RSF-1 for ZFS Quick Start Guide

For RSF-1 Version 3.12

High-Availability.com Limited
Pentland House,
Village Way,
Wilmslow,
Cheshire,
SK9 2GH,
United Kingdom.

http://www.high-availability.com

Normal hours   +44 (0)844 736 1434
Outside hours   +44 (0)844 736 1974

........................................................................................................................................................................
Table Of Contents

1. What is RSF-1 for ZFS?........................................................................................................................................3
1.1. RSF-1 Communication................................................................................................................................3
1.2. RSF-1 Services ..............................................................................................................................................3
1.3. RSF-1 Service States ....................................................................................................................................4
1.4. RSF-1 Run Modes ..........................................................................................................................................4
1.5. RSF-1 Switchover Modes ..............................................................................................................................5
1.6. RSF-1 Startup and Shutdown Scripts ..........................................................................................................5
2. SYSTEM REQUIREMENTS .............................................................................................................................6
2.1. Operating Systems ........................................................................................................................................6
2.2. Hardware ......................................................................................................................................................6
2.3. Network and Firewalls .................................................................................................................................6
2.4. Storage .........................................................................................................................................................7
2.5. IP Addresses ................................................................................................................................................7
2.6. Required Packages .......................................................................................................................................7
3. GETTING STARTED (Solaris and derivatives) .............................................................................................8
3.1. Getting the RSF-1 Package ..........................................................................................................................8
3.2. Installing the RSF-1 Package ......................................................................................................................8
3.3. RSF-1 Licences ............................................................................................................................................9
3.4. Starting RSF-1 ............................................................................................................................................11
3.5. RSF-1 Processes ........................................................................................................................................11
3.6. RSF-1 Configuration and Management ....................................................................................................12
3.7. RSF-1 User Authentication and Security ................................................................................................12
4. RSF-1 CONFIGURATION ...............................................................................................................................12
4.1. Example Topology ......................................................................................................................................12
4.2. RSF-1 Configuration using rsfadm ............................................................................................................14
4.3. RSF-1 Administration Commands ...........................................................................................................20
5. Starting RSF-1 Services ...............................................................................................................................22
6. RSF-1 Log Files ............................................................................................................................................24
7. Pool Services ................................................................................................................................................27
8. Testing RSF-1 ................................................................................................................................................29
8.2. Faking Heartbeat Failures ..........................................................................................................................31
8.3. Testing for Split-Brain Scenarios ................................................................................................................31
9. Configuring with the GUI ............................................................................................................................33
9.1. Licensing RSF-1 with the GUI ....................................................................................................................34
9.2. RSF-1 Configuration through the GUI .......................................................................................................35
9.3. Creating Services and Adding Volumes ....................................................................................................37
9.4. RSF-1 Status with the GUI ........................................................................................................................39
9.5. Moving Services between Cluster Nodes ................................................................................................39
9.6. Viewing Cluster Heartbeats with the GUI ..................................................................................................41
9.7. Adding Additional Network Heartbeats with the GUI .............................................................................41
1. What is RSF-1 for ZFS?

RSF-1 for ZFS is an Enterprise proven High Availability Cluster product that manages the availability of critical ZFS storage pools. Each RSF-1 Service contains one or more ZFS storage pools with associated file and block services. An RSF-1 ZFS Cluster consists of two or more servers that have any number of RSF-1 Services (pools) configured.

RSF-1 provides high availability of ZFS pools by managing the start-up and failover of RSF-1 Services within an RSF-1 Cluster.

A typical Active/Active 2-node RSF-1 Cluster configuration consists of two RSF-1 Services, each of which have an independent ZFS Pool and a single associated VIP. Under normal operation, each node is responsible for providing services to one ZFS pool, and in the event of either node failing, the surviving node will run both. When the failed node has been repaired and restarted, it will rejoin the cluster and the administrator can control when the ZFS pools are redistributed. To provide optimum uptime, RSF-1 does not automatically failback RSF-1 Services.

1.1. RSF-1 Communication

Each node in the Cluster communicates with all others via a number of Heartbeat mechanisms:

- Network – Using TCP/IP
- Serial – using a back-to-back RS232-C cable
- Disk – using a private physical region on shared disk

NOTE - On ZFS RSF-1 Clusters, it is most common for each node to have two independent network heartbeats (one private back-to-back connection, and a public network) and two or more disk heartbeats. Serial heartbeats are deprecated.

RSF-1 detects system failure when no updates from a node have been seen across all heartbeat mechanisms for a given configurable time period.

1.2. RSF-1 Services

Each RSF-1 Service consists of:

- A list of RSF-1 nodes that are able to run the service
- One or more ZFS pools
Any number of VIPs (or none for Fibre Channel only services)  
A number of disk devices that can be reserved (to ensure data protection)  
A sequence of Start-up / Shut down scripts (that describe how to secure and release physical storage, import and export the ZFS pools, start and file and block services, VIPs etc.)  
Any number of Monitor dependencies (to describe any underlying resource dependency, e.g. a public network interface)  
A number of Timeout parameters (to configure how long nodes should wait before failing over services)

### 1.3. RSF-1 Service States

RSF-1 Services are managed independently from one another and have a number of possible States per node of which the most important are:

<table>
<thead>
<tr>
<th>Service State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting</td>
<td>The Start-up scripts associated with the service are being executed</td>
</tr>
<tr>
<td>Running</td>
<td>The Start-up scripts successfully completed and the service is operating</td>
</tr>
<tr>
<td>Stopping</td>
<td>The Shut down scripts associated with the Service are being executed</td>
</tr>
<tr>
<td>Stopped</td>
<td>The Shut down scripts successfully completed and the service is not running</td>
</tr>
<tr>
<td>broken_safe</td>
<td>A problem occurred while starting the service on the server, but it has been stopped safely and is safe to be started again elsewhere</td>
</tr>
<tr>
<td>broken_unsafe</td>
<td>A fatal problem occurred while starting or stopping the service on the server. The service cannot be run on any other server in the cluster until it has been repaired (by operator intervention)</td>
</tr>
<tr>
<td>panicking</td>
<td>rsfmon has detected the service in an active state on more than one server. It is now running the panic scripts</td>
</tr>
<tr>
<td>panicked</td>
<td>rsfmon has finished running panic scripts after detecting the service in an active state on more than one node. Manual intervention is required to fix this state</td>
</tr>
</tbody>
</table>

### 1.4. RSF-1 Run Modes

Each RSF-1 Service also has an independent Run Mode per node:

<table>
<thead>
<tr>
<th>Run Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocked</td>
<td>The RSF-1 Service is disabled from being started on this node</td>
</tr>
<tr>
<td>Unblocked</td>
<td>The RSF-1 Service is enabled to be started on this node</td>
</tr>
</tbody>
</table>
Run Modes are controlled by other processes within the RSF-1 framework to prevent RSF-1 Service start-up if certain dependencies have not been met (for example, the requirement for a public network interface to be available). The usage of this facility is beyond the scope of this document but further information is available in the RSF-1 Administrator's Guide.

