XC™ Series SMW HA Administration Guide

(SLEHA12.SP3.UP07)

S-2551

Contents

1 About the XC™ Series SMW HA Administration Guide.... 4
1.1 Related Publications.... 6
2 SMW HA Overview.... 8
2.1 SMW Cluster Configuration.... 8
2.2 Network Connections for an SMW HA System.... 9
2.3 Shared Storage on the Boot RAID.... 10
2.4 Storage for the Power Management Database (PMDB).... 10
2.5 Synchronized Files.... 11
2.6 Cluster Resources.... 11
2.7 Limitations of SMW Failover.... 13
2.8 About Maintenance Mode.... 14
2.9 Operational Differences on an SMW HA System.... 14
3 About SMW HA Commands.... 16
3.1 The crm Command.... 17
3.2 The crm_mon Command.... 18
4 Manage an SMW HA System.... 20
4.1 Log into the SMW HA Cluster.... 20
4.2 Identify the Active SMW.... 21
4.3 Display SMW HA Cluster and Resource Status.... 21
4.4 Display SMW Power Status.... 23
4.5 Monitor the fsync Resource.... 25
4.6 Boot an SMW HA Cluster.... 27
4.7 Reboot the Passive SMW in Maintenance Mode.... 30
4.8 Change SMW, iDRAC, and STONITH Passwords.... 31
4.8.1 Change the Default iDRAC Password.... 32
4.9 Update the NIMS Map on the Passive SMW.... 32
4.10 Apply Firewall Rule Changes to the Passive SMW.... 33
4.11 Revert to a Previous SMW HA Snapshot.... 34
4.12 Enable Multipath on an Installed SMW HA System.... 35
5 Handle Failover.... 41
5.1 Critical Events That Cause SMW HA Failover.... 41
5.2 Examine the SMW HA Log File to Determine SMW Failover Cause.... 42
5.3 Restore Normal Operations After SMW Failover.... 43
5.4 Perform a Manual Failover.... 44
6 Change the SMW HA Cluster Configuration.... 47

6.1 Change Failover Notification 47
6.2 Add Site-specific Files to the Synchronization List 48
6.3 Set the Migration Threshold for a Resource 49
7 Troubleshoot an SMW HA System 51
7.1 Cray HA System Components 51
7.2 Group Resources 55
7.3 Cluster Resource Dependence 57
7.4 Check Cluster Status 58
7.5 Snapshot Failure 62
7.6 Reasons for Failover 63
7.7 Debug a Cluster Problem or Failover 65
7.8 Debug Failure of a Preinstalled SMW HA System 67
7.9 Restart Stopped Resources 67
7.10 Return an SMW to the HA Cluster After It Has Been Powered Off 70
7.11 Boot SMW HA When Both Are Powered Off 73
7.12 Check Heartbeat Cables for Repeated STONITH Failures 75
7.13 Clear an HSS Lock After Failover Occurs During a Cray System Boot 75
7.14 Recover System Settings After Failover During Discovery 76
7.15 Recover From Loss of Power to the Active SMW and iDRAC 76
7.16 Stop Repeated "Ping Pong" Failovers 78
7.17 PostgreSQL Does Not Start 78
7.18 Reinitialize the PMDB 81
7.19 Fix an Outdated DRBD Disk 83

1 About the XC™ Series SMW HA Administration Guide

Scope and Audience

The *XC™ Series SMW HA Administration Guide* (S-2551) describes how to monitor and manage a Cray XC System Management Workstation (SMW) High Availability (HA) system and summarizes the features of this system. An SMW HA system is a Cray XC system with two second-generation rack-mount SMWs, either Dell PowerEdge R815s or Dell PowerEdge R630s. The SMWs run the SUSE Linux Enterprise High Availability (SLEHA) Extension and the Cray SMW High Availability Extension release package, also called the SMW HA package. The two SMWs must have the same hardware, software, and configuration settings.

The information and HA-specific procedures of this guide supplement the Cray system administration information in the *XC™ Series System Administration Guide* (S-2393).

This publication is intended for system administrators of a Cray XC Series system. It assumes some familiarity with standard Linux and open source tools (e.g., zypper/yum for RPMs, Ansible, YAML/JSON configuration data).

SMW HA SLEHA12.SP3.UP07 Release

XC™ Series SMW HA Administration Guide (SLEHA12.SP3.UP07) S-2551 supports Cray software release SMW HA SLEHA12.SP3.UP07 for Cray XC™ Series systems, released on 12 July 2018.

In previous releases, this publication was titled *SMW HA Administration Guide* (S-2551-B) and *Installing, Configuring, and Managing SMW Failover on the Cray XC System* (S-0044-D).

New in this release

- The troubleshooting section has been expanded.

Host Name Conventions for SMW HA Systems

These host name conventions are used to refer to the SMWs in an HA cluster:

`smw1#`	Specifies the currently active SMW.
`smw2#`	Specifies the currently passive SMW.
`virtual-smw#`	Specifies the virtual (active) SMW, which could be either `smw1` or `smw2`.

Command Prompt Conventions

Host name and account in	The host name in a command prompt indicates where the command must be run. The account that must run the command is also indicated in the prompt.

command prompts

- The `root` or super-user account always has the # character at the end of the prompt.
- Any non-`root` account is indicated with *`account`*`@`*`hostname`*`>`. A user account that is neither `root` nor `crayadm` is referred to as `user`.

`smw#`	Run the command on the SMW as `root`.
`cmc#`	Run the command on the CMC as `root`.
`sdb#`	Run the command on the SDB node as `root`.
`crayadm@boot>`	Run the command on the boot node as the `crayadm` user.
`user@login>`	Run the command on any login node as any non-`root` user.
`hostname``#`	Run the command on the specified system as `root`.
`user@`*`hostname`*`>`	Run the command on the specified system as any non-`root` user.
`smw1#` `smw2#`	For a system configured with the SMW failover feature there are two SMWs—one in an active role and the other in a passive role. The SMW that is active at the start of a procedure is smw1. The SMW that is passive is smw2.
`smwactive#` `smwpassive#`	In some scenarios, the active SMW is smw1 at the start of a procedure—then the procedure requires a failover to the other SMW. In this case, the documentation will continue to refer to the formerly active SMW as smw1, even though smw2 is now the active SMW. If further clarification is needed in a procedure, the active SMW will be called smwactive and the passive SMW will be called smwpassive.

Command prompt inside chroot

If the `chroot` command is used, the prompt changes to indicate that it is inside a chroot environment on the system.

```
smw# chroot /path/to/chroot
chroot-smw#
```

Directory path in command prompt

Example prompts do not include the directory path, because long paths can reduce the clarity of examples. Most of the time, the command can be executed from any directory. When it matters which directory the command is invoked within, the `cd` command is used to change into the directory, and the directory is referenced with a period (`.`) to indicate the current directory.

For example, here are actual prompts as they appear on the system:

```
smw:~ # cd /etc
smw:/etc# cd /var/tmp
```

```
smw:/var/tmp# ls ./file
smw:/var/tmp# su - crayadm
crayadm@smw:~> cd /usr/bin
crayadm@smw:/usr/bin> ./command
```

And here are the same prompts as they appear in this publication:

```
smw# cd /etc
smw# cd /var/tmp
smw# ls ./file
smw# su - crayadm
crayadm@smw> cd /usr/bin
crayadm@smw> ./command
```

Typographic Conventions

`Monospace`	Indicates program code, reserved words, library functions, command-line prompts, screen output, file/path names, and other software constructs.
`Monospaced Bold`	Indicates commands that must be entered on a command line or in response to an interactive prompt.
`Oblique` or *`Italics`*	Indicates user-supplied values in commands or syntax definitions.
Proportional Bold	Indicates a **GUI Window**, **GUI element**, cascading menu (**Ctrl**→**Alt**→**Delete**), or key strokes (press **Enter**).
\ (backslash)	At the end of a command line, indicates the Linux® shell line continuation character (lines joined by a backslash are parsed as a single line).

Trademarks

1.1 Related Publications

The following publications contain additional information that may be helpful. Unless otherwise noted, these publications can be found at *http://pubs.cray.com*.

- *SMW HA Release Errata* and the *SMW HA README* (provided with the SMW HA software release package)
- *CLE Release Errata* and the *CLE README* (provided with the CLE release software)
- *SMW Release Errata* (includes notice of any day-1 patches) and the *SMW README*, which are provided with the SMW release software

- *XC™ Series SMW HA Installation Guide* (S-0044)
- SUSE Linux Enterprise High Availability (SLEHA) Extension 12 documentation, which provides information on the SUSE HA software, the Pacemaker Cluster Resource Manager (CRM), and related tools. SUSE manuals can be found in the `docu` directory of the SLEHA installation media, or in the directory `/usr/share/doc/` on the installed system (if installed).
- *XC™ Series System Administration Guide* (S-2393)
- *XC™ Series Configurator User Guide* (S-2560)
- *XC™ Series Lustre® Administration Guide* (S-2648)
- *XC™ Series Power Management and SEDC Administration Guide* (S-0043)
- *XC™ Series System Environment Data Collections (SEDC) Guide* (S-2491)
- *XC™ Series DataWarp™ Installation and Administration Guide* (S-2564), which supersedes *DataWarp Installation Guide* (S-2547)
- *Cray Compiling Environment Release Overview and Installation Guide*
- *XC™ Series eLogin Installation Guide* (S-2556)
- *XC™ Series SEC and check_xt Software Configuration Guide* (S-2542) (Simple Event Correlator)
- *XC™ Series Aries™ Network Resiliency Guide* (S-0041)
- *XC™ Series DVS Administration Guide* (S-0005)

2 SMW HA Overview

The Cray System Management Workstation (SMW) High Availability system supports SMW failover. An SMW High Availability (HA) system is a Cray XC system with two second-generation high-end SMWs (also called rack-mount SMWs) that run the SUSE Linux Enterprise High Availability (SLEHA) Extension and the Cray SMW High Availability Extension (SMWHA) release package. The two SMWs must be installed and configured as specified in the install guide.

The SMW failover feature provides improved reliability, availability, and serviceability (RAS) of the SMW, allowing the mainframe to operate correctly and at full speed. This feature adds SMW failover, fencing, health monitoring, and failover notification. Administrators can be notified of SMW software or hardware problems in real time and be able to react by manually shutting down nodes, or allowing the software to manage the problems. In the event of a hardware failure or `rsms` (HSS) daemon failure, the software will fail over to the passive SMW node, which becomes the active node. The failed node, once repaired, can be returned to the configuration as the passive node.

The SUSE Pacemaker Cluster Resource Manager (CRM) provides administration and monitoring of the SMW HA system with a command-line interface (`crm`). With this interface and associated commands, the SMW administrator can display cluster status, monitor the HSS daemons (configured as cluster resources), configure automatic failover notification by email, and customize the SMW failover thresholds for each resource. The Cray SMWHA software includes scripts and commands to simplify the CRM interface and help with specific tasks on an SMW HA system.

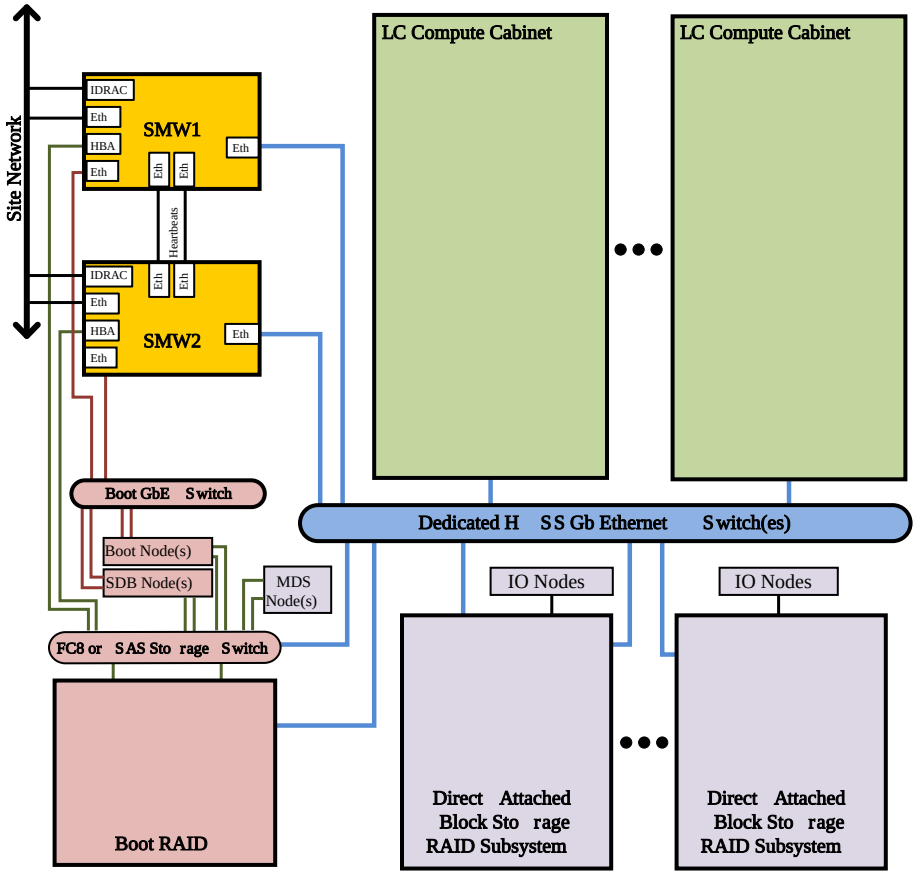
The Pacemaker Cluster Resource Manager uses the term node to refer to a host in a CRM cluster. On an SMW HA system, a CRM node is an SMW, not a Cray XC compute or service node.

2.1 SMW Cluster Configuration

Both SMWs are connected to the boot RAID, and are connected to each other with heartbeat cables between the `eth2` and `eth4` ports on each SMW. The heartbeat connection monitors the health of the cluster. In addition, each SMW is connected to the boot RAID (through FC or SAS cards), to the site network through `eth0`, to the HSS network through `eth1`, and to the boot node through `eth3`. (For more information, see *Network Connections for an SMW HA System* on page 9.) An Integrated Dell™ Remote Access Controller (iDRAC) is required on both SMWs.

The following figure shows the major connections between components in an SMW HA system.

Figure 1. SMW HA Hardware Components for a Cray XC System



In a Cray SMW HA cluster, the two SMWs are configured in an active/passive configuration. This configuration lets the passive node take over the SMW functions if a software or hardware fault occurs on the active node. All HSS daemons run on the active SMW. (An additional `stonith` daemon, which monitors SMW health, runs on both SMWs.) At failover, all daemons move to the passive SMW, which then becomes the active one.

During initial installation, the first SMW that is installed and configured becomes the active SMW. The second SMW that is installed and configured becomes the passive SMW. However, either SMW can be active during normal operation. The cluster configuration does not remember which SMW was initially configured to be active.

2.2 Network Connections for an SMW HA System

An SMW HA system uses `eth2` and `eth4` (on the second Ethernet card) for heartbeat connections to the other SMW, in addition to the network connections required for single SMW.

Each SMW must be connected to the customer network through `eth0`, to the HSS network through `eth1`, to the boot node through `eth3`, and to the boot RAID through the Fibre Channel (FC) or SAS card. In addition, `eth2` and `eth4` must directly connect the two SMWs to each other as heartbeat monitoring channels.

Each SMW must have the following private network connections:

`eth0` - To the customer network
`eth1` - To the HSS network

`eth2` - To the other SMW (heartbeat connection)
`eth3` - To the boot node
`eth4` - To the other SMW (redundant heartbeat connection)
`eth5` - To the other SMW for the Power Management Database (PMDB) replication

2.3 Shared Storage on the Boot RAID

The SMW HA system shares the following file systems on the boot RAID for data that must be highly available. These shared file systems are mounted only on the active SMW. When a failover occurs, access to these directories is automatically transferred to the other SMW as part of the failover process.

Device	Mount point	Description
`/dev/mapper/smw_node_vg-log`	`/var/opt/cray/disk/1`	Log disk. The following directories symbolically link to the Log disk: • `/var/opt/cray/debug` • `/var/opt/cray/dump` • `/var/opt/cray/log`
`/dev/mapper/smw_node_vg-home`	`/home`	SMW home directories.
`/dev/mapper/smw_node_vg-db`	`/var/lib/mysql`	Maria DB (MySQL) HSS database. Although the database is shared, the HSS database server runs on the active SMW only.
`/dev/mapper/smw_node_vg-repos`	`/var/opt/cray/repos`	Location of the SLE, SMW, and CLE RPM repositories.
`/dev/mapper/smw_node_vg-imps`	`/var/opt/cray/imps`	Location of global and CLE config sets, CLE boot images, and CLE image roots.

Do not use tune2fs on these `xfs` and `btrfs` file systems. Because these file systems are shared between the two SMWs, an SMW HA system has a slightly increased risk for double-mount problems. Do **not** mount any of these on both SMWs at the same time.

2.4 Storage for the Power Management Database (PMDB)

The Power Management Database (PMDB) is a PostgreSQL database that contains power management data, event router file system (`erfs`) data, and (optionally) System Environment Data Collections (SEDC) data. The directory `/var/lib/pgsql` is the mount point for the PMDB storage.

On an SMW HA system, the PMDB is kept in sync between the two SMWs using database replication and rewind. In this configuration, the active SMW mounts `/var/lib/pgsql` and communicates database insertions over a private TCP/IP connection (eth5) to the passive SMW.

2.5 Synchronized Files

For files not located on shared storage (boot RAID), the SLEHA Extension software includes the `csync2` utility to synchronize (sync) important files between the two SMWs. When a file changes on the active SMW, it is automatically synchronized to the passive SMW. The `csync2` utility synchronizes the required files and directories for the SMW HA cluster, such as `/etc/passwd` and `/opt/cray/hss/*/etc/*`.

File synchronization is automatically configured during initial installation. The file `/etc/csync2/csync2_cray.cfg` lists the Cray-specific files and directories that must be synchronized, as well as small files that are convenient to keep in sync.

File synchronization happens in one direction only: from the active SMW to the passive SMW. If a synchronized file changes on the passive SMW, the change will not be propagated to the active SMW in the course of normal operations and could be overwritten on the passive SMW later if there is a subsequent change to the corresponding file on the active SMW. However, if a failover occurs, the previously passive SMW becomes the active SMW. If the change is still in place, the changed file becomes a candidate for propagation to the other SMW (subject to the rules of file conflict resolution).

The `fsync` resource controls file synchronization operations. Every 100 seconds, `fsync` checks for files that need to be synchronized.

> **IMPORTANT:** If a failover occurs before a file synchronization operation has completed, it could result in the loss of the latest updates.

Very large files are explicitly excluded from synchronization (such as `/opt/cray/hss-images/master`). The `csync2` utility is designed to synchronize small amounts of data. If `csync2` must monitor many directories or synchronize a large amount of data, it can become overloaded and failures may not be readily apparent. Cray recommends that sites do not change the list of synchronized files (or add only small files); instead, copy large files and directories manually to the other SMW.

2.6 Cluster Resources

A resource is any type of service or application that is managed by the Pacemaker Cluster Resource Manager, such as a daemon or file system. In an SMW HA system, the HSS (rsms) daemons are configured as resources.

