

Fujitsu Software PRIMECLUSTER

Cluster Foundation Configuration and Administration Guide 4.7

Linux

J2UL-2823-01ENZ0(00)
February 2023

Preface

The Cluster Foundation (CF) provides a comprehensive base of services that user applications and other PRIMECLUSTER services need to administrate and communicate in a cluster.

Target Readers

This manual is intended for all users who use PRIMECLUSTER 4.7 and perform cluster system installation and operation management.

Configuration of This Documentation

This manual is organized as follows:

Chapter	Description
Chapter 1 Cluster Foundation	Describes the administration and configuration of the Cluster Foundation.
Chapter 2 CF Registry and Integrity Monitor	Discusses purpose and physical characteristics of the CF synchronized registry, and it discusses the purpose and implementation of the Cluster Integrity Monitor (CIM).
Chapter 3 Cluster resource management	Discusses the database which is a synchronized clusterwide database holding information specific to several PRIMECLUSTER products.
Chapter 4 GUI administration	Describes the administration features in the CF portion of the Cluster Admin graphical user interface (GUI).
Chapter 5 LEFTCLUSTER state	Discusses the LEFTCLUSTER state, relation between LEFTCLUSTER state and other states. How LEFTCLUSTER state is caused in various ways is also described.
Chapter 6 CF topology table	Discusses the layout and the use of the CF topology table.
Chapter 7 Shutdown Facility (SF)	Describes the components and advantages of PRIMECLUSTER SF and provides administration information.
Chapter 8 Diagnostics and troubleshooting	Provides help for troubleshooting and problem resolution for PRIMECLUSTER Cluster Foundation.
Chapter 9 CF over IP	Provides the overview of CF over IP and how CF over IP is configured.

Related Documentation

Refer to the following manuals as necessary when setting up the cluster:

- PRIMECLUSTER Concepts Guide
- PRIMECLUSTER Installation and Administration Guide
- PRIMECLUSTER Installation and Administration Guide Cloud Services
- PRIMECLUSTER Web-Based Admin View Operation Guide
- PRIMECLUSTER Reliant Monitor Services (RMS) with Wizard Tools Configuration and Administration Guide
- PRIMECLUSTER Global Disk Services Configuration and Administration Guide
- PRIMECLUSTER Global File Services Configuration and Administration Guide
- PRIMECLUSTER Global Link Services Configuration and Administration Guide: Redundant Line Control Function
- PRIMECLUSTER Messages
- FJQSS (Information Collection Tool) User's Guide



Note

The PRIMECLUSTER documentation includes the following documentation in addition to those listed above:

- PRIMECLUSTER Software Release Guide and Installation Guide

This Software Release Guide and Installation Guide are provided with each PRIMECLUSTER product package.

The data is stored on "DVD" of each package. For details on the file names, see the documentation.

Manual Printing

If you want to print a manual, use the PDF file found on the DVD for the PRIMECLUSTER product. The correspondences between the PDF file names and manuals are described in the Software Release Guide for PRIMECLUSTER that comes with the product.

Adobe Reader is required to read and print this PDF file. To get Adobe Reader, see Adobe Systems Incorporated's website.

Conventions

Notation

Prompts

Command line examples that require system administrator (or root) privileges to execute are preceded by the system administrator prompt, the hash sign (#). Entries that do not require system administrator rights are preceded by a dollar sign (\$).

In some examples, the notation `node#` indicates a root prompt on the specified node. For example, a command preceded by `shasta2#` would mean that the command was run as user root on the node named shasta2.

The keyboard

Keystrokes that represent nonprintable characters are displayed as key icons such as [Enter] or [F1]. For example, [Enter] means press the key labeled Enter; [Ctrl-b] means hold down the key labeled Ctrl or Control and then press the [B] key.

Typefaces

The following typefaces highlight specific elements in this manual.

Typeface / Symbol	Usage
Constant Width	Computer output and program listings; commands, file names, manual page names and other literal programming elements in the main body of text.
<i>Italic</i>	Variables in a command line that you must replace with an actual value. May be enclosed in angle brackets to emphasize the difference from adjacent text; for example, <code><nodename>RMS</code> ; unless directed otherwise, you should not enter the angle brackets. The name of an item in a character-based or graphical user interface. This may refer to a menu item, a radio button, a checkbox, a text input box, a panel, or a window title.
Bold	Items in a command line that you must type exactly as shown.

Example 1

Several entries from an `/etc/passwd` file are shown below:

```
sysadm:x:0:0:System Admin.:/usr/admin:/usr/sbin/sysadm
setup:x:0:0:System Setup:/usr/admin:/usr/sbin/setup
daemon:x:1:1:0000-Admin(0000):/:
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:daemon:/sbin:/bin/bash
lp:x:4:7:lp daemon:/var/spool/lpd:/bin/bash
```

Example 2

To use the `cat` command to display the contents of a file, enter the following command line:

```
$ cat file
```

Command syntax

The command syntax observes the following conventions.

Symbol	Name	Meaning
[]	Brackets	Enclose an optional item.
{ }	Braces	Enclose two or more items of which only one is used. The items are separated from each other by a vertical bar ().
	Vertical bar	When enclosed in braces, it separates items of which only one is used. When not enclosed in braces, it is a literal element indicating that the output of one program is piped to the input of another.
()	Parentheses	Enclose items that must be grouped together when repeated.
...	Ellipsis	Signifies an item that may be repeated. If a group of items can be repeated, the group is enclosed in parentheses.

Notation symbols

Material of particular interest is preceded by the following symbols in this manual:



.....
Contains important information about the subject at hand.
.....



.....
Describes an item to be noted.
.....



.....
Describes operation using an example.
.....



.....
Describes reference information.
.....



.....
Provides the names of manuals to be referenced.
.....

Abbreviations

Red Hat OpenStack Platform is abbreviated as RHOSP.

PRIMEQUEST 3000 Series are abbreviated as PRIMEQUEST.

FUJITSU Hybrid IT Service FJcloud-O is abbreviated as FJcloud-O.

FUJITSU Hybrid IT Service FJcloud-V is abbreviated as FJcloud-V.

FUJITSU Hybrid IT Service FJcloud-Baremetal is abbreviated as FJcloud-Baremetal.

FUJITSU Hybrid IT Service for Microsoft Azure is abbreviated as "for Azure."

FUJITSU Hybrid IT Service for AWS is abbreviated as "for AWS."

FJcloud-V sold by FUJITSU LIMITED and NIFCLOUD sold by FUJITSU CLOUD TECHNOLOGIES LIMITED are abbreviated as "NIFCLOUD" in this manual.

"for Azure" sold by FUJITSU LIMITED and Microsoft Azure sold by Microsoft Corporation in the United States are abbreviated as "Azure" in this manual.

"for AWS" sold by FUJITSU LIMITED and AWS (Amazon Web Services) sold by Amazon.com, Inc. are abbreviated as "AWS" in this manual.

Export Controls

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Trademarks

UNIX is a registered trademark of The Open Group in the United States and other countries.

Red Hat and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the U.S. and other countries.

Linux(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions.

Amazon Web Services is a registered trademark of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

Microsoft and Azure are trademarks of the Microsoft group of companies.

All other hardware and software names used are trademarks of their respective companies.

Requests

- No part of this documentation may be reproduced or copied without permission of FUJITSU LIMITED.
- The contents of this documentation may be revised without prior notice.

Date of publication and edition

February 2023, First edition

Copyright notice

Copyright Fujitsu Limited 2023.

Contents

Chapter 1 Cluster Foundation.....	1
1.1 CF, CIP, and CIM configuration.....	1
1.1.1 Differences between CIP and CF over IP.....	3
1.1.2 cfset.....	5
1.1.3 CF security.....	7
1.1.3.1 cfcp/cfsh.....	7
1.1.3.2 sshconf.....	7
1.1.4 Example of creating a cluster.....	8
1.1.5 Adding a new node to CF.....	30
1.1.6 Example of CF configuration by CLI.....	31
1.2 CIP configuration file.....	34
Chapter 2 CF Registry and Integrity Monitor.....	35
2.1 CF Registry (CFREG).....	35
2.2 Cluster Integrity Monitor(CIM).....	35
2.2.1 Configuring CIM.....	35
2.2.2 Query of the quorum state.....	36
2.2.3 Reconfiguring quorum.....	36
Chapter 3 Cluster resource management.....	38
3.1 Resource Database configuration.....	38
3.2 Startup synchronization.....	39
3.2.1 Startup synchronization and the new node.....	40
Chapter 4 GUI administration.....	41
4.1 Starting Cluster Admin GUI and logging in.....	41
4.2 Main CF table.....	43
4.3 CF route tracking.....	44
4.4 Node details.....	46
4.5 Displaying the topology table.....	47
4.6 Starting and stopping CF.....	50
4.6.1 Starting CF.....	52
4.6.2 Stopping CF.....	53
4.7 Marking nodes DOWN.....	55
4.8 Using PRIMECLUSTER log viewer.....	55
4.8.1 Search based on time filter.....	56
4.8.2 Search based on keyword.....	56
4.8.3 Search based on severity levels.....	57
4.9 Displaying statistics.....	57
4.10 Heartbeat monitor.....	59
4.11 Adding and removing a node from CIM.....	60
4.12 Unconfigure CF.....	61
4.13 CIM Override.....	62
Chapter 5 LEFTCLUSTER state.....	64
5.1 Description of the LEFTCLUSTER state.....	64
5.2 Recovering from LEFTCLUSTER.....	65
5.2.1 Caused by a panic/hung node.....	66
5.2.2 Caused by staying in the kernel debugger too long.....	66
5.2.3 Caused by a cluster partition.....	67
Chapter 6 CF topology table.....	68
6.1 Basic layout.....	68
6.2 Selecting devices.....	69
6.3 Examples of topology table.....	69
Chapter 7 Shutdown Facility.....	72

7.1 Overview.....	72
7.2 Available SAs.....	73
7.2.1 IPMI.....	73
7.2.2 kdump.....	74
7.2.3 iRMC.....	74
7.2.4 ICMP.....	74
7.2.5 VMCHKHOST.....	74
7.2.6 libvirt.....	75
7.2.7 VMware vCenter Server Functional Cooperation.....	75
7.2.8 FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API.....	75
7.2.9 OpenStack API.....	75
7.2.10 AWS CLI.....	75
7.2.11 Azure CLI.....	75
7.2.12 NIFCLOUD API.....	76
7.3 SF split-brain handling.....	76
7.3.1 Administrative LAN.....	76
7.3.2 SF split-brain handling.....	76
7.3.2.1 RMS ShutdownPriority attribute.....	76
7.3.2.2 Shutdown Facility weight assignment.....	76
7.3.3 Runtime processing.....	77
7.3.4 Configuration notes.....	77
7.4 Configuring the Shutdown Facility.....	78
7.4.1 Shutdown Daemon.....	78
7.4.2 Shutdown Agents.....	79
7.5 SF administration.....	88
7.5.1 Starting and stopping SF.....	88
7.5.1.1 Starting and stopping SF manually.....	88
7.5.1.2 Starting and stopping SF automatically.....	89
7.5.2 Checking SA status.....	89
7.6 Debugging.....	90
Chapter 8 Diagnostics and troubleshooting.....	92
8.1 Beginning the process.....	92
8.2 Symptoms and solutions.....	94
8.2.1 Join-related problems.....	94
8.2.1.1 Identifying join-related problems.....	94
8.2.1.2 Solving join-related problems.....	94
8.3 Collecting Troubleshooting Information.....	98
Chapter 9 CF over IP.....	99
9.1 Overview.....	99
9.2 Configuring CF over IP.....	100
Glossary.....	102
Index.....	114

Chapter 1 Cluster Foundation

This chapter describes the administration and configuration of the Cluster Foundation (CF).

1.1 CF, CIP, and CIM configuration

You must configure CF before any other cluster services, such as Reliant Monitor Services (RMS). CF defines which nodes are in a given cluster. In addition, after you configure CF and CIP, the Shutdown Facility (SF) and RMS can be run on the nodes.

The Shutdown Facility (SF) is responsible for node elimination. This means that even if RMS is not installed or running in the cluster, missing CF heartbeats will cause SF to eliminate nodes.

You can use the Cluster Admin CF Wizard to easily configure CF, CIP, and CIM for all the nodes in the cluster.