1.5. RSF-1 Switchover Modes

Each RSF-1 Service has an independent Switchover Mode associated per node:

<table>
<thead>
<tr>
<th>Switchover Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>The service will automatically start if not blocked and not already running elsewhere</td>
</tr>
<tr>
<td>Manualk</td>
<td>The service will not automatically start</td>
</tr>
</tbody>
</table>

NOTE - Under normal operating mode, each RSF-1 Service will be in Automatic mode on all cluster nodes. For maintenance purposes, the administrator can prevent automatic failovers or start-ups by changing the Switchover Mode to Manual.

1.6. RSF-1 Startup and Shutdown Scripts

For each RSF-1 service, the following startup / shutdown scripts (located in /opt/HAC/RSF-1/etc/rc.appliance.c/) are executed in ascending order of script file name (in the same method used for init.d):

<table>
<thead>
<tr>
<th>Startup Script</th>
<th>Shutdown Script</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S01announce</td>
<td>K01announce</td>
<td>Write to log file (start of Startup / Shutdown sequence)</td>
</tr>
<tr>
<td>S02ApplianceStarting</td>
<td>K02ApplianceStopping</td>
<td>Event notify for Nexenta</td>
</tr>
<tr>
<td>S14res_drives</td>
<td>K79res_drives</td>
<td>Create res_drives file if it doesn’t exist, or Test the drives listed in res_drives file if it does exist</td>
</tr>
<tr>
<td>S15zfs_mhdc</td>
<td>K85zfs_mhdc</td>
<td>Set / Release SCSI reservations</td>
</tr>
<tr>
<td>S20zfs</td>
<td>K80zfs</td>
<td>Import / Export pool(s) For COMSTAR, restore / backup views</td>
</tr>
<tr>
<td>S21res_drives</td>
<td>K86res_drives</td>
<td>Spawn a background process to</td>
</tr>
</tbody>
</table>
2. SYSTEM REQUIREMENTS

2.1. Operating Systems

The supported Operating Systems are as follows:

<table>
<thead>
<tr>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris</td>
</tr>
<tr>
<td>OpenSolaris / Open Indiana</td>
</tr>
<tr>
<td>Illumos derivatives including: OmniOS, SmartOS, NexentaStor</td>
</tr>
</tbody>
</table>

2.2. Hardware

RSF-1 can be deployed on real or virtual servers. As RSF-1 is a very lightweight process, there are no minimal hardware requirements. It is however expected that the servers to be used have a reasonable amount of memory and CPU power to provide ZFS storage services.

2.3. Network and Firewalls

It is recommended that two separate network ports be used for heartbeats: a private connection (using exclusive back-to-back Ethernet cable) and a public network.

As a minimum, each node in the cluster must be able to utilize port 1195 using both udp and tcp across all network heartbeats.

NOTE - If firewalls are deployed, a rule should be added to allow tcp and udp access via port 1195 between the cluster nodes

<table>
<thead>
<tr>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>refresh res_dives file</td>
</tr>
<tr>
<td>refresh res_dives file</td>
</tr>
<tr>
<td>K98ApplianceStopped</td>
</tr>
<tr>
<td>Event notify for Nexenta</td>
</tr>
<tr>
<td>S99announce</td>
</tr>
<tr>
<td>Write to log file</td>
</tr>
<tr>
<td>Start VIP(s)</td>
</tr>
<tr>
<td>Start all VIPs associated with service</td>
</tr>
</tbody>
</table>
2.4. Storage

Shared storage devices (real or virtual) must be visible to all nodes in the cluster that are to be configured to be capable of running the associated ZFS pools. There is however no requirement for device naming to be identical across all nodes.

Other than iSCSI devices, where virtual storage devices are being used for testing, disk reservations using SCSI-2 or PGR3 are not supported and should be disabled.

2.5. IP Addresses

Each RSF-1 cluster node requires at least one fixed IP address.

Except in the case of Fibre Channel only, at least one spare IP address is also required for each RSF-1 Service, per network that ZFS services will be available.

2.6. Required Packages

In addition to the core RSF-1 product set, you will also need to install the COMSTAR stack, which can be done as follows:

```
# pkg install -v storage-server
```

If you are installing RSF-1 on OmniOS, you will also need to install the iSCSI target framework because it is packaged separately:

```
# pkg install -v iscsi/target
```
3. GETTING STARTED (Solaris and derivatives)

3.1. Getting the RSF-1 Package

You can download the latest RSF-1 Package and documentation via http or anonymous ftp from:

- http://www.high-availability.com/package-downloads
- ftp://ftp.high-availability.com

```bash
# ftp ftp.high-availability.com
Connected to ftp.high-availability.com (213.171.204.157).
220-(<>)=--:. (( Welcome to PureFTPd 1.1.0 )) .:.-(<>)=--
220-You are user number 8 of 50 allowed
220-Local time is now 11:35 and the load is 0.05. Server port: 21.
220 You will be disconnected after 15 minutes of inactivity.
Name (ftp.high-availability.com:root): anonymous
230 Anonymous user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

```
ls
150 Accepted data connection
drwxr-xr-x 4 0 0 4096 Oct 2 11:35 .
drwxr-xr-x 4 0 0 4096 Oct 2 11:35 ..
drwxrxrwvx 2 0 0 4096 Mar 28 2013 Docs
drwxrwrwxv 5 0 0 4096 Apr 19 2013 RSF-1 Gui
-rw-r--r-- 1 0 0 2180703 Sep 23 11:24 rsf-1-3.9.1-7-CentOS-6.4.x86_64.rpm
-rw-r--r-- 1 0 0 27474432 Dec 11 17:47 rsf-1-solaris-5.11-x86.3.9.10.2014-12-04.pkg
```

```
ftp> get rsf-1-solaris-5.11-x86.3.9.10.2014-12-04.pkg
local: rsf-1-solaris-5.11-x86.3.9.10.2014-12-04.pkg remote: rsf-1-solaris-5.11-x86.3.9.10.2014-12-04.pkg
229 Extended Passive mode OK (|||44589|)
150-Accepted data connection
150 26830.5 kbytes to download
100% **************************** 26830 KiB 3.78 MiB/s 00:00 ETA
226-File successfully transferred
226 6.898 seconds (measured here), 3.80 Mbytes per second
27474432 bytes received in 00:06 (3.78 MiB/s)
ftp>
```

Online documentation is also available at:

- http://www.high-availability.com/resources

3.2. Installing the RSF-1 Package

Copy the downloaded package (actual package name will vary with future releases) onto both servers and, as root, install as follows:

```bash
# pkgadd -d rsf-1-solaris-5.11-x86.3.9.10.2014-12-04.pkg
```
Once the pkgadd has completed, the product will be installed in the directory hierarchy /opt/HAC and /opt/HAC/RSF-1, and the final output line from the pkgadd command will advise how to connect to the GUI:

```
install_http_server: generating HTML for HTTP server (standalone)
Server is running
Server is running
    install_http_server: RSF-1 GUI now available on http://romulus:8020/
Installation of <rsf-1> was successful.
```

You can now choose to complete the installation and configuration using the Graphical User Interface (GUI), described in section 9, or the Command Line Interface (CLI) following here.

### 3.3. RSF-1 Licences

The only difference between evaluation and production versions of RSF-1 is the licence string used to activate the product. Evaluation licenses normally have an expiry of 30 days from issue whereas permanent (purchased) licence keys are perpetual. Evaluation licenses can be upgraded to permanent ones without reconfiguration or service disruption.

The licence string is generated from the node’s hostid, which can be determined once the base package has been installed as follows:

```
# /opt/HAC/bin/hac_hostid
390ea5a5
```

**NOTE** - A unique licence is required for each RSF-1 node in the cluster, and therefore the hostid of each server is required.

Evaluation licences may be obtained as follows:

Using the CLI:

Send an email to support@high-availability.com with the following body:
Subject: RSF-1 Evaluation License Request

de1: <node1_hostid>
de2: <node2_hostid>
type: temp
custref: <organisation name, your name>
os: <Operating System and version>

By return, you will receive an email with the license strings required to activate
the product together with: install script, RSF-1 password files and End User Licence Agreement (EULA). By installing the licence keys, you are agreeing to the terms of the EULA. On both nodes, temporarily save the received attachments in
/tmp and install with install_lic.sh

# sh /tmp/install_lic.sh
Installing licenses for cluster nodes with hostid 42029b314 and 49e5e6b0
License successfully installed on this node.
#

To verify the licence strings, use rsfmon -v

# /opt/HAC/RSF-1/bin/rsfmon -v
RSF-1 monitor release 3.9.9 (06 Oct 2014 15:43) for 19:solaris (built on
5.11)
Copyright High-Availability.Com Ltd
[29786 Oct 6 16:56:56] Using machine ID 0x42029b314
This copy of RSF-1 is licensed for up to 128 services.
The licence expires on 2014-12-22
[29786 Oct 6 16:56:56] This host is licenced for automatic service startup
#

NOTE – If RPC services are not already enabled, rsfmon -v will warn as follows, and you should enable and the start the service as shown:

RSF-1: Note: SMF property for network/rpc/bind/local_only is set to true.
RSF-1: This means remote cluster RPC operations will fail!”
RSF-1: Resolve by issuing:
RSF-1: svcfg -s svc:/network/rpc/bind setprop config/local_only = false
RSF-1: svcadm refresh network/rpc/bind:default
RSF-1: To allow remote RPC operations.
# svcfg -s svc:/network/rpc/bind setprop config/local_only = false
# svcadm refresh network/rpc/bind:default
3.4. Starting RSF-1

Once the licenses have been verified, RSF-1 can now be started in bootstrap mode. Run the following on each node:

```
# /opt/HAC/RSF-1/bin/rsfctl start
RSF-1: (1) Registering ZFS sysevent watcher: /opt/HAC/RSF-1/bin/rsf-zfs-event

RSF-1 is now ready for configuration.

3.5. RSF-1 Processes

At this stage, a number of RSF-1 processes should be running:

```
# ps -ef | grep -i rsf
root  1168     1   0 09:54:24 ? 0:00 /opt/HAC/RSF-
1/bin/rsfpmon -v -W 1 -l /var/run/rsfpmon_rphasvc /opt/HAC/RSF-1/
root  1173  1168   0 09:54:24 ? 0:00 /opt/HAC/RSF-
1/bin/rphasvc
root  1186  1185   0 09:54:24 ? 0:02 python /opt/HAC/RSF-
1/bin/rpc_server.pyc
root  1185     1   0 09:54:24 ? 0:00 /opt/HAC/RSF-
1/bin/rsfpmon -v -W 1 -l /var/run/rsfpmon_rpc_py python /opt/HAC/R
root  1207  1202   0 09:54:24 ? 0:01 rsfmon -i
root  1202     1   0 09:54:24 ? 0:01 rsfmon -i
root  1208  1202   0 09:54:24 ? 0:02 rsfmon -i
root  1209  1202   0 09:54:24 ? 0:03 rsfmon -i
root  3290     1   0 09:54:24 ? 0:00 /bin/sh /opt/HAC/RSF-
1/bin/rsf-zpool-cache-sync pool=poola
root  3291     1   0 09:54:54 ? 0:00 /bin/sh /opt/HAC/RSF-
1/bin/rsf-zpool-cache-sync pool=poolb
#```
3.6. RSF-1 Configuration and Management

RSF-1 can be configured and administered via both Graphical User Interface (via a browser) and Command Line Interface rsfcli. Optional additional Application Programmable Interfaces (APIs) are also available for integration with other toolsets.

3.7. RSF-1 User Authentication and Security

RSF-1 uses special user names and passwords to administer the cluster. These are separate to those used by the underlying operating system and must be maintained separately, and are installed as part of the license generation procedure described previously. Further information about managing RSF-1 usernames and passwords is in the RSF-1 Administration Guide.

**NOTE** - It is possible to bypass the need to input the password each time you use the `rsfcli` command using `rsfcli --i0`.

4. RSF-1 CONFIGURATION

4.1. Example Topology

In this example, we have two servers: *romulus* and *remus* and three ZFS pools. The ZFS pools (POOLA, POOLB and POOLC) have all been created on top of the underlying shared storage and both servers have the capability to import and export them.

We are going to use two network heartbeats: one private (shown below as a red dotted line) and one public (shown in orange). We’ll also use six independent disk heartbeats, two per ZFS pool, (shown as black dotted lines).
The system hostnames and identities on the public network are romulus and remus respectively, and romulus_priv and remus_priv respectively on the private network.

Let's assume we want to set up an Active/Active cluster configuration with two RSF-1 Services, the first (which we'll call POOLA) consisting of ZFS pools A and C, and the second (called POOLB) just ZFS pool B.

For service POOLA, we'll associate the VIP sales_staff-public and for service POOLB, we'll use support_staff-public. User access to ZFS pools A and C is therefore via the sales_staff-public VIP address, and access to ZFS pool B via the support_staff-public VIP address.

NOTE - When the POOLA service is failed over between servers, both ZFS pools A and C and the VIP sales_staff-public will migrate as part of that service. When POOLB service is failed over, only pool B is moved together with the support_staff-public VIP.

Before we start to configure RSF-1, make sure that the ZFS pools can be successfully imported and exported between the two servers.
4.2. RSF-1 Configuration using rsfadm

The command `rsfadm` (/opt/HAC/RSF-1/bin/rsfadm) can be used to easily configure the cluster. It makes changes to the cluster using an HTTP API, storing the configuration parameters in a database before writing them to a config file. Because of this, the config file should not be edited manually, since that would cause it to hold different information to the database.

**NOTE** - Ensure that all ZFS pools are imported on one node. In our example, POOLA and POOLC are imported on romulus and POOLB is imported on remus. Also ensure the VIPs are not already in use elsewhere before proceeding with RSF-1 configuration.

Before the cluster can be initialised, rsfmon should be running in bootstrap mode. This can be checked using the `show` subcommand - it should show a cluster of one node, with the cluster name ‘Ready_For_Cluster_Configuration’:

```
root@romulus:~# rsfadm show
Global information:
   Cluster name : Ready_For_Cluster_Configuration
   Poll time    : 2
   Config CRC   : 7f14
Nodes:
   0 : romulus (romulus) available
      RSF-1 release 3.11.0p9, built on 16-Sep-2015-16:13
Services:
   (none)
Heartbeats:
   (none)
root@romulus:~#
```

To make sure RSF-1 can communicate with RSF-1 on the remote node, the discover-nodes subcommand can be used:

```
root@romulus:~# rsfadm discover-nodes
Node 0: remus
root@romulus:~#
```

Now, to create the cluster, the hostnames of the two nodes will need to be provided to the init subcommand:

```
root@romulus:~# rsfadm init romulus remus
Oct 22 15:23:07 romulus RSF-1[907]: [ID 702911 local0.alert] RSF-1 hot restart: services may be running.
root@romulus:~#
```
Now that the cluster is initialised, the 'show' subcommand should show a 2 node cluster with a single network heartbeat between the nodes:

```
rroot@romulus:~# rsfadm show
Global information:
  Cluster name : HA-Cluster
  Poll time    : 1
  Config CRC   : 9e82
Nodes:
  0 : remus (remus) available
      RSF-1 release 3.11.0p9, built on 16-Sep-2015-16:13
  1 : romulus (romulus) available
      RSF-1 release 3.11.0p9, built on 16-Sep-2015-16:13
Services:
  (none)
Heartbeats:
  0: NET    remus --> romulus VIA romulus: Up - last heartbeat #6 (updated Thu Oct 22 15:26:44)
  1: NET    romulus --> remus VIA remus: Up - last heartbeat #5 (updated Thu Oct 22 15:26:44)
2 heartbeats configured, 2 up, 0 down
root@romulus:~#
```

A second network heartbeat can be added (optionally) to increase resiliency. In this example, romulus_priv and remus_priv will be used for the network addresses:

```
rroot@romulus:~# \r
   rsfadm create-hb -t net -d romulus:romulus_priv,remus:remus_priv
Machine: romulus, Interface: romulus_priv
Oct 22 15:29:12 romulus RSF-1[29192]: [ID 702911 local0.alert] RSF-1 hot restart: services may be running.
root@romulus:~#
rroot@romulus:~# rsfadm show heartbeats
  0: NET    remus --> romulus VIA romulus_priv: Up - last heartbeat #12 (updated Thu Oct 22 15:29:25)
  1: NET    romulus --> remus VIA remus_priv: Up - last heartbeat #12 (updated Thu Oct 22 15:29:24)
  2: NET    romulus --> remus VIA remus: Up - last heartbeat #12 (updated Thu Oct 22 15:29:26)
  3: NET    romulus --> remus VIA remus_priv: Up - last heartbeat #12 (updated Thu Oct 22 15:29:26)
4 heartbeats configured, 4 up, 0 down
root@romulus:~#
```
The cluster is now set up with two network heartbeat channels, so we can now add a service to control the zpool POOLA (currently imported on romulus). As in the example in the previous section, the service will use the floating IP address ‘sales_staff-public’ (defined in /etc/hosts on both nodes). Also, the primary server for this service will be set to romulus, and the vip will be plumbed into net1:

```
root@romulus:~# \
rsfadm create-svc POOLA -v sales_staff-public -i net1 -p romulus
Oct 22 16:53:58 romulus RSF-1[29880]: [ID 702911 local0.alert] RSF-1 hot restart: services may be running.
root@romulus:~#
root@romulus:~#
root@romulus:~#
root@romulus:~# rsfadm show
Global information:
  Cluster name : HA-Cluster
  Poll time    : 1
  Config CRC   : 9fbb
Nodes:
  0 : remus (remus) available
      RSF-1 release 3.11.0p9, built on 16-Sep-2015-16:13
  1 : romulus (romulus) available
      RSF-1 release 3.11.0p9, built on 16-Sep-2015-16:13
Services:
  0 : POOLA, IP address sales_staff-public, "RSF-1 cluster service"
     stopped automatic unblocked on remus
     running automatic unblocked on romulus
Heartbeats:
  0: NET    remus ---> romulus VIA romulus_priv: Up - last heartbeat #203
      (updated Thu Oct 22 16:57:22)
  1: NET    remus ---> romulus VIA remulus: Up - last heartbeat #203 (updated
      Thu Oct 22 16:57:23)
  2: DISC   remus ---> romulus VIA
      id1,sd@n60018400000055659e5a0001/a,raw:512,id1,sd@n60018400000055659e5a0001/
      a,raw:512: Up - last heartbeat #203 (updated Thu Oct 22 16:57:23)
  3: DISC   remus ---> romulus VIA
      id1,sd@n60018400000055659e5b0002/a,raw:512,id1,sd@n60018400000055659e5b0002/
      a,raw:512: Up - last heartbeat #203 (updated Thu Oct 22 16:57:23)
  4: NET    romulus ---> remus VIA remus_priv: Up - last heartbeat #205
      (updated Thu Oct 22 16:57:22)
  5: NET    romulus ---> remus VIA remus: Up - last heartbeat #204 (updated
      Thu Oct 22 16:57:22)
  6: DISC   romulus ---> remus VIA
      id1,sd@n60018400000055659e5a0001/a,raw:512,id1,sd@n60018400000055659e5a0001/
      a,raw:512: Up - last heartbeat #204 (updated Thu Oct 22 16:57:22)
  7: DISC   romulus ---> remus VIA
      id1,sd@n60018400000055659e5b0002/a,raw:512,id1,sd@n60018400000055659e5b0002/
      a,raw:512: Up - last heartbeat #204 (updated Thu Oct 22 16:57:22)
8 heartbeats configured, 8 up, 0 down
root@romulus:~#```

Next is to add the second service - POOLT, which will run on remus by default and use floating IP hostname 'support_staff-public', which will also be plumbed into net1:
From the `show` command, you can see that when each service is added, `rsfadm` chooses two disks from the corresponding zpool to use for heartbeats, so now that we have two services, there are 4 disk heartbeats.

