent for Network Appliance SnapMirror”](#) on page 16.

- Delete the file

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

- Install the latest version of the agent.  
See “Installing the agent for Network Appliance SnapMirror” on page 13.
- Copy the file `NetAppSnapMirrorTypes.cf` from the `/etc/VRTSvcs/conf/` to the `/etc/VRTSvcs/conf/config` directory.
- Repeat these four substeps that are mentioned earlier on each node.
- From a node in the cluster, edit the `/etc/VRTSvcs/conf/config/main.cf` configuration file.  
Configure the new attributes, if applicable.
- Verify the configuration by running the following command:  
  

```
# hacf -verify config
```

## 5 Start the agent.

- For NetAppSnapMirror agent:

```
# haagent -start NetAppSnapMirror -sys system
```

- For NetAppSnapMirrorSnap agent:

```
# haagent -start NetAppSnapMirrorSnap -sys system
```

## 6 Verify the status of the agent and ensure that it is running.

- For NetAppSnapMirror agent:

```
# haagent -display NetAppSnapMirror | grep Running
```

The command output resembles the following:

```
NetAppSnapMirror Running Yes
```

- For NetAppSnapMirrorSnap agent:

```
# haagent -display NetAppSnapMirrorSnap | grep Running
```

The command output resembles the following:

```
NetAppSnapMirrorSnap Running Yes
```

# Removing the agent for Network Appliance SnapMirror

Before you attempt to remove the agent, make sure the application service group is not online.

You must remove the SnapMirror agent from each node in the cluster.

## To uninstall the agent in a VCS environment

- ◆ To remove the agent, type the following command on each node. Answer prompts accordingly:

AIX	# installp -u VRTSvcsna.rte
HP-UX	# swremove VRTSvcsna
Linux	# rpm -e VRTSvcsna
Solaris	# pkgrm VRTSvcsna



# Configuring the agent for Network Appliance SnapMirror

This chapter includes the following topics:

- [Configuration concepts for Network Appliance SnapMirror agent](#)
- [Before you configure the agent for Network Appliance SnapMirror](#)
- [Configuring the agent for Network Appliance SnapMirror](#)
- [Configuring the agent for SnapMirror replication over multiple paths](#)

## Configuration concepts for Network Appliance SnapMirror agent

Review the resource type definition and the attribute definitions for the agent.

### Resource type definition for the NetAppSnapMirror agent

The resource type definition defines the agent in VCS.

```
type NetAppSnapMirror (
    static int ActionTimeout = 120
    static keylist SupportedActions = { update,
    reestablish, swap, PreSwitch, GetCurrentRPO }
    static int MonitorInterval = 300
    static int OpenTimeout = 180
```

```
static int NumThreads = 1
static str ArgList[] = { VolumeName, SnapMirrorSchedule,
LocalFilerName, LocalFilerUserName, LocalFilerPword,
RemoteFilerName, RemoteFilerUserName, RemoteFilerPword,
OpsThrottle, VisibilityFrequency, VServer, CModeCluster }
str VolumeName
str SnapMirrorSchedule
str LocalFilerName
str LocalFilerUserName="root"
str LocalFilerPword
str RemoteFilerName
str RemoteFilerUserName="root"
str RemoteFilerPword
str OpsThrottle
int VisibilityFrequency = 180
int ComputeDRSLA = 1
str VServer
str CModeCluster
)
```

## Attribute definitions for the NetAppSnapMirror agent

Review the description of the agent attributes. You must assign values to the following attributes.

### Required attributes

**Note:** Some required attributes are applicable only to configurations with Clustered Data ONTAP.

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

Attribute	Description
Name: LocalFilerName	The DNS resolvable name of the local NetApp filer.
Type: string	You must use the same value for the LocalFilerName as in the NetAppSnapMirror configuration. If the filer's name is not fully qualified in the NetAppSnapMirror configuration, do not use the fully qualified name to set this attribute.
Dimension: scalar	
	For example, if the filer name is defined as netapp3.example.com in the configuration, use the same filer's name to set the attribute.