A CF configuration consists of the following main attributes:

- Cluster name - This can be any name that you choose as long as it is 31 characters or less per name and each character comes from the set of printable ASCII characters, excluding white space, newline, and tab characters. Cluster names are always mapped to upper case.
- Interconnects - Set of interfaces on each node in the cluster used for CF networking. An Ethernet device on the local node is one example of an interface.
- CF node name - By default, in Cluster Admin, the CF node names are the same as the Web-Based Admin View names; however, you can use the CF Wizard to change them. CF node names are converted to lower case.

The dedicated network connections used by CF are known as interconnects. They typically consist of some form of high speed networking such as 100 MB or Gigabit Ethernet links. These interconnects must meet the following requirements if they are to be used for CF:

- The network links used for interconnects must have low latency and low error rates. This is required by the CF protocol. Private switches and hubs will meet this requirement. Public networks, bridges, and switches shared with other devices may not necessarily meet these requirements, and their use is not recommended.

It is recommended that each CF interface be connected to its own private network with each interconnect on its own switch or hub.

- The interconnects should not be used on any network that might experience network outages of 5 seconds or more. A network outage of 10 seconds will, by default, cause a route to be marked as DOWN. *cfset(1M)* can be used to change the 10 second default. Refer to the section "[1.1.2 cfset](#)" for additional information.

Since CF automatically attempts to bring up downed interconnects, the problem with split clusters only occurs if all interconnects experience a 10-second outage simultaneously. Nevertheless, CF expects highly reliable interconnects.

CF can also be run over IP. Any IP interface on the node can be chosen as an IP device, and CF will treat this device much as it does an Ethernet device. However, all the IP addresses for all the cluster nodes on that interconnect must be on the same IP subnetwork, and their IP broadcast addresses must be the same (see "[Chapter 9 CF over IP](#)" for more information).

The IP interfaces used by CF must be completely configured by the System Administrator before they are used by CF. You can run CF over both Ethernet devices and IP devices. Higher level services, such as RMS, SF, Global File Services (hereinafter GFS), and so forth, will not notice any difference when CF is run over IP.

You should carefully choose the number of interconnects you want in the cluster before you start the configuration process. If you decide to change the number of interconnects after you have configured CF across the cluster, you can either bring down CF on each node to do the reconfiguration. Bringing down CF requires that higher level services, like RMS, SF, and applications, be stopped on that node, so the reconfiguration process is neither trivial nor unobtrusive.



Note

Your configuration should specify at least two interconnects to avoid a single point of failure in the cluster.

Before you begin the CF configuration process, ensure that all of the nodes are connected to the interconnects you have chosen and that all of the nodes can communicate with each other over those interconnects. For proper CF configuration using Cluster Admin, all of the interconnects should be working during the configuration process.

CIP configuration involves defining virtual CIP interfaces and assigning IP addresses to them. Up to eight CIP interfaces can be defined per node. These virtual interfaces act like normal TCP/IP interfaces except that the IP traffic is carried over the CF interconnects. Because CF

is typically configured with multiple interconnects, the CIP traffic will continue to flow even if an interconnect fails. This helps eliminate single points of failure as far as physical networking connections are concerned for intracluster TCP/IP traffic.

Except for their IP configuration, the eight possible CIP interfaces per node are all treated identically. There is no special priority for any interface, and each interface uses all of the CF interconnects equally. For this reason, many system administrators may choose to define only one CIP interface per node.

To ensure that you can communicate between nodes using CIP, the IP address on each node for a specific CIP interface should use the same subnet. Besides, if you use an IPv6 address, use the IPv6 address assigned to the CIP interface for communications. Communications using the link local address are not available.

CIP traffic is really intended only to be routed within the cluster. The CIP addresses should not be used outside of the cluster. Because of this, you should use addresses from the non-routable reserved IP address range.

For the IPv4 address, Address Allocation for Private Internets (RFC 1918) defines the following address ranges that are set aside for private subnets:

Subnets (s)	Class	Subnetmask
10.0.0.0	A	255.0.0.0
172.16.0.0 ... 172.31.0.0	B	255.255.0.0
192.168.0.0 ... 192.168.255.0	C	255.255.255.0

For the IPv6 address, the range where Unique Local IPv6 Unicast Addresses (RFC 4193) defined with the prefix FC00::/7 is used as the address (Unique Local IPv6 Unicast Addresses) which can be allocated freely within the private network.

For CIP nodenames, it is strongly recommended that you use the following convention for RMS:

*cfname*RMS

cfname is the CF name of the node and RMS is a literal suffix. This will be used for one of the CIP interfaces on a node. This naming convention is used in the Cluster Admin GUI to help map between normal node names and CIP names. In general, you only need to configure one CIP interface per node.



Note

In the CIP configuration, CIP names are stored in `/etc/hosts`. `/etc/nsswitch.conf(4)` should be set to use files as the first criteria when looking up nodes.

The recommended way to configure CF, CIP and CIM is to use the Cluster Admin GUI. You can use the CF/CIP Wizard in the GUI to configure CF, CIP, and CIM on all the nodes in the cluster in just a few screens. Before running the wizard, however, you must complete the following steps:

1. CF/CIP, Web-Based Admin View, and Cluster Admin should be installed on all the nodes in the cluster.
2. If you are running CF over Ethernet, then all of the interconnects in the cluster should be physically attached to their proper hubs or networking equipment and should be working.
3. If you are running CF over IP, then all interfaces used for CF over IP should be properly configured and be up and running. See "[Chapter 9 CF over IP](#)" for details.
4. Web-Based Admin View configuration must be done. Refer to "2.4.1 Management server configuration" in "PRIMECLUSTER Web-Based Admin View Operation Guide" for details.

To start the Cluster Admin screen, install the Java application ("PRIMECLUSTER Web-Based Admin View Startup") on a client.

Refer to "3.1.3.1 Installing Java application" and "3.1.3.2 Setting up Java application" in "PRIMECLUSTER Web-Based Admin View Operation Guide" to install and set up the Java application.

Log in to the Web-Based Admin View from the shortcut of the Java application (PRIMECLUSTER Web-Based Admin View Startup) and start the Cluster Admin screen from the menu.

In the *cf*tab in Cluster Admin, make sure that the CF driver is loaded on that node. Press the *Load Driver* button if necessary to load the driver. Then press the *Configure* button to start the CF Wizard.

The CF/CIP Wizard is automatically started by selecting the node where CF has not been configured to start Cluster Admin.

In PRIMECLUSTER, it is recommended that you configure the administrative LAN and cluster interconnects on different NICs. However, if you cannot make such a configuration due to restrictions on hardware in KVM environment or VMware environment, the configuration which shares the administrative LAN and cluster interconnects on the NIC is also supported.

KVM environment

In the configuration which shares the administrative LAN and cluster interconnects on the NIC, you must conform all the following conditions for network and Global Link Services (hereinafter GLS):

- Make two NICs redundant by GLS Virtual NIC mode on the Host OS.
- Create the necessary number of the VLAN interfaces for the Host OS, the administrative LAN for the Host OS, public LAN, and cluster interconnects on the virtual interface.
- Create cluster interconnects for the Host OS and guest OS on their VLAN interfaces. They are not made redundant on the cluster interconnect side.
- For the public LAN, create GLS resources on the guest OS and RMS on the guest OS monitors them.

This configuration requires the CF configuration by CLI. For the configuration method, see "[1.1.6 Example of CF configuration by CLI](#)".



Note

In this configuration, there are the following notes:

- Availability in the event of a double failure of network switch
If both network switches where two NICs are connected fail, the administrative LAN, public LAN, and cluster interconnects will enter the fault state. In this state, the Host OS and guest OS cannot be forcibly stopped and no switchover of applications occur. Note that if a double failure occurs on the NIC of a server, switchover of applications occurs because they can be forcibly stopped from the other server.
- Restriction on the timeout value of cluster interconnects
In GLS Virtual NIC mode, it takes 20 seconds to switch a path. On the other hand, the time to detect the failure of cluster interconnects is 10 seconds (default value). Therefore, with the default value, the failure of cluster interconnects will be detected first if one NIC failure occurs.
To solve this problem, change the timeout value (CLUSTER_TIMEOUT) to 40 seconds for the Host OS and 30 seconds for the guest OS.
By this setting change, the time to detect failures of cluster interconnects will be longer (from 10 seconds to 40 seconds).
- Cluster switchover due to overload of the public LAN
If a communication timeout which is more than 30 seconds occurs, PRIMECLUSTER detects a failure of cluster interconnects, forcibly stops the Host OS or guest OS, and a cluster switchover may occur.
- Restriction on the starting and stopping of GLS, and the rebooting for network service of system
When stopping and starting GLS, or rebooting the network service of System, stop CF beforehand. For instructions on stopping CF, refer to the section "[4.6 Starting and stopping CF](#)".

VMware environment

When sharing NIC with administrative LAN and cluster interconnect in the VMware environment, separate the network allocated to the virtual machine using VMware's function. In this case, CF configuration can be conducted from GUI.

1.1.1 Differences between CIP and CF over IP

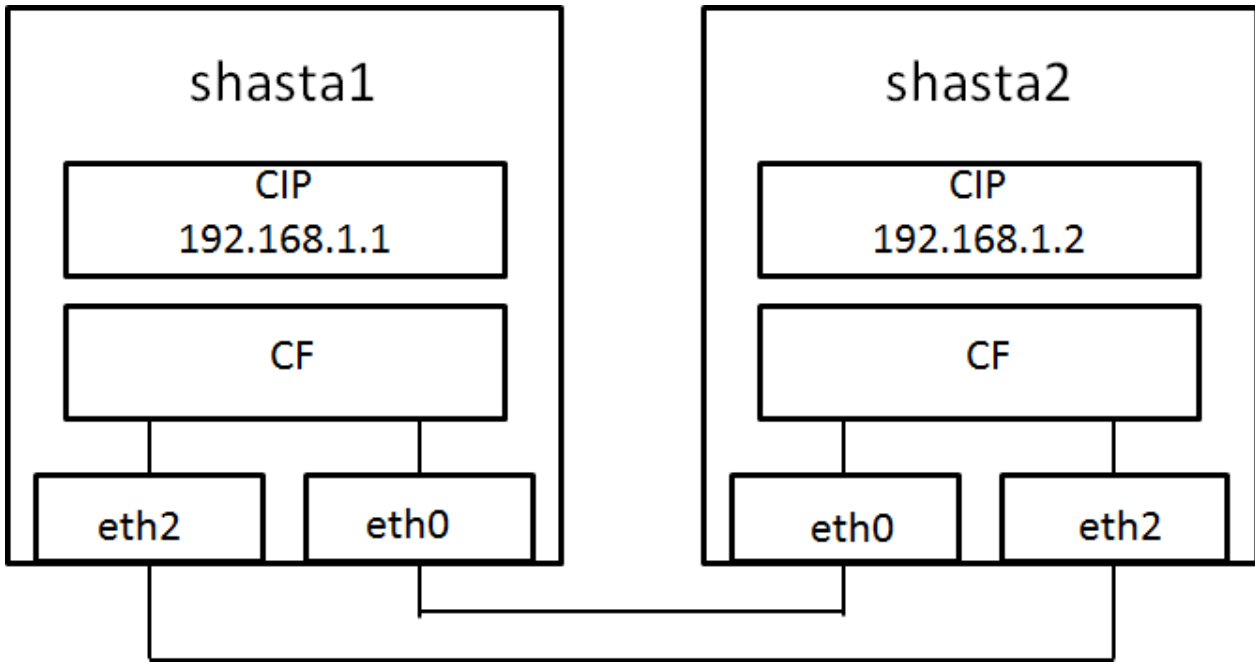
Although the two terms "CF over IP" and "CIP" (also known as "IP over CF") sound similar, they are completely different functions.

In general, the cluster interconnects are separate from the public network and not used by the TCP/IP stack. To allow applications to use TCP/UDP protocols on top of CF, CF uses the CIP driver.

CIP routes the TCP/IP traffic through the cluster interconnects to the other nodes in the cluster. CIP uses the interfaces configured for CF and does failover and load balancing if multiple interfaces are available.

CIP defines a reliable IP interface for applications on top of the Cluster Foundation (CF).

Figure 1.1 CIP diagram



CIP should not be used in a CF over IP configuration. Instead of creating an additional CIP address for applications like RMS, take the IP address assigned to CF directly.

CF over IP is used for configurations where the nodes are separated by long distances and where standard Ethernet wiring cannot be used. This can be due to the maximum cable length or different segments connected through routers. In this case, CF sends its protocol messages through the IPv4 stack instead of using the low-level network interfaces. This is not operated on IPv6. The IP stack needs to be configured up front and must be available before you start the cluster.