Each time a resource fails, it is automatically restarted and its failcount is raised. If the failcount exceeds the defined migration threshold for the resource, a failover occurs and management of all cluster resources migrates to the other SMW, making it the active SMW. The original SMW will no longer be allowed to run the failed resource, so no failback can occur until the resource's failcount is reset for that SMW.

Reset failcounts with the `clean_resources` or `clear_failcounts` command. For more information, see *Restart Stopped Resources* on page 67.

An SMW HA system includes the following resources:

`clo_PostgreSQL` Controls and monitors the cloned PostgreSQL resource that now runs on both the Active and Passive SMWs.

ClusterIP, ClusterIP1, ClusterIP2, ClusterIP3, ClusterIP4, and ClusterIP5	Control and monitor the Ethernet connections (`eth0`, `eth1`, `eth2`, `eth3`, and `eth4`, respectively).
ClusterMonitor	Records failcounts and failed actions in the log file `/var/log/smwha.log` at cluster startup, then clears the failure data from `crm` (for example, in the output of `crm_mon -r1`).
ClusterTimeSync	Monitors the kernel time on each SMW. If the difference is greater than 60 seconds, both SMWs are synchronized with the NTP server. If the time difference is greater than 10 hours, the time is not synchronized.
cray-ansible	Starts/stops the `/etc/init.d/cray-ansible` script when the cluster starts and stops.
cray-cfgset-cache	Monitors the config set services.
cray-ids-service	Controls and monitors the Image Distribution Service.
cray-syslog	Controls and monitors Lightweight Log Management (LLM).
dhcpd	Controls and monitors `dhcpd` as used by the SMW HA feature.
EnableRsyslog	Controls enabling and disabling of the `rsyslog` service.
fsync	Provides file synchronization using `csync2`.
homedir	Mounts and unmounts the shared `/home` directory.
HSSDaemonMonitor	Periodically monitors all HSS daemons. If a daemon crashes, `systemd` will restart it. At the same time, the failcount for this resource is increased by one. When the failcount reaches the failcount threshold, a failover is triggered.
hss-daemons	Controls and monitors HSS daemons; corresponds to the `/etc/init.d/rsms` startup script.
IMPSFilesystemConfig	Refreshes stale mount points from the SMW on the boot node.
imps-fs	Mounts and unmounts the `/var/opt/cray/imps` directory.
LOGFilesystemConfig	Refreshes stale mount points from the SMW on the boot node.
NFSServer	Controls and monitors the NFS server, `nfsserver`.
Notification	Provides automatic notification email when a failover occurs.
md-fs	Mounts, unmounts, and monitors the shared MySQL database, `/var/lib/mysql`.
ml-fs	Mounts, unmounts, and monitors the shared log directory, `/var/opt/cray/disk/1`, which symbolically links to the `dump`, `install`, and `log` subdirectories in `/var/opt/cray/`.
mysqld	Controls and monitors MySQL.

postgreSQL	Controls and monitors the Power Management Database (PMDB) PostgreSQL server, `postgresqld`.
repos-fs	Mounts `/var/opt/cray/repos` to the boot RAID connected to the active SMW and ensures that it is not mounted on the passive SMW. This is very important because a double mount (mount from both active and passive SMW) will corrupt the disk.
ResourceInit	Scans `/etc/fstab` to find the mount options for the `mysql`, `/home`, `log`, `imps`, and `repos` disks. Then it configures the cluster resources to match the options defined in `/etc/fstab`. Note that the disk options defined in `/etc/fstab` include the subvols, which are pointed to the current snapshot. **IMPORTANT:** Ensure that both the active and passive SMW have the same current snapshot name so that that the same subvols are used after a failover.
stonith-1 and stonith-2	Monitors the health of the other SMW. Each SMW monitors its peer and has the ability to power off that peer at failover time using the STONITH capability. STONITH failovers are used when the state of the failing SMW cannot be determined. A STONITH failover powers off the failing SMW to guarantee that the newly active SMW has exclusive access to all cluster managed resources.
syslog.socket	Controls and monitors the syslog.socket portion of `rsyslog`, `syslog.socket`.

2.7 Limitations of SMW Failover

The SMW HA failover feature has the following limitations:

- Both SMWs must be of the same type, either Dell PowerEdge 815s or Dell PowerEdge 630s.
- Both SMWs must run the same versions of SLES and SMW/HSS software.
- System administration of an SMW HA environment is more complex than administration of a system with a single SMW.
- HSS Commands on the SMW do not work immediately after a failover. The failover takes time to complete. If an HSS command is issued on the active SMW before the HA system has settled, it will fail. The system must complete the failover before HSS commands are executed.

  After a failover, there is a delay before the cluster is fully available. In the first 30 seconds after failover, resources may appear to be started, then change to another state. Although logging in via the virtual IP address is possible before this period is over, the cluster is not ready for use until all resources are fully started. Cray recommends waiting for 30 - 60 seconds before using a command that interacts with the HSS daemons, to ensure that all cluster resources have started and that the shared file systems are mounted on the active SMW.
- SMW and CLE upgrades in an HA environment require some duplication of effort, with portions of the procedure done individually to each SMW. System down-time requirements for operating system upgrades are somewhat longer as a result.
- There is no support for seamless failover (also called double failure) if errors occur while the system is doing error handling for another system component. If an HSS daemon or other SMW process were doing some type of error handling that got interrupted by an (unrelated) failover, when that daemon restarts on the new

SMW it may not be able to resume operation where it left off and complete the recovery from the first error. In this case, even though a failover occurs, manual intervention might still be required to return the system to an operational state.

- There is no support for seamless failover during operational commands. All user commands that were started from the active SMW are terminated. These commands must be restarted on the new active SMW. The restarted commands might not start with the same internal states, if those commands do not provide persistent capabilities. An interrupted operation such as `xtbootsys`, shutdown, dump, warm-swap, or flash will need to be reissued after failover has completed and the other SMW becomes active.
- Partial migration of managed resources is not supported. For example, the SMW HA system does not support migration of individual HSS daemons or resources to the other SMW. A particular SMW is either active, with complete responsibility for all HSS daemons, or passive with no HSS daemons running.
- If both SMWs are started (powered on) at the same time, a race condition can develop that could result in one SMW being powered off via the STONITH capability. Before starting the second SMW, wait until the first SMW has completed startup and initialized all cluster resources.
- During failover, if there is no communication between the SMW and the Cray mainframe for about 30 seconds, workload throttling can occur; therefore, auto-throttling of applications is likely while an actual SMW failover is taking place. Blades begin to auto-throttle if essential HSS daemons (`erd`, `state-manager`, or `xtnlrd`) are unavailable and lasts until those daemons resume operation on the other SMW. On a single-cabinet system, the throttled period was fairly consistent, lasting approximately 120 seconds. The throttled period may increase for larger systems.

2.8 About Maintenance Mode

When the cluster is in maintenance mode all resources in the cluster are not controlled by the cluster, which means:

- The cluster will not restart a resource if it is stopped.
- Users can control the resources, bypassing the cluster software.
- If the active SMW is rebooted, no resources will be started.
- No failover is triggered if the cluster is in maintenance mode.
- There is no protection from double mounts.

2.9 Operational Differences on an SMW HA System

The SMW HA system includes the following operational differences from running a single SMW:

- On an SMW HA system, control the `rsms` daemon as `root` rather than as `crayadm`. In addition, restarting `rsms` behaves differently than on a system with a single SMW. Running `/etc/init.d/rsms restart` does not display the expected output, because the HA cluster returns immediately rather than waiting for the HSS daemons to start. To display the daemon status, run `/etc/init.d/rsms status`.
- Key system services (also called resources) are controlled by the cluster manager (see *Cluster Resources* on page 11). Do not start or stop these services individually. Instead, use cluster management tools to start and stop these services. For more information, see *About SMW HA Commands* on page 16.

- Auto-throttling of applications is likely while an actual SMW failover is taking place. Blades begin to auto-throttle if essential HSS daemons (`erd`, `state-manager`, or `xtnlrd`) are unavailable and lasts until those daemons resume operation on the other SMW. On a single-cabinet system, the throttled period was fairly consistent, lasting 37 seconds. The throttled period may increase for larger systems.
- Because several file systems are shared between the two SMWs, an SMW HA system has a slightly increased risk for double-mount problems. Do not mount the shared file systems from the boot RAID on both SMWs at the same time.

3 About SMW HA Commands

The SUSE Pacemaker Cluster Resource Manager (CRM) software includes several commands to monitor and manage a cluster.

`crm`	Pacemaker command line interface for configuration and management.
`crm_mon`	Provides a summary of cluster's current state.
`crm_resource`	Perform tasks related to cluster resources. Allows resources to be queried (definition and location), modified, and moved around the cluster.

The Cray SMW HA software includes additional commands for an SMW HA cluster.

`clean_resources`	Cleans up all HA failover resources on both SMWs by setting the status of each resource to the default clean state and the failcount (number of failures) to 0. If some resources did not start after system boot or are marked as unclean after failover, use this command to quickly clean up all resources on both SMWs. The command `crm resource cleanup` also cleans up resources, but requires entering each resource name separately. After running `clean_resources`, wait several minutes for cluster activity to settle. **IMPORTANT:** Before running the `clean_resources` command, log in directly as `root` (instead of using `su` from a `crayadm` login), because `clean_resources` terminates all non-root user sessions.
`clear_failcounts`	Resets the failcounts and failed action data for all SMW failover resources, instead of running multiple `crm` or `crm_failcount` commands.
`ha_health`	Displays a summary of the SMW HA cluster health.
`ha_sync_nims_map`	Synchronizes the Node Image Mapping Service (NIMS) links on the active SMW to the passive SMW.
`maintenance_mode_configure`	Enables or disables maintenance mode for the cluster and checks the cluster maintenance status.
`show_failcount`	Displays the failcount of a specific SMW failover resource (any type of service or application that is managed by the Pacemaker Cluster Resource Manager, such as a daemon or file system).
`show_failcounts`	Displays the failcounts of all SMW failover resources on both SMWs, instead of running multiple `crm` or `crm_failcount` commands.
`set_migration_threshold`	Sets the migration threshold for an SMW failover resource. A migration threshold is defined as the maximum number of failures (the failcount)

	allowed for the resource. If the failcount exceeds this threshold, a failover occurs and management of all cluster resources migrates to the other SMW, making it the active SMW. The original SMW will no longer be allowed to run the failed resource until the resource's failcount is reset for that SMW. By default, the migration threshold is `1000000`.
`show_migration_threshold`	Displays the migration threshold for an SMW failover resource. The migration threshold must be set (with `set_migration_threshold`); otherwise, this command displays an error message.
`SMWHAconfig`	Configures SMW failover on both SMWs in an SMW HA cluster when installing or updating the Cray SMW HA software.

Only the `root` user can execute the Cray SMW HA commands. These commands are included in the `ha-smw` module, which is automatically loaded when the `root` user logs in. If necessary, use the following command to load the `ha-smw` module:

```
smw1# module load ha-smw
```

3.1 The crm Command

The `crm` command provides a command-line interface to the SUSE Pacemaker Cluster Resource Manager (CRM). This command can be used either as an interactive shell or as a single command entered on the command line.

For example, execute the following command to display a list of all cluster resources on the system.

```
smw1# crm resource show
 ClusterIP      (ocf::heartbeat:IPaddr2):       Started
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP5     (ocf::heartbeat:IPaddr2):       Started
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started
 HSSDaemonMonitor       (ocf::smw:HSSDaemonMonitor):    Started
 Notification   (ocf::heartbeat:MailTo):        Started
 ResourceInit   (ocf::smw:ResourceInit):        Started
 cray-cfgset-cache      (systemd:cray-cfgset-cache):    Started
 dhcpd  (systemd:dhcpd.service):        Started
 fsync  (ocf::smw:fsync):       Started
 hss-daemons    (lsb:rsms):     Started
 stonith-1      (stonith:external/ipmi):        Started
 stonith-2      (stonith:external/ipmi):        Started
 Resource Group: HSSGroup
     mysqld     (ocf::heartbeat:mysql): Started
 Resource Group: IMPSGroup
     cray-ids-service   (systemd:cray-ids-service):     Started
     cray-ansible       (systemd:cray-ansible): Started
     IMPSFilesystemConfig       (ocf::smw:FileSystemConfig):    Started
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started
 Resource Group: SharedFilesystemGroup
```

```
    homedir    (ocf::heartbeat:Filesystem):    Started
    md-fs      (ocf::heartbeat:Filesystem):    Started
    imps-fs    (ocf::heartbeat:Filesystem):    Started
    ml-fs      (ocf::heartbeat:Filesystem):    Started
    repos-fs   (ocf::heartbeat:Filesystem):    Started
Resource Group: SystemGroup
    NFSServer  (systemd:nfsserver):    Started
    EnableRsyslog      (ocf::smw:EnableRsyslog):       Started
    syslog.socket      (systemd:syslog.socket):        Started
Clone Set: clo_PostgreSQL [PostgreSQL]
    Started: [ smw1 smw2 ]
```

To display the status of a single resource, such as `fsync`, execute the following command:

```
smw1# crm resource status fsync
resource fsync is running on: smw1
```

To display the same information with the interactive interface:

```
smw1# crm
crm(live)# resource
crm(live)resource# status fsync
resource fsync is running on: smw1
crm(live)resource# end
crm(live)# quit
bye
smw1#
```

The `crm` command has multiple levels. Use the `help` keyword to display the commands at each level and the valid options and arguments for each command. For example, the following commands display different levels of help:

- `crm help`
- `crm resource help`
- `crm resource failcount help`

For more information, see the `crm(8)` man page and the SUSE Linux Enterprise High Availability Extension documentation.

3.2 The crm_mon Command

The SUSE `crm_mon` command helps monitor cluster status and configuration. The output includes the number of nodes, host names, SMW status, the resources configured in the cluster, the current status of each resource, and any failed actions.

By default (if no options are specified), `crm_mon` shows only the active (started) resources and runs continuously, redisplaying the cluster status every 15 seconds. To specify the number of repeats, enter a number as an option. To include inactive (stopped) resources, use the `-r` option. This example uses `-r1` to display one snapshot of cluster status including all resources.

```
smw1# crm status
Stack: unknown
Current DC: smw1 (version unknown) - partition with quorum
Last updated: Fri Feb  9 14:38:17 2018
Last change: Fri Feb  9 06:28:13 2018 by hacluster via crmd on smw1
```

```
2 nodes configured
36 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP5     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
 HSSDaemonMonitor       (ocf::smw:HSSDaemonMonitor):    Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 ResourceInit   (ocf::smw:ResourceInit):        Started smw1
 cray-cfgset-cache      (systemd:cray-cfgset-cache):    Started smw1
 dhcpd  (systemd:dhcpd.service):        Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 Resource Group: HSSGroup
     mysqld     (ocf::heartbeat:mysql): Started smw1
 Resource Group: IMPSGroup
     cray-ids-service   (systemd:cray-ids-service):     Started smw1
     cray-ansible       (systemd:cray-ansible): Started smw1
     IMPSFilesystemConfig       (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started purie1-smw
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
 Clone Set: clo_PostgreSQL [PostgreSQL]
     Started: [ smw1 smw2 ]
 ClusterIP6     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP7     (ocf::heartbeat:IPaddr2):       Started smw1
 esd    (systemd:esd.service):  Started smw1
```

`crm_mon` may display different resource names, group names, or resource order on the system.

The `crm_mon` output marks one of the nodes as the `Current DC`, which stands for designated coordinator. This is a Pacemaker CRM concept that is not related to the SMW's current active or passive role. The active SMW is not necessarily the CRM designated coordinator.

For more information, see the `crm_mon(8)` man page and the SUSE Linux Enterprise High Availability Extension documentation.

4 Manage an SMW HA System

Most administration tasks for an SMW HA system are the same as those for a system with a single SMW. The HA-specific procedures include:

- *Log into the SMW HA Cluster* on page 20
- *Identify the Active SMW* on page 21
- *Display SMW HA Cluster and Resource Status* on page 21
- *Display SMW Power Status* on page 23
- *Monitor the fsync Resource* on page 25
- *Boot an SMW HA Cluster* on page 27
- *Reboot the Passive SMW in Maintenance Mode* on page 30
- *Change SMW, iDRAC, and STONITH Passwords* on page 31
- *Update the NIMS Map on the Passive SMW* on page 32
- *Apply Firewall Rule Changes to the Passive SMW* on page 33
- *Revert to a Previous SMW HA Snapshot* on page 34
- *Enable Multipath on an Installed SMW HA System* on page 35

All SMW HA commands and procedures require `root` permissions unless otherwise noted.

4.1 Log into the SMW HA Cluster

About this task

In general, Cray recommends using the virtual host name to connect to the SMW cluster. Avoid connecting to an SMW by specifying the actual host names, except for host-specific maintenance. In the event of a failover, all connections made using the virtual host name will be terminated. A connection to the active SMW via the actual host name could be confusing after a failover occurs, because the login session would remain open, but there is no indication that the SMW is now passive.

Procedure

1. To log on to the active SMW, specify the virtual SMW host name (in this example, *`virtual-smw`*).

```
remote-system% ssh root@virtual-smw
smw1#
```

2. Note the host name of the SMW after logging in. The prompt displays the host name of the active SMW (in this example, `smw1`).

Use the actual host name of the SMW (such as `smw1` or `smw2`) to log on to a specific SMW.

4.2 Identify the Active SMW

About this task

Use one of these methods to identify the active SMW in an HA cluster.

Procedure

1. The easiest way to find the active SMW is to log in using the virtual SMW host name and look at the host name in the prompt.

```
remote-system% ssh root@virtual-smw
smw1#
```

2. Another way to find the active SMW is to determine where the HA cluster resources are running. One `stonith` resource runs on each SMW to monitor the other SMW. All other resources run only on the active SMW.

   As `root` on either SMW, execute the following command. Specify any HA resource, such as `homedir` or `fsync`.

```
smw1# crm_mon -r1 | grep homedir
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
```

3. Or use the `ha_health` command as `root` on either SMW.

```
smw1# ha_health | grep -i active
Active Node                  : smw1
```

4.3 Display SMW HA Cluster and Resource Status

About this task

Use some or all of the following steps to check the health of the SMW HA cluster. Execute the CRM and Cray SMW HA commands as `root`. Unless otherwise noted, you can execute these commands on either SMW.

Procedure

1. Verify that both SMWs are online. This procedure assumes that `smw1` is the active SMW

```
smw1# crm_mon -r1 | grep Online
Online: [ smw1 smw2 ]
```

2. Display the cluster status with `crm_mon -r1`.

```
smw1# crm status
Stack: unknown
Current DC: smw2 (version unknown) - partition with quorum
Last updated: Tue Aug 29 16:25:47 2017
Last change: Tue Aug 29 09:03:49 2017 by root via crm_attribute on smw2

2 nodes configured
35 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
.
.
.
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started smw1
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
 ip-drbd-pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
 Master/Slave Set: ms-drbd-pgsql [drbd-pgsql]
     Masters: [ smw1 ]
     Slaves: [ smw2 ]
```

Note that `crm_mon` may display different resource names, group names, or resource order on the system.