**Table 3-1** Required attributes for the SnapMirror agent (*continued*)

Attribute	Description
Name: LocalFilerUserName Type: string Dimension: scalar	The local NetApp filer user who has permissions to perform SnapMirror actions using ONTAP SDK.  
Name: LocalFilerPword Type: string Dimension: scalar	A valid encrypted password for the user on the local filer.  If you enter the password from the command line, run the <code>vcseencrypt</code> command to encrypt the password. Do not encrypt passwords when entering them from the GUI.  <b>Note:</b> In a 7-mode configuration, if you do not specify a password, the agent uses the <code>hosts.equiv</code> file on the filer to determine the access rights.
Name: RemoteFilerName Type: string Dimension: scalar	The DNS resolvable name of the remote NetApp filer.  You must use the same value for the RemoteFilerName as in the NetAppSnapMirror configuration. If the filer's name is not fully qualified in the NetAppSnapMirror configuration, do not use the fully qualified name to set this attribute.  For example, if the filer name is defined as <code>netapp3.example.com</code> in the configuration, use the same filer's name to set the attribute.
Name: RemoteFilerUserName Type: string Dimension: scalar	Remote NetApp filer user name:  Have a permission to perform SnapMirror actions using ONTAP SDK.
Name: RemoteFilerPword Type: string Dimension: scalar	A valid encrypted password for the root user on the remote filer.  If you enter the password from the command line, run the <code>vcseencrypt</code> command to encrypt the password. Do not encrypt passwords when entering them from the GUI.  <b>Note:</b> In a 7-mode configuration, if you do not specify a password, the agent uses the <code>hosts.equiv</code> file on the filer to determine the access rights.

**Table 3-1** Required attributes for the SnapMirror agent (*continued*)

Attribute	Description
Name: VolumeName Type: string Dimension: scalar	Specifies the name of the replicated volume or qtree on the filer. You must use same case for the volume name as on the filer because the attribute value is case-sensitive.  <b>Note:</b> For qtrees, you must use the full path name, for example, /vol/myvol/myqtree.
Name: VServer Type: string Dimension: scalar	(Cluster-mode only)  Specifies the Vserver name of the SnapMirror relationship in cluster-mode.
Name: CModeCluster Type: string Dimension: scalar	(Cluster-mode only)  Specifies the cluster name of the SnapMirror relationship in cluster-mode.

## Optional attributes

**Note:** The optional attributes are not applicable to cluster-mode configurations.

**Table 3-2** Optional attributes for the SnapMirror agent:

Attribute	Description
Name: OpsThrottle Type: string Dimension: scalar	Specifies how out of sync the local and remote filer can run. You must set this attribute to configure SnapMirror in synchronous mode.  You can specify the attribute value in one of the following units: <ul style="list-style-type: none"> <li>■ ops - Number of I/O operations</li> <li>■ s - Number of seconds</li> <li>■ ms - Number of milliseconds</li> </ul> Example: "5ops"  Note that to configure SnapMirror in synchronous mode, you must set the value as "0s", "0ms" or "0ops".

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

Attribute	Description
Name: SnapMirrorSchedule Type: string Dimension: scalar	Specifies the schedule that the destination uses for updating data. This attribute can be left empty to configure SnapMirror in synchronous mode.  You must set this attribute to configure SnapMirror in asynchronous mode.  Example: 0-59/5 * * *
Name: VisibilityFrequency Type: integer Dimension: scalar	Specifies the time when the source snapshot is visible at the destination mirror.  Default: 180 seconds
Name: ComputedRSLA Type: integer Dimension: scalar	Used to enable Recovery Point Objective (RPO) computation. This attribute cannot be edited.  Default value : 1

## Sample configuration for the Network Appliance SnapMirror agent

You can configure the SnapMirror agent with the Mount agent. You can also use the NetAppFiler and NetAppExport agents provided by Network Appliance.