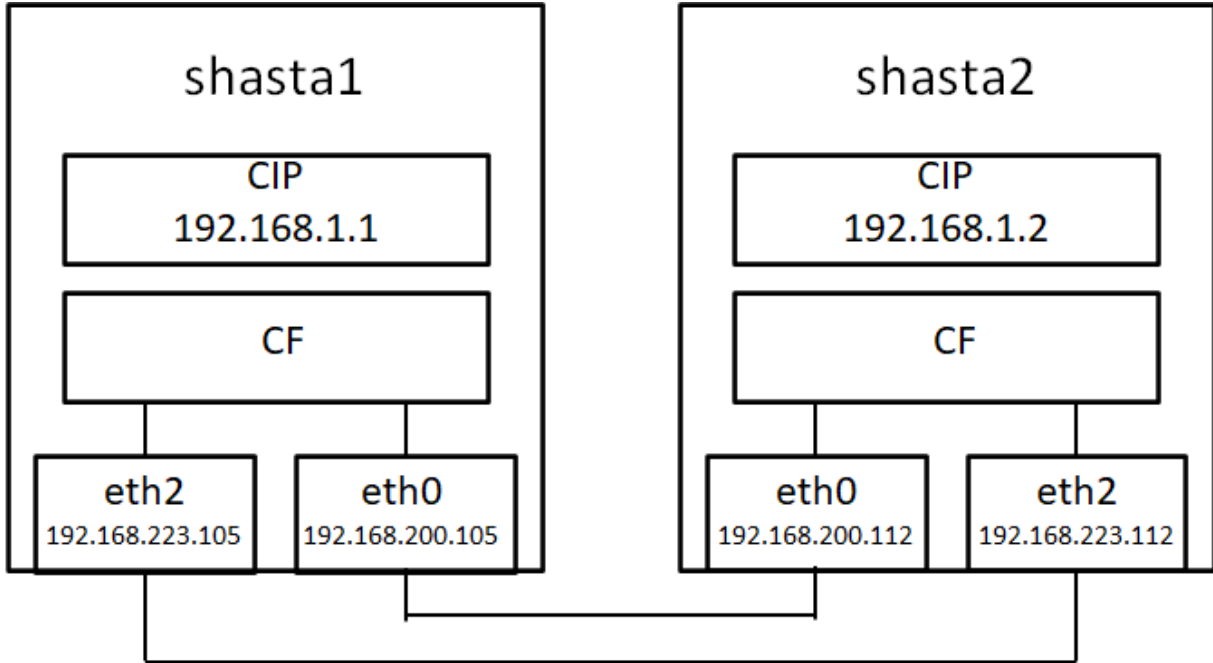
The CF node discovery is done by a JOIN message. This message is sent to the configured destination address. This can be the address of the remote node or the broadcast address of this subnet. Broadcast messages do not travel across routers and subnets. If the remote node is on a different subnet the broadcast address cannot be used or the cluster will not join.

It is recommended to separate the CF networks from the public network. Up to four IP devices can be configured for CF but the best way would be to configure only one IP device and let the IP layer deal with multi path or bonding.

CIP traffic can only be routed within a cluster. Do not use an externally used network segment for the CIP address.

Also, make sure that the network segment is different from the IP addresses of all the network interfaces (include eth0 and eth2 in the figure) on the OS.

Figure 1.2 CIP diagram (CF over IP)



Note

Use CF with the Ethernet link-level connection whenever possible because CF over IP implies additional network/protocol information and usually will not perform as well.

See

Refer to the section "[Chapter 9 CF over IP](#)" for more details.

1.1.2 cfset

The `cfset(1M)` command can be used to set certain tunable parameters in the CF driver. The values are stored in `/etc/default/cluster.config`. The `cfset(1M)` command can be used to retrieve and display the values from the kernel or the file as follows:

- A new file under `/etc/default` called `cluster.config` is created.
- The values defined in `/etc/default/cluster.config` can be set or changed using the GUI (for `cfcp` and `cfsh` during initial cluster configuration) or by using a text editor.
- The file consists of the following tuple entries, Name and Value:

Name:

- This is the name of a CF configuration parameter. It must be the first token in a line.
- Maximum length for Name is 31 bytes. The name must be unique.
- Duplicate names will be detected and reported as an error when the entries are applied by `cfconfig -l` and by the `cfset(1M)` command (`cfset -r` and `-f` option). This will log invalid and duplicate entries to `/var/log/messages`.
- `cfset(1M)` can change the Value for the Name in the kernel if the driver is already loaded and running.

Value:

- This represents the value to be assigned to the CF parameter. It is a string, enclosed in double quotes or single quotes. Maximum length for Value is 4 kilobytes (4K) characters.

- New lines are not allowed inside the quotes.
- A new line or white space marks the close of a token.
- However, if double quotes or single quotes start the beginning of the line, *cfset* treats the line as a continuation value from the previous value.
- The maximum number of Name/Value pair entries is 100.
- The hash sign (#) is used for the comment characters. It must be the first character in the line, and it causes the entries on that line to be ignored.
- Single quotes can be enclosed in double quotes or vice versa.

cfset(1M) options are as follows:

```
cfset [ -r | -f | -a | -o name | -g name | -h ]
```



See

For details on the "cfset(1M)" command, see the manual pages of "cfset(1M)."

The tuneables are as follows:

- CLUSTER_TIMEOUT (refer to "Example 1 Use cfset(1M) to tune timeout as follows:" that follows)
- CFSH (refer to the following section "1.1.3 CF security")
- CFPC (refer to the following section "1.1.3 CF security")
- CLUSTER_IP_TTL (refer to "Example 2 To set tuneables to non-default values:" that follows)

This parameter becomes enabled when using CF over IP.

Tunable Description: This is the value of the TTL field in the IP-header for all CF packets.

Default value: 64

Valid values: 1-255

- CLUSTER_IP_CTRL_TOS (refer to "Example 2 To set tuneables to non-default values:" that follows)

This parameter becomes enabled when using CF over IP.

Tunable Description: This is the value of the TOS 8-bit field in the IP-header for all CF control packets. This includes cluster heartbeat packets. The default value shown below is a best-fit default which sets the 6-bit DSCP field to binary 100010. This is a DSCP forwarding 4F AF class (compatible with older IP precedence) and specifies the lowest AF drop precedence (least likely to be dropped when congestion is encountered).

Default value: 0x88 (136)

Valid values: 0-255

- CLUSTER_IP_DATA_TOS (refer to "Example 2 To set tuneables to non-default values:" that follows)

This parameter becomes enabled when using CF over IP.

Tunable Description: This is the value of the TOS 8-bit field in the IP-header for all CF data packets (non-control packets). The default value shown below is a best-fit default which sets the 6-bit DSCP field to binary 001010. This is a DSCP forwarding 1F AF class (compatible with older IP precedence) and specifies the lowest AF drop precedence (least likely to be dropped when congestion is encountered).

Default value: 0x28 (40)

Valid values: 0-255

After any change to cluster.config, run the *cfset(1M)* command as follows:

```
# cfset -r
```

Example

Example 1 Use `cfset(1M)` to tune timeout as follows:

```
CLUSTER_TIMEOUT "30"
```

This changes the default 10-second timeout to 30 seconds. The minimum value is 1 second. There is no maximum. It is strongly recommended that you use the same value on all cluster nodes.

`CLUSTER_TIMEOUT` represents the number of seconds that one cluster node waits for a heartbeat response from another cluster node. Once `CLUSTER_TIMEOUT` seconds has passed, the non-responding node is declared to be in the `LEFTCLUSTER` state. The default value for `CLUSTER_TIMEOUT` is 10, which experience indicates is reasonable for most `PRIMECLUSTER` installations. We allow this value to be tuned for exceptional situations, such as networks which may experience long switching delays.

Example 2 To set tuneables to non-default values:

1. Edit the `/etc/default/cluster.config` file and add entries for each tunable:

```
CLUSTER_IP_TTL "64"  
CLUSTER_IP_CTRL_TOS "0x88"  
CLUSTER_IP_DATA_TOS "0x28"
```

2. Run `cfset -f` to verify settings in file.
3. Run `cfset -r` to load new values to CF.
4. Run `cfset -a` to verify values in kernel.

1.1.3 CF security

`PRIMECLUSTER` includes the following facilities for cluster communications if you do not want to use `.rhosts`:

- `cfcp/cfsh`
- `sshconf` (not supported by Wizard Tools)

These tools are provided to allow cluster configuration in an environment which does not permit `rsh` and `rcp`. They are specialized utilities that do not provide all the functionality of `rsh` and `rcp` and are not intended as replacements.

1.1.3.1 cfcp/cfsh

CF includes the ability to allow cluster nodes to execute commands on another node (`cfsh`) and to allow cluster nodes to copy files from one node to another (`cfcp`). However, this means that your cluster interconnects must be secure since any node that can join the cluster has access to these facilities. Because of this, these facilities are disabled by default.

`PRIMECLUSTER 4.1` and higher offers a chance to configure these facilities. As one of the final steps of the CF Configuration Wizard in the Cluster Adm GUI, there are two checkboxes. Checking one enables remote file copying and checking the other enables remote command execution.

To enable remote access using `cfcp/cfsh`, set the following parameters in `cluster.config`:

```
CFCP "cfcp"  
CFSH "cfsh"
```

To deactivate, remove the settings from the `/etc/default/cluster.config` file and run `cfset -r`.

Refer to the section "[1.1.2 cfset](#)" in this chapter for more information.

1.1.3.2 sshconf

You can use the `sshconf` tool to set up non-interactive `ssh` access among a list of nodes. Running `sshconf` is similar to setting up the `.rhosts` file for `rsh`.

`sshconf` uses the RSA authentication method and protocol version 2. If it exists, `sshconf` uses the default authentication key `$HOME/.ssh/id_rsa`, or it creates the key if it does not already exist.

Example

Examples of the sshconf tool are as follows:

- Enable one way access between nodes:

```
shasta1# sshconf shasta2 shasta3 shasta4
```

Running this command on shasta1 sets up one way ssh access from shasta1 to shasta2, shasta3, and shasta4 respectively.

- Disable one-way access to a node:

```
shasta1# sshconf -d shasta2 shasta3 shasta4
```

Running this command on shasta1 disables ssh access from shasta1 to shasta2, shasta3, and shasta4. This means that shasta1 does not have ssh access to shasta2, shasta3, and shasta4; however, shasta2, shasta3, and shasta4 still have the same ssh access as before running the command.

- Enable two-way access without password:

```
shasta1# sshconf -c shasta2 shasta3 shasta4
```

Running this command on shasta1 sets up ssh access among shasta2, shasta3, and shasta4 without being asked for a password. Note that shasta1 (where the command is run) is not automatically included. shasta1 only has one-way ssh access to shasta2, shasta3, and shasta4.

Note

ssh is not supported by Wizard Tools.

1.1.4 Example of creating a cluster

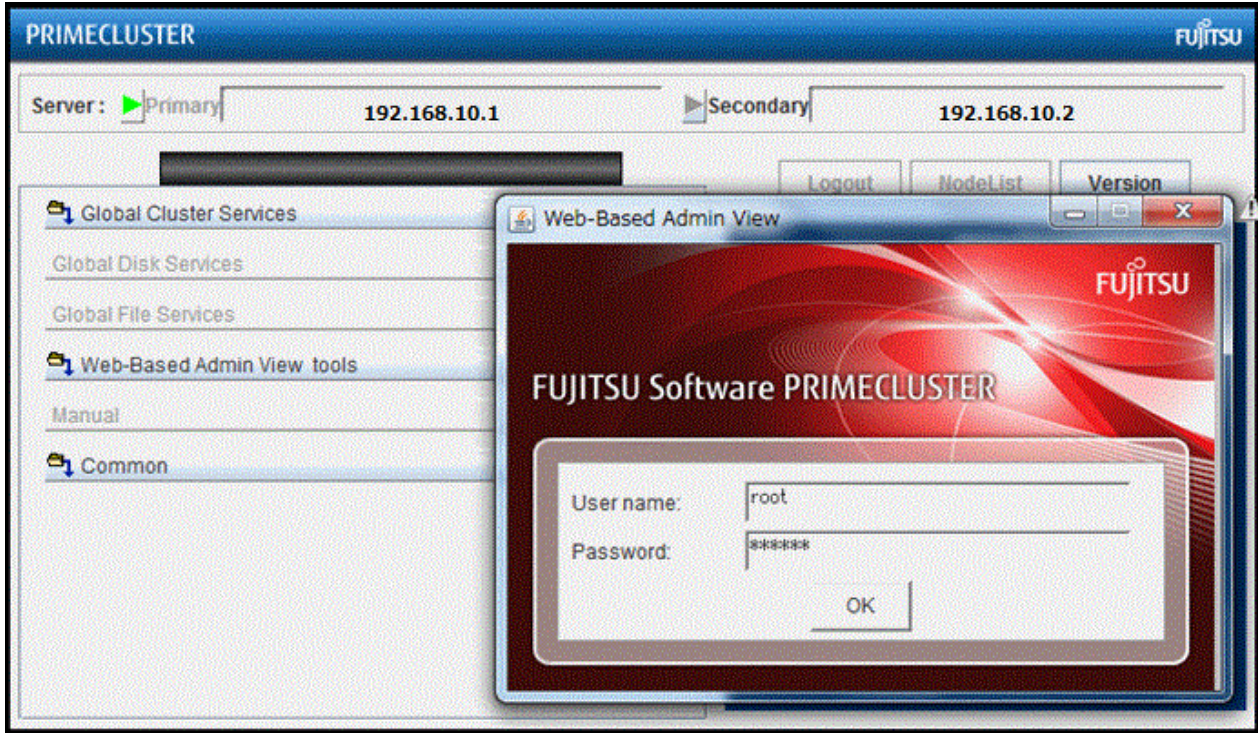
The following example shows what the Web-Based Admin View and Cluster Admin screens would look like when creating a two-node cluster. The nodes involved are named shasta1 and shasta2, and the cluster name is SHASTAS.