All resources run only on the active SMW (except for one `stonith` resource, which is a special case). Failed actions can be cleared by using the `clear_failcounts` command. Any failed actions that display again indicate issues with the resources. For more information, see *Restart Stopped Resources* on page 67.

3. Display the status of the cluster resources.

```
smw1# crm resource status
 ClusterIP      (ocf::heartbeat:IPaddr2):       Started
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started
```

```
HSSDaemonMonitor       (ocf::smw:HSSDaemonMonitor):    Started
Notification   (ocf::heartbeat:MailTo):        Started
cray-cfgset-cache      (systemd:cray-cfgset-cache):    Started
dhcpd  (systemd:dhcpd.service):        Started
fsync  (ocf::smw:fsync):       Stopped
hss-daemons    (lsb:rsms):     Started
.
.
.
```

4. Display failcount data for all resources.

```
smw1# show_failcounts
node=smw1 scope=status  name=fail-count-ClusterIP value=0
node=smw1 scope=status  name=fail-count-ClusterIP1 value=0
node=smw1 scope=status  name=fail-count-ClusterIP2 value=0
node=smw1 scope=status  name=fail-count-ClusterIP3 value=0
node=smw1 scope=status  name=fail-count-ClusterIP4 value=0
node=smw1 scope=status  name=fail-count-ClusterMonitor value=0
node=smw1 scope=status  name=fail-count-ClusterTimeSync value=0
node=smw1 scope=status  name=fail-count-HSSDaemonMonitor value=0
node=smw1 scope=status  name=fail-count-Notification value=0
node=smw1 scope=status  name=fail-count-cray-cfgset-cache value=0
node=smw1 scope=status  name=fail-count-dhcpd value=0
node=smw1 scope=status  name=fail-count-fsync value=1000001
node=smw1 scope=status  name=fail-count-hss-daemons value=0
 .
 .
 .
```

5. Test file synchronization.

a. Create a temporary file in a synchronized directory on the active SMW, then check for it on the passive SMW.

```
smw1# cp /etc/motd /opt/cray/hss/default/etc/my_test_file
smw1# ls -l /opt/cray/hss/default/etc/my_test_file
smw1# md5sum /opt/cray/hss/default/etc/my_test_file

... (wait about 2 minutes for the next file synchronization  operation to complete) ...

smw1# ssh smw2
...
smw2# ls -l /opt/cray/hss/default/etc/my_test_file
smw2# md5sum /opt/cray/hss/default/etc/my_test_file
```

b. Return to the active SMW to delete the test file. Within several minutes, the file will be automatically removed from the passive SMW.

c. If file synchronization does not appear to be working, see *Monitor the fsync Resource* on page 25.

4.4 Display SMW Power Status

About this task

If you are not near the SMWs to check the LEDs, you can use one of the following methods to display the power status for the SMWs:

Procedure

1. As `root` on either SMW, use the `crm_mon` command to check the HA cluster status.

```
smw1# crm status
Stack: unknown
Current DC: smw2 (version unknown) - partition with quorum
Last updated: Tue Aug 29 16:25:47 2017
Last change: Tue Aug 29 09:03:49 2017 by root via crm_attribute on smw2

2 nodes configured
35 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
.
.
.
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started smw1
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
 ip-drbd-pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
 Master/Slave Set: ms-drbd-pgsql [drbd-pgsql]
     Masters: [ smw1 ]
     Slaves: [ smw2 ]
```

2. As `root` on either SMW, use the `ipmitool` command to check the power status of a specific SMW. Replace *`smw-iDRAC-IP-addr`* with the SMW's iDRAC IP address.

```
smw1# /usr/bin/ipmitool -I lanplus -U root -H smw-iDRAC-IP-addr -a chassis power status
Password:
Chassis Power is on
```

At the password prompt, enter the `root` password for the iDRAC.

4.5 Monitor the fsync Resource

About this task

For files not located on shared storage (boot RAID), the SLEHA Extension software includes the `csync2` utility to synchronize (sync) important files between the two SMWs. When a file changes on the active SMW, it is automatically synchronized to the passive SMW. The `csync2` utility synchronizes the required files and directories for the SMW HA cluster, such as `/etc/passwd` and `/opt/cray/hss/*/etc/*`.

The `fsync` resource controls file synchronization operations. Every 100 seconds, `fsync` checks for files that need to be synchronized. Use the following procedure to ensure that the `fsync` resource is running and that file synchronization is occurring.

Procedure

1. Check the status of the `fsync` resource by executing the following command as `root` on either SMW:

```
smw1# crm_mon -r1 | grep fsync
 fsync  (ocf::smw:fsync):       Started smw1
```

2. If `fsync` is stopped, display the failcount data for this resource. The status `Stopped` is usually caused by exceeding the failcount for a resource.

```
smw1# show_failcounts | grep fsync
node=smw1 scope=status  name=fail-count-fsync value=13
node=smw2 scope=status  name=fail-count-fsync value=0
```

3. If necessary, clear the failcount data for the `fsync` resource.

```
smw1# clear_failcounts
Clearing failcount on node smw1
Clearing failcount on node=smw1 for resource=ClusterIP
Clearing failcount on node=smw1 for resource=ClusterIP1
Clearing failcount on node=smw1 for resource=ClusterIP2
Clearing failcount on node=smw1 for resource=ClusterIP3
Clearing failcount on node=smw1 for resource=ClusterIP4
Clearing failcount on node=smw1 for resource=ClusterMonitor
Clearing failcount on node=smw1 for resource=ClusterTimeSync
Clearing failcount on node=smw1 for resource=HSSDaemonMonitor
Clearing failcount on node=smw1 for resource=Notification
 .
 .
 .
Clearing failcount on node=smw2 for resource=cray-ansible
Clearing failcount on node=smw2 for resource=IMPSFilesystemConfig
Clearing failcount on node=smw2 for resource=ml-fs
Clearing failcount on node=smw2 for resource=rsyslog
```

```
Clearing failcount on node=smw2 for resource=cray-syslog
Clearing failcount on node=smw2 for resource=LOGFilesystemConfig
Clearing failcount on node=smw2 for resource=ip_drbd_pgsql
Clearing failcount on node=smw2 for resource=drbd_pgsql:0
Clearing failcount on node=smw2 for resource=drbd_pgsql:1
```

After all failcounts have been cleared, the resource should be up and running.

4. Check the cluster status again to verify that the resource has been restarted.

```
smw1# crm resource status
Stack: unknown
Current DC: smw2 (version unknown) - partition with quorum
Last updated: Tue Aug 29 16:25:47 2017
Last change: Tue Aug 29 09:03:49 2017 by root via crm_attribute on smw2

2 nodes configured
35 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
 HSSDaemonMonitor       (ocf::smw:HSSDaemonMonitor):    Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 ResourceInit   (ocf::smw:ResourceInit):        Started smw1
 cray-cfgset-cache      (systemd:cray-cfgset-cache):    Started smw1
 dhcpd  (systemd:dhcpd.service):        Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 Resource Group: HSSGroup
     postgresqld        (systemd:postgresql):   Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
 Resource Group: IMPSGroup
     cray-ids-service   (systemd:cray-ids-service):     Started smw1
     cray-ansible       (systemd:cray-ansible): Started smw1
     IMPSFilesystemConfig       (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started smw1
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
```

```
 ip-drbd-pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
 Master/Slave Set: ms-drbd-pgsql [drbd-pgsql]
     Masters: [ smw1 ]
     Slaves: [ smw2 ]
```

5. If not all resources have started, execute the `clean_resources` command.

```
smw1# clean_resources
Cleaning resources on node smw1
Cleaning resource on node=smw1 for resource=ClusterIP
Cleaning up ClusterIP on smw1
Cleaning up ClusterIP on smw2
Waiting for 2 replies from the CRMd.. OK
Cleaning resource on node=smw1 for resource=ClusterIP1
Cleaning up ClusterIP1 on smw1
Cleaning up ClusterIP1 on smw2
Waiting for 2 replies from the CRMd.. OK
 .
 .
 .
Cleaning resource on node=smw1 for resource=syslog.socket
Cleaned up NFSServer on smw1
Cleaned up NFSServer on smw2
Cleaned up EnableRsyslog on smw1
Cleaned up EnableRsyslog on smw2
Cleaned up syslog.socket on smw1
Cleaned up syslog.socket on smw2
Multiple attributes match name=target-role
  Value: started        (id=SystemGroup-meta_attributes-target-role)
  Value: Started        (id=NFSServer-meta_attributes-target-role)
  Value: Started        (id=EnableRsyslog-meta_attributes-target-role)
  Value: Started        (id=syslog.socket-meta_attributes-target-role)
Waiting for 6 replies from the CRMd..... OK
Cleaning resource on node=ethel for resource=PostgreSQL:0
Cleaned up PostgreSQL:0 on smw1
Cleaned up PostgreSQL:0 on smw2
Waiting for 2 replies from the CRMd.. OK
Cleaning resource on node=smw1 for resource=PostgreSQL:1
Cleaned up PostgreSQL:0 on smw1
Cleaned up PostgreSQL:0 on smw2
Waiting for 2 replies from the CRMd.. OK
```

6. After running `clean_resources`, wait several minutes for cluster activity to settle. Ensure that all resources have started by checking the cluster status with `crm_mon -r1`.

4.6 Boot an SMW HA Cluster

About this task

IMPORTANT: When SMW HA is enabled, do not start both SMWs at the same time. Doing so can cause a race condition that could result in one SMW being powered off via the STONITH capability. Before starting the second SMW, wait until the first SMW has completed startup and initialized all cluster resources.

Follow these steps to boot or reboot both SMWs.

Procedure

1. Boot smw1 (or the SMW that should be active).

2. Wait until smw1 has rejoined the cluster. After the SMW responds to a ping command, log into smw1, sleep for at least 2 minutes, then execute the crm_mon -r1 command to verify that smw1 is online.

```
smw1# crm status
Stack: unknown
Current DC: smw2 (version unknown) - partition with quorum
Last updated: Tue Aug 29 16:25:47 2017
Last change: Tue Aug 29 09:03:49 2017 by root via crm_attribute on smw2

2 nodes configured
35 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
.
.
.
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started smw1
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
 ip-drbd-pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
 Master/Slave Set: ms-drbd-pgsql [drbd-pgsql]
     Masters: [ smw1 ]
     Slaves: [ smw2 ]
```

   Note that crm status may display different resource names, group names, or resource order on the system.

3. Boot smw2 (or the SMW that will be passive).

4. Wait until smw2 has rejoined the cluster. After the SMW responds to a ping command, log into smw2, sleep for at least 2 minutes, then execute the crm_mon -r1 command to verify that smw2 is online.

5. Examine the `crm_mon -r1` output to verify that each resource has started. Look for `Started smw1` or `Started smw2`. Also look for any failed actions at the end of the output.

```
smw1# crm status
Stack: unknown
Current DC: smw2 (version unknown) - partition with quorum
Last updated: Tue Aug 29 16:25:47 2017
Last change: Tue Aug 29 09:03:49 2017 by root via crm_attribute on smw2

2 nodes configured
35 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
.
.
.
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started smw1
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
 ip-drbd-pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
 Master/Slave Set: ms-drbd-pgsql [drbd-pgsql]
     Masters: [ smw1 ]
     Slaves: [ smw2 ]
```

6. If not all resources have started or if any failed actions are displayed, execute the `clean_resources` command on either SMW.

> **IMPORTANT:** Before running the `clean_resources` command, log in directly as `root` (instead of using `su` from a `crayadm` login), because `clean_resources` terminates all non-root user sessions.

```
smw1# clean_resources
Cleaning resources on node smw1
Cleaning resource on node=smw1 for resource=ClusterIP
Cleaning up ClusterIP on smw1
Cleaning up ClusterIP on smw2
Waiting for 2 replies from the CRMd.. OK
Cleaning resource on node=smw1 for resource=ClusterIP1
Cleaning up ClusterIP1 on smw1
Cleaning up ClusterIP1 on smw2
```

```
Waiting for 2 replies from the CRMd.. OK
 .
 .
 .
Cleaning resource on node=smw1 for resource=syslog.socket
Cleaned up NFSServer on smw1
Cleaned up NFSServer on smw2
Cleaned up EnableRsyslog on smw1
Cleaned up EnableRsyslog on smw2
Cleaned up syslog.socket on smw1
Cleaned up syslog.socket on smw2
Multiple attributes match name=target-role
  Value: started        (id=SystemGroup-meta_attributes-target-role)
  Value: Started        (id=NFSServer-meta_attributes-target-role)
  Value: Started        (id=EnableRsyslog-meta_attributes-target-role)
  Value: Started        (id=syslog.socket-meta_attributes-target-role)
Waiting for 6 replies from the CRMd..... OK
Cleaning resource on node=ethel for resource=PostgreSQL:0
Cleaned up PostgreSQL:0 on smw1
Cleaned up PostgreSQL:0 on smw2
Waiting for 2 replies from the CRMd.. OK
Cleaning resource on node=smw1 for resource=PostgreSQL:1
Cleaned up PostgreSQL:0 on smw1
Cleaned up PostgreSQL:0 on smw2
Waiting for 2 replies from the CRMd.. OK
```

7. After running `clean_resources`, wait several minutes for cluster activity to settle (check cluster status again with the `crm_mon -r1` command). If the output of this command shows only a subset of the SMW HA services, wait for another minute, then check again. For more information, see the `clean_resources(8)` man page.

4.7 Reboot the Passive SMW in Maintenance Mode

About this task

When an HA system is running in production mode and it is necessary to reboot the passive SMW, the cluster must first be placed in maintenance mode. When the reboot is completed, disable maintenance mode.

Procedure

1. Log in to the active SMW as `root`, using the virtual host name for the HA cluster. This procedure assumes that *smw1* is active and *smw2* is passive.

```
remote-system% ssh root@virtual-smw
smw1#
```

2. Enable maintenance mode for the SMW HA cluster.

```
smw1# maintenance_mode_configure enable
```

3. Verify the status of maintenance mode.

```
smw1# maintenance_mode_configure status
Maintenance mode is enabled
```

4. Log in to the passive SMW as `root`.

```
remote-system% ssh root@smw2
smw2#
```

5. Reboot the passive SMW.

```
smw2# reboot
```

   Wait several minutes for the passive SMW to completely reboot.

6. Disable maintenance mode.

```
smw1# maintenance_mode_configure disable
```

4.8 Change SMW, iDRAC, and STONITH Passwords

About this task

This procedure changes the SMW root password and the hacluster and stonith passwords. To change the iDRAC password, use the procedure in *Change the Default iDRAC Password* on page 32.

The passwords for an SMW HA system must follow these rules:

- The SMW root password must be the same on each SMW.
- The Integrated Dell™ Remote Access Controller (iDRAC) root password must be the same on each iDRAC.
- The iDRAC root password can be different than the SMW root password.
- The hacluster password on each SMW must be the same as the SMW root password.
- The HA stonith resource passwords must be the same as the iDRAC root password.

Procedure

1. Log into the active SMW (for example, `smw1`) as `root`, using the virtual SMW host name (such as `virtual-smw`). After login, the prompt displays the host name of the active SMW.

2. Change the SMW root and hacluster passwords on the active SMW (`smw1`).

   The hacluster password must be the same as the SMW root password.

```
smw1# passwd root
smw1# passwd hacluster
```

3. Change the stonith-1 and stonith-2 passwords on the active SMW (`smw1`).

   The stonith resource passwords must be the same as the iDRAC root password.

```
smw1# crm resource param stonith-1 set passwd new-passwd
smw1# crm resource param stonith-2 set passwd new-passwd
```

4. Change the SMW root and hacluster passwords on the passive SMW (`smw2`), using the same root password as on `smw1`.

   The hacluster password must be the same as the SMW root password.

   ```
   smw2# passwd root
   smw2# passwd hacluster
   ```

4.8.1 Change the Default iDRAC Password

About this task

This procedure describes how to log in to the iDRAC web interface and change a user password.

Procedure

1. Bring up a web browser.
2. Go to: `https://cray-drac`, where *`cray-drac`* is the name of the iDRAC.

   A login screen appears.
3. Log in to the web interface as `root`.
4. Select **iDRAC settings** on the left navigation bar.
5. Expand **iDRAC settings** on the left navigation bar.
6. Select **User Authentication**.
7. Select the user whose password is changing. To change the root password, select userid 2.
8. Select **Next**.
9. Select the **Change Password** box and enter the new password in the boxes below it.
10. Select **Apply** to complete the password change.

The password change is complete.

Alternative. Another approach to changing the iDRAC root password is to use `ipmitool` on the SMW command line interface.

```
smw# ipmitool -U root -I lanplus -H drac-ip-addr -P old-drac-password \
user set password 2 new-drac-password
```

4.9 Update the NIMS Map on the Passive SMW

About this task

The Node Image Mapping Service (NIMS) maps a node to boot attributes, which are used when the node is booted. NIMS maps are stored in the `/var/opt/cray/imps` directory, which only the active SMW can access. Links to the NIMS maps are in the `/etc/opt/cray/nims/active_maps` directory on each SMW, but only the active SMW's links are automatically updated after a map change. The administrator must manually synchronize the NIMS links to the passive SMW. Otherwise, the other SMW could have trouble booting the CLE nodes after failover because it would be linked to a different NIMS map.

Use this procedure after making changes to the NIMS map, such as adding new hardware, making manual updates to the NIMS map, or specifying a different active NIMS map.

Procedure

1. Log in to the active SMW as `root`.
2. Execute the `ha_sync_nims_map` command.

   ```
   smw1# ha_sync_nims_map
   ```

3. Log into the passive SMW as `root` in a separate terminal session.
4. Verify that both SMWs are linked to the same NIMS map.

   On smw1:

   ```
   smw1# ls -l /etc/opt/cray/nims/active_maps
   total 4
   lrwxrwxrwx 1 crayadm crayadm 86 Jun  9 10:46 p0 -> /var/opt/cray/imps/config/
   sets/global/nims/maps/cle6.0.96-PS04-08.created20160609-l2.7
   ```

   On smw2:

   ```
   smw2# ls -l /etc/opt/cray/nims/active_maps
   total 4
   lrwxrwxrwx 1 crayadm crayadm 86 Jun 10 19:31 p0 -> /var/opt/cray/imps/config/
   sets/global/nims/maps/cle6.0.96-PS04-08.created20160609-l2.7
   ```

4.10 Apply Firewall Rule Changes to the Passive SMW

About this task

Changes to firewall rules are not automatically synchronized to the passive SMW. The firewall rules are located in the `/var/opt/cray/imps/` directory, which is mounted only on the active SMW in an SMW HA system.

After making firewall rule changes on an SMW HA system, use this procedure to manually apply the changed firewall rules to the passive SMW.

Procedure

1. Log in to the active SMW as `root`, using the virtual SMW host name (for example, *`virtual-smw`*). This procedure assumes that *`smw1`* is active and *`smw2`* is passive.