Sample configuration for the Network Appliance SnapMirror agent

- Sample configuration with the Mount agent

```
Mount ntap (
    MountPoint = "/ntap"
    BlockDevice = "netappl:/vol/unixvol"
    FSType = nfs
    FsckOpt = "-n"
)

NetAppSnapMirror sm (
    VolumeName = unixvol
    SnapMirrorSchedule = "0-59/5 * * * *"
    LocalFilerName = netappl
    LocalFilerUserName = root
    LocalFilerPword = xxxx
    RemoteFilerName = netapp2
    RemoteFilerUserName = root
)
```

```
RemoteFilerPword = xxxx
)
```

```
ntap requires sm
```

#### ■ Sample configuration with the NetAppExport and NetAppFiler agents

```
group ntap (
    SystemList = { primary1 = 0, primary2 = 1 }
    ClusterList = { secondary = 0, primary = 1 }
)

NetAppExport nfs (
    FilerResName = filer
    FilerPathName = "/vol/unixvol"
    FilerOptions @primary1 = "access=primary1,root=primary1"
    FilerOptions @primary2 = "access=primary2,root=primary2"
    MountPathName = "/ntap"
    MountOptions = rw
)

NetAppFiler filer (
    FilerName = priflr
)

NetAppSnapMirror sm (
    VolumeName = unixvol
    SnapMirrorSchedule = "0 23 * *"
    LocalFilerName = priflr
    LocalFilerUserName = root
    LocalFilerPword = xxxx
    RemoteFilerName = secflr
    RemoteFilerUserName = root
    RemoteFilerPword = xxxx
)

nfs requires sm
sm requires filer
```

#### Sample configuration for the Network Appliance SnapMirror agent in cluster-mode:

```
group SG_NTAP (
    SystemList = { dl380g9-50vm5 = 0, dl380g9-50vm6 = 1 }
    ClusterList = { PRIM_CLUS = 0, SEC_CLUS = 1 }
    Authority = 1
```

```

AutoStartList = { dl380g9-50vm5 }
ClusterFailOverPolicy = Auto
)

Mount res_mnt (
    MountPoint = "/mnt1"
    BlockDevice =
        "10.209.92.147:/NAS_IAFAS3240_vol_n2_sr226876_DL380G9_50vm5_8_2gb_1_m
    FSType = nfs
    FsckOpt = "-n"
)

NetAppSnapMirror res_ntp (
    VolumeName =
        NAS_IAFAS3240_vol_n2_sr226876_DL380G9_50vm5_8_2gb_1_mirror
    LocalFilerName = "10.209.92.171"
    LocalFilerUserName = vrpadmin
    LocalFilerPword = xxxx
    RemoteFilerName = "10.209.92.170"
    RemoteFilerUserName = vrpadmin
    RemoteFilerPword = xxxx
    CModeCluster = IA2552-02
    VServer = NAS_IAFAS2552
)

res_mnt requires res_ntp

// resource dependency tree
//
//   group SG_NTAP
//   {
//       Mount res_mnt
//       {
//           NetAppSnapMirror res_ntp
//       }
//   }

```

## Before you configure the agent for Network Appliance SnapMirror

Before you configure the agent, review the following information:

- Verify that you have installed the agent on all the cluster nodes.
- Verify the hardware setup for the agent.  
 See [“Typical Network Appliance SnapMirror setup in a VCS cluster”](#) on page 7.
- Make sure that the cluster has an effective heartbeat mechanism in place.  
 See [“About cluster heartbeats”](#) on page 24.

You can automate the mounting of filer volumes or qtrees through VCS. Configure a resource of type Mount as the parent of the NetAppSnapMirror resource and set the FSType attribute of the Mount resource to nfs. You must also configure the filer to export the volumes or qtrees to all the nodes in the cluster.

## About cluster heartbeats

In a global cluster, VCS sends ICMP pings over the public network between the two sites for network heartbeating. To minimize the risk of split-brain, VCS sends ICMP pings to highly available IP addresses. VCS global clusters also notify the administrators when the sites cannot communicate.

# Configuring the agent for Network Appliance SnapMirror

You can adapt most clustered applications to a disaster recovery environment.

If you plan to configure the agent in a global cluster, make sure that you have configured the global service group for the application.

After configuration, the application service group must follow the dependency diagram.

See [“Sample configuration for the Network Appliance SnapMirror agent”](#) on page 21.

## Configuring the agent manually in a global cluster

Configuring the agent manually in a global cluster involves the following tasks:

### To configure the agent in a global cluster

- 1 Start Cluster Manager (Java Console) and log on to the cluster.
- 2 If the agent resource type (NetAppSnapMirror) is not added to your configuration, add it. From the Cluster Explorer **File** menu, choose **Import Types**, and select:  
 /etc/VRTSvc/conf/NetAppSnapMirrorTypes.cf
- 3 Click **Import**.