This example assumes that the Web-Based Admin View configuration has already been done. shasta1 is assumed to be configured as the primary management server for Web-Based Admin View, and shasta2 is the secondary management server.

Start the Web-Based Admin View screen from the shortcut of the Java application (PRIMECLUSTER Web-Based Admin View Startup).

After a few moments, a login pop-up appears asking for a user name and password (similar to the screen below).

Figure 1.3 Login pop-up



Since you will be running the Cluster Admin CF Wizard, which does configuration work, you will need a privileged user ID such as root. There are three possible categories of users with sufficient privilege:

- The user root

You can enter root for the user name and root's password on shasta1. The user root is always given the maximum privilege in Web-Based Admin View and Cluster Admin.

- A user in group clroot

You can enter the user name and password for a user on shasta1 who is part of the UNIX group clroot. This user will have maximum privilege in Cluster Admin, but will be restricted in what Web-Based Admin View functions they can perform. This should be fine for CF configuration tasks.

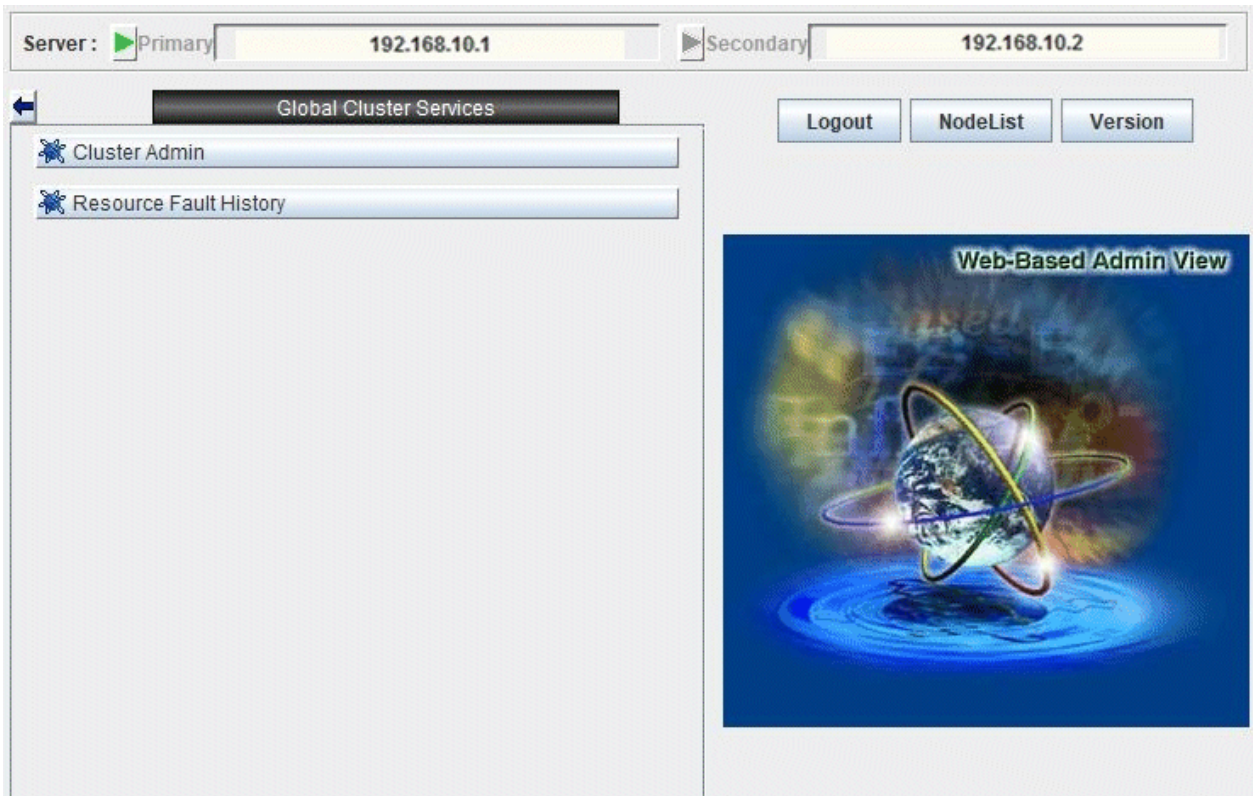
- A user in group wvroot

You can enter the user name and password for a user on shasta1 who is part of the UNIX group wvroot. Users in wvroot have maximum Web-Based Admin View privileges and are also granted maximum Cluster Admin privileges.

For further details on Web-Based Admin View and Cluster Admin privilege levels, refer to "4.3.1 Assigning Users to Manage the Cluster" in "PRIMECLUSTER Installation and Administration Guide."

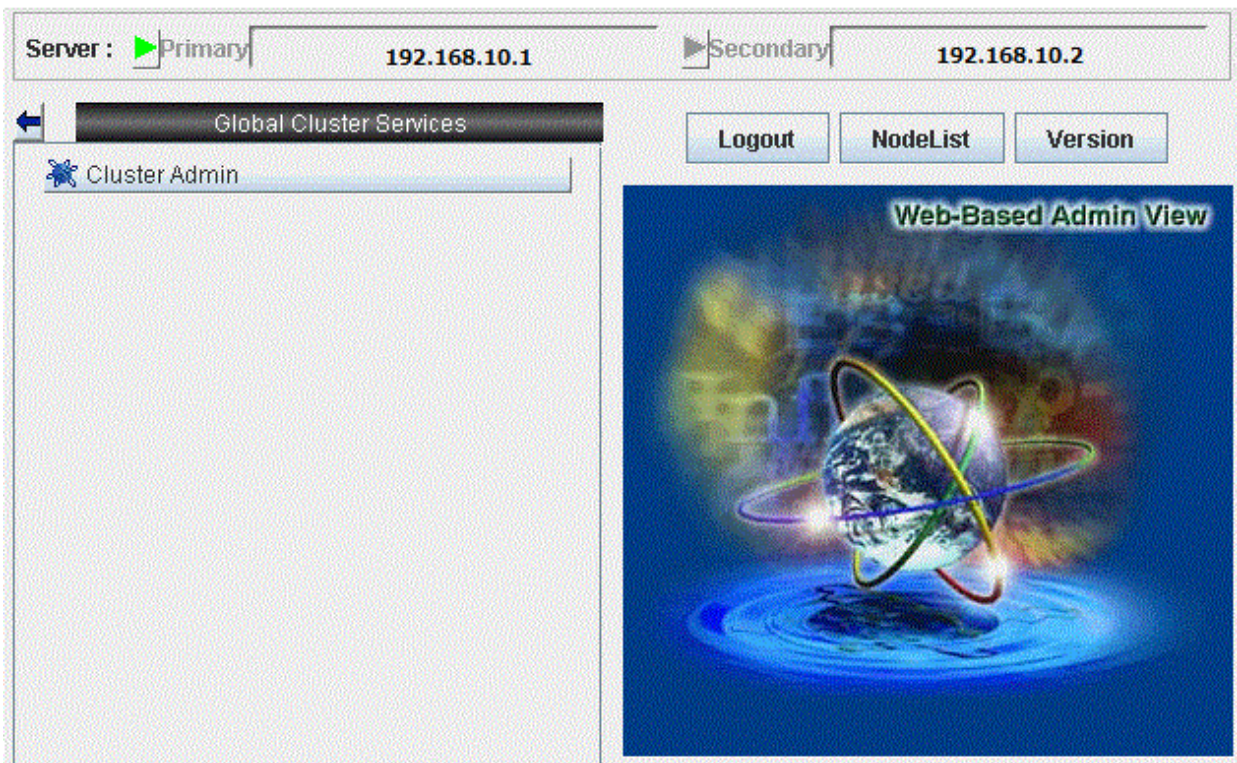
After clicking on the *OK* button, the top menu appears. Click on the button labeled *Global Cluster Services*.

Figure 1.4 Main Web-Based Admin View window after login



The Cluster Admin selection window appears.

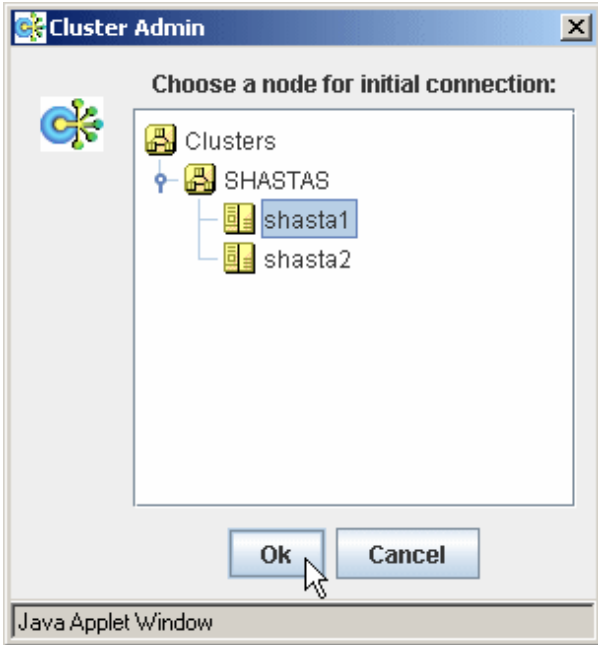
Figure 1.5 Global Cluster Services window in Web-Based Admin View



Click on the button labeled *Cluster Admin* to launch the Cluster Admin GUI.

The *Choose a node for initial connection* window appears.