   ```
   remote-system% ssh root@virtual-smw
   smw1#
   ```

2. Check that both SMWs are online.

   ```
   smw1# crm status | grep Online
   Online: [ smw1 smw2 ]
   ```

3. Log in to the passive SMW as `root`.

   ```
   remote-system% ssh root@smw2
   smw2#
   ```

4. Apply the firewall rules on the passive SMW.

   ```
   smw2# ansible-playbook -v /etc/ansible/firewall_init.yaml
   ```

4.11 Revert to a Previous SMW HA Snapshot

Prerequisites

Before beginning this procedure, ensure that the CLE system is shut down.

About this task

This procedure reverts an SMW HA system to a previous snapshot, if necessary. Note that this procedure restores only the SMW snapshot; it does not affect the CLE node snapshots.

CAUTION: When reverting to a previous snapshot, use only an HA snapshot — that is, a snapshot created after the SMW HA software was installed and configured. It is dangerous to boot a non-HA snapshot on an HA system because there is a risk of double-mounting the shared file systems, which could cause file system corruption.

Procedure

1. Log in to both SMWs as `root`. This procedure assumes that *`smw1`* is the active SMW, and *`smw2`* is the passive SMW. The previous snapshot is shown as *`HA-snapshot-name-to-restore`*.

2. Set the default snapshot on both SMWs to the name of the HA snapshot to be restored. Do not use a non-HA snapshot.

   On *`smw1`*:

   ```
   smw1# snaputil default HA-snapshot-name-to-restore
   ```

On *smw2*:

```
smw2# snaputil default HA-snapshot-name-to-restore
```

3. Shut down the passive SMW.

```
smw2# shutdown -h now
```

4. Reboot the active SMW.

```
smw1# reboot
```

Wait several minutes until the active SMW completely reboots and all cluster services have started.

5. Power on the passive SMW.

After the passive SMW has finished rebooting, the SMW HA cluster has been reverted to *HA-snapshot-name-to-restore*.

6. Verify that both SMWs have booted the correct snapshot.

On *smw1*:

```
smw1# snaputil list | grep 'cur,def'cur,def
HA-snapshot-name-to-restore                2016-05-18         07:11:28
```

On *smw2*:

```
smw2# snaputil list | grep 'cur,def'cur,def
HA-snapshot-name-to-restore                2016-05-18         07:11:28
```

7. Migrate the PMDB volume to the current DRBD version.

If the system has been rebooted from an SMW snapshot with version 8.0.UP04 or later to a snapshot with version 8.0.UP03 or earlier, a DRBD version 9.x to version 8.4 migration is required. To do this, repeat the section 3.8.10 procedure (steps a through k). This procedure is also needed following a reboot from a snapshot with version 8.0.UP03 and earlier to a snapshot with version 8.0.UP04 or later.

4.12 Enable Multipath on an Installed SMW HA System

Prerequisites

This procedure assumes that the Cray XC system has already been installed and configured as an SMW HA system without multipath having been enabled. If performing a fresh install, this procedure is not necessary: use the procedures in XC™ Series Software Installation and Configuration Guide instead.

About this task

This procedure describes how to enable multipath on a Cray XC system that has already been installed and configured as an SMW HA system. Note that multipath does NOT need to be fully cabled to be used. The multipath driver can handle using one path or many.

> **IMPORTANT:** If this system has partitions, repeat any steps that modify 'p0' for each partition. Multipath must be enabled everywhere or nowhere; enabling it on only part of the system causes problems.

Procedure

1. Start the multipath daemon now on each SMW in the HA cluster (active and passive).

```
smw1# systemctl start multipathd

smw2# systemctl start multipathd
```

   Later in this procedure, the `cray-ansible` command will be used to enable the multipath daemon on the active SMW, and `systemctl enable` will be used to enable it on the passive SMW.

2. Obtain the host ID of each SMW in the HA cluster and the cnames of any nodes in the system that are connected to the boot RAID with an HBA (host bus adapter).

   The system should be bounced or booted for `xtcheckhss` to return a proper list. Run `hostid` for each SMW in the SMW HA system.

```
smw1# hostid
{8 digit hostid}
smw1# xtcheckhss --detail=f --pci

smw2# hostid
{8 digit hostid}
smw2# xtcheckhss --detail=f --pci
```

   Look for cnames with HBAs like 'QLogic_ISP2532_8Gb_Fibre_Channel_HBA.'

3. Use the configurator to enable and customize multipath in the global config set on the active SMW.

```
smw1# cfgset update -s cray_multipath -m interactive global
```

   a. Enable multipath.

      Enter **E** at the configurator prompt to toggle the enable status of the multipath service, which is disabled by default.

```
Cray Multipath Configuration Service Menu [default: save & exit - Q] $ E
```

   b. Add the host IDs and cnames obtained earlier in this procedure.

      At the prompt, enter **1** to select the node_list setting, then enter **C** to configure it. At the prompt for that setting, enter values **+** to add node_list entries: add the host IDs and cnames obtained in step two, one per line. When finished, press **Ctrl-d** and then **<cr>** to set the entries.

      Remember to add the host ID of both SMWs.

```
Cray Multipath Configuration Service Menu [default: save & exit - Q] $ 1
...
Cray Multipath Configuration Service Menu [default: configure - C] $ C
...
cray_multipath.settings.multipath.data.node_list
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add node_list (Ctrl-d to exit) $
```

4. Use the configurator to update cray_bootraid in the global config set on the active SMW.

```
smw1# cfgset update -s cray_bootraid -m interactive global
```

a. Select the storage sets setting to configure it.

```
Boot RAID Configuration Service Menu [default: save & exit - Q] $ 1
...
Boot RAID Configuration Service Menu [default: configure - C] $ C
```

b. For each device in the cledefault and smwdefault storage sets, modify the path name from scsi to dm-uuid-mpath.

This example shows selecting the cledefault (1) volume group (a) boot_node_vg (1) devices (b) field. The * indicates that the selection is to be edited.

```
cray_bootraid.settings.storage_sets
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ 1a1b*
```

Remove the "scsi path name, and replace it with the dm-uuid-mpath name.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg
.devices
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ 1-

cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg
.devices
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add devices (Ctrl-d to exit) $ /dev/disk/by-id/dm-uuid-
mpath-3600a0980009ec0750000010a5762af70
Add devices (Ctrl-d to exit) $ <Ctrl-d>
```

Set the entries for the boot_node_vg volume group.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg
.devices
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <cr>
```

Repeat substep b for each device in the cledefault and smwdefault storage sets. Enter * at the prompt to see all storage set entries.

- To select the next cledefault volume group device (sdb_node_vg), enter **1a2b*** at the prompt. If there are more cledefault volume groups, increment the third character to select each one (**1a3b***, **1a4b***, and so forth).

- To select the first smwdefault volume group device (smw_node_vg), enter **2a1b*** at the prompt. If there are more smwdefault volume groups, increment the third character to select each one (**2a2b***, **2a3b***, and so forth).

c. Set the storage set entries, then save and exit the configurator.

```
cray_bootraid.settings.storage_sets
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <cr>
...
Boot RAID Configuration Service Menu [default: save & exit - Q] $ Q
```

5. Use the configurator to set up inheritance for multipath in the CLE config set of the active SMW.

This example uses 'p0' as the name of the CLE config set. Substitute the actual name used for this system.

```
smw1# cfgset update -s cray_multipath -m interactive p0
```

Enter **I** at the configurator prompt to toggle the inherit status of the multipath service, which is disabled by default. This means that multipath settings in the global config set will be used instead of multipath settings in the CLE config set.

```
Cray Multipath Configuration Service Menu [default: save & exit - Q] $ I
```

Repeat this step for each CLE config set.

6. Validate the config sets and run cray-ansible on the active SMW to apply the config set changes.

a. Validate the config sets.

```
smw1# cfgset validate global

smw1# cfgset validate p0
```

b. Run cray-ansible.

```
smw1# /etc/init.d/cray-ansible start
```

7. Update the `dal.fs_defs` file for systems using direct-attached Lustre (DAL).

Repeat for each partition.

a. Locate the current `fs_defs` files (typically stored in `/home/crayadm`).

```
smw1# find /home/crayadm -name "*fs_defs*"
```

b. Find the `fs_defs` files that are currently installed and compare with the one found in `/home/crayadm`.

```
smw1# find /var/opt/cray/imps/config/sets/p0 -name "*fs_defs*"
```

```
smw1# diff /home/crayadm/dal.fs_defs /var/opt/cray/\
imps/config/sets/p0/lustre/.lctrl/dal.fs_defs.20160205.1454685527
```

c. Edit the `dal.fs_defs` file to ensure that it has the proper mpath paths in it.

```
smw1# cd /home/crayadm

smw1# sed -i.nompath 's/\/dev\/disk\/by-id\/scsi/\/dev\/disk\/by-id\/dm-uuid-
mpath/g' \
dal.fs_defs

smw1# cp -p dal.fs_defs dal.fs_defs.mpath
```

d. Install the new `dal.fs_defs` file using `lustre_control`.

```
smw1# lustre_control install -c p0 /home/crayadm/dal.fs_defs
```

8. Shut down all partitions of the Cray system (service and compute nodes).

9. Check whether `/etc/lvm/lvm.conf` and `/etc/multipath.conf` were synced to the passive SMW.

They should be synced automatically, but if they are not the same, `scp` both files from the active SMW to the passive SMW.

On the active SMW:

```
smw1# stat /etc/lvm/lvm.conf
smw1# stat /etc/multipath.conf
```

On the passive SMW (smw2 in the example):

```
smw2# stat /etc/lvm/lvm.conf
smw2# stat /etc/multipath.conf
```

10. Enable multipath on the passive SMW.

```
smw2# systemctl enable multipathd
```

11. Put the SMW HA system into maintenance mode.

```
smw1# maintenance_mode_configure enable
```

12. Reboot both SMWs at the same time.

```
smw1# reboot

smw2# reboot
```

13. Disable maintenance mode and check cluster status after both SMWs have completed booting.

```
smw1# maintenance_mode_configure disable
smw1# sleep 300
smw1# crm status
```

14. Boot the Cray system.

5 Handle Failover

When failover occurs on an SMW HA system, the essential system services (file system mounts and daemons, also called cluster resources) fail over to the passive SMW node, which becomes the active node. For a recoverable problem, the failed node automatically rejoins the cluster as the passive node. For most issues, however, the administrator must diagnose and fix the problem, then manually return the failed SMW to the cluster.

Use the following procedures when failover occurs:

- *Critical Events That Cause SMW HA Failover* on page 41
- *Examine the SMW HA Log File to Determine SMW Failover Cause* on page 42
- *Restore Normal Operations After SMW Failover* on page 43
- *Perform a Manual Failover* on page 44
- *Limitations of SMW Failover* on page 13

5.1 Critical Events That Cause SMW HA Failover

The following critical events cause a failover from the active SMW to the passive SMW:

- Hardware fault on the active SMW.
- Lost heartbeat between the two SMWs.
- Kernel fault (panic) on the active SMW.
- Failed resource (HSS daemon or cluster service). If a resource stops, the cluster manager automatically restarts it and increments the failcount by 1. When the failcount exceeds the migration threshold (by default, `1,000,000`), a failover occurs.

The failover type (STONITH or non-STONITH) depends upon whether the newly active SMW can determine the health of the failing SMW. A STONITH failover occurs only if there is no other way for the new SMW to ensure the integrity of the cluster.

- In the case of STONITH failover, the original SMW is powered off (via the STONITH capability) if it is not already off. This guarantees that file synchronization is stopped and the failed SMW no longer holds any cluster-managed resources so that the new SMW will have exclusive access to those resources.
- In the case of non-STONITH failover, the original SMW is still powered up. In addition:
  - HSS daemons are stopped on the original SMW.
  - Lightweight Log Manager (LLM) logging to shared disk is stopped.
  - File synchronization (`csync2`) between SMWs is stopped.
  - The shared file systems on the boot RAID are unmounted on the original SMW.
  - Network connections using the `eth0`, `eth1`, `eth2`, `eth3`, and `eth4` virtual IP addresses are dropped and those interfaces begin accepting connections to their actual IP addresses only.

For both types of failover, the following actions then occur on the new SMW:

- The `eth0`, `eth1`, `eth2`, `eth3`, `eth4`, and `eth5` (optional) interfaces begin accepting connections using the virtual IP addresses in addition to their actual IP addresses.
- The shared file systems on the boot RAID are mounted on the new SMW.
- File synchronization (`csync2`) between SMWs usually resumes (depending on the reason for failover).
- LLM logging to the shared disk resumes.
- The HSS database (MySQL) is started on the new SMW.
- HSS daemons are started on the new SMW (including, if necessary, any `xtbootsys`-initiated daemons).
- Failcounts and failed actions are written to the log file `/var/log/smwha.log` on the newly active SMW.

  IMPORTANT: When the SMW HA system is running, do not remove power to the active SMW and its iDRAC to force a failover. Doing so will put the HA cluster in a frozen state where no resources are online. The HA cluster cannot recover automatically from this state.

5.2 Examine the SMW HA Log File to Determine SMW Failover Cause

The log file `/var/log/smwha.log` contains cluster status and resource failure data that can help determine the cause of a failover. At system startup (such as after a failover), the `ClusterMonitor` resource records failcounts and failed actions in the log file, then clears this failure information from `crm` (for example, in the output of `crm_mon -1`).

The log file `/var/log/smwha.log` is not shared. Entries are recorded only on the active SMW.

This example shows the format of entries in the log file.

```
************************************************************
smw1 acted as active SMW at Fri May 27 15:52:15 CDT 2016
************************************************************
node=smw1 scope=status  name=fail-count-ClusterIP value=0
node=smw1 scope=status  name=fail-count-ClusterIP1 value=0
node=smw1 scope=status  name=fail-count-ClusterIP2 value=0
node=smw1 scope=status  name=fail-count-ClusterIP3 value=0
node=smw1 scope=status  name=fail-count-ClusterIP4 value=0
node=smw1 scope=status  name=fail-count-ClusterMonitor value=0
node=smw1 scope=status  name=fail-count-ClusterTimeSync value=0
node=smw1 scope=status  name=fail-count-HSSDaemonMonitor value=0
node=smw1 scope=status  name=fail-count-Notification value=0
node=smw1 scope=status  name=fail-count-cray-cfgset-cache value=0
node=smw1 scope=status  name=fail-count-dhcpd value=0
node=smw1 scope=status  name=fail-count-fsync value=0
node=smw1 scope=status  name=fail-count-hss-daemons value=0
node=smw1 scope=status  name=fail-count-repos-fs value=0
node=smw1 scope=status  name=fail-count-stonith-1 value=0
node=smw1 scope=status  name=fail-count-stonith-2 value=0
node=smw1 scope=status  name=fail-count-ResourceInit value=0
node=smw1 scope=status  name=fail-count-homedir value=0
node=smw1 scope=status  name=fail-count-md-fs value=0
node=smw1 scope=status  name=fail-count-pm-fs value=0
node=smw1 scope=status  name=fail-count-postgresqld value=0
node=smw1 scope=status  name=fail-count-mysqld value=0
node=smw1 scope=status  name=fail-count-imps-fs value=0
node=smw1 scope=status  name=fail-count-cray-ids-service value=0
```

```
node=smw1 scope=status  name=fail-count-cray-ansible value=0
node=smw1 scope=status  name=fail-count-IMPSFilesystemConfig value=0
node=smw1 scope=status  name=fail-count-ml-fs value=0
node=smw1 scope=status  name=fail-count-rsyslog value=0
node=smw1 scope=status  name=fail-count-cray-syslog value=0
node=smw1 scope=status  name=fail-count-LOGFilesystemConfig value=0
node=smw1 scope=status  name=fail-count-ip_drbd_pgsql value=0
node=smw1 scope=status  name=fail-count-drbd_pgsql:0 value=0
node=smw1 scope=status  name=fail-count-drbd_pgsql:1 value=0
node=smw2 scope=status  name=fail-count-ClusterIP value=0
node=smw2 scope=status  name=fail-count-ClusterIP1 value=0
node=smw2 scope=status  name=fail-count-ClusterIP2 value=0
node=smw2 scope=status  name=fail-count-ClusterIP3 value=0
node=smw2 scope=status  name=fail-count-ClusterIP4 value=0
node=smw2 scope=status  name=fail-count-ClusterMonitor value=0
node=smw2 scope=status  name=fail-count-ClusterTimeSync value=0
node=smw2 scope=status  name=fail-count-HSSDaemonMonitor value=0
node=smw2 scope=status  name=fail-count-Notification value=0
node=smw2 scope=status  name=fail-count-cray-cfgset-cache value=0
node=smw2 scope=status  name=fail-count-dhcpd value=0
node=smw2 scope=status  name=fail-count-fsync value=0
node=smw2 scope=status  name=fail-count-hss-daemons value=0
node=smw2 scope=status  name=fail-count-repos-fs value=0
node=smw2 scope=status  name=fail-count-stonith-1 value=0
node=smw2 scope=status  name=fail-count-stonith-2 value=0
node=smw2 scope=status  name=fail-count-ResourceInit value=0
node=smw2 scope=status  name=fail-count-homedir value=0
node=smw2 scope=status  name=fail-count-md-fs value=0
node=smw2 scope=status  name=fail-count-pm-fs value=0
node=smw2 scope=status  name=fail-count-postgresqld value=0
node=smw2 scope=status  name=fail-count-mysqld value=0
node=smw2 scope=status  name=fail-count-imps-fs value=0
node=smw2 scope=status  name=fail-count-cray-ids-service value=0
node=smw2 scope=status  name=fail-count-cray-ansible value=0
node=smw2 scope=status  name=fail-count-IMPSFilesystemConfig value=0
node=smw2 scope=status  name=fail-count-ml-fs value=0
node=smw2 scope=status  name=fail-count-rsyslog value=0
node=smw2 scope=status  name=fail-count-cray-syslog value=0
node=smw2 scope=status  name=fail-count-LOGFilesystemConfig value=0
node=smw2 scope=status  name=fail-count-ip_drbd_pgsql value=0
node=smw2 scope=status  name=fail-count-drbd_pgsql:0 value=0
node=smw2 scope=status  name=fail-count-drbd_pgsql:1 value=0
```

5.3 Restore Normal Operations After SMW Failover

About this task

While a failover is automatic, adding the failed SMW back into the cluster requires manual intervention to identify the reason for failover, take corrective action if needed, and return the failed SMW to an online state. Another failover (that is, a "failback" to the originally active SMW) is not possible until the failed SMW returns to online status and its failcounts are cleared so that it is eligible to run all cluster resources.

Procedure

1. Identify and fix the problems that caused the failover (such as a hardware fault, kernel panic, or HSS daemon issues). Use the following methods to help diagnose problems:

   a. Examine the log file `/var/log/smwha.log` on the new active SMW. For more information, see *Examine the SMW HA Log File to Determine SMW Failover Cause* on page 42.

   b. Execute the `show_failcounts` command and note any resources with non-zero failcounts.

   c. From the active SMW, examine `/var/opt/cray/log/smwmessages-`*yyyymmdd* for relevant messages.

   d. Examine the failing SMW for additional clues.

   e. For a non-STONITH failover: In most cases, the failing SMW will still be running; additional clues may be available in `dmesg` or via other commands.

   f. For a STONITH failover: The failing SMW will be powered off. Before powering it back on, place the SMW into standby mode so that it does not automatically try to rejoin the cluster at startup before ensuring that the node is healthy. For more information, see *Restart Stopped Resources* on page 67.