Further `rsfadm` commands can be used to add or remove disk heartbeats if necessary.
Finally, the third pool - POOLC - needs to be added to the first service - POOLA - so that POOLA and POOLC fail over together:

```
root@romulus:/rsf-1# rsfadm modify-svc POOLA create-pool POOLC
Oct 22 17:44:23 romulus RSF-1[2090]: [ID 702911 local0.alert] RSF-1 hot restart: services may be running.

root@romulus:/rsf-1# rsfadm show
```

```
Global information:
Cluster name : HA-Cluster
Poll time : 1
Config CRC : 1c47

Nodes:
0 : remus (remus) available
  RSF-1 release 3.11.0p9, built on 16-Sep-2015-16:13
1 : romulus (romulus) available
  RSF-1 release 3.11.0p9, built on 16-Sep-2015-16:13

Services:
0 : POOLA, IP address sales_staff-public, "RSF-1 cluster service"
  stopped automatic unblocked on remus
  running automatic unblocked on romulus
1 : POOLB, IP address support_staff-public, "RSF-1 cluster service"
  stopped automatic unblocked on romulus
  running automatic unblocked on remus

Heartbeats:
0: NET remus --> romulus VIA romulus_priv: Up - last heartbeat #53
  (updated Thu Oct 22 17:45:16)
1: NET remus --> romulus VIA romulus: Up - last heartbeat #53 (updated Thu Oct 22 17:45:17)
2: DISC remus --> romulus VIA
id,sl@n60018400000055659e5b0007/a,raw:518,id,sl@n60018400000055659e5b0007/a,raw:512: Up - last heartbeat #53 (updated Thu Oct 22 17:45:17)
3: DISC remus --> romulus VIA
id,sl@n60018400000055659e5a0001/a,raw:518/id,sl@n60018400000055659e5a0001/a,raw:512: Up - last heartbeat #53 (updated Thu Oct 22 17:45:17)
4: DISC remus --> romulus VIA
id,sl@n60018400000055659e5b0002/a,raw:512/id,sl@n60018400000055659e5b0002/a,raw:518: Up - last heartbeat #53 (updated Thu Oct 22 17:45:17)
5: DISC remus --> romulus VIA
id,sl@n60018400000055659e5b0004/a,raw:512/id,sl@n60018400000055659e5b0004/a,raw:518: Up - last heartbeat #53 (updated Thu Oct 22 17:45:17)
6: DISC remus --> romulus VIA
id,sl@n60018400000055659e5b0005/a,raw:512/id,sl@n60018400000055659e5b0005/a,raw:518: Up - last heartbeat #53 (updated Thu Oct 22 17:45:17)
7: DISC remus --> romulus VIA
id,sl@n60018400000055659e5b0008/a,raw:518/id,sl@n60018400000055659e5b0008/a,raw:512: Up - last heartbeat #53 (updated Thu Oct 22 17:45:17)
```

16 heartbeats configured, 16 up, 0 down
Again from the show command, two more disk heartbeats have been added.

The full list of rsfadm commands can be found by running ‘rsfadm --h’:

```
root@romulus:~# rsfadm --h
rsfadm     CLI for administration of an RSF-1 cluster
Usage:
   rsfadm [OPTIONS] <command> [command options]
Options:
   -v|--verbose
       increase output debug level
   -V|--version
       print version
Subcommands:
   discover-nodes
   init
   destroy
   reset
   show
   create-hb
   delete-hb
   create-svc
   delete-svc
   modify-svc
       create-vip
       delete-vip
       create-pool
       delete-pool
       create-hb
       delete-hb
   control-svc
   help
root@romulus:~#
```

The ‘help’ subcommand can be used to get a summary of all commands, or more information about one command:

```
root@romulus:~# rsfadm help destroy
destroy     Destroy cluster
Usage:
   rsfadm destroy [OPTIONS]
Options:
   -f|--force
       Force destroy the cluster
root@romulus:~#
```
4.3. RSF-1 Administration Commands

When `config_dist` has been used to distribute the config file, all newly configured services will be started on all nodes and set in auto mode. You can use the `rsfcli` to see the various service statuses:

```
romulus# /opt/HAC/RSF-1/bin/rsfcli -v list
romulus:
POOLA running auto unblocked sales_staff-public bge0 20 8
POOLB running auto unblocked support_staff-public bge0 20 8
remus:
POOLA running auto unblocked sales_staff-public bge0 20 8
POOLB running auto unblocked support_staff-public bge0 20 8
```

Heartbeat status can also be checked as follows:

```
romulus# /opt/HAC/RSF-1/bin/rsfcli -v heartbeats
remus : net=2 disc=4 serial=0
romulus net remus
romulus disc /dev/rdsk/c3t20000011C6CBCAD2d0s0:518,
remus disc /dev/rdsk/c3t20000011C6CBCAD2d0s0:518,
romulus disc /dev/rdsk/c3t40000012F5E33CA0Dd0s0:518,
remus disc /dev/rdsk/c3t40000012F5E33CA0Dd0s0:518,
romulus disc /dev/rdsk/c3t40000012F5E38D6EEd0s0:518,
remus disc /dev/rdsk/c3t40000012F5E38D6EEd0s0:518,
romulus disc /dev/rdsk/c3t40000012F5E38D6EEd0s0:518,
remus disc /dev/rdsk/c3t40000012F5E38D6EEd0s0:518,
romulus disc /dev/rdsk/c3t60000016D55EAD32F0d0s0:518,
remus disc /dev/rdsk/c3t60000016D55EAD32F0d0s0:518,
romulus disc /dev/rdsk/c3t60000016D55EAD32F0d0s0:518,
remus disc /dev/rdsk/c3t60000016D55EAD32F0d0s0:518,
```

To see the full service situation including heartbeat status and time-stamps:

```
romulus# /opt/HAC/RSF-1/bin/rsfcli stat
Contacted 127.0.0.1 in cluster "HA-Cluster", CRC = 0x692c, ID = <none>
Host romulus (192.168.33.91) UP, service startups enabled,
  RSF-1 release 3.9.8, built on 01-Oct-2014-10:41 "3.9.8".
Host remus (192.168.33.92) UP, service startups enabled,
  RSF-1 release 3.9.8, built on 01-Oct-2014-10:41 "3.9.8".
2 nodes configured, 2 online.
```
0 Service POOLA, IP Address sales_staff-public, "Sales Staff Pools":
stopped auto unblocked on romulus
running auto unblocked on remus

1 Service POOLB, IP Address support_staff-public, "Support Staff Pools":
running auto unblocked on romulus
stopped auto unblocked on remus

2 services configured
1 service instances stopped
1 service instances running

Heartbeats:

<table>
<thead>
<tr>
<th>Service</th>
<th>Status</th>
<th>Heartbeat ID</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>net</td>
<td>Up</td>
<td>#2138</td>
<td>Thu 2014-10-02 16:58:18 BST</td>
</tr>
<tr>
<td>disc</td>
<td>Up</td>
<td>#2138</td>
<td>Thu 2014-10-02 16:58:18 BST</td>
</tr>
</tbody>
</table>

Errors:
No errors detected

Assuming there are no configuration issues, all newly added services should be running. If however there are any errors reported, you can view the RSF-1 text log file at /opt/HAC/RSF-1/log/rsfmon.log for further information, errors or warnings.