Figure 1.6 Initial connection pop-up

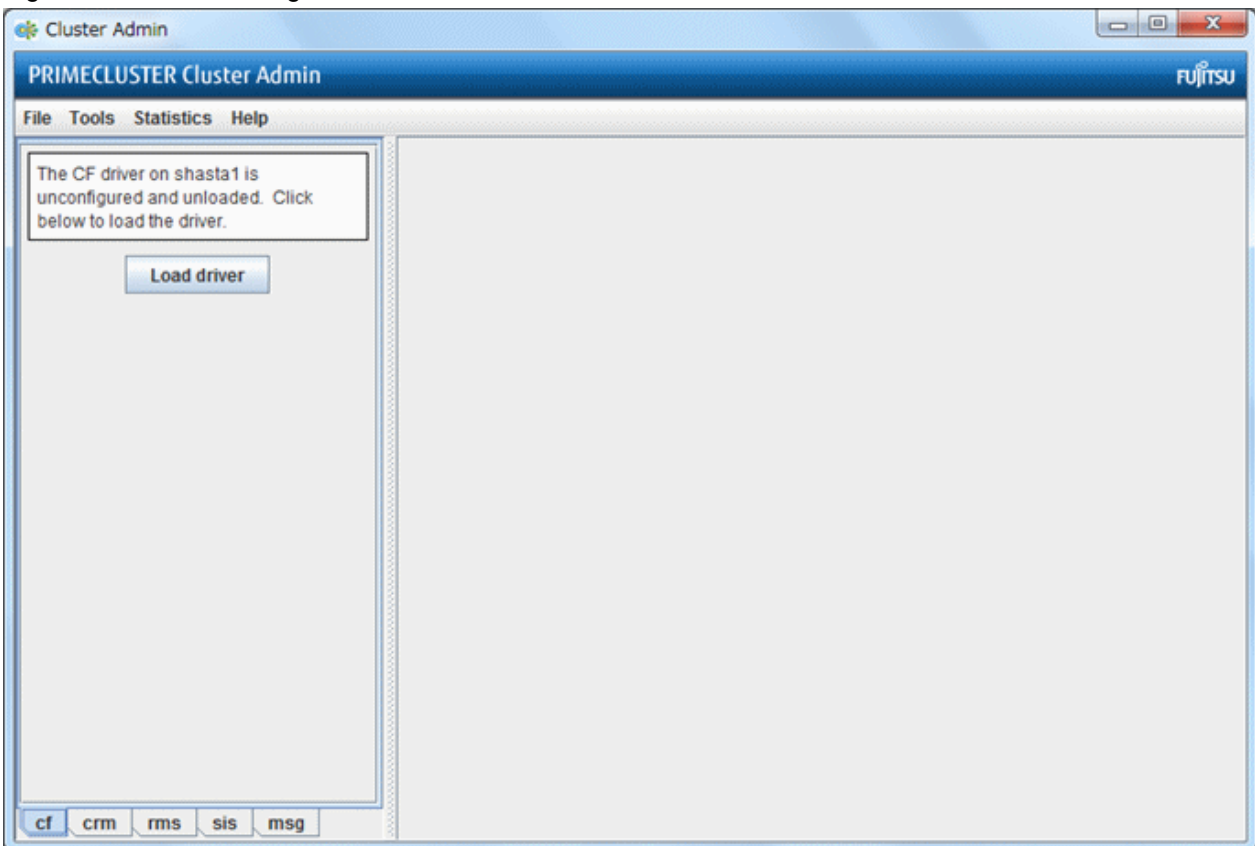


The *Choose a node for initial connection* window lists the nodes that are known to the Web-Based Admin View management station. If you select a node where CF has not yet been configured, then Cluster Admin will let you run the CF Wizard on that node.

In this example, neither shasta1 nor shasta2 have had CF configured, so either would be acceptable as a choice. In Figure 1.6, shasta1 is selected. Clicking on the *OK* button causes the main Cluster Admin GUI to appear. Since CF is not configured on shasta1, a window similar to the following figure appears.

When cancelling the startup of the Cluster Admin GUI, click the <Cancel> button in the initial connection pop-up screen.

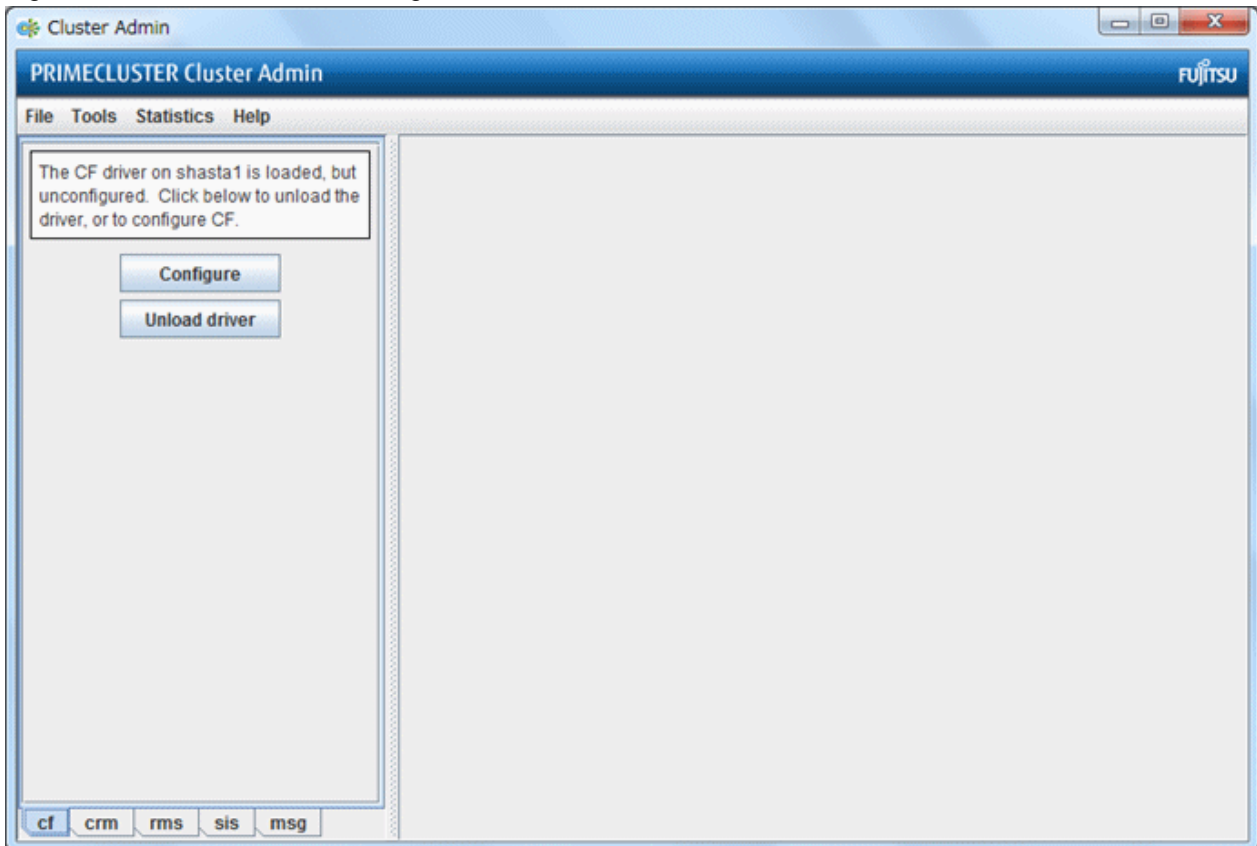
Figure 1.7 CF is unconfigured and unloaded



Click on the *Load driver* button to load the CF driver.

A window indicating that CF is loaded but not configured appears.

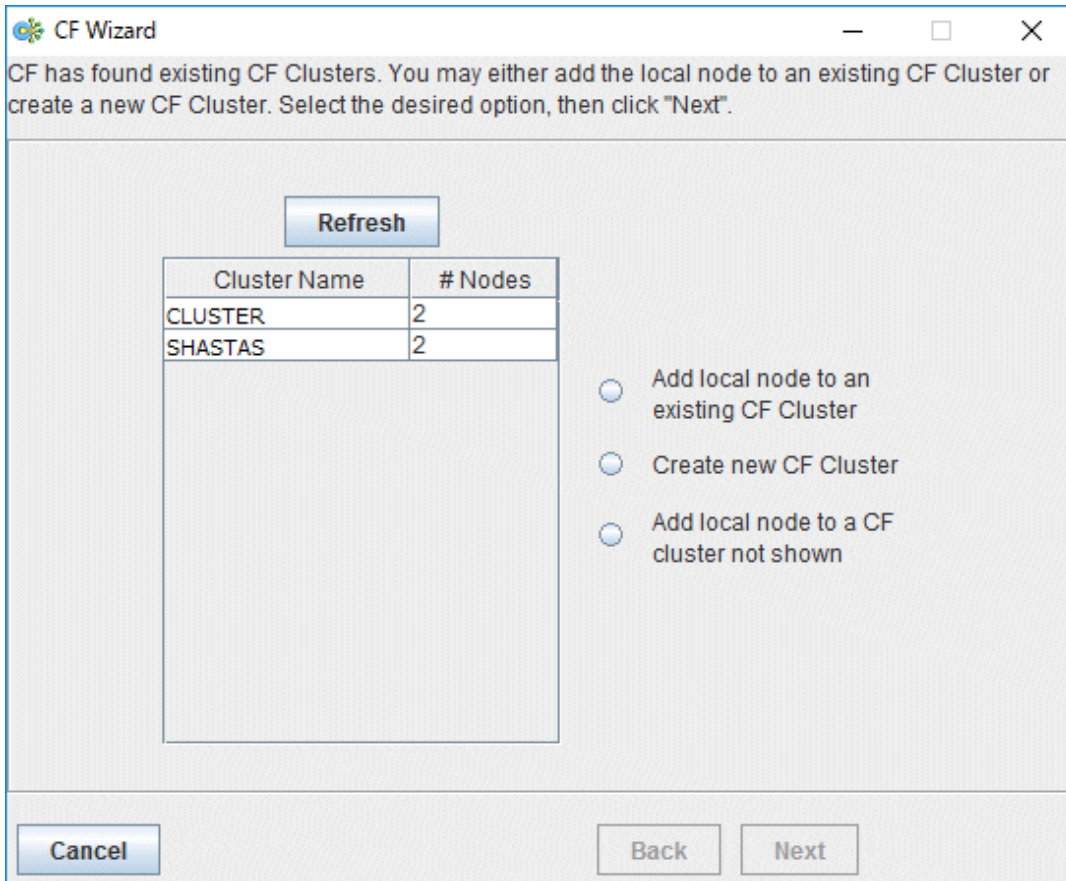
Figure 1.8 CF loaded but not configured



Click on the Configure button to bring up the CF Wizard. The CF Wizard scans for existing clusters.

After the CF Wizard finishes looking for clusters, a window similar to Figure 1.9 appears.

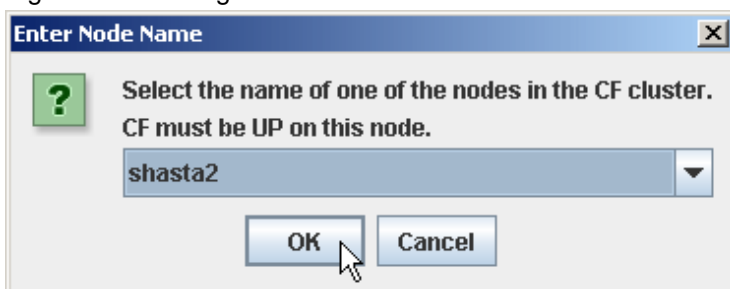
Figure 1.9 Creating or joining a cluster



This window lets you decide if you want to join an existing cluster or create a new one.

A pure CF over IP cluster will not show up in the *Cluster Name* column. To join a CF over IP cluster, select the *Add local node to a CF cluster not shown* radio button and click *Next*.

Figure 1.10 Adding a local node to a CF cluster not shown



Enter the node name of the CF cluster that you want to join. Click *OK* to proceed. After scanning the node and retrieving the existing cluster's details, the CF wizard takes you to the window for joining an existing cluster.

To create a new cluster as shown in "Figure 1.9 Creating or joining a cluster", make sure that the [Create new CF Cluster] radio button is selected. Then, click <Next>.

The window for creating a new cluster or for joining an existing cluster appears, depending on your previous selection.