2. Log on to the failing SMW (either from the console or remotely by using the actual host name). Identify the reason for the failure and take corrective action as needed. This might include administrative actions such as freeing space on a file system that has filled up or hardware actions such as replacing a failing component.

3. After the SMW is ready to rejoin the cluster, run the `clean_resources` command as described in *Restart Stopped Resources*. This command also resets all failcounts to zero.

   After running `clean_resources`, wait several minutes for cluster activity to settle. You can check cluster status with the `crm_mon -r1` command.

4. Return the SMW to online status as the passive SMW.

   Replace *smw2* with the host name of the failed SMW.

   ```
   smw1# crm node online smw2
   ```

5. If the boot node mounts any SMW directories, and passwordless access between the boot node and SMW is not configured, the mount point on the boot node to the SMW is stale. To refresh the mount point:

   a. Log into the boot node.

   b. Unmount then remount the SMW directories.

   c. Restart `bnd`.

   ```
   boot# /etc/init.d/bnd restart
   ```

5.4 Perform a Manual Failover

About this task

IMPORTANT: When the SMW HA system is running, do not remove power to the active SMW and its iDRAC to force a failover. Doing so will put the HA cluster in a frozen state where no resources are online. The HA cluster cannot recover automatically from this state. To restore normal cluster operation after this type of power loss, see *Recover From Loss of Power to the Active SMW and iDRAC* on page 76.

Procedure

1. As `root` on the active SMW, move the resources away from the active SMW. This command forces a failover, which stops all resources on the active SMW and moves them to the passive SMW.

   ```
   smw1# crm resource unmove ClusterIP
   smw1# crm resource move ClusterIP TargetSMW
   ```

   At this point, the target SMW (e.g., `smw2`) is now the active SMW.

2. Bring the previously active SMW (`smw1`) online as the passive SMW.

   ```
   smw1# crm node online smw1
   ```

3. Check the cluster status.

   ```
   smw1# crm status
   Stack: unknown
   Current DC: smw2 (version unknown) - partition with quorum
   Last updated: Tue Aug 29 16:25:47 2017
   Last change: Tue Aug 29 09:03:49 2017 by root via crm_attribute on smw2

   2 nodes configured
   35 resources configured

   Online: [ smw1 smw2 ]

   Full list of resources:

    ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
    ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
   .
   .
   .
    Resource Group: LogGroup
        cray-syslog        (systemd:llmrd.service):        Started smw1
        LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
    Resource Group: SharedFilesystemGroup
        homedir    (ocf::heartbeat:Filesystem):    Started smw1
        md-fs      (ocf::heartbeat:Filesystem):    Started smw1
        imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
        ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
   ```

```
    repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
    pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
Resource Group: SystemGroup
    NFSServer  (systemd:nfsserver):    Started smw1
    EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
    syslog.socket      (systemd:syslog.socket):        Started smw1
ip-drbd-pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
Master/Slave Set: ms-drbd-pgsql [drbd-pgsql]
    Masters: [ smw1 ]
    Slaves: [ smw2 ]
```

If `crm_mon` shows resource problems, see *Restore Normal Operations After SMW Failover* on page 43 and *Restart Stopped Resources* on page 67.

6 Change the SMW HA Cluster Configuration

The SMW HA system is configured during installation. You can customize the system by changing the failover notification address, resource migration threshold, and list of synchronized files.

When customizing the SMW HA system, follow these rules:

- Do not change the basic cluster configuration, except for the migration threshold (maximum failcount value). You can set the migration threshold for each resource by using the `set_migration_threshold` command.
- Do not attempt to migrate a single resource. All resources must migrate as a group.
- Do not change the system list of synchronized files. You can define which local (site-specific) files are synchronized or excluded from synchronization, but do not add large files or directories to the local list of synchronized files.

See *About SMW HA Commands* on page 16 for information on the SMW HA commands used for cluster configuration.

6.1 Change Failover Notification

Prerequisites

Failover notification requires email to be configured on both SMWs. For information about configuring email, see *http://www.postfix.org/BASIC_CONFIGURATION_README.html*.

About this task

The SMW HA software includes a `Notification` resource that automatically sends email when a failover occurs. Failover notification is usually configured during initial installation, but can be changed after the HA system is installed and running.

Procedure

1. Determine the email address for failover notification. Only one email address is allowed. To send notifications to multiple addresses, create a group email alias that includes all necessary email addresses.

2. Execute the following `crm resource` command as `root` on either SMW. Substitute the actual address for *`address@thedomain.com`*.

   ```
   smw1# crm resource param Notification set email address@thedomain.com
   ```

3. Verify the setting.

```
smw1# crm resource param Notification show email
address@thedomain.com
```

6.2 Add Site-specific Files to the Synchronization List

About this task

The file /etc/csync2/csync2_cray.cfg specifies the Cray-specific files and directories that must be synchronized, as well as small files that are convenient to keep in sync. For information about the default contents of the synchronization list, see *Synchronized Files* on page 11.

The csync2 utility is designed to synchronize small amounts of data. If csync2 must monitor many directories or synchronize a large amount of data, it can become overloaded and failures may not be readily apparent. For example, do not synchronize the following files or directories:

- /home
- /home/crayadm/.ssh/authorized_keys
- /etc/hosts
- Very large files

Procedure

1. Ensure that the file or directory is small enough for the synchronization list. Cray recommends adding only small files to /etc/csync2/csync2_cray.cfg. Use these other methods for large files:
   - Use scp to copy a large, static file to the passive SMW, as in this example:

     ```
     smw1# scp -pr /path/file smw2:/path/file
     ```
   - Use the rsync command for directories and files that may change during the copy operation.
2. Ensure that the parent directory exists on the passive SMW for each file or directory on the active SMW that requires synchronization. Some cases require either manually creating directories on the passive SMW or copying the directory structure from the active SMW. With either method, be sure that owner, group, and permissions are maintained, as csync2 can be sensitive to mismatches.
3. Edit the file /etc/csync2/csync2_cray.cfg as root on the active SMW.
4. Add the full path (one entry per line) to /etc/csync2/csync2_cray.cfg to add a file or directory. Comments in this file explain how to make changes.

   For a symbolic link, only the link itself is synchronized, not the content (destination) of the symbolic link.
5. Save changes and exit the editor.

   The fsync resource will synchronize the additional files and directories the next time it runs.
6. Manually copy /etc/hosts to /etc/hosts on smw2 if there are local changes to /etc/hosts on smw1. The customized entries must be above the first section of "XT Cabinet x - y".

```
smw2# cp /etc/hosts /etc/hosts.sav
smw2# scp smw1:/etc/hosts /etc/hosts
```

Then edit the `/etc/hosts` file on smw2:

a. Change IP addresses 10.1.1.*x*, 10.2.1.*x*, 10.3.1.*x*, and 10.4.1.*x* to 10.1.1.*y*, 10.2.1.*y*, 10.3.1.*y*, and 10.4.1.*y* where if *x* is 2 *y* is 3 and if *x* is 3 *y* is 2.

b. Change the line `smw1-ip smw1 smw1` to `smw2-ip smw2 smw2` .

6.3 Set the Migration Threshold for a Resource

About this task

The `set_migration_threshold` command sets the migration threshold for a resource in an SMW HA cluster. A migration threshold is defined as the maximum number of failures (the failcount) allowed for the resource. If the failcount exceeds this threshold, a failover occurs and management of all cluster resources migrates to the other SMW, making it the active SMW. By default, the migration threshold is `1,000,000`.

> **IMPORTANT:** Cray recommends that you either leave migration thresholds at the default values or set them to a very high value until you have experience with SMW HA operation. Migration threshold settings that are too low could cause the resource to be ineligible to run if the failcount exceeds that value on both SMWs. If lower settings are used, Cray recommends that you monitor failcounts regularly for trends and clear the failcount values as appropriate. Otherwise, transient errors over time could push failcount values beyond the migration threshold, which could lead to one of the following scenarios:
>
> - Failovers could be triggered by a transient error condition that might otherwise have been handled by a less disruptive mechanism.
> - Failovers might not be possible because both SMWs have exceeded the migration threshold.

Execute these commands as `root` on either SMW.

Procedure

1. Determine the resource name.

   To display a list of resource names and the status of those resources, use the `crm_resource` command.

   ```
   smw1# crm_resource -l
   ```

2. Use the `set_migration_threshold` command to change the migration threshold for a resource.

   For *`resource`*, specify a particular resource name. For *`value`*, specify an integer in the range of `0 - 1000000`.

   ```
   smw1# set_migration_threshold resource value
   ```

3. Verify the change.

```
smw1# show_migration_threshold resource
```

For more information, see the `set_migration_threshold(8)` man page.

7 Troubleshoot an SMW HA System

The following procedures describe how to troubleshoot issues on an SMW HA system.

- *Cray HA System Components* on page 51
- *Group Resources* on page 55
- *Cluster Resource Dependence* on page 57
- *Check Cluster Status* on page 58
- *Snapshot Failure* on page 62
- *Reasons for Failover* on page 63
- *Debug a Cluster Problem or Failover* on page 65
- *Debug Failure of a Preinstalled SMW HA System* on page 67
- *Restart Stopped Resources* on page 67
- *Return an SMW to the HA Cluster After It Has Been Powered Off* on page 70
- *Check Heartbeat Cables for Repeated STONITH Failures* on page 75
- *Clear an HSS Lock After Failover Occurs During a Cray System Boot* on page 75
- *Recover System Settings After Failover During Discovery* on page 76
- *Recover From Loss of Power to the Active SMW and iDRAC* on page 76
- *Stop Repeated "Ping Pong" Failovers* on page 78
- *PostgreSQL Does Not Start* on page 78
- *Reinitialize the PMDB* on page 81
- *Fix an Outdated DRBD Disk* on page 83

7.1 Cray HA System Components

This section describes all HA controlled resources with their names, functionalities, and the dependencies between them.

The Cray HA system configuration is complex, and issues will arise if customer installations have not followed the correct procedures in installing and configuring the hardware and software. The Cray HA system controls and monitors a variety of resources to ensure the active SMW is in a healthy state. If a resource fails and the number of failures reaches the designated failcount threshold, a failover will be triggered.

Cluster Configuration

Below is the output of `crm status`. For the sake of this example, smw1 is the active SMW and smw2 is the passive SMW.

```
smw1# crm status
Stack: unknown
Current DC: smw2 (version unknown) - partition with quorum
Last updated: Tue Aug 29 16:25:47 2017
Last change: Tue Aug 29 09:03:49 2017 by root via crm_attribute on smw2

2 nodes configured
35 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
 HSSDaemonMonitor       (ocf::smw:HSSDaemonMonitor):    Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 ResourceInit   (ocf::smw:ResourceInit):        Started smw1
 cray-cfgset-cache      (systemd:cray-cfgset-cache):    Started smw1
 dhcpd  (systemd:dhcpd.service):        Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 Resource Group: HSSGroup
     postgresqld        (systemd:postgresql):   Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
 Resource Group: IMPSGroup
     cray-ids-service   (systemd:cray-ids-service):     Started smw1
     cray-ansible       (systemd:cray-ansible): Started smw1
     IMPSFilesystemConfig       (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started smw1
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
 ip-drbd-pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
 Master/Slave Set: ms-drbd-pgsql [drbd-pgsql]
     Masters: [ smw1 ]
     Slaves: [ smw2 ]
```

The cluster is in a healthy state. This can be checked with `ha_health`.

```
smw1# ha_health
Cluster State
----------------------------------------
Health State                : Healthy
Active Node                 : smw1
Node-1                      : smw1 (online)
Node-2                      : smw2 (online)
Number of Resources         : 33
Number of Resources Running : 33
Number of Resources Stopped : 0
Maintenance Mode            : disabled
Stonith Mode                : enabled
```

Cluster IP Addresses

There are five IP addresses under cluster control.

Cluster IP

This resource exposes the cluster virtual IP to users and redirects the incoming traffic to the active SMW. If an error is detected or the eth0 interface is configured down, a failover can be triggered.

ClusterIP1

This resource provides the virtual IP address 10.1.1.1, which directs the traffic to eth1 of the active SMW. The IP address of eth1 on SMW1 is 10.1.1.2. The IP address of eth1 on SMW2 is 10.1.1.3. Port eth1 is connected to the HSS network on the mainframe.

Cluster IP2

This resource provides the virtual IP address 10.2.1.1, which directs the traffic to eth2 on the active SMW. The IP address of eth2 on SMW1 is 10.2.1.2. The IP address of eth2 on SMW2 is 10.2.1.3. Ports of eth2 on both SMWs are directly connected with a cable to provide the heartbeat communication channel for corosync.

Cluster IP3

This resource provides the virtual IP address 10.3.1.1, which directs the traffic to eth3 of the active SMW. The IP address of eth3 on SMW1 is 10.3.1.2. The IP address of eth3 on SMW2 is 10.3.1.3. Port eth3 is connected to the boot node on the mainframe.

Cluster IP4

This resource provides the virtual IP address 10.4.1.1, which directs the traffic to eth4 of the active SMW. The IP address of eth4 on SMW1 is 10.4.1.2. The IP address of eth4 on SMW2 is 10.4.1.3. Ports of eth4 on both SMWs are directly connected with a cable to provide the redundant heartbeat communication channel for corosync.

Cluster IP5

This resource provides the virtual IP address 10.5.1.1, which directs the traffic to eth5 of the active SMW. The IP address of eth5 on SMW1 is 10.4.1.2. The IP address of eth4 on SMW2 is 10.4.1.3. Ports of eth5 on both SMWs are directly connected with a cable to provide a backdoor channel to do database replication.

Cluster Resources

ClusterMonitor

This resource monitors the cluster at cluster startup time. It populates `/etc/hosts` by calling `/opt/cray/hss/default/bin/discoverhosts`. This ensures that after a failover `/etc/hosts` on

the new active SMW includes the latest routes added on the previous active SMW. ClusterMonitor also dumps the cluster status after a failover and dumps the failcount status to `/var/log/smwha.log`, which provides the information for failover debugging.

ClusterTimeSync

This resource monitors the RTCs on the active SMW and the passive SMW and synchronizes them to match if the time difference is 10 seconds or more.

HSSDaemonMonitor

This resource periodically monitors all HSS daemons. If a daemon crashes, systemd will restart the failed daemon, and the HA software will simultaneously increase the failcount for the HSSDaemonMonitor by one. When the failcount reaches the failcount-threshold, a failover is triggered.

Notification

This resource sends an email to the system administrator when a failover occurs. The email address is configurable.

ResourceInit

This resource scans `/etc/fstab` to find the mount options for mysql, /home, log, imps, and repos disks, then configures the cluster resources to match those options. Please note the disk options defined in `/etc/fstab` include the subvols, which are pointed to the current snapshot. Both active and passive SMWs must have the same current snapshot name so the same subvols are used after a failover.

cray-cfgset-cache

This resource manages the IMPS cfgset-cache daemon via systemd.

dhcpd

This resource monitors dhcpd on the active SMW and ensures that dhcpd is stopped on the passive SMW. Failed dhcpd can force a failover.

fsync

This resource synchronizes the files and directories listed in `/etc/csync2/csync2_cray.cfg` from the active SMW to the passive SMW. The synchronization interval is 100 seconds.

hss-daemons

This resource starts/stops the following HSS daemons and xtbootsys-started background commands on the active SMW and ensures that none of them are running on the passive SMW. The daemons include:

```
erd erdh state_manager nid_mgr bootmanager sedc_manager xtpmd
erfsd xtremoted xtpowerd nimsd xtsnmpd xtdiagd
xtconsole xtnetwatch xtconsumer xthwerrlogd xtnlrd dumpd xtpcimon
```

The HSS daemons can be controlled using the following equivalent commands:

- `crm resource start|stop|status hss-daemons`
- `rsms status`
- `systemct1 status rsms`

Status can be obtained with the following commands, with different output:

- On either SMW: `crm resource status hss-daemons`

- On the active SMW: `systemctl status rsms`
- On the active SMW: `rsms status` or (`or /etc/init.d/rsms status`)

stonith-1 and stonith-2

Both the active SMW and the passive SMW has its own running STONITH. Each resource is associated with the iDRAC on its respective SMW. If a critical failure is detected on the active SMW, the passive SMW will STONITH (power off) the active SMW, causing a failover. The SMW that received the STONITH is now powered off and will need to be powered back on before it can rejoin the HA configuration as the passive SMW.

PostgreSQL

This resource controls and monitors the Power Management Database (PMDB) PostgreSQL server, `postgresqld`.

clo_PostgreSQL

This resource controls and monitors the cloned PostgreSQL resource that now runs on both the Active and Passive SMWs.

7.2 Group Resources

This section describes group resources, which group a set of sub-resources together. Within a group, stopping one sub-resource will stop all sub-resources. Additionally, the start or stop sequence is forced based on the location of a sub-resource in the group. These sub-resources start from top to bottom and stop from bottom to top.

HSSGroup

postgresqld

This resource starts and stops the daemon postgresqld on the active SMW. If postgresqld fails, a failover can be triggered. The HA software will attempt to restart a resource a few times. If it determines that it cannot start the resource on the active SMW it will initiate a failover.

mysqld

This resource starts and stops the daemon mysqld on the active SMW. If mysqld fails, a failover can be triggered.

IMPSGroup

cray-ids-service

This resource triggers the script `/etc/init.d/cray-ids-service` to start and stop DIOD (Distributed I/O Daemons) processes on the SMW. The DIOD process exports content from the local filesystem and allows 9pfs clients to mount and use the associated content. Additionally, cray-ids-service also monitors the process. If a failure is detected, a failover can be triggered.

cray-ansible

This resource starts the `'/etc/init.d/cray-ansible` script when the cluster starts and stops the script when the cluster stops.

IMPSFilesystemConfig

This resource operates on `/var/opt/cray/imps`and controls boot node mount points from the SMW. When the resource starts on the active SMW, it first unexports all exported directories on the SMW and then re-exports them. Then it uses `ssh` to the boot node and mounts all smw:/* directories on the boot node. When the resource stops on the active SMW, it un-exports all NFS directories with `exportfs -au` and then uses `ssh` to the boot node to unmount all smw:/* directories.

SharedFileSystemGroup

homedir

This resource mounts `/home` on the active SMW to the RAID. If a failure is detected, a failover can be triggered..

md-fs

This resource mounts `/var/lib/mysql` on the active SMW to the RAID. If a failure is detected, a failover can be triggered..

imps-fs

This resource mounts `/var/opt/cray/imps` on the shared RAID and monitors it. If a failure is detected, a failover can be triggered.

ml-fs

This resource mounts `/var/opt/cray/disk/1` to the shared RAID and monitors its status. If an error is detected, a failover may be triggered.

repos-fs

This resource mounts `/var/opt/cray/repos` to the shared RAID connected to the active SMW and ensures it is not mounted on the passive SMW. A double mount (mount from both active and passive SMW) will corrupt the RAID.

pm-fs

This resource mounts `/var/lib/pgsql` on the active SMW to the RAID. If the DRBD disks are used, it mounts `/var/lib/pgsql` on the master DRBD disk from the active SMW. If the resource fails, a failover can be triggered.