21
5. Starting RSF-1 Services

If services are set to stopped and manual on the cluster nodes, they can be started by simply putting the service into Automatic mode on each node.

NOTE - It is advisable to put the primary node mode to Automatic before doing this on the other node, so for example, put POOLA service in Automatic mode on romulus, and POOLB service in Automatic mode on remus.

```bash
romulus# /opt/HAC/RSF-1/bin/rsfcli -i0 auto POOLA
remus# /opt/HAC/RSF-1/bin/rsfcli -i0 auto POOLB
```

rsfcli can again be used to verify the services are now running:

```
romulus# /opt/HAC/RSF-1/bin/rsfcli -v list
romulus:
  POOLA running auto unblocked sales_staff-public bge0 20 8
  POOLB stopped manual unblocked support_staff-public bge0 20 8
remus:
  POOLA stopped manual unblocked sales_staff-public bge0 20 8
  POOLB running auto unblocked support_staff-public bge0 20 8
romulus#
```

It is now safe to put all services in automatic mode across the cluster:

```
romulus# /opt/HAC/RSF-1/bin/rsfcli -i0 auto POOLB
remus# /opt/HAC/RSF-1/bin/rsfcli -i0 auto staff_staff
```

```
romulus# /opt/HAC/RSF-1/bin/rsfcli -v list
romulus:
  POOLA running auto unblocked sales_staff-public bge0 20 8
  POOLB stopped auto unblocked support_staff-public bge0 20 8
remus:
  POOLA stopped auto unblocked sales_staff-public bge0 20 8
  POOLB running auto unblocked support_staff-public bge0 20 8
romulus#
```
If either of the services are not shown as running, view the RSF-1 log files to look for reasons. If a service is shown as broken_safe, it can be repaired and reset to automatic as follows:

```
romulus# /opt/HAC/RSF-1/bin/rsfcli -i0 repair POOLA
romulus# /opt/HAC/RSF-1/bin/rsfcli -i0 auto POOLA
```

If however a service is showing as broken_unsafe, further investigation is required before a retry should be attempted. Please refer to the RSF-1 Administrator’s Guide for further information and detail.

If the services are successfully running, we are now ready to run some failure and failover tests.
6. RSF-1 Log Files

All RSF-1 log files are located in /opt/HAC/RSF-1/log, and are readable text files written sequentially in ascending date order. A new log file is created when RSF-1 is started, and will remain current until RSF-1 is stopped.

For troubleshooting purposes, examining the log files on both cluster nodes, is required.

Example log file:

```
[28430 Oct  6 16:46:24] ----------------- RSF-1 starting -----------------
[28430 Oct  6 16:46:24] Using machine ID 0x42029b314
[28430 Oct  6 16:46:24] INFO: This copy of RSF-1 expires on 2014-12-22
[28430 Oct  6 16:46:24] INFO: This copy of RSF-1 is licenced for automatic service startup
[28430 Oct  6 16:46:24] Configuration file parsed OK, CRC is 0xe10d
[28430 Oct  6 16:46:24] Running as machine nextest1 on host nextest1
(5436EH9AD)
[28541 Oct  6 16:46:24] Running at realtime RR scheduling priority 1
[28541 Oct  6 16:46:24] NOTICE: Service tank1 is now stopped on nextest1 (was unknown)
[28541 Oct  6 16:46:24] Socket state change. UDP port 1195 was down, is up
[28541 Oct  6 16:46:24] Socket state change. TCP port 1195 was down, is up
[28541 Oct  6 16:46:24] Starting main heartbeat loop
[28541 Oct  6 16:46:24] network interface monitor process (re)started, pid = 28542
[28541 Oct  6 16:46:24] event notification process (re)started, pid = 28543
[28541 Oct  6 16:46:24] disc heartbeat process (re)started, pid = 28544
machine=nextest1 state=start
service=tank1 state=stopped mode=manual block=unblocked
heartbeat=3 type=net from=nextest2 state=Unavailable
nextest2.tank1 unknown/manual/unblocked -> stopped/manual/unblocked
heartbeat=3 type=net from=nextest2 state=Up latest=1412610385
remote=nextest2 state=Up
[28541 Oct  6 16:46:26] bge0 network interface state change: unknown -> unblocked
netdevice=bge0 state=OK
[28541 Oct  6 16:46:29] NOTICE: disc heartbeat (4, seq 40) from nextest2 OK
```
Whilst the above shows a typical startup sequence, log-file analysis is more helpful when diagnosing or debugging specific service startup and shutdown sequences. The following is the initial startup of a service tank1:

```
[28334 Nov 18 15:16:53] NOTICE: Service tank1 not running, start in 6 seconds
[28334 Nov 18 15:16:59] Time to start tank1
[28334 Nov 18 15:16:59] Running start scripts for nextest2.tank1, pid = 16528
[28334 Nov 18 15:16:59] NOTICE: Service tank1 is now starting on nextest2 (was stopped)
[28334 Nov 18 15:17:00] INFO: event-notify: LOG_INFO RSF_SERVICE service=tank1 state=starting mode=automatic block=unblocked
[28334 Nov 18 15:17:00] Service tank1 is now starting on nextest2 (was stopped on nextest1)
[16915 Nov 18 15:17:01] User _rsfadmin setting service tank1 on nextest2 to manual
[16528 Nov 18 15:17:02] [tank1 rsfexec] Running /opt/HAC/RSF-1/etc/rc.appliance.c/S01announce start 1

```

[16960 Nov 18 15:17:02] [tank1 S02ApplianceStarting] ========= ifconfig =========

```
[16528 Nov 18 15:17:09] [tank1 rsfexec] =========      ifconfig/netstat complete        =========
[16528 Nov 18 15:17:09] [tank1 rsfexec] Total run time for service start: 7 seconds
[17577 Nov 18 15:17:09] [tank1 S21res_drives] Service running on this node - refreshing /opt/HAC/RSF-1/etc/.res_drives.tank1...
[28334 Nov 18 15:17:10] Script child 16528 exited with code 0
[28334 Nov 18 15:17:10] Service tank1 start scripts succeeded (pid 16528 exit 0)
[28334 Nov 18 15:17:10] NOTICE: Service tank1 is now running on nextest2 (was starting)