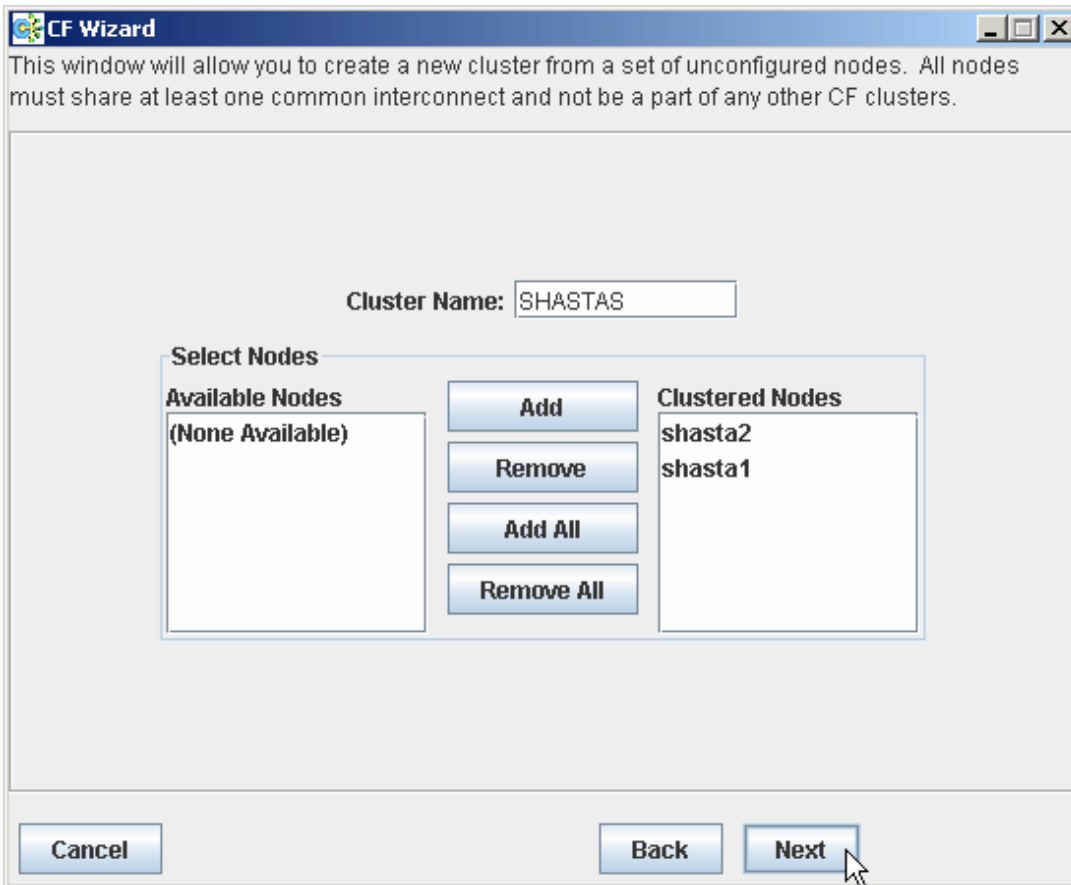
The following figure shows the window for creating a new cluster. The window for joining an existing cluster has almost the same configuration as the window for creating a new cluster except that the cluster name cannot be changed.

Note

Make sure to select [Create new CF Cluster] in the following environments:

- For a cloud environment
- For an RHOSP environment
- When configuring an interconnect between cluster nodes with different network segments
- When a device that only supports IP exists in the route tracking of the interconnect

Figure 1.11 Selecting cluster nodes and the cluster name



This window lets you choose the cluster name and also determine what nodes will be in the cluster. In the example above, we have chosen SHASTAS for the cluster name.

Below the cluster name are two boxes. The one on the right, under the label *Clustered Nodes*, contains all the nodes that you want to become part of this CF cluster. The box on the left, under the label *Available Nodes*, contains all the other nodes known to the Web-Based Admin View management server. You should select nodes in the left box and move them to the right box using the *Add* or *Add All* button. If you want all of the nodes in the left box to be part of the CF cluster, then just click on the *Add All* button.

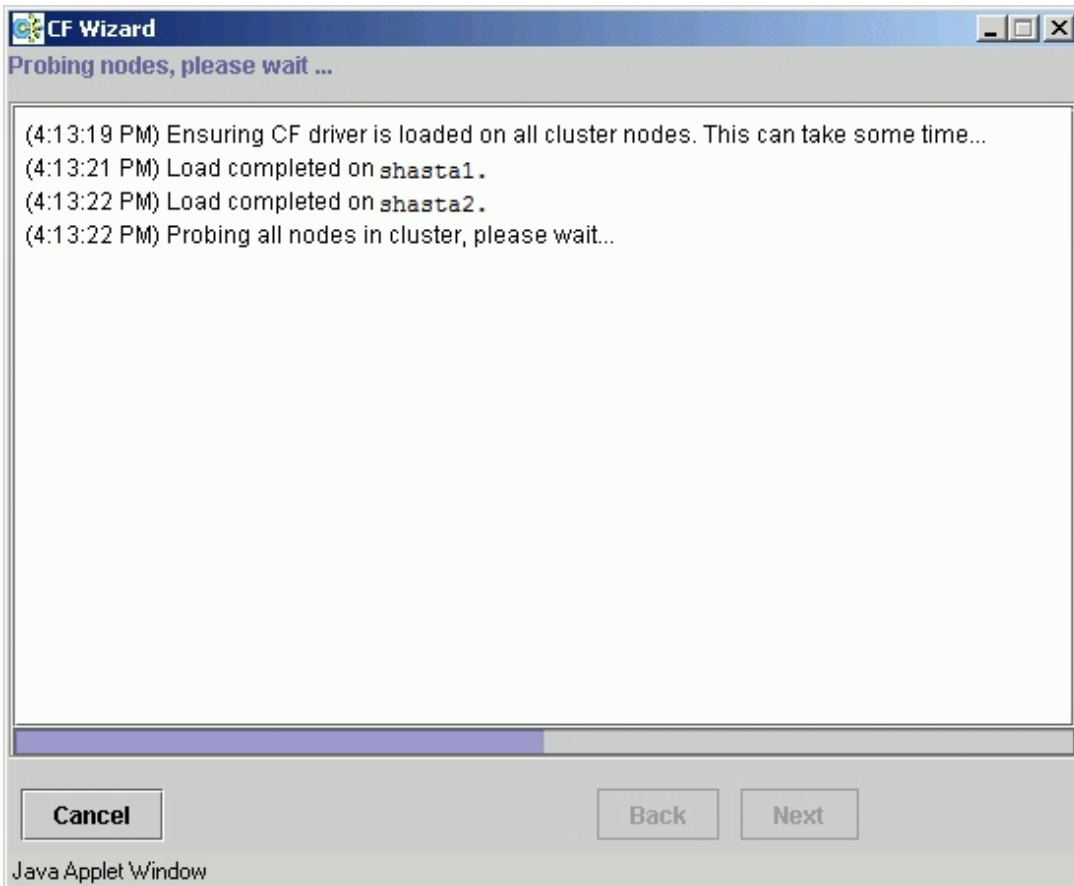
If you get to this window and you do not see all of the nodes that you want to be part of this cluster, then there is a very good chance that you have not configured Web-Based Admin View properly. When Web-Based Admin View is initially installed on the nodes in a potential cluster, it configures each node as if it were a primary management server independent of every other node. If no additional Web-Based Admin View configuration were done, and you started up Cluster Admin on such a node, then Figure 1.11 would show only a single node in the right-hand box and no additional nodes on the left-hand side. If you see this, then it is a clear indication that proper Web-Based Admin View configuration has not been done.

Refer to "4.3 Preparations for Starting the Web-Based Admin View Screen" in "PRIMECLUSTER Installation and Administration Guide."

After you have chosen a cluster name and selected the nodes to be in the CF cluster, click on the *Next* button.

The CF Wizard then loads CF on all the selected nodes and does CF pings to determine the network topology. While this activity is going on, a window similar to Figure 1.12 appears.

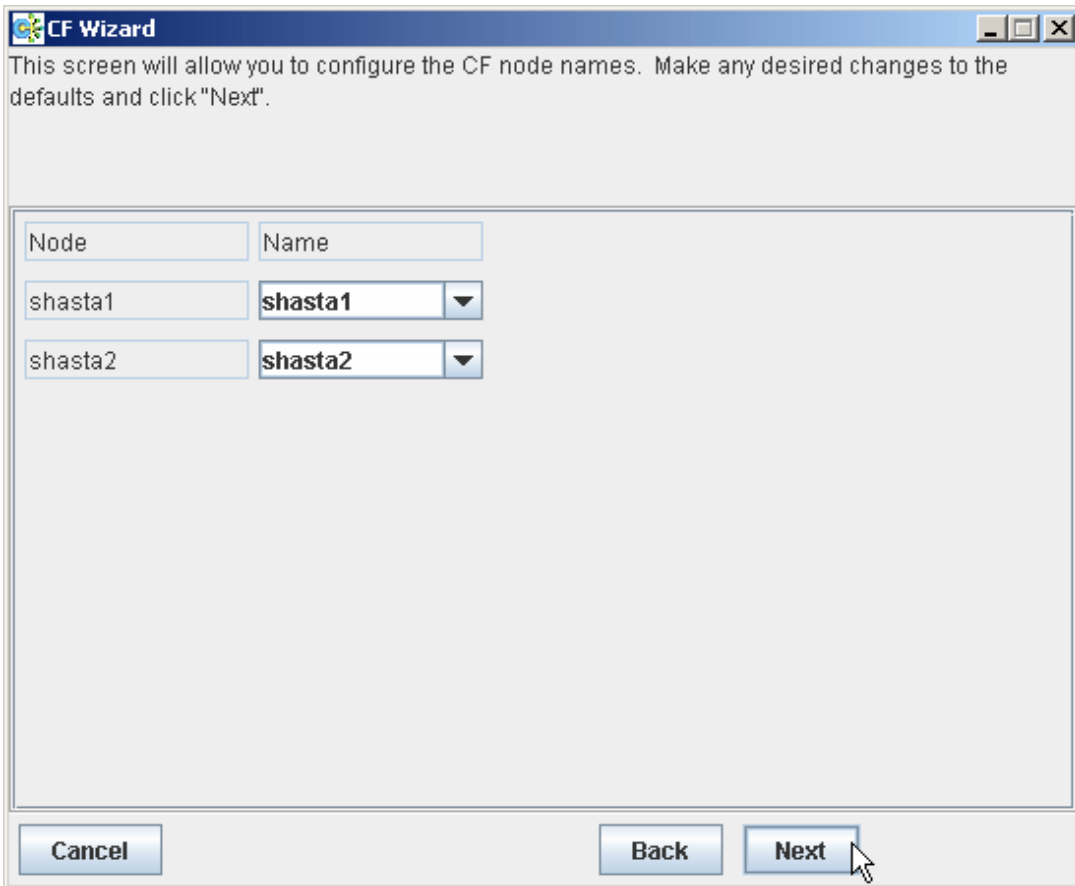
Figure 1.12 CF loads and pings



Usually, loading the CF driver is a relatively quick process. However, on some systems which use large disk arrays, the first CF load can take several minutes.

The window that allows you to edit the CF node names for each node appears. By default, the CF node names, which are shown in the right-hand column, are the same as the Web-Based Admin View names which are shown in the left-hand column.

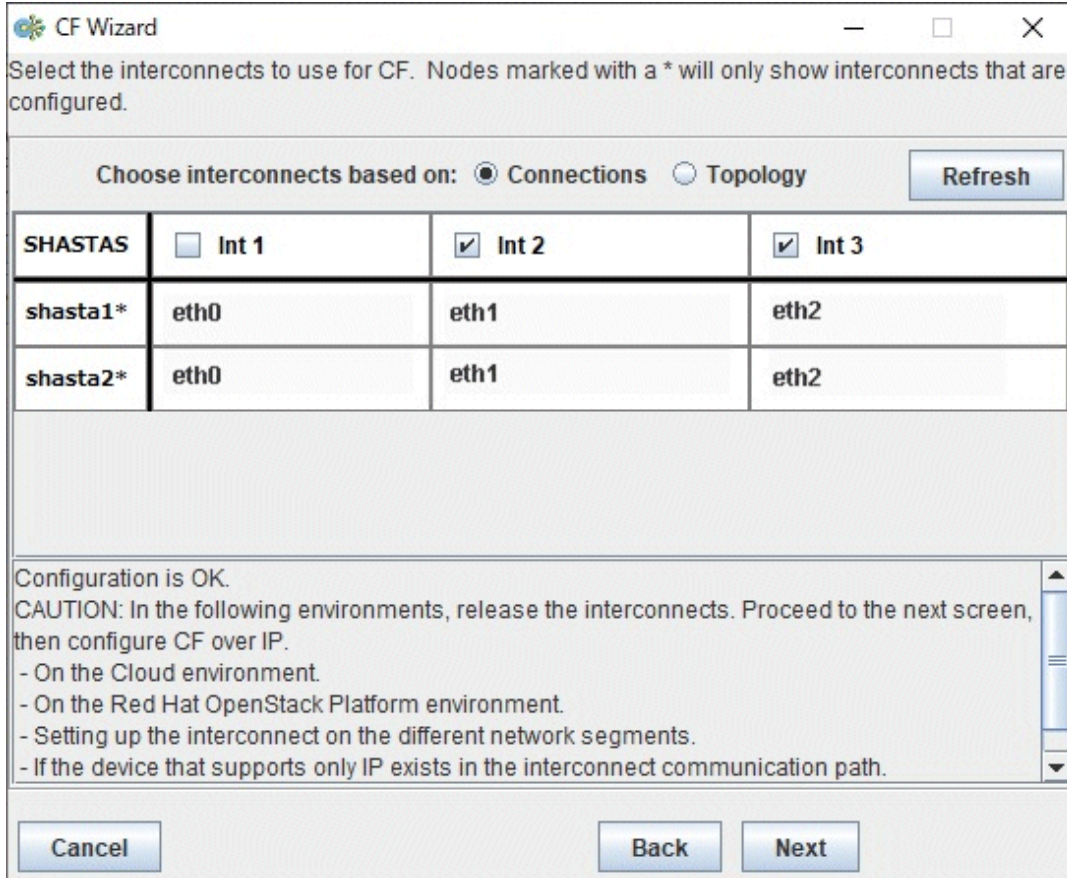
Figure 1.13 Edit CF node names



Make any changes to the CF node name and click *Next*.