- 4 Save the configuration.
- 5 Add a resource of type NetAppSnapMirror at the bottom of the service group.
- 6 Configure the attributes of the NetAppSnapMirror resource.
- 7 If the service group is not configured as a global service group, configure the service group using the Global Group Configuration Wizard.  
  
Refer to the *Cluster Server Administrator's Guide* for more information.
- 8 Change the ClusterFailOverPolicy attribute from the default, if necessary. Veritas recommends keeping the default, which is Manual, to minimize the chance of failing over on a split-brain.
- 9 Repeat step 5 through step 8 for each service group in each cluster that uses replicated data.

## Configuring the agent to compute RPO

In a global cluster environment, the agent for Network Appliance SnapMirror can compute the recovery point objective (RPO), which is a disaster recovery (DR) SLA. In a DR configuration where data is replicated asynchronously to the DR site, the DR site data is not always as current as the primary site data.

RPO is the maximum acceptable amount of data loss in case of a disaster at the primary site. The agent computes RPO in terms of time, that is, in seconds.

Before you configure the agent to compute the RPO, ensure that the following pre-requisites are met:

- The service group containing the NetAppSnapMirror resource and the VxVM disk group resource are online at the production site.
- The disk group resource is dependent on the NetAppSnapMirror resource.

**To configure the agent to compute the RPO:**

- 1 In the DR cluster, on any one of the nodes where devices are asynchronously replicated and where the service group is configured, run the following command to start the RPO computation:

```
hares -modify NetAppSnapMirror_resource_name ComputeDRSLA 1 -sys
system_name.
```

- 2 Run the following command on the same node in the DR cluster:

```
hares -action NetAppSnapMirror_resource_name GetCurrentRPO -sys
system_name
```

The action entry point displays the RPO. The agent does not store the computed RPO; make a note of the RPO for future reference.

If the RPO is not reported, it indicates that the agent needs more time to finish computing the RPO. Wait for some more time before you run the GetCurrentRPO action function again.

- 3 To stop RPO computation, run the following command:

```
hares -modify NetAppSnapMirror_resource_name ComputeDRSLA 0 -sys
system_name
```

## Configuring the agent for SnapMirror replication over multiple paths

The agent for Network Appliance SnapMirror supports multiple paths for replication between source and destination filer. The agent can work for up to two paths for a given SnapMirror relationship. Agent detects configured multiple paths on secondary filer and can handle it accordingly during a failover. The agent supports both Multiplexing mode and Failover mode of SnapMirror over multiple paths. The paths can be Ethernet, Fibre Channel or a combination of Ethernet and Fibre Channel.

We recommend ONTAP 7.3.3 for agent support for SnapMirror over Multiple paths. ONTAP versions before version 7.3.3 needs patch from NetApp to fix issue in ONTAP API `snapmirror-set-connection`.

### Configuring multiple paths in SnapMirror

Set up SnapMirror over multiple paths on destination filer. Ensure valid paths between source filer and destination filer. Edit the `snapmirror.conf` file on the destination system to add a connection line and relation entry to specify the volumes or qtrees to be copied and the schedule on which the destination is updated. Use

the connection name specified as the source system. The format of the line in snapmirror.conf as follows:

```
name = failover (src_system1, dest_system1) (src_system2,dest_system2)
#connection entry name:srcvol1 dest_filer:destvol1 - 0-55/5 * * *
#Relationship entry
```

## The agent behavior when SnapMirror over multiple path is configured

Filer1 is source having two interfaces:source\_ip\_addr1 and source\_ip\_addr2.  
 Filer2 is a destination filer having interfaces: dest\_ip\_addr1 and dest\_ip\_addr2.  
 Two SnapMirror relations are configured between srcvol1 to destvol1 and srcvol2 to destvol2 using failover mode of multiple paths. Following table describes agent behavior:

**Table 3-3** Agent behaviour

Action	Source system	Destination system
Initial setup: There are two actively running SnapMirror relations between filer1 and filer2 with reference to connection My-conn. Agent has relations configured as resource1 and resource2 accordingly.		My-conn = failover (source_ip_addr1,dest_ip_addr1) (source_ip_addr2,dest_ip_addr2)  My-conn:srcvol1 filer2:destvol1 - 0-55/5 * * *  My-conn:srcvol2 filer2:destvol2 - 0-55/5 * * *

**Table 3-3** Agent behaviour (*continued*)

Action	Source system	Destination system
First relation(resource1) is migrated to Filer2: Agent detects multiple paths configured and derived connection name with resource1. Agent adds new connection entry My-conn to filer1 if existing connection is not there with same name. If there is existing connection with name My-conn, it will cross check if they are identical.	My-conn = failover (dest_ip_addr1, source_ip_addr1) (dest_ip_addr2, source_ip_addr2)  My-conn:destvol1 filer1:sourcevol1 - 0-55/5 * * *	My-conn = failover (source_ip_addr1,dest_ip_addr1) (source_ip_addr2,dest_ip_addr2)  My-conn:srcvol2 filer2:destvol2 - 0-55/5 * * *
First relation is migrated back to filer1: Check and add relationship entry. The agent does not delete connection entry on filer1.	My-conn = failover (dest_ip_addr1, source_ip_addr1) (dest_ip_addr2, source_ip_addr2)	My-conn = failover (source_ip_addr1,dest_ip_addr1) (source_ip_addr2,dest_ip_addr2)  My-conn:srcvol2 filer2:destvol2 - 0-55/5 * * *  My-conn:srcvol1 filer2:destvol1 - 0-55/5 * * *
Migrating both relations to Filer2: Check and add relationship entry. The agent won't delete connection entry on filer2.	My-conn = failover (dest_ip_addr1, source_ip_addr1) (dest_ip_addr2, source_ip_addr2)  My-conn:destvol1 filer1:sourcevol1 - 0-55/5 * * *  My-conn:destvol2 filer1:sourcevol2 - 0-55/5 * * *	My-conn = failover (source_ip_addr1,dest_ip_addr1) (source_ip_addr2,dest_ip_addr2)

# Managing and testing clustering support for Network Appliance SnapMirror

This chapter includes the following topics:

- [Performing a disaster test](#)
- [Failure scenarios for Network Appliance SnapMirror](#)

## Performing a disaster test

Test how robust your cluster is in case of a disaster.

### To perform a disaster test

- 1 Shut down the host on which the service group is online as well as the filer to which it is attached.  
  
If you cannot shut down the filer, disconnect it from the network to make it invisible to the remote site.
- 2 In a global cluster, the administrator is notified of the failure. The administrator can then initiate the failover.
- 3 Reboot the host and restart the filer. After the filer comes up, issue the swap action to resynchronize all data back to the source. After the resynchronization completes, switch the application back to the primary node.

# Failure scenarios for Network Appliance SnapMirror

Review the failure scenarios and agent behavior in response to failure.

## Site disaster

In a global cluster environment, if the site running the application fails, VCS detects the outage and recommends failing over to the DR site.

The agent for SnapMirror attempts a data migration, which means taking over the target data and reversing the role of replication. The data migration during a site disaster involves:

- Breaking the replication to enable write access to the target (through the NetAppSnapMirror agent).
- Updating the SnapMirror configuration to demote the original source to a target, and begin replicating from the new source (also through the NetAppSnapMirror agent).
- Starting the application services through the VCS agents for various applications.

If the original filer returns, then you must resynchronize the filer.

## All host or all application failure

In global cluster environments, multiple service groups can fail over in parallel.

In a global cluster environment, VCS detects the outage and recommends failing over to the DR site in one of the following cases:

- The primary host that runs the application fails
- The application itself fails

In this case, the VCS agents for SnapMirror attempt a data migration because both filers are healthy. The agent takes over the target data and reverses the role of replication. The data migration during a host or application failure involves:

- Breaking the replication to enable write access to the target (through the NetAppSnapMirror agent).
- Updating the SnapMirror configuration to demote the original source to a target, and begin replicating from the new source (also through the NetAppSnapMirror agent).
- Starting the application services through the VCS agents for various applications.