LOGGroup

This resource group monitors and controls the system log disk and daemons.

cray-syslog

This resource starts and stops Cray's `llm` logger and monitors its status. If an error is detected, a failover can be triggered.

LOGFilesystemConfig

This resource operates on `/var/opt/cray/disk/1`. When this resource starts on the active SMW it first un-exports all exported directories on SMW and then re-exports them. Then it uses `ssh` to the boot node and mounts all smw:/* directories on the boot node. When the resource stops on the active SMW, it unexports all NFS directories with `exportfs -au` and then uses `ssh` to the boot node and unmounts all smw:/* directories.

SystemGroup

This resource group monitors and controls system level daemons.

nfsserver

This resource starts and stops the `nfsserver` on the active SMW. If an error is detected, a failover can be trigged.

EnableRsyslog

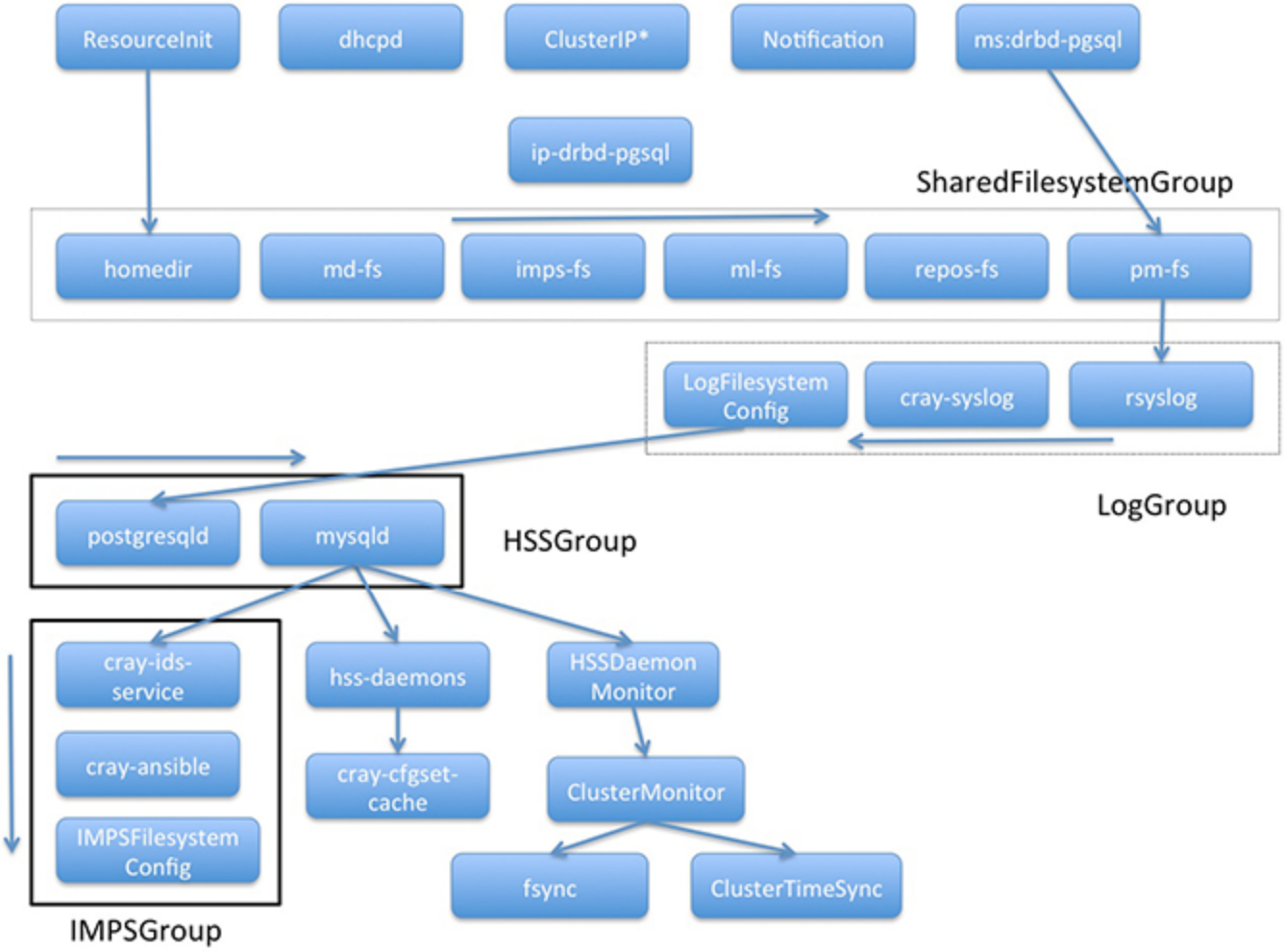
This resource enables and disables the `rsyslog` daemon on the active SMW.

syslog.socket

This resource starts and stops the `syslog.socket` service on the active SMW. syslog.socket controls stopping and starting of the rsyslog daemon in the background. If an error is detected, a failover can be triggered.

7.3 Cluster Resource Dependence

The following diagram indicates the cluster dependence and their start sequence (from top to bottom) when the cluster starts on the active SMW.



7.4 Check Cluster Status

About this task

For the SMW cluster to work correctly, both SMWs should be in the online state and all resources should be in the Started state.

Procedure

1. Verify that both SMWs are online.

```
smw1# crm status | grep " Online:"
Online: [ smw1 smw2 ]
```

2. Verify health state.

   If Health State is Healthy, all resources are running.

```
smw1# ha_health
Cluster State
----------------------------------------
Health State                : Healthy
Active Node                 : smw2
Node-1                      : smw1 (online)
Node-2                      : smw2 (online)
Number of Resources         : 33
Number of Resources Running  : 33
Number of Resources Stopped : 0
Maintenance Mode            : disabled
Stonith Mode                : enabled
----------------------------------------
```

3. Verify that the SMW cluster is working correctly by verifying both SMWs are online and all resources are in the Started state.

```
smw1# check_config smw1 smw2 smw1-drac-ip smw2-drac-ip

Please type iDRAC root password and press [ENTER]:

Checking configuration. Please wait...

Logging output to /var/log/ha-check-config-20170427090155.log

HA service (pacemaker/corosync) is running.

This SMW is: ACTIVE

...

Verify hssbootlink pointers are in sync...
    OK

System is configured correctly.
```

4. The `check_config` command is running tests on the cluster configuration to determine if it is correct, and will report any errors found.

```
smw1# check_config smw1 smw2 smw1-drac-ip smw2-drac-ip
Please type iDRAC root password and press [ENTER]:

Checking configuration. Please wait...

Logging output to /var/log/ha-check-config-20170427090155.log

HA service (pacemaker/corosync) is running.

This SMW is: ACTIVE

Verify SMW ping...
    ICMP ping to smw1 succeeded
    ICMP ping to smw2 succeeded

Verify passwordless access via ssh to both SMWs...
    SMW smw1 - OK
    SMW smw2 - OK

Verify software revisions match on both SMWs...
    /opt/cray/hss/default/etc/smw-release - OK
    /etc/SuSE-release - OK
    /etc/opt/cray/release/cle-release - OK
    /etc/os-release - OK

Verify iDRAC ping...
    ICMP ping to mars1-drac succeeded
    ICMP ping to mars2-drac succeeded

Verify iDRAC configuration on both SMWs...
    iDRAC mars1-drac configured correctly
    iDRAC mars2-drac configured correctly

Verify /etc/fstab shared entries match on both SMWs...
    /dev/mapper/smw_node_vg-home: OK
    /dev/mapper/smw_node_vg-imps: OK
    /dev/mapper/smw_node_vg-db: OK
    /dev/mapper/smw_node_vg-log: OK
    /dev/mapper/smw_node_vg-repos: OK

Check DRBD configurtation...
    DRBD is configured; validating /etc/drbd.d/r0.res file...
    smw1 OK
    smw2 OK

Check DRBD status...
    DRBD is configured; checking /proc/drbd for status...
    OK

Verify Ethernet interfaces are not managed by cfgset tool...
    OK

Verify STONITH IP addresses are the given iDRAC IP addresses...
    drac-1: 172.30.12.156    stonith-1: 172.30.12.156 - OK
    drac-2: 172.30.12.156    stonith-1: 172.30.12.156 - OK

Verify STONITH IP addresses are the IP addresses known to each iDRAC...
```

```
    stonith-1: 172.30.12.156    iDRAC: 172.30.12.156 - OK
    stonith-2: 172.30.12.158    iDRAC: 172.30.12.158 - OK

Verify that Ethernet interfaces have link detected...
    SMW smw1: interface 0 link UP
    SMW smw1: interface 1 link UP
    SMW smw1: interface 2 link UP
    SMW smw1: interface 3 link UP
    SMW smw1: interface 4 link UP
    SMW smw1: interface 5 link UP
    SMW smw2: interface 0 link UP
    SMW smw2: interface 1 link UP
    SMW smw2: interface 2 link UP
    SMW smw2: interface 3 link UP
    SMW smw2: interface 4 link UP
    SMW smw2: interface 5 link UP

Verify that file sync between SMWs is working...
    OK

Verify IP addresses for Ethernet interfaces...
    smw1    eth1 - OK
    smw1    eth2 - OK
    smw1    eth3 - OK
    smw1    eth4 - OK
    smw1    eth5 - OK
    smw2    eth1 - OK
    smw2    eth2 - OK
    smw2    eth3 - OK
    smw2    eth4 - OK
    smw2    eth5 - OK

Verify secondary (virtual) IP addresses on active SMW...
    smw1    eth1 - OK
    smw1    eth2 - OK
    smw1    eth3 - OK
    smw1    eth4 - OK
    smw1    eth5 - OK

Verify ping to real IP addresses on Ethernet interfaces...
    smw1:    ICMP ping to 10.1.1.3 succeeded
    smw1:    ICMP ping to 10.2.1.3 succeeded
    smw1:    ICMP ping to 10.3.1.3 succeeded
    smw1:    ICMP ping to 10.4.1.3 succeeded
    smw1:    ICMP ping to 10.5.1.3 succeeded
    smw2:    ICMP ping to 10.1.1.2 succeeded
    smw2:    ICMP ping to 10.2.1.2 succeeded
    smw2:    ICMP ping to 10.3.1.2 succeeded
    smw2:    ICMP ping to 10.4.1.2 succeeded
    smw2:    ICMP ping to 10.5.1.2 succeeded

Verify passwordless access to the boot node from both SMWs...
    SMW smw1 OK
    SMW smw2 OK

Check the boot node for stale NFS mountpoints from the SMW...
    OK - None found

Verify that the NIMS map link is in sync...
    NIMS map links are identical
```

```
Run ha_health and report status...
    OK - ha_health reports SMWs are healthy


    Cluster State
    -----------------------------------------
    Health State                : Healthy
    Active Node                 : smw1
    Node-1                      : smw1 (online)
    Node-2                      : smw2 (online)
    Number of Resources         : 35
    Number of Resources Running : 35
    Number of Resources Stopped : 0
    Maintenance Mode            : disabled
    Stonith Mode                : enabled
    -----------------------------------------

Check permissions for MariaDB...
    OK: /var/lib/mysql permissions 755 owner mysql:mysql
    OK: /etc/my.cnf permissions 644 owner root:root

Check status of HSS daemons...
    OK

PID        DAEMON               STATE                UPTIME
9344       erd                  running              Wed 2017-04-19 13:48:33 CDT
9656       erdh                 running              Wed 2017-04-19 13:48:36 CDT
10136      state_manager        running              Wed 2017-04-19 13:48:36 CDT
10393      nid_mgr              running              Wed 2017-04-19 13:48:37 CDT
10575      bootmanager          running              Wed 2017-04-19 13:48:38 CDT
11710      xtpmd                running              Wed 2017-04-19 13:48:47 CDT
11984      erfsd                running              Wed 2017-04-19 13:48:51 CDT
12604      xtremoted            running              Wed 2017-04-19 13:48:55 CDT
12947      xtpowerd             running              Wed 2017-04-19 13:48:59 CDT
13439      nimsd                running              Wed 2017-04-19 13:49:02 CDT
13625      xtsnmpd              running              Wed 2017-04-19 13:49:06 CDT
13841      xtdiagd              running              Wed 2017-04-19 13:49:10 CDT

Check that HSS state manager is responding...
    OK - state manager is ready

Check for filesystem out-of-space conditions...
    OK - none found

Verify SMW clocks are in sync...
    OK - SMWs are within 20 seconds of each other

Check corosync configuration...
    OK - On smw1 bindnetaddr is correct
    OK - On smw1 mcastaddr is correct
    OK - On smw1 mcastport is correct
    OK - On smw2 bindnetaddr is correct
    OK - On smw2 mcastaddr is correct
    OK - On smw2 mcastport is correct

Check for mounts of shared storage on the passive SMW...
    OK - none found

Verify hssbootlink pointers are in sync...
    OK
```

```
System is configured correctly.
```

7.5 Snapshot Failure

Creating an extra copy of the current configuration ensures that a user can return to that configuration if unwanted changes are made. However, there are several instances in which a snapshot will not work.

snaputil only works on the active SMW

Only run `snaputil create` on the active SMW because all shared btrfs disks are only mounted on the active SMW. Running `snaputil create` on the passive SMW will fail.

snaputil only works when all shared disks are mounted

snaputil requires that all shared disks are mounted. They are listed as:

```
smw1# df | grep mapper
/dev/mapper/smw_node_vg-home   104806400  33350408  71455992  32% /home
/dev/mapper/smw_node_vg-db      10485760   1593288   6817848  19% /var/lib/mysql
/dev/mapper/smw_node_vg-imps  1048576000 380721888 654627296  37% /var/opt/cray/imps
/dev/mapper/smw_node_vg-log    628838400  95869440 532968960  16% /var/opt/cray/disk/1
/dev/mapper/smw_node_vg-repos  104857600  35752308  67355564  35% /var/opt/cray/repos
```

If the resources for the shared disks are not running, a snapshot cannot be created.

Shared btrfs disks show different contents after failover

Sometimes the shared btrfs disk(s) show different contents after a failover. While there is no generic way to detect such an event, one way it can be discovered is if the system behaves differently when the active SMW switches. For example, the release number is different, a user configuration file has different contents, etc. This most likely occurs because the active and passive SMWs are pointing to different current snapshot names.

The cluster fs resources, like md-fs and repos-fs, are configured to point to the current snapshot subvol as follows:

```
primitive md-fs Filesystem \
  operations $id=md-fs-operations \
  op monitor interval=20 timeout=40 \
  op start interval=0 timeout=60 \
  op stop interval=0 timeout=60 \
  params device="/dev/mapper/rhinegreen_smw_node_vg-db" directory="/var/lib/
mysql" \
  fstype=btrfs options="x-cray.managed,noauto,x-cray.snapshot,subvol=snapshots \
  /smw-8.0.19_cle-6.0.19.201601141018-ha-trigger,nofail" \
  meta target-role=Started
```

The cluster retrieves the subvol name from `/etc/fstab` whenever it starts on an SMW. When a snapshot is created, snaputil updates `/etc/fstab` to set the subvol to point to the newly created snapshot. If the active and passive SMWs have different snapshot names, `/etc/fstab` will include a different subvol. Consequently, the cluster will point to a separate subvol after the cluster moves to the other SMW.

7.6 Reasons for Failover

Once a failover has occurred, it is important to understand what caused the failover, in order to prevent future failovers, or take any other necessary measures before resuming regular use.

Disk full failover

When a shared disk or local disk is full, the SMW runs slowly. This can cause the cluster heartbeat communication to be interrupted and trigger a failover. Monitor the disk usage on a regular basis to avoid the disks becoming too full.

SMWHAconfig

`SMWHAconfig` is the HA configuration command to do a fresh HA software installation or HA software updates. Log into the active SMW using its actual hostname. Do not run this script on the passive SMW as failures will result. If `SMWHAconfig` fails, it can be run again without causing problems, once the problem causing the failure has been addressed. `SMWHAconfig` logs its output to a date-stamped file in `/var/log`, for example: `/var/log/SMWHAconfig.log.20170129`. `SMWHAconfig --update` creates a shadow CIB (Cluster Information Base) and applies updates to it. Only after all operations are successful is the live CIB updated to the shadow CIB. If the update fails the current live CIB is not affected.

Login session terminated

There are three hostnames on an HA system: smw1, smw2, and the virtual hostname smw. Regular users log in using the virtual hostname smw, while the system administator typically logs in using the actual hostnames smw1 and smw2. If a failover happens when logged in to the cluster virtual hostname, the current login session will be terminated. However, if the actual hostname is used, the login session will continue. Consequently, administrators should log in using the actual SMW hostname to run the installer when installing software on the SMW to ensure the login session continues during the software installation.

Cluster ping/pong

If a cluster resource fails on both SMWs, the cluster can fall into a ping/pong state. This occurs when a state that is in common on both the SMWs causes a failover, and a STONITH does not occur. Examples of this include failure of a shared file system, or erroneous cluster configuration. The cluster fails from the active SMW to the passive SMW. Since the same resource will also fail on the passive SMW, the cluster will then fail back to the active SMW from the passive SMW. This is a limitation of the current cluster software. To fix this problem, one of the SMWs must be moved to standby with the following command:

```
smw1# crm node standby smw2
```

After one SMW has been moved to standby, the cluster resource will remain on smw1 and the the problem can be debugged.

Corosync not started

After reboot, the SMW cluster daemon corosync may not start. This event has been observed to happen after a fresh HA software installation and can be identified by checking the cluster status:

```
smw1# crm status
ERROR: status: crm_mon (rc=107): Connection to cluster failed:
Transport endpoint is not connected
smw2:~ # crm status
ERROR: status: crm_mon (rc=107): Connection to cluster failed:
Transport endpoint is not connected
```

Typically, such an occurrence is caused by an incorrect corosync interface configuration.
Check `/etc/corosync/corosync.conf` to ensure bindnetaddr and mcastport are configured as defined in the HA installation procedure S-0044-*. For the sake of this example, only the necessary section is shown:

```
interface {
   #Network Address to be bind for this interface setting
   bindnetaddr:    10.2.1.0
   #The multicast address to be used
   mcastaddr:      226.0.0.1
   #The ringnumber assigned to this interface setting
   ringnumber:     0
   #The multicast port to be used
   mcastport:      1694
   #Time-to-live for cluster communication packets
   ttl:    1
}
interface {
    #The multicast address to be used
    mcastaddr:      225.0.0.1
    #Network Address to be bind for this interface setting
    bindnetaddr:    10.4.1.0
    #The multicast port to be used
    mcastport:      1694
    #The ringnumber assigned to this interface setting
    ringnumber:     1
 }
```

After correcting the error (if it exists), reboot both SMWs and the cluster will come up as expected. Verify the cluster configuration after rebooting with `checkconfig`.

xtdiscover failover

The `xtdiscover` command interacts with the SMW MySQL database and restarts the HSS daemons during the discover operation. There is a possibility that this activity could cause an unexpected failover. If the failover occurs while xtdiscover is updating the database, the database will need to be restored manually.