```

[Nov 18 15:17:12] [tank1 disk_list] Created /opt/HAC/RSF-1/etc/.res_drives.tank1.new with 9 drives
```

[28334 Nov 18 15:17:14] INFO: event-notify: ZFS_INFO IMPORT_COMPLETE machine=nextest2 service=tank1
The following log-file segment shows an example startup error where the service goes broken_safe during startup due to a disk reservation issue:

```
[Nov 11 19:06:15] [cvol disk_list] Unknown disk
[2376 Nov 11 19:06:15] [cvol S14res_drives] All reservation drives are either missing or return I/O error!
[2376 Nov 11 19:06:15] [cvol S14res_drives] Aborting because of PROP_ABORT_UNFENCED_IMPORT: true
[1288 Nov 11 19:06:16] Service cvol start scripts set broken safe state (pid 1427 exit 4)
[1288 Nov 11 19:06:16] NOTICE: Service cvol is now broken_safe on nextest2 (was starting)
```
7. Pool Services

ZFS zvols can be used to create shares using NFS, CIFS/SMB, iSCSI or Fibre Channel.

The following shows how to create a simple share with both NFS and SMB:

```
# zfs create POOLA/voll
# zfs set sharesmb=on POOLA/voll
# zfs set sharennfs=on POOLA/voll
```

As the ZFS properties are being used to create the shares, nothing else needs to be done to enable failover of the shares.

The following shows how to create a COMSTAR LU and expose it to clients using iSCSI. Note that RSF-1 handles failover of the LU and views, but you will need to ensure any host groups, target groups, target portal groups and targets are identical on both nodes. The groups are optional depending on your needs, but if used, they must be identical on both nodes. Assume POOLA service is running on romulus, stopped on remus:

**NOTE** – The following commands need to be executed on both nodes (as shown).

Create TPG for VIP (sales_staff-public = 192.168.33.105):

```
root@romulus:~# itadm create-tpg poola-tpg 192.168.33.105
root@remus:~# itadm create-tpg poola-tpg 192.168.33.105
```

Create target using TPG:

```
root@romulus:~# itadm create-target -t poola-tpg
Target iqn.1986-03.com.sun:02:af94b9da-6905-e22a-f5ed-981dbd97c5c6 successfully created

root@remus:~# itadm create-target -t poola-tpg -n iqn.1986-03.com.sun:02:af94b9da-6905-e22a-f5ed-981dbd97c5c6
Target iqn.1986-03.com.sun:02:af94b9da-6905-e22a-f5ed-981dbd97c5c6 successfully created
```

**NOTE** – The following commands manipulate the ZFS pools and so only need to be executed on the node that has the pool imported.
Create ZFS volume to be shared:

```bash
root@romulus:~# zfs create -V 1G POOLA/vol1
```

Create LU:

```bash
root@romulus:~# stmadm create-lu /dev/zvol/rdsk/POOLA/vol01
Logical unit created: 600144F08C1BC4000000546F44110001
```

Create view:

```bash
root@romulus:~# stmadm add-view 600144F08C1BC4000000546F44110001
root@romulus:~# stmadm list-view -l 600144F08C1BC4000000546F44110001
View Entry: 0
Host group : All
Target group : All
LUN : 0
```

Back up view information for RSF-1 failover:

```bash
root@romulus:~# stmha backup POOLA
root@romulus:~# ls -l /POOLA/.mapping/
total 3
-rw------- 1 root root 44 2014-11-21 13:59 @@RAN475204875@@POOLA@-vol01
-rw-r--r-- 1 root root 0 2014-11-21 13:59 timeOfCreation68366902028957
root@romulus:~# cat /POOLA/.mapping/@@RAN475204875@@POOLA@-vol01
BeginMap
index=0
TG=All
HG=All
LUN=0
EndMap
```

**NOTE** - Any clients should always connect to the VIP address for the HA service, rather than the fixed IP address of either of the servers.
8. Testing RSF-1

The following tests show how RSF-1 can be used to manually move services around the cluster and perform automatic failovers in the event of failures. The user may find it useful to open two terminal sessions per node whilst undertaking these tests; one for monitoring the RSF-1 log files, and the other to issue the various commands.

NOTE - When connecting to cluster nodes, use the fixed IP addresses, not the service VIPs as these sessions will hang or exit during failover tests

The recommended way to monitor log files is as follows (on each cluster node):

```bash
romulus# tail -f /opt/HAC/RSF-1/log/rsfmon.log
[9555 Oct 3 09:15:02] Process mlocked in memory
[9556 Oct 3 09:15:02] Process mlocked in memory
[9552 Oct 3 09:15:03] INFO: event-notify: LOG_INFO RSF_DAEMON
machine=romulus state=start
[9552 Oct 3 09:15:03] INFO: event-notify: LOG_INFO RSF_SERVICE
service=POOLA state=stopped mode=manual block=unblocked
[9552 Oct 3 09:15:03] INFO: event-notify: LOG_INFO RSF_SERVICE
service=POOLB state=stopped mode=manual block=unblocked
[9552 Oct 3 09:15:03] NOTICE: net heartbeat (7, seq 3) from remus OK
[9552 Oct 3 09:15:03] CRIT: Established contact with remus
stopped/manual/unblocked ->
romulus.POOLA unknown/manual/unblocked ->
stopped/manual/unblocked
[9552 Oct 3 09:15:03] NOTICE: Service POOLA not running, is manual/unblocked, not starting it.
[9552 Oct 3 09:15:03] NOTICE: Service POOLB not running, is manual/unblocked, not starting it.
```

8.1. Manual Failover Testing

To move a service from one node to another, the `rsfcli` command can be used in two ways:

Using `rsfcli` move:

```bash
romulus# /opt/HAC/RSF-1/bin/rsfcli -i0 move POOLA remus
```

This causes the following to happen:

<table>
<thead>
<tr>
<th>Step</th>
<th>on romulus:</th>
<th>on remus:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Disable the `sales_staff-public VIP`
2. Run the POOLA shutdown scripts
3. Place switchover mode for POOLA to manual
4. Instruct `remus` to immediately start the service
5. Check the service is not already running elsewhere
6. Run the POOLA startup scripts
7. Enable the `sales_staff-public VIP`

Or using `rsfcli stop`:

```
romulus# /opt/HAC/RSF-1/bin/rsfcli -i0 stop POOLA
```

This alternate method simply tells `romulus` to stop the service (running through steps 1-3 above). If `remus` has switchover mode for POOLA set to automatic, as soon as it sees the service is not running anywhere (and cannot run anywhere else), it will start the service after its `RUNTIMEOUT` countdown timer has expired.

Once a service has been moved elsewhere and is running successfully, the switchover mode of the previous node should be reset to `automatic`:

```
romulus# /opt/HAC/RSF-1/bin/rsfcli -i0 auto POOLA
```

The above procedures can be used to manually move RSF-1 services around the cluster. Other than for initial testing purposes, these procedures can be used for planned maintenance situations, load redistribution etc.

### Automatic Failover Testing

Automatic failover testing is best achieved by faking a system crash of one of the nodes that is running one or more services using one of the following methods:

- `halt`
- `init 0`
- `uadmin 2 1`
As soon as communication via all heartbeat channels ceases, the surviving node, assuming switchover modes are set to automatic, will initiate countdown and failover of those services lost by the crash.

8.2. Faking Heartbeat Failures

In order to trigger an automatic failover situation, all heartbeats must fail. Individual heartbeat mechanisms can be tested by temporarily disconnecting cables (e.g. back-to-back network), or by temporarily unplumbing network interfaces.

8.3. Testing for Split-Brain Scenarios

A split-brain situation occurs when both nodes believe the other has gone away, and both try to start and manage the storage service. RSF-1 uses disk ring-fencing mechanisms to prevent this and force one or both servers to panic in order to protect the pool and prevent data corruption.

Assuming romulus is currently running the POOLA service, and remus has its switchover mode for the staff_sales service set to automatic, the easiest way to test this is to hang romulus using the use mdb command. This causes the Operating System to lock up until resumed by the operator.

```
romulus# mdb -K
```

remus will detect this failure and will initiate failover of the POOLA. Once this has completed, hung romulus can be reawakened:

```
::quit -u
romulus#
```
As romulus resumes, and assumes it is still running the POOLA service, attempts to access the underlying disks (now reserved by remus) will cause the ring-fencing mechanism to kick in and immediately cause romulus to panic and safely reboot.

```
root@romulus:~# panic[cpu1]/thread=ffffffffc80b5ac20: Reservation Conflict
Disk: /ethdrv/sd@1,0

ffffffffc80b5aa80 fffffffff797526d ()
ffffffffc80b5ae0 sd:sd_mhd_watch_cb+b6 ()
ffffffffc80b5ab30 scsi:scsi_watch_request_intr+144 ()
ffffffffc80b5ab60 ethdrv:ethdrv_retire+c5 ()
ffffffffc80b5ac00 genunix:taskq_thread+22e ()
ffffffffc80b5ac10 unix:thread_start+8 ()

syncing file systems... done
dumping to /dev/zvol/dsk/rpool/dump, offset 65536, content: kernel
```

When romulus restarts, it will rejoin the cluster and will see all services are running on remus and will remain in stopped automatic mode for all services.
9. Configuring with the GUI

To connect to the RSF-1 GUI, direct your browser to hostname:port as shown in the output of the pkgadd installation procedure as shown in section 3.2.

The first stage is to tell the GUI configurator the name of the other node to be added to the cluster. Before the above dropdown menu appears, the GUI scans the local network for other servers running RSF-1 in cluster-discovery mode and populates the dropdown menu with the name of any host it finds.

NOTE – A node cannot enter RSF-1 cluster-discovery mode until it has been licensed and therefore it will be necessary to enter the other cluster node name manually in the box provided.

Once the other node name has been entered, you will be asked to confirm selection before the configuration process starts.
Once confirm is clicked, the first stage process will quickly complete.

9.1. Licensing RSF-1 with the GUI

At this point, it is necessary to license the product on both nodes. If you have already been provided with a temporary or permanent license, this can be entered manually by selecting the manual option; alternatively, a temporary evaluation license can be obtained by clicking Auto.

In either case, the RSF-1 End User License Agreement (EULA) must be accepted before proceeding.

Once the terms and conditions have been accepted, enter a valid email address to receive the temporary licenses.
To manually install the license, select the Manual option and enter the codes exactly as provided in the respective fields and click confirm. This will install licenses on both cluster nodes and only needs to be performed once.

Once the license keys have been successfully installed, RSF-1 will automatically restart and both cluster nodes can be configured.

9.2. RSF-1 Configuration through the GUI

Once RSF-1 has automatically restarted, the cluster is ready to be configured.
On Romulus, three ZFS pools, POOLA, POOLB and POOLC, have already been created and imported using the `zpool create` command. To begin configuration, click the Configure button.

Once initialization has completed, you can begin to configure services and add ZFS volumes.
9.3. Creating Services and Adding Volumes

Click the Volumes tab on the left-hand column to go to the Cluster Volume Operations page, and select a volume to cluster from the pull down list as shown:

![ROMULUS: HA CLUSTER VOLUME OPERATIONS](image)

In this example, we are going to configure a service for POOLA

![Configure volume service](image)

Once selected, complete the fields for IPV4 Failover hostname and click Confirm. In the following example, a second service, POOLB, is configured with the associated VIP support-staff-public.
In this example, we also want to add a second pool, POOLC, to the POOLA service. Select the Advanced Operations tab on the left column to access the Advanced Operations page, and then click on the Add/Remove Volumes tab. You can now drag-and-drop the additional pool, POOLC, on to the service meaning that both POOLA and POOLC are associated with the same RSF-1 service and will move together through the cluster.

Click Confirm to complete this configuration.
9.4. RSF-1 Status with the GUI

To view the current cluster status, click on the Status tab on the left-hand column to access the Cluster Status page.

This screen shows the current location of each service and the respective Volume states and Failover modes and allows the operator to stop, start and move services throughout the cluster. Right-click the mouse over each of the pools to see what operations are possible. The following screenshot shows the operations available on remus for POOLB.

9.5. Moving Services between Cluster Nodes

In this example, we want to move the service POOLB from remus to romulus.
During this process, POOLB will be cleanly stopped on remus and then restarted on romulus.

Once this move has completed, romulus is running POOLB and remus POOLA with POOLC.
9.6. Viewing Cluster Heartbeats with the GUI

In addition to the cluster status page, you can also view the cluster heartbeats page by clicking the Heartbeats tab on the left column. The first tab, Volume Heartbeats, shows the disk heartbeats.

The second tab, Appliance Heartbeats, shows the network heartbeats.

9.7. Adding Additional Network Heartbeats with the GUI

We can also add additional network heartbeats on this page. In the worked example, we also have a private network connection between the two servers named romulus-priv and remus-priv respectively. To add network
heartbeats using this private interface, right-click over the server name to see the option:

In this example, we are going to add a heartbeat connection between remus and romulus-priv:

and between romulus and remus-priv:
The status now shows two separate network heartbeats connecting both nodes:

Click **Save Settings** to confirm the addition.
The basic cluster is now configured. For additional File/Block service configuration, please refer to section 7, and to section 8 for failover testing.

Please also consult the RSF-1 Administrators Guide available from http://www.high-availability.com/resources/