# Setting up a fire drill

This chapter includes the following topics:

- [About fire drills](#)
- [About the NetAppSnapMirrorSnap agent](#)
- [Before you configure the fire drill service group](#)
- [Configuring the fire drill service group](#)
- [Verifying a successful fire drill](#)
- [Sample configuration for a fire drill service group](#)

## About fire drills

A fire drill procedure verifies the fault-readiness of a disaster recovery configuration. This procedure is done without stopping the application at the primary site and disrupting user access.

A fire drill is performed at the secondary site using a special service group for fire drills. The fire drill service group is identical to the application service group, but uses a fire drill resource in place of the replication agent resource. The fire drill service group uses a copy of the data that is used by the application service group.

In clusters employing Network Appliance SnapMirror, the NetAppSnapMirrorSnap resource manages the replication relationship during a fire drill.

Bringing the fire drill service group online demonstrates the ability of the application service group to come online at the remote site when a failover occurs.

---

**Note:** The NetAppSnapMirrorSnap agent does not support fire drills for qtree.

---

**Note:** The VCS agent for SnapMirror supports only the Network File Systems (NFS) protocol.

# About the NetAppSnapMirrorSnap agent

The NetAppSnapMirrorSnap agent is the fire drill agent for Network Appliance SnapMirror.

The agent manages the replication relationship between the source and target arrays when running a fire drill. Configure the NetAppSnapMirrorSnap resource in the fire drill service group, in place of the NetAppSnapMirror resource.

The agent is configured in the fire drill service group and supports fire drills only for flexvols. The agent exports configured volumes as nfs shares.

## NetAppSnapMirrorSnap agent functions

The NetAppSnapMirrorSnap agent performs the following functions:

**Table 5-1** Agent functions

Function	Description
online	<ul style="list-style-type: none"><li>■ Takes a clone of the target volume that is used by the application.</li><li>■ Breaks the mirror relation between the source and target volumes.</li><li>■ Takes a clone of the target volume and exports it to bring up the application.</li><li>■ Breaks the mirror relation between the source and target volumes.</li><li>■ Exports the target volume without taking a snapshot.</li><li>■ Creates a lock file to indicate that the resource is online.</li></ul> <p>After the online operation is completed, the following nfs share becomes available: &lt;Filename&gt;:/vol/&lt;volname&gt;_fd</p>
offline	<p>Reestablishes the mirror relationship between the source and target volumes by doing a resync.</p> <p>Removes the lock file created by the Online operation.</p>
monitor	<p>Verifies the existence of the lock file to make sure the resource is online.</p>



**Table 5-1** Agent functions (*continued*)

Function	Description
clean	<ul style="list-style-type: none"> <li>■ Reestablishes the mirror relationship between the source and target volumes by doing a resync.</li> <li>■ Removes the lock file created by the Online operation.</li> </ul>
open	<p>If the lock file does not exist, the agent takes no action.</p> <p>If the lock file exists:</p> <ul style="list-style-type: none"> <li>■ If any resources that depend on the NetAppSnapMirrorSnap are online, the agent does not take any action.</li> <li>■ If any resources that depend on the NetAppSnapMirrorSnap are not online, the agent removes the lock file.</li> </ul>

## Resource type definition for the NetAppSnapMirrorSnap agent

Following is the resource type definition for the NetAppSnapMirrorSnap agent:

```
type NetAppSnapMirrorSnap (
    static int ActionTimeout = 120
    static keylist SupportedActions = { }
    static int MonitorInterval = 300
    static int NumThreads = 1
    static str ArgList[] = { TargetResName, FilerName,
        FilerPword, MountSO, UseSO, RequiredSO, SnapshotName}
    str TargetResName
    str FilerName
    str FilerPword
    str MountSO
    str UseSO
    str RequiredSO
    str SnapshotName
)
```

## Attribute definitions for the NetAppSnapMirrorSnap agent

To customize the behavior of the NetAppSnapMirrorSnap agent, configure the following attributes:

**Table 5-2** Agent attributes

Attribute	Description
TargetResName	<p>The name of the VCS resource that manages the object to be snapshot. This can be a SnapMirror type resource or a NetApp type resource.</p> <p>Type-Dimension: string-scalar</p>
FilerName	<p>The name of the filer to which the target volumes belong.</p> <p>Type-Dimension: string-scalar</p>
FilerPword	<p>The password to access the filer specified by the attribute FilerName.</p> <p>Type-Dimension: string-scalar</p>
UseSnapshot	<p>Specifies whether the NetAppSnapMirrorSnap resource takes a local snapshot of the target array. Set this attribute to 1.</p> <p>Type-Dimension: integer-scalar</p>
RequireSnapshot	<p>Specifies whether the NetAppSnapMirrorSnap resource must take a snapshot before coming online.</p> <p>Set this attribute to 1 if you want the resource to come online only after it succeeds in taking a snapshot.</p> <p>Type-Dimension: integer-scalar</p> <p><b>Note:</b> Set this attribute to 1 only if UseSnapshot is set to 1.</p>
MountSnapshot	<p>Specifies whether the resource uses the snapshot to bring the service group online. Set this attribute to 1.</p> <p>Type-Dimension: integer-scalar</p> <p><b>Note:</b> Set this attribute to 1 only if the UseSnapshot attribute is set to 1.</p>
SnapshotName	<p>For Gold configuration only.</p> <p>Specifies the name of the snapshot used to clone the target volume. For Example: hourly.0, nightly.0.</p> <p>Type-Dimension: string-scalar</p>

## Before you configure the fire drill service group

Before you configure the fire drill service group, ensure that the following pre-requisites are met:

- Make sure the application service group is configured with a NetAppSnapMirror resource.
- Make sure the infrastructure to take volume clones is properly configured between the source and target volumes.
- You must have a NetApp FlexClone license as well as NFS licensed and enabled to test the creation of a snapshot of the target flexvol.

## Configuring the fire drill service group

At the secondary site, the initial steps create a fire drill service group that closely follows the configuration of the original application service group. The fire drill service group uses a point-in-time copy of the data at the primary site. Bringing the fire drill service group online at the secondary site demonstrates the ability of the application service group to fail over and come online at the secondary site, should the need arise.

See [“Sample configuration for a fire drill service group”](#) on page 38.

You can create the fire drill service group using one of the following methods:

- Cluster Manager (Java Console)  
See [“Creating the fire drill service group using Cluster Manager \(Java Console\)”](#) on page 35.
- Fire Drill Setup wizard  
This text-based wizard is available at /opt/VRTSvcs/bin/fdsetup-ntap.  
See [“Creating the fire drill service group using the Fire Drill SetUp Wizard”](#) on page 37.

## Creating the fire drill service group using Cluster Manager (Java Console)

Open the Cluster Manager and perform the following procedure:

### To create the fire drill service group

- 1 Log on to the cluster and click **OK**.
- 2 Click the **Service Group** tab in the left pane and click the **Resources** tab in the right pane.

- 3 Right-click the cluster in the left pane and click **Add Service Group**.
- 4 In the **Add Service Group** dialog box, provide information about the new service group.
  - In Service Group name, enter a name for the fire drill service group, for example: **MyApplication\_AppSG\_FD**.
  - Select systems from the Available Systems box and click the arrows to add them to the Systems for Service Group box.
  - Click **OK**.
- 5 After the fire drill service group is successfully created, set the Failover attribute to **false** so that the fire drill service group does not fail over to another node during a test.

#### **To disable the AutoFailOver attribute**

- 1 Click the **Service Group** tab in the left pane and select the fire drill service group.
- 2 Click the **Properties** tab in the right pane.
- 3 Click the **Show all attributes** button.
- 4 Double-click the **AutoFailOver** attribute.
- 5 In the **Edit Attribute** dialog box, clear the **AutoFailOver** check box.
- 6 Click **OK** to close the **Edit Attribute** dialog box.
- 7 Click the **Save and Close Configuration** icon in the toolbar.

## **Adding resources to the fire drill service group**

Add resources to the new fire drill service group to re-create key aspects of the application service group.

#### **To add resources to the service group**

- 1 In Cluster Explorer, click the **Service Group** tab in the left pane, click the application service group and click the **Resources** tab in the right pane.
- 2 Right-click the resource at the top of the tree, select **Copy > Self and Child Nodes**.
- 3 In the left pane, click the fire drill service group.
- 4 Right-click the right pane, and click **Paste**.