To avoid a potential failover, the cluster should be set into maintenance mode during an update:

```
smw1# maintenance_mode_configure enable
```

Maintenance mode should be disabled after xtdiscover is finished:

```
smw1# maintenance_mode_configure disable
```

Other commands that change the state of hardware and which may result in incomplete changes if a failover occurs while the command is executing include `xtbootsys`, `xtzap`, and `snaputil delete`.

Manual install of smw-ha*.rpm breaks the HA system

The SMW HA installers, SMWHAinstall and SMWHAconfig, must be used to update the SMW HA software. Manually installing smw-ha-*.rpm using `rpm` will wipe out the HA configuration files configured by SMWHAconfig.

If an HA rpm has been accidentally installed manually and the cluster stops working, the cluster can be restored:

```
smw1# crm configure load update /opt/cray/ha-smw \
/default/hainst/_CLUSTER_CONFIG_BACKUP_yyyymmddhhmmss
```

Rerun SMWHAinstall and SMWHAconfig to install the desired HA software.

HSS Firmware Boot

The symbolic link to the HSS controller boot image, set by `hssbootlink`, must point to the same boot image on both the active and passive SMWs. This ensures that after an SMW failover and a reboot of CCs, the controllers wil load the same HSS image. If the link is different between the two SMWs, an incorrect image may be loaded when the CCs boot after an SMW failover. This requires that the HSS image must be built on both SMWs and `hssbootlink` executed to point to the same image.

Unexpected failover

An unexpected one time failover are known to occur without obvious reasons. Try to find clues in `/var/opt/cray/log/smwmessages`, `/var/log/smwha.log`, `/var/log/pacemaker.log` or dmesg. Search for "timed out," "fence," "GetUnit," and "STONITH" within these files to find the relative logs.

7.7 Debug a Cluster Problem or Failover

Prerequisites

A failover has recently taken place.

About this task

After a failover occurs, there are several methods for discovering what triggered it.

Procedure

1. Check `/var/log/smwha.log` for failover information.

   This log file on the current active SMW includes the Time Stamp indication when a failover happened and the SMW hostname of the new active SMW, the failcounts before the failover, and the failcount history.

   ```
   ************************************************************
   smw1 acted as active SMW at Sun Mar 6 14:51:07 CST 2016   <--- WHEN FAILOVER HAPPENED
   ************************************************************
   node=smw2 scope=status  name=fail-count-ClusterIP value=0
   node=smw2 scope=status  name=fail-count-ClusterIP1 value=0
   node=smw2 scope=status  name=fail-count-ClusterIP2 value=0
   node=smw2 scope=status  name=fail-count-ClusterIP3 value=0
   node=smw2 scope=status  name=fail-count-ClusterIP4 value=0
   node=smw2 scope=status  name=fail-count-ClusterMonitor value=0
   node=smw2 scope=status  name=fail-count-ClusterTimeSync value=0
   node=smw2 scope=status  name=fail-count-HSSDaemonMonitor value=0
   node=smw2 scope=status  name=fail-count-Notification value=0
   node=smw2 scope=status  name=fail-count-dhcpd value=0
   node=smw2 scope=status  name=fail-count-fsync value=0
   node=smw2 scope=status  name=fail-count-hss-daemons value=0
   node=smw2 scope=status  name=fail-count-repos-fs value=0
   ```

```
node=smw2 scope=status  name=fail-count-stonith-1 value=0
node=smw2 scope=status  name=fail-count-stonith-2 value=0
node=smw2 scope=status  name=fail-count-ResourceInit value=0
node=smw2 scope=status  name=fail-count-homedir value=0
node=smw2 scope=status  name=fail-count-md-fs value=0
node=smw2 scope=status  name=fail-count-mysqld value=0
node=smw2 scope=status  name=fail-count-imps-fs value=0
node=smw2 scope=status  name=fail-count-cray-ids-service value=0
node=smw2 scope=status  name=fail-count-cray-ansible value=0
node=smw2 scope=status  name=fail-count-IMPSFilesystemConfig value=0
node=smw2 scope=status  name=fail-count-ml-fs value=0
node=smw2 scope=status  name=fail-count-rsyslog value=0
node=smw2 scope=status  name=fail-count-cray-syslog value=0
node=smw2 scope=status  name=fail-count-LOGFilesystemConfig value=0
node=smw1 scope=status  name=fail-count-ClusterIP value=0
node=smw1 scope=status  name=fail-count-ClusterIP1 value=0
node=smw1 scope=status  name=fail-count-ClusterIP2 value=0
node=smw1 scope=status  name=fail-count-ClusterIP3 value=0
node=smw1 scope=status  name=fail-count-ClusterIP4 value=0
node=smw1 scope=status  name=fail-count-ClusterMonitor value=0
node=smw1 scope=status  name=fail-count-ClusterTimeSync value=0
node=smw1 scope=status  name=fail-count-HSSDaemonMonitor value=0
node=smw1 scope=status  name=fail-count-Notification value=0
node=smw1 scope=status  name=fail-count-dhcpd value=0
node=smw1 scope=status  name=fail-count-fsync value=0
node=smw1 scope=status  name=fail-count-hss-daemons value=0
node=smw1 scope=status  name=fail-count-repos-fs value=0
node=smw1 scope=status  name=fail-count-stonith-1 value=0
node=smw1 scope=status  name=fail-count-stonith-2 value=INFINITY  <---- WHAT CAUSED
FAILOVER
node=smw1 scope=status  name=fail-count-ResourceInit value=0
node=smw1 scope=status  name=fail-count-homedir value=0
node=smw1 scope=status  name=fail-count-md-fs value=0
node=smw1 scope=status  name=fail-count-mysqld value=0
node=smw1 scope=status  name=fail-count-imps-fs value=0
node=smw1 scope=status  name=fail-count-cray-ids-service value=0
node=smw1 scope=status  name=fail-count-cray-ansible value=0
node=smw1 scope=status  name=fail-count-IMPSFilesystemConfig value=0
node=smw1 scope=status  name=fail-count-ml-fs value=0
node=smw1 scope=status  name=fail-count-rsyslog value=0
node=smw1 scope=status  name=fail-count-cray-syslog value=0
node=smw1 scope=status  name=fail-count-LOGFilesystemConfig value=0
smw1:~ # crm status            <---- CLUSTER STATUS AFTER FAILOVER
Last updated: Sun Mar  6 14:51:23 2016
Last change: Sun Mar  6 13:44:49 2016
Stack: corosync
Current DC: smw1 (167903490) - partition with quorum
Version: 1.1.12-ad083a8
2 Nodes configured
27 Resources configured
Online: [ smw1 smw2 ]
ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
Notification   (ocf::heartbeat:MailTo):        Started smw1
dhcpd  (systemd:dhcpd.service):        Started smw1
fsync  (ocf::smw:fsync):       Started smw1
hss-daemons    (lsb:rsms):     Started smw1
repos-fs       (ocf::heartbeat:Filesystem):    Started smw1
stonith-1      (stonith:external/ipmi):        Started smw2
Resource Group: HSSGroup
    ResourceInit       (ocf::smw:ResourceInit):        Started [ smw1 ]
    homedir    (ocf::heartbeat:Filesystem):    Started smw1
    md-fs      (ocf::heartbeat:Filesystem):    Started smw1
    mysqld     (ocf::heartbeat:mysql): Started smw1
```

```
Resource Group: IMPSGroup
    imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
    cray-ids-service   (lsb:cray-ids-service): Started smw1
    cray-ansible       (lsb:cray-ansible):     Stopped
    IMPSFilesystemConfig       (ocf::smw:FileSystemConfig):    Stopped
Resource Group: LOGGroup
    ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
    rsyslog    (systemd:rsyslog.service):      Started smw1
    cray-syslog        (systemd:llmrd.service):        Started smw1
    LOGFilesystemConfig        (ocf::smw:FileSystemConfig):    Started [ smw1 ]
```

2. Search `/var/opt/cray/log/smwmessages-*` for failover information.

   All cluster related messages are logged in `smwmessages-*`, which can be examined to find the possible reasons for the failover. It is generally possible to determine when a failover happened from file `/var/log/smwha.log`. Search `smwmessages-*` for the failover time to see the events that forced the failover. If the reason for the failover is not apparent, contact a Cray representative.

3. Check `/var/log/pacemaker.log` for failover information.

   This log file contains log messages from pacemaker and is helpful in debugging the failover transaction.

4. Use `crm_report` to gather HA logs and configuration data for use by SUSE in order to diagnose problems.

   ```
   smw1# crm_report -f 00:00
   ```

   The output is a compressed tarball in `/root`, with a filename of the form *`pcmk-<date>.tar.bz2`*.

7.8 Debug Failure of a Preinstalled SMW HA System

A preinstalled and tested SMW HA system may be found to be not working after being shipped to a customer site. This may be due to the following reasons:

1. The SMW HA system IP addresses have not been reconfigured.

   At Cray, all IP addresses are configured for the Cray environment (network). At a customer site, all IP addresses must be changed for the customer network. Update the following IP addresses:

   - smw1 IP address
   - smw2 IP address
   - cluster virtual IP address
   - smw1 iDRAC IP address
   - smw2 iDRAC IP address

2. The SMW HA system has not been cabled correctly.

   See *XC™ Series SMW HA Installation Guide* (S-0044) for details on how to reconfigure an SMW HA system that was preinstalled in the factory.

7.9 Restart Stopped Resources

About this task

A successful failover should restart all cluster-managed resources on the other SMW. Unless the cluster has been placed into maintenance mode, stopped resources do not occur during normal operation. A stopped resource usually indicates that the resource's failcount on the active SMW has exceeded the migration threshold for that resource and that no failover is possible because the passive SMW is offline, unclean, or is also not eligible to run that resource (typically, because its failcount has also exceeded the migration threshold).

If this cluster has stopped resources, use this procedure on either the active or passive SMW. Execute the commands in this procedure as `root`.

Procedure

1. Execute the following command on either SMW as `root`.

```
smw1# crm resource status
 ClusterIP      (ocf::heartbeat:IPaddr2):       Started
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started
 HSSDaemonMonitor       (ocf::smw:HSSDaemonMonitor):    Started
 Notification   (ocf::heartbeat:MailTo):        Started
 cray-cfgset-cache      (systemd:cray-cfgset-cache):    Started
 dhcpd  (systemd:dhcpd.service):        Started
 fsync  (ocf::smw:fsync):       Stopped
 hss-daemons    (lsb:rsms):     Started
 .
 .
 .
```

The status `Stopped` is usually caused by exceeding the failcount for a resource.

2. Display the failcount data for all resources.

```
smw1# show_failcounts
node=smw1 scope=status  name=fail-count-ClusterIP value=0
node=smw1 scope=status  name=fail-count-ClusterIP1 value=0
node=smw1 scope=status  name=fail-count-ClusterIP2 value=0
node=smw1 scope=status  name=fail-count-ClusterIP3 value=0
node=smw1 scope=status  name=fail-count-ClusterIP4 value=0
node=smw1 scope=status  name=fail-count-ClusterMonitor value=0
node=smw1 scope=status  name=fail-count-ClusterTimeSync value=0
node=smw1 scope=status  name=fail-count-HSSDaemonMonitor value=0
node=smw1 scope=status  name=fail-count-Notification value=0
node=smw1 scope=status  name=fail-count-cray-cfgset-cache value=0
node=smw1 scope=status  name=fail-count-dhcpd value=0
node=smw1 scope=status  name=fail-count-fsync value=1000001
node=smw1 scope=status  name=fail-count-hss-daemons value=0
 .
 .
 .
```

Or use the `show_failcount` command to display the failcount data for a single resource on the specified SMW. This example shows the data for `fsync` on *`smw1`*.

```
smw1# show_failcount smw1 fsync
node=smw1 scope=status  name=fail-count-fsync value=1000001
```

3. Clear the failcounts and return all values to zero.

```
smw1# clear_failcounts
Clearing failcount on node smw1
Clearing failcount on node=smw1 for resource=ClusterIP
Clearing failcount on node=smw1 for resource=ClusterIP1
Clearing failcount on node=smw1 for resource=ClusterIP2
Clearing failcount on node=smw1 for resource=ClusterIP3
Clearing failcount on node=smw1 for resource=ClusterIP4
Clearing failcount on node=smw1 for resource=ClusterMonitor
Clearing failcount on node=smw1 for resource=ClusterTimeSync
Clearing failcount on node=smw1 for resource=HSSDaemonMonitor
Clearing failcount on node=smw1 for resource=Notification
 .
 .
 .
Clearing failcount on node=smw2 for resource=cray-ansible
Clearing failcount on node=smw2 for resource=IMPSFilesystemConfig
Clearing failcount on node=smw2 for resource=ml-fs
Clearing failcount on node=smw2 for resource=rsyslog
Clearing failcount on node=smw2 for resource=cray-syslog
Clearing failcount on node=smw2 for resource=LOGFilesystemConfig
Clearing failcount on node=smw2 for resource=ip_drbd_pgsql
Clearing failcount on node=smw2 for resource=drbd_pgsql:0
Clearing failcount on node=smw2 for resource=drbd_pgsql:1
```

4. After all failcounts have been cleared, the resource should be up and running. Check the cluster status again to verify that the resource has been restarted.

```
smw1# crm resource status
 ClusterIP      (ocf::heartbeat:IPaddr2):       Started
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started
 HSSDaemonMonitor       (ocf::smw:HSSDaemonMonitor):    Started
 Notification   (ocf::heartbeat:MailTo):        Started
 cray-cfgset-cache      (systemd:cray-cfgset-cache):    Started
 dhcpd  (systemd:dhcpd.service):        Started
 fsync  (ocf::smw:fsync):       Stopped
 hss-daemons    (lsb:rsms):     Started
 .
 .
 .
```

5. If not all resources have started, execute the `clean_resources` command.

```
smw1# clean_resources
Cleaning resources on node smw1
Cleaning resource on node=smw1 for resource=ClusterIP
Cleaning up ClusterIP on smw1
```

```
Cleaning up ClusterIP on smw2
Waiting for 2 replies from the CRMd.. OK
Cleaning resource on node=smw1 for resource=ClusterIP1
Cleaning up ClusterIP1 on smw1
Cleaning up ClusterIP1 on smw2
Waiting for 2 replies from the CRMd.. OK
 .
 .
 .
Cleaning resource on node=smw1 for resource=syslog.socket
Cleaned up NFSServer on smw1
Cleaned up NFSServer on smw2
Cleaned up EnableRsyslog on smw1
Cleaned up EnableRsyslog on smw2
Cleaned up syslog.socket on smw1
Cleaned up syslog.socket on smw2
Multiple attributes match name=target-role
  Value: started        (id=SystemGroup-meta_attributes-target-role)
  Value: Started        (id=NFSServer-meta_attributes-target-role)
  Value: Started        (id=EnableRsyslog-meta_attributes-target-role)
  Value: Started        (id=syslog.socket-meta_attributes-target-role)
Waiting for 6 replies from the CRMd...... OK
Cleaning resource on node=ethel for resource=PostgreSQL:0
Cleaned up PostgreSQL:0 on smw1
Cleaned up PostgreSQL:0 on smw2
Waiting for 2 replies from the CRMd.. OK
Cleaning resource on node=smw1 for resource=PostgreSQL:1
Cleaned up PostgreSQL:0 on smw1
Cleaned up PostgreSQL:0 on smw2
Waiting for 2 replies from the CRMd.. OK
```

6. After running `clean_resources`, wait several minutes for cluster activity to settle, then check cluster status with the `crm_mon -r1` command to ensure that all resources have started.

7.10 Return an SMW to the HA Cluster After It Has Been Powered Off

Procedure

1. As `root` on either SMW, check the cluster status with the `crm_mon` command. This procedure assumes that *smw1* is the active SMW and that *smw2* is the passive SMW.

```
smw1# crm status Stack: unknown
Current DC: smw2 (version unknown) - partition with quorum
Last updated: Tue Aug 29 16:25:47 2017
Last change: Tue Aug 29 09:03:49 2017 by root via crm_attribute on smw2

2 nodes configured
35 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
```

```
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
.
.
.
 Resource Group: HSSGroup
     postgresqld        (systemd:postgresql):   Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
 Resource Group: IMPSGroup
     cray-ids-service   (systemd:cray-ids-service):     Started smw1
     cray-ansible       (systemd:cray-ansible): Started smw1
     IMPSFilesystemConfig       (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started smw1
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
 ip-drbd-pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
 Master/Slave Set: ms-drbd-pgsql [drbd-pgsql]
     Masters: [ smw1 ]
     Slaves: [ smw2 ]
```

2. Determine the cause of the problem and resolve it before continuing with this procedure.

3. On the active SMW, put the passive SMW into standby mode.

   ```
   smw1# crm node standby smw2
   ```

4. Check the power status of the passive SMW. Replace *smw2-iDRAC-IP-addr* with the passive SMW's iDRAC IP address.

   ```
   smw1# /usr/bin/ipmitool -I lanplus -U root -H smw2-iDRAC-IP-addr \
   -a chassis power status
   Password:
   Chassis Power is off
   ```

   At the `Password:` prompt, enter the `root` password for the iDRAC.

5. If the power status is `off`, use the following command to turn power on. Replace *smw2-iDRAC-IP-addr* with the passive SMW's iDRAC IP address.

   ```
   smw1# /usr/bin/ipmitool -I lanplus -U root -H smw2-iDRAC-IP-addr \
   -a chassis power on
   ```

6. Verify the changed power status. Replace *smw2-iDRAC-IP-addr* with the passive SMW's iDRAC IP address.

```
smw1# /usr/bin/ipmitool -I lanplus -U root -H smw2-iDRAC-IP-addr \
-a chassis power status
Password:
Chassis Power is on
```

At the `Password:` prompt, enter the `root` password for the iDRAC.

7. Wait for the SMW to reboot.

   Before continuing, wait until the SMW has rejoined the cluster. After the SMW responds to a `ping` command, log into the SMW, sleep for at least 2 minutes, then execute the `crm_mon -r1` command to verify that the active SMW is online.

8. Join the passive SMW to the cluster.

```
smw1# crm node online smw2
```

9. Display the cluster status to verify that all resources are running. Examine the output to verify that each resource has started by looking for `Started smw1` or `Started smw2`. Also look for any failed actions at the end of the output.

```
smw1# crm status
Stack: unknown
Current DC: smw2 (version unknown) - partition with quorum
Last updated: Tue Aug 29 16:25:47 2017
Last change: Tue Aug 29 09:03:49 2017 by root via crm_attribute on smw2

2 nodes configured
35 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
.
.
.
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started smw1
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
 ip-drbd-pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
```

```
 Master/Slave Set: ms-drbd-pgsql [drbd-pgsql]
     Masters: [ smw1 ]
     Slaves: [ smw2 ]
```

Note that `crm status` may display different resource names, group names, or resource order on the system.

10. If not all resources have started or if any failed actions are displayed, execute the `clean_resources` command on either SMW.