After the CF Wizard has finished the loads and the pings, the CF topology and connection table appears.

Figure 1.14 CF topology table/CF connection table



Point

Use CF over IP for the following:

- For a cloud environment
- For an RHOSP environment
- When configuring an interconnect between cluster nodes with different network segments
- When a device that only supports IP exists in the route tracking of the interconnect

When using CF over IP, uncheck all the checkboxes of the interconnects and click [Next].

Before explaining the CF topology table and the connection table in this screen, the following terms are defined:

- Full interconnect - An interconnect where CF communication is possible to all the nodes in the cluster.
- Partial interconnect - An interconnect where CF communication is possible between at least two nodes, but not to all the nodes. If the devices on a partial interconnect are intended for CF communications, then there is a networking or cabling problem somewhere.
- Unconnected devices - These devices are potential candidates for CF configuration, but are not able to communicate with any other nodes in the cluster.

The CF Wizard determines all the full interconnects, partial interconnects, and unconnected devices in the cluster using CF pings. If there are one or more full interconnects, then it will display the connection table shown in "[Figure 1.14 CF topology table/CF connection table.](#)"

Connections table

The connection table lists all full interconnects. Each column with an Int header represents a single interconnect. Each row represents the devices for the node whose name is given in the left-most column. The name of the CF cluster is given in the upper-left corner of the table.

In "Figure 1.14 CF topology table/CF connection table", for example, Int 1 has eth0 on shasta1 and shasta2 attached to it. The cluster name is SHASTA.

Note

The connections and topology tables typically show devices that are on the public network. Using devices on a public network is a security risk; therefore, in general, do not use any devices on the public network as a CF interconnect. Instead, use devices on a private network.

Although the CF Wizard may list Int 1, Int 2, and so on, it should be pointed out that this is simply a convention in the GUI. CF itself does not number interconnects. Instead, it keeps track of point-to-point routes to other nodes.

Occasionally, there may be problems setting up the networking for the cluster. Cabling errors may mean that there are no full interconnects. If you click on the button next to *Topology*, the CF Wizard will display all the full interconnects, partial interconnects, and unconnected devices it has found. If a particular category is not found, it is omitted. For example, in "Figure 1.14 CF topology table/CF connection table", only full interconnects are shown because no partial interconnects or unconnected devices were found on shasta1 or shasta2.

To configure CF using the connection table, click on the interconnects that have the devices that you want to use.

When you are satisfied with your choices, click on *Next* to go to the CF over IP configuration window.

Topology table

The topology table gives more flexibility in configuration than the connection table. In the connection table, you could only select an interconnect, and all devices on that interconnect would be configured. In the topology table, you can individually select devices.

While you can configure CF using the topology table, you may wish to take a simpler approach. If no full interconnects are found, then display the topology table to see what your networking configuration looks like to CF. Using this information, correct any cabling or networking problems that prevented the full interconnects from being found. Then go back to the CF Wizard window where the cluster name was entered and click on *Next* to cause the Wizard to reprobe the interfaces. If you are successful, then the connection table will show the full interconnects, and you can select them. Otherwise, you can repeat the process.

The text area at the bottom of the window lists problems or warnings concerning the configuration.

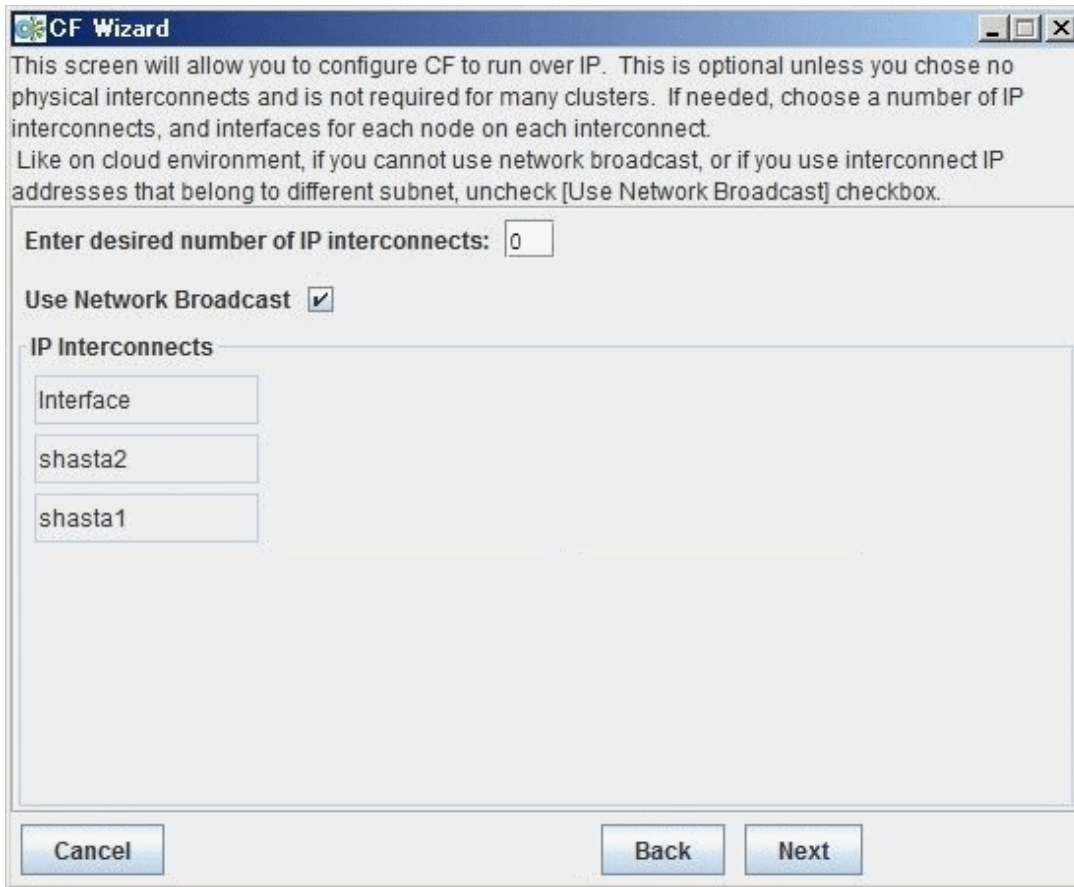
When you are done, click on *Next* to go to the CF over IP configuration window.

CF can use either Ethernet packets or IP for its communication. The topology table and connection table discussed previously allow you to configure CF to use Ethernet packets. This is the preferred CF configuration since CF over Ethernet is significantly faster than CF over IP.

However, CF over Ethernet requires Ethernet link-level connectivity between the nodes in a cluster. In certain disaster recovery scenarios, there may only be IP connectivity between hosts. This is typically the case when the hosts are separated by large geographical distances.

- For physical environment, KVM environment and VMware environment (Configuration when not using CF over IP)

Figure 1.15 CF over IP configuration window



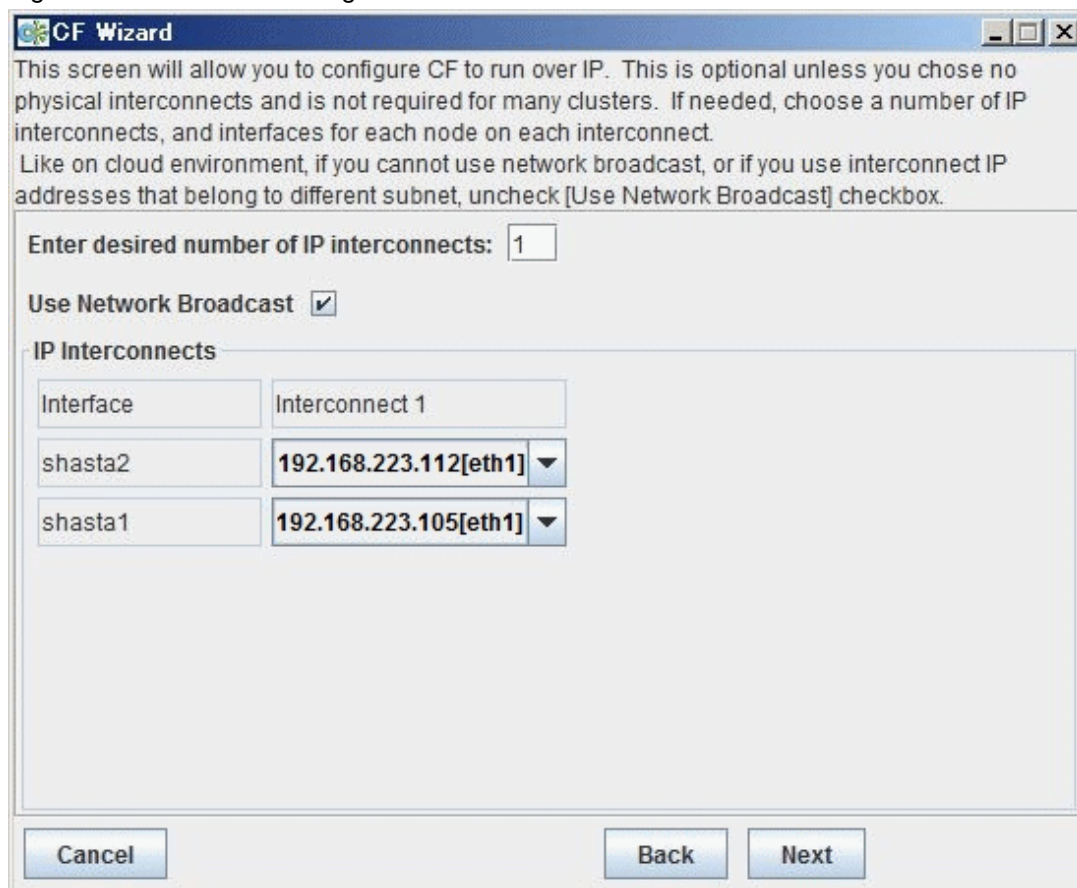
If you have already configured CF over Ethernet in the topology table or the connection table, you do not have to change any settings on this screen. Just click [Next].

- For physical environment, KVM environment and VMware environment (Configuration when using CF over IP)

 Note

.....
If the interconnection of each node has the same network segment, perform this configuration. See "[- For cluster nodes placed between different network segments](#)" when reaching between cluster nodes with different network segments.
.....

Figure 1.16 CF over IP configuration window



Use the configuration window above to configure CF to run over the IP network.

CF over IP uses IP subnetworks in the same way that CF over Ethernet uses physical interconnects. Each IP interconnect must correspond to exactly one subnetwork. For example, suppose that your nodes had the following IP interfaces:

node	subnet1	subnet 2	subnet 3
shasta1	172.25.222.105	192.168.223.105	185.33.48.105
shasta2	172.25.222.112	192.168.223.112	185.33.48.112

Using CF over IP, you might configure one IP interconnect to use the IP addresses 192.168.223.105 and 192.168.223.112. You could configure a second IP interface using the addresses 185.33.48.105 and 185.33.48.112.

But if you need CF over IP, then set [Enter desired number of IP interconnects] to "2" (or more if desired). Up to 4 IP interconnects can be set. The Wizard will propose IP interconnects. The IP interconnects are conveniently sorted by subnetwork. If you think that a particular subnetwork is missing a node, then double check that the netmask and broadcast addresses are properly configured for all the nodes on subnetwork.

Select the subnetworks that you want to use as your IP interconnects. You should avoid using addresses on your public network. CF allows promiscuous joins without any limits, so it is best to use private subnetworks for your IP interconnects.

With this setting, CF can be configured to run over the IP interface. After entering the number of required IP interconnects, the CF Wizard will display interconnects sorted by available subnetworks, netmasks, and broadcast addresses.

All the IP addresses for all the nodes on a given IP interconnect must be on the same IP subnetwork and should have the same netmask and broadcast address.

Confirm that [Use Network Broadcast] checkbox is checked.

Choose the IP interconnects from the combo boxes on this window, and click on [Next]. The CIP Wizard windows like "[Figure 1.19 CIP wizard \(IPv4\) window](#)" and "[Figure 1.20 CIP wizard \(IPv6\) window](#)" appear.