- 5 In the **Name Clashes** dialog box, specify a way for the resource names to be modified, for example, insert an **\_FD** suffix. Click **Apply**.
- 6 Click **OK**.

## Configuring resources for fire drill service group

Edit the resources in the fire drill service group so they work properly with the duplicated data. The attributes must be modified to reflect the configuration at the remote site. Bringing the service group online without modifying resource attributes is likely to result in a cluster fault and interruption in service.

### To configure resources for the fire drill service group

- 1 In Cluster Explorer, click the **Service Group** tab in the left pane.
- 2 Click the fire drill service group in the left pane and click the **Resources** tab in the right pane.
- 3 Right-click the NetAppSnapMirror resource and click **Delete**.
- 4 Add a resource of type NetAppSnapMirrorSnap and configure its attributes.
- 5 Right-click the resource to be edited and click **View > Properties View**. If a resource to be edited does not appear in the pane, click **Show All Attributes**.
- 6 Edit the attributes to reflect the configuration at the remote site. For example, change the Mount resources so that they point to the volumes that are used in the fire drill service group.

## Creating the fire drill service group using the Fire Drill SetUp Wizard

This section describes how to use the Fire Drill SetUp Wizard to create the fire drill service group.

### To create the fire drill service group

- 1 Start the Fire Drill SetUp Wizard.  
  
`/opt/VRTSvcs/bin/fdsetup-ntap`
- 2 Enter the name of the application service group for which you want to configure a fire drill service group.
- 3 Select the supported snapshot configurations:  
  
Gold, Silver, or Bronze
- 4 Choose whether to run a Bronze fire drill, if the snapshot fails with Gold or Silver configurations.

If snapshot fails, should bronze be used? [y,n,q](n)

- 5 Press **Return** to verify the snapshot infrastructure.
- 6 Enter **y** to create the fire drill service group.

The wizard runs various commands to create the fire drill service group.
- 7 In Linux clusters, verify that the StartVolumes attribute for each DiskGroup type resource in the fire drill group is set to 1. If not, modify the resource to set the value to 1.
- 8 Schedule fire drill for the service group by adding the following command to the crontab to be run at regular intervals.

```
/opt/VRTSvc/bin/fdsched-netappsnapmirror
```

- 9 Make fire drill highly available by adding the following command to the crontab on every node in this cluster.

```
fdsched-netappsnapmirror
```

## Verifying a successful fire drill

Run the fire drill routine periodically to verify that the application service group can fail over to the remote node.

### To verify a successful fire drill

- 1 Bring the fire drill service group online on a node at the secondary site where the application is not running.

If the fire drill service group comes online, it validates your DR configuration. The service group at the primary site can fail over to the secondary site in the event of an actual failure or a disaster at the primary site.
- 2 If the fire drill service group does not come online, review the VCS engine log for more information.
- 3 Take the fire drill service group offline after its functioning has been verified.

Failing to take this service group offline may cause failures in your environment. For example, if the application service group fails over to the node where the fire drill service group is hosted, both the service groups may fault owing to resource conflicts.

## Sample configuration for a fire drill service group

The sample configuration of a fire drill service group is identical to an application service group with a hardware replication resource. However, in a fire drill service

group, the NetAppSnapMirrorSnap resource replaces the NetAppSnapMirror resource.

In the mount resource, the name of the volume gets replaced by <volname>\_fd.

You can configure a resource of type NetAppSnapMirrorSnap in the `main.cf` file as follows:

```
group firedrill_sg (
    SystemList = { thoribm90mp4 = 0 }
)
Mount firedrill_mount (
    MountPoint = "/ntap_sec_fd"
    BlockDevice = "netapp3:/vol/gmvolsec_aix_fd"
    FSType = nfs
)
NetAppSnapMirrorSnap ntap_sec_fd (
    TargetResName = ntap_sec
    FilerName = netapp3
    FilerPword = xxxx
    MountSnapshot = 1
    UseSnapshot = 0
    RequireSnapshot = 0
)
firedrill_mount requires ntap_sec_fd

// resource dependency tree
//
// group firedrill_sg
// {
// Mount firedrill_mount
// {
// NetAppSnapMirrorSnap ntap_sec_fd
// }
// }
```