> **IMPORTANT:** Before running the `clean_resources` command, log in directly as `root` (instead of using `su` from a `crayadm` login), because `clean_resources` terminates all non-root user sessions.

```
smw1# clean_resources
Cleaning resources on node smw1
Cleaning resource on node=smw1 for resource=ClusterIP
Cleaning up ClusterIP on smw1
Cleaning up ClusterIP on smw2
Waiting for 2 replies from the CRMd.. OK
Cleaning resource on node=smw1 for resource=ClusterIP1
Cleaning up ClusterIP1 on smw1
Cleaning up ClusterIP1 on smw2
Waiting for 2 replies from the CRMd.. OK
 .
 .
 .
Cleaning resource on node=smw1 for resource=syslog.socket
Cleaned up NFSServer on smw1
Cleaned up NFSServer on smw2
Cleaned up EnableRsyslog on smw1
Cleaned up EnableRsyslog on smw2
Cleaned up syslog.socket on smw1
Cleaned up syslog.socket on smw2
Multiple attributes match name=target-role
  Value: started        (id=SystemGroup-meta_attributes-target-role)
  Value: Started        (id=NFSServer-meta_attributes-target-role)
  Value: Started        (id=EnableRsyslog-meta_attributes-target-role)
  Value: Started        (id=syslog.socket-meta_attributes-target-role)
Waiting for 6 replies from the CRMd..... OK
Cleaning resource on node=ethel for resource=PostgreSQL:0
Cleaned up PostgreSQL:0 on smw1
Cleaned up PostgreSQL:0 on smw2
Waiting for 2 replies from the CRMd.. OK
Cleaning resource on node=smw1 for resource=PostgreSQL:1
Cleaned up PostgreSQL:0 on smw1
Cleaned up PostgreSQL:0 on smw2
Waiting for 2 replies from the CRMd.. OK
```

11. After running `clean_resources`, wait several minutes for cluster activity to settle (check cluster status again with the `crm_mon -r1` command). If the output of this command shows only a subset of the SMW HA services, wait for another minute, then check again. For more information, see the `clean_resources(8)` man page.

7.11 Boot SMW HA When Both Are Powered Off

About this task

There might be an instance where both of the SMWs in an HA pair have been powered down. This could be due to an install, upgrade, planned test, or a power outage. In any of these events, it is best practice to follow a simple procedure to bring the SMWs back online in the quickest and least impactive way.

Procedure

1. Power on the first SMW.
2. Wait for the first SMW to complete its boot and start all cluster resources.

   The SMW now should have started all cluster resources except for STONITH on the first SMW. If it hasn't, refer to the troubleshooting section in this document.
3. Validate the status of the SMWs if the SMWs were powered down due to a failure.
   a. Check dmesg for errors.

      If any errors are found, correct them before continuing onto the next step.
   b. Check `/var/opt/cray/log/smwmessages-<DATE>` for errors.

      If any errors are found, correct them before continuing onto the next step.
4. Check Boot RAID filesystems using fsck if SMWs were powered down due to failure.

   Run the tool in query only mode since the filesystems are mounted.

   If a problem is found, run the following substeps:
   a. Put the SMW into standby to stop all resources.

      ```
      smw1# crm node standby smw1
      ```
   b. Execute fsck or equivalent on the filesystem to repair any corruption.
   c. Once all filesystems have been recovered, bring the SMW back online

      ```
      smw1# crm node online smw1
      ```
5. Power on the second SMW.

   The SMW now should have started all cluster resources except for STONITH on the second SMW. If it hasn't, refer to the troubleshooting section in this document.
6. Wait for the second SMW to complete its boot and start the remaining cluster resources.
7. Once both SMWs are booted and all resources have started check the cluster state

   ```
   smw1# check_config smw1 smw2 smw1-drac-ip smw2-drac-ip
   ```

   Ignore warnings about not being able to contact the boot node.

   If `check_config` reports errors, resolve the errors and execute `check_config` again. If errors continue to be reported, contact support.

When no errors are reported, continue with other site specific SMW checks and CLE boot.

7.12 Check Heartbeat Cables for Repeated STONITH Failures

If the cluster manager repeatedly kills one or both SMWs with the STONITH capability, it usually means that the cluster has lost the heartbeat because of a communication issue. In this situation, check that the `eth2` and `eth4` cables are connected correctly on each SMW. For more information, see *Network Connections for an SMW HA System* on page 9.

7.13 Clear an HSS Lock After Failover Occurs During a Cray System Boot

About this task

If a failover occurs when the Cray system is booting, use this procedure to clear an HSS lock, if necessary.

Procedure

1. As `crayadm` on the active SMW, determine the lock ID.

```
crayadm@smw1> xtcli lock show
Network topology: class 2
======== SM Session Info ========
:3:s0:  mtoken=0
session id:     1
time       :    Sat Feb  2 11:22:16 2013
target type:    rt_node
members:        c0-0
---------------------------------
```

   In this example, the line "`:3:s0: mtoken=0`" indicates that service number 3 (boot manager) holds a lock. The lock ID is shown in the line "`session id: 1`", indicating a lock ID of `1`.

2. Manually clear the lock on the active SMW. In the following command, replace *`id-number`* with the actual lock ID.

```
crayadm@smw1> xtcli lock -u id-number
Network topology: class 2
```

3. Verify that the lock has been cleared.

```
crayadm@smw1> xtcli lock show
Network topology: class 2
======== SM Session Info ========
No session found in the SM.
```

4. If the lock remains in place, log on to the active SMW as `root` and restart the RSMS service. Replace *`smw1`* with the host name of the active SMW.

```
crayadm@smw1> ssh root@smw1
Password:
...
smw1# /etc/init.d/rsms restart
```

5. Ensure that CLE is not running (that is, the boot node is not partially or fully booted) before running `xtbootsys` again.

```
crayadm@smw> ping boot
```

7.14 Recover System Settings After Failover During Discovery

Procedure

1. Restore the previously saved HSS database, as described in the NOTES section of the `xtdiscover(8)` man page. The recovery procedure is the same as that for a system with a single SMW.
2. Rerun `xtdiscover`.

7.15 Recover From Loss of Power to the Active SMW and iDRAC

About this task

IMPORTANT: When the SMW HA system is running, do not remove power to the active SMW and its iDRAC to force a failover. Doing so will put the HA cluster in a frozen state where no resources are online. The HA cluster cannot recover automatically from this state.

If the active SMW and its iDRAC has lost power, use the following procedure to manually recover from this state. This procedure assumes that *smw1* lost power.

Procedure

1. Log into the other SMW (for example, *smw2*).
2. Clear the state of the SMW that lost power (for example, *smw1*).

```
smw2# crm node clearstate smw1
Do you really want to drop state for node smw1 (y/n)? y
```

3. Restore power to smw1 and its iDRAC. This should return the cluster back to normal operational state.
4. Check the cluster status with one of the following commands.
   - Use the `ha_health` command.

```
smw2# ha_health
Cluster State
```

```
----------------------------------------
Health State                : Healthy
Active Node                 : smw1
Node-1                      : smw1 (online)
Node-2                      : smw2 (online)
Number of Resources         : 33
Number of Resources Running : 33
Number of Resources Stopped : 0
Maintenance Mode            : disabled
Stonith Mode                : enabled
----------------------------------------
```

- Execute `crm_mon -r1`.

```
smw2# crm status
Stack: unknown
Current DC: smw2 (version unknown) - partition with quorum
Last updated: Tue Aug 29 16:25:47 2017
Last change: Tue Aug 29 09:03:49 2017 by root via crm_attribute on smw2

2 nodes configured
35 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
.
.
.
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started smw1
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
 ip-drbd-pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
 Master/Slave Set: ms-drbd-pgsql [drbd-pgsql]
     Masters: [ smw1 ]
     Slaves: [ smw2 ]
```

- Use the `crm resource show` command.

```
smw2# crm resource show
 ClusterIP      (ocf::heartbeat:IPaddr2):       Started
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started
```

```
ClusterIP2     (ocf::heartbeat:IPaddr2):       Started
ClusterIP3     (ocf::heartbeat:IPaddr2):       Started
ClusterIP4     (ocf::heartbeat:IPaddr2):       Started
ClusterMonitor (ocf::smw:ClusterMonitor):      Started
ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started
HSSDaemonMonitor       (ocf::smw:HSSDaemonMonitor):    Started
Notification   (ocf::heartbeat:MailTo):        Started
cray-cfgset-cache      (systemd:cray-cfgset-cache):    Started
.
.
.
ip_drbd_pgsql  (ocf::heartbeat:IPaddr2):       Started
Master/Slave Set: ms_drbd_pgsql [drbd_pgsql]
    Masters: [ smw1 ]
    Slaves: [ smw2 ]
```

5. If the cluster is still not in a normal state, reboot *smw2*.

7.16 Stop Repeated "Ping Pong" Failovers

About this task

If a resource fails on both SMWs, a "ping-pong" condition can occur with repeated failovers from one SMW to the other. For example, if a resource fails on smw1, it causes the cluster to fail over to smw2. If the same resource also fails on smw2, it causes cluster failback to smw1. This process can repeat indefinitely.

Use this procedure to fix a ping-pong failover condition.

Procedure

1. Log in as root on either SMW.
2. Put the other SMW (for example, *smw2*) in standby mode.

   ```
   smw1# crm node standby smw2
   ```

   After this SMW is stopped, all resources stay on the first SMW so that the problem can be debugged.
3. Resolve the problem with the failed resource. If necessary, restart the stopped resource as described in *Restart Stopped Resources* on page 67.
4. When the resource is running reliably, take the other SMW out of standby mode.

   ```
   smw1# crm node online smw2
   ```

7.17 PostgreSQL Does Not Start

There may be several situations in which PostgreSQL will fail to start. The status of PostgreSQL can be checked with `crm-status`.

PostgreSQL Failure on Passive SMW Due to Synchronization

If the output of `crm-status` is as follows:

```
2 nodes configured
33 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP5     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
 HSSDaemonMonitor       (ocf::smw:HSSDaemonMonitor):    Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 ResourceInit   (ocf::smw:ResourceInit):        Started smw1
 cray-cfgset-cache      (systemd:cray-cfgset-cache):    Started smw1
 dhcpd  (systemd:dhcpd.service):        Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 Resource Group: HSSGroup
     mysqld     (ocf::heartbeat:mysql): Started smw1
 Resource Group: IMPSGroup
     cray-ids-service   (systemd:cray-ids-service):     Started smw1
     cray-ansible       (systemd:cray-ansible): Started smw1
     IMPSFilesystemConfig       (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: LogGroup
     cray-syslog        (systemd:llmrd.service):        Started smw1
     LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
 Resource Group: SharedFilesystemGroup
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
 Resource Group: SystemGroup
     NFSServer  (systemd:nfsserver):    Started smw1
     EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
     syslog.socket      (systemd:syslog.socket):        Started smw1
 Clone Set: clo_PostgreSQL [PostgreSQL]
     Started: [ smw1 ]
     Stopped: [ smw2 ]
```

The last line indicates that PostgreSQL is not running on SMW2. If SMW2 is the passive SMW, this may indicate that the SMW2 is still synchronizing with the active SMW. Once synchronization is complete, PostgreSQL will have started.

To confirm that synchronization is the issue, run `journalctl -u pmdb_util`. The following lines should appear near the bottom of the output:

```
Nov 30 14:43:27 smw1 pmdb_util[40749]: [init_standby()]:
INFO: Initializing HA standby system...
Nov 30 14:43:40 smw1 pmdb_util[40749]: [init_standby()]:
INFO: Old data directory removed.
Nov 30 14:43:40 smw1 pmdb_util[40749]: [init_standby()]:
INFO: Synchronizing this standby with master. This might take a while!
```

This indicates that synchronization is not yet complete, and more wait time is needed before verifying the status of PostgreSQL.

Reported Failures in `journalctl -u pmdb_util`

If the last line of `journalctl -u pmdb_util` is `Initial replication NOT successful!`, this indicates that synchronization has failed. Look over the entire output in order to find where the problem is. Resolution of the issue depends on the problem listed.

Reported Failures in `crm_status` on the Passive SMW

Problems with PostgreSQL may also be found in the output of `crm_status`. PostgreSQL may have timed out, as seen below:

```
PostgreSQL_start_0 on smw2 'unknown error' (1): call=615, status=Timed Out,
exitreason='none',
    last-rc-change='Fri Dec  8 11:57:41 2017', queued=0ms, exec=200002ms
```

It may also have failed, as seen in the following output:

```
PostgreSQL_start_0 on ethel 'not running' (7): call=7480, status=complete,
exitreason='none',
    last-rc-change='Wed Feb 21 12:29:12 2018', queued=0ms, exec=3141ms
```

Since smw2 is the passive SMW, the most likely reason for this failure is it was not able to replicate the data from the master fast enough. The administrator will need to re-initialize the passive SMW manually using `pmdb_util ha --init_standby`. Once this is complete, run `clear_failcounts` to remove the reported failure from `crm-status`. The cluster is now in a healthy state.

Investigate PostgresSQL failure on the Active SMW

If the output of `crm-status` is as follows:

```
2 nodes configured
33 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
```

```
ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
ClusterIP5     (ocf::heartbeat:IPaddr2):       Started smw1
ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
ClusterTimeSync        (ocf::smw:ClusterTimeSync):     Started smw1
HSSDaemonMonitor       (ocf::smw:HSSDaemonMonitor):    Started smw1
Notification   (ocf::heartbeat:MailTo):        Started smw1
ResourceInit   (ocf::smw:ResourceInit):        Started smw1
cray-cfgset-cache      (systemd:cray-cfgset-cache):    Started smw1
dhcpd  (systemd:dhcpd.service):        Started smw1
fsync  (ocf::smw:fsync):       Started smw1
hss-daemons    (lsb:rsms):     Started smw1
stonith-1      (stonith:external/ipmi):        Started smw2
stonith-2      (stonith:external/ipmi):        Started smw1
Resource Group: HSSGroup
    mysqld     (ocf::heartbeat:mysql): Started smw1
Resource Group: IMPSGroup
    cray-ids-service   (systemd:cray-ids-service):     Started smw1
    cray-ansible       (systemd:cray-ansible): Started smw1
    IMPSFilesystemConfig       (ocf::smw:FileSystemConfig):    Started smw1
Resource Group: LogGroup
    cray-syslog        (systemd:llmrd.service):        Started smw1
    LogFilesystemConfig        (ocf::smw:FileSystemConfig):    Started smw1
Resource Group: SharedFilesystemGroup
    homedir    (ocf::heartbeat:Filesystem):    Started smw1
    md-fs      (ocf::heartbeat:Filesystem):    Started smw1
    imps-fs    (ocf::heartbeat:Filesystem):    Started smw1
    ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
    repos-fs   (ocf::heartbeat:Filesystem):    Started smw1
Resource Group: SystemGroup
    NFSServer  (systemd:nfsserver):    Started smw1
    EnableRsyslog      (ocf::smw:EnableRsyslog):       Started smw1
    syslog.socket      (systemd:syslog.socket):        Started smw1
Clone Set: clo_PostgreSQL [PostgreSQL]
    Stopped: [ smw1, smw2 ]
    Stopped: [ smw2 ]
```

The last line indicates that PostgreSQL is not running on either SMW, which means PostgreSQL is not running on the active SMW. This requires further research to find the error. Investigate the logs with `journalctl`. Resolution of the issue depends on the problem listed.

Resolve PostgresSQL failure on the Active SMW

If the problem with PostgreSQL can't be found, the active SMW will need to be reinitialized manually. For this procedure, see *Reinitialize the PMDB* on page 81.

For further assistance with PostgreSQL, contact a Cray representative.

7.18 Reinitialize the PMDB

About this task

When troubleshooting SMW-HA, there may be instances in which the SMW requires to be manual reinitialization.

Procedure

Initialize the replicated PMDB on both SMWs.

a. Enable maintenance mode.

```
smw1# maintenance_mode_configure enable
 Maintenance mode was enabled
```

b. Initialize the active SMW.

```
smw1# pmdb_util ha --init_master
...
[initialize()]:     INFO: ------------------------------------------------
[initialize()]:     INFO: PMDB Initialization SUCCEEDED
[initialize()]:     INFO: ------------------------------------------------
```

CAUTION: Ensure that the last three lines of the output look as above. Do not proceed until the initialization has finished.

c. Start the service for the active SMW.

```
smw1# systemctl start postgresql
```

d. Initialize the passive SMW.

```
smw2# pmdb_util ha --init_standby
[main()]:     INFO: Initializing standby...
[init_standby()]:     INFO: Initializing HA standby system...
[init_standby()]:     INFO: Old data directory removed.
[init_standby()]:     INFO: Synchronizing this standby with master. This might take a
while!
[init_standby()]:     INFO: Initial replication successful! Full output:
[init_standby()]:     INFO:         NOTICE:  WAL archiving is not enabled;
you must ensure that all required WAL segments are copied through other means to
complete the backup
[init_standby()]:     INFO: Standby successfully initialized!
```

CAUTION: Ensure that the last line of the output look as above. Do not proceed until the initialization has finished.

e. Start the service for the passive SMW.

```
smw2# systemctl start postgresql
```

f. Disable maintenance mode.

```
smw1# # maintenance_mode_configure disable
 Maintenance mode was disabled
```

g. Check `crm status`.

If you see any failed actions, proceed to the next step.

```
Clone Set: clo_PostgreSQL [PostgreSQL]
     Started: [ smw1 ]
     Stopped: [ smw2 ]
Failed Actions:
* PostgreSQL_start_0 on smw2 'not running' (7): call=7480, status=complete,
exit reason='none',
    last-rc-change='Wed Feb 21 12:29:12 2018', queued=0ms, exec=3141ms
```

If the `clo_PostgreSQL` group near the bottom shows both SMWs as "Started," then the set up is complete.

```
Clone Set: clo_PostgreSQL [PostgreSQL]
    Started: [ smw1 smw2 ]
```

h. Clear any "Failed Actions."

```
smw1# clear_failcounts
```

i. Recheck `crm status` to verify that the "Failed Actions" have cleared.

7.19 Fix an Outdated DRBD Disk

About this task

Sometimes DRBD disks can find themselves in an "Outdated"state. As a precaution, DRBD will put each instance into "Standalone" mode to prevent any communication between the two nodes. To correct this situation, manual intervention is required.

A system in this state will exhibit the following status.

```
smw1# drbdsetup status
r0 role:Primary
  disk:UpToDate
  SMW2 connection:StandAlone

smw2:~ # drbdsetup status
r0 role:Secondary
  disk:Outdated
  SMW2 connection:StandAlone
```

As can be seen from the output, both SMWs will report a "StandAlone" connection state on the secondary SMW with the secondary SMW's disk having a status of "Outdated." In this state, SMW1 is the only one that can start all of the resources as its disk is the only one "UpToDate." To resolve this issue, the two instances must be reconnected.

Procedure

1. From the Primary SMW, attempt to reconnect the instances.

   ```
   smw1# drbdadm connect all
   ```

2. Check the status.

   ```
   smw1# drbdsetup status
   ```

   If the connection status is "Connected," the issue is resolved. Stop now.

   If the connect status remains "Standalone," the secondary disk must be resynchronized. Continue to the next step.

3. From the secondary SMW, use the following command to resynchronize the secondary disk.

```
smw2# drbdadm -- --discard-my-data connect all
```

Note that resynchronizing the secondary disk can take a very long time: up to 12 hours.

4. Monitor the connection status.

```
smw1# drdbsetup status
```

When the connection status is "Connected" and the disk status is "UpToDate", the issue is resolved.