- For cluster nodes placed between different network segments

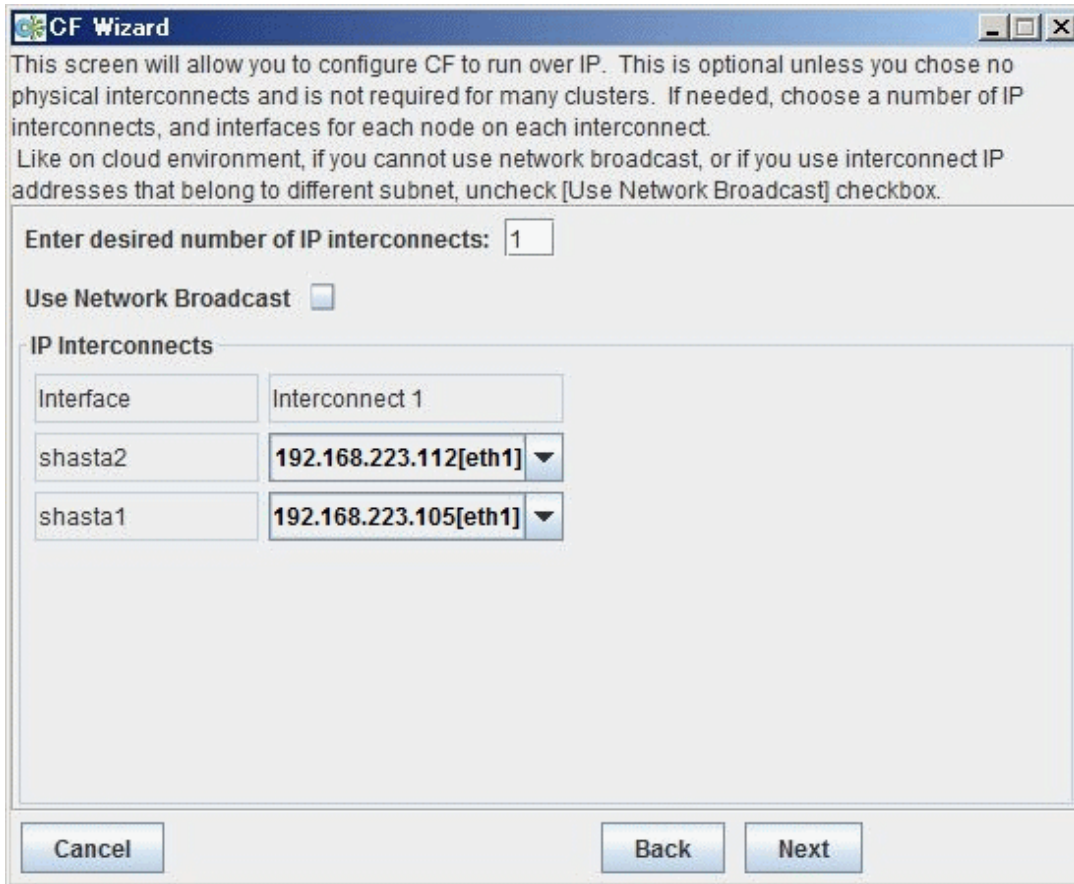
Set [Enter desired number of IP interconnects] to at least "1."

If it is "0," click [Back] to return to the CF topology and connection table screen, and deselect the interconnects.

Deselect [Use Network Broadcast].

Select the IP address and network interface of cluster interconnect in interconnect, and click [Next].

Figure 1.17 CF over IP configuration window



- For a cloud environment or an RHOSP environment

Make sure that [Enter desired number of IP interconnects] is "1."

If it is "0," click [Back] to return to the CF topology and connection table screen, and deselect the interconnects.

Deselect [Use Network Broadcast].

Select the IP address and network interface ("192.168.223.105 [eth1]" for shasta1, "192.168.223.112 [eth1]" for shasta2) of cluster interconnect in interconnect1, and click [Next].

Figure 1.18 CF over IP configuration window

The image shows a 'CF Wizard' window with a title bar containing a gear icon and the text 'CF Wizard'. The window has standard minimize, maximize, and close buttons. The main content area contains the following text and controls:

This screen will allow you to configure CF to run over IP. This is optional unless you chose no physical interconnects and is not required for many clusters. If needed, choose a number of IP interconnects, and interfaces for each node on each interconnect.
Like on cloud environment, if you cannot use network broadcast, or if you use interconnect IP addresses that belong to different subnet, uncheck [Use Network Broadcast] checkbox.

Enter desired number of IP interconnects:

Use Network Broadcast

IP Interconnects

Interface	Interconnect 1
shasta1	192.168.223.105 [eth1] ▼
shasta2	192.168.223.112 [eth1] ▼

At the bottom of the window, there are three buttons: 'Cancel', 'Back', and 'Next'.

Figure 1.19 CIP wizard (IPv4) window

The screenshot shows a window titled "CF Wizard" with a standard Windows title bar. Below the title bar is a text area containing the following instructions: "This screen will allow to configure IP over CF. Choose the number of subnets you would like, and for each subnet, choose a naming scheme, and an IP range. You may also mark one subnet for use by RMS."

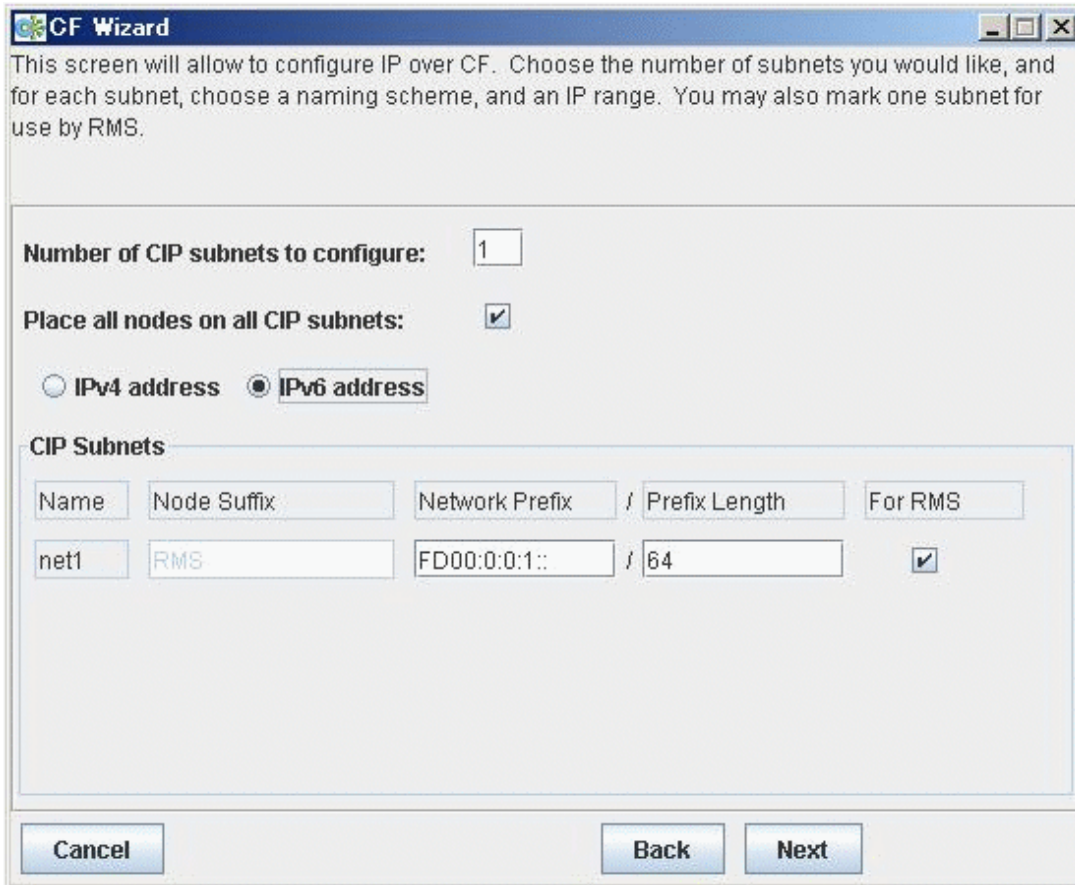
The main configuration area contains the following elements:

- Number of CIP subnets to configure:** A text input field containing the number "1".
- Place all nodes on all CIP subnets:** A checkbox that is checked.
- Address Type:** Two radio buttons: "IPv4 address" (selected) and "IPv6 address".
- CIP Subnets:** A table with the following columns: Name, Node Suffix, Subnet Number, Subnet Mask, and For RMS.

Name	Node Suffix	Subnet Number	Subnet Mask	For RMS
net1	RMS	192.168.1.0	255.255.255.0	<input checked="" type="checkbox"/>

At the bottom of the window are three buttons: "Cancel", "Back", and "Next".

Figure 1.20 CIP wizard (IPv6) window



This window allows you to configure CIP. You can enter a number in the box after *Number of CIP* subnets to *configure* to set the number of CIP subnets to configure. The maximum number of CIP subnets is 8.

For each defined subnet, the CIP Wizard configures a CIP interface on each node defined in the CF cluster.

Set either IPv4 or IPv6 as the IP address to set to the CIP interface.

By selecting either of the [IPv4 address] or [IPv6 address] radio button, you can switch "[Figure 1.19 CIP wizard \(IPv4\) window](#)" and "[Figure 1.20 CIP wizard \(IPv6\) window](#)".

When using IPv4 for CIP interface

The following values are assigned for CIP interface:

- The IP address will be a unique IP number on the subnet specified in the *Subnet Number* field. The node portions of the address start at 1 and are incremented by 1 for each additional node.
The CIP Wizard will automatically fill in a default value for the *Subnet Number* field for each CIP subnetwork requested. The default values are taken from the private IP address range specified by RFC 1918. Note that the values entered in the *Subnet Number* field have 0 for their node portion even though the CIP Wizard starts the numbering at 1 when it assigns the actual node IP addresses.
- The IP name of the interface will be of the form *cnameSuffix* where *cname* is the name of a node from the CF Wizard, and the *Suffix* is specified in the field *Node Suffix*. If the checkbox *For RMS* is selected, then the *Node Suffix* will be set to RMS and will not be editable. If you are using RMS, one CIP network must be configured for RMS.
- The *Subnet Mask* will be the value specified.

In "[Figure 1.19 CIP wizard \(IPv4\) window](#)", the system administrator has selected 1 CIP network. The *For RMS* checkbox is selected, so the RMS suffix will be used. Default values for the *Subnet Number* and *Subnet Mask* are also selected. The nodes defined in the CF cluster are shasta1 and shasta2. This will result in the following configuration:

- On shasta1, a CIP interface will be configured with the following:

```
CIP nodename: shasta1RMS  
  
IP address: 192.168.1.1  
  
Subnet Mask: 255.255.255.0
```

- On shasta2, a CIP interface will be configured with the following:

```
CIP nodename: shasta2RMS  
  
IP address: 192.168.1.2  
  
Subnet Mask: 255.255.255.0
```

When using IPv6 for CIP interface

The following values are assigned for CIP interface:

- The IP address will be a unique IP number on the network prefix specified in the *Prefix* field. Interface ID of the address start at 1 and are incremented by 1 for each additional node.

The CIP Wizard will automatically fill in a default value for the Prefix field for each CIP subnetwork requested. The default values are taken from the Unique Local Unicast Address range specified by RFC 4193. Note that the values entered in the *Prefix* field have 0 for their interface ID portion even though the CIP Wizard starts the numbering at 1 when it assigns the actual node IP addresses.

- The IP name of the interface is represented in *cfnameSuffix* format.
 - *cfname* is the name of a node given by the CF Wizard
 - *Suffix* is specified in the field *Node Suffix*.

If the checkbox *For RMS* is selected, then the *Node Suffix* will be set to RMS and will not be editable. If you are using RMS, one CIP network must be configured for RMS.

- The *Prefix Length* will be the value specified.

In "Figure 1.20 CIP wizard (IPv6) window", the system administrator has selected 1 CIP network. The *For RMS* checkbox is selected, so the RMS suffix will be used. Default values for the *Prefix* and *Prefix Length* are also selected. The nodes defined in the CF cluster are shasta1 and shasta2. This will result in the following configuration:

- On shasta1, a CIP interface will be configured with the following:

```
CIP nodename: shasta1RMS  
  
IPv6 address: FD00:0:0:1::1  
  
Prefix Length: 64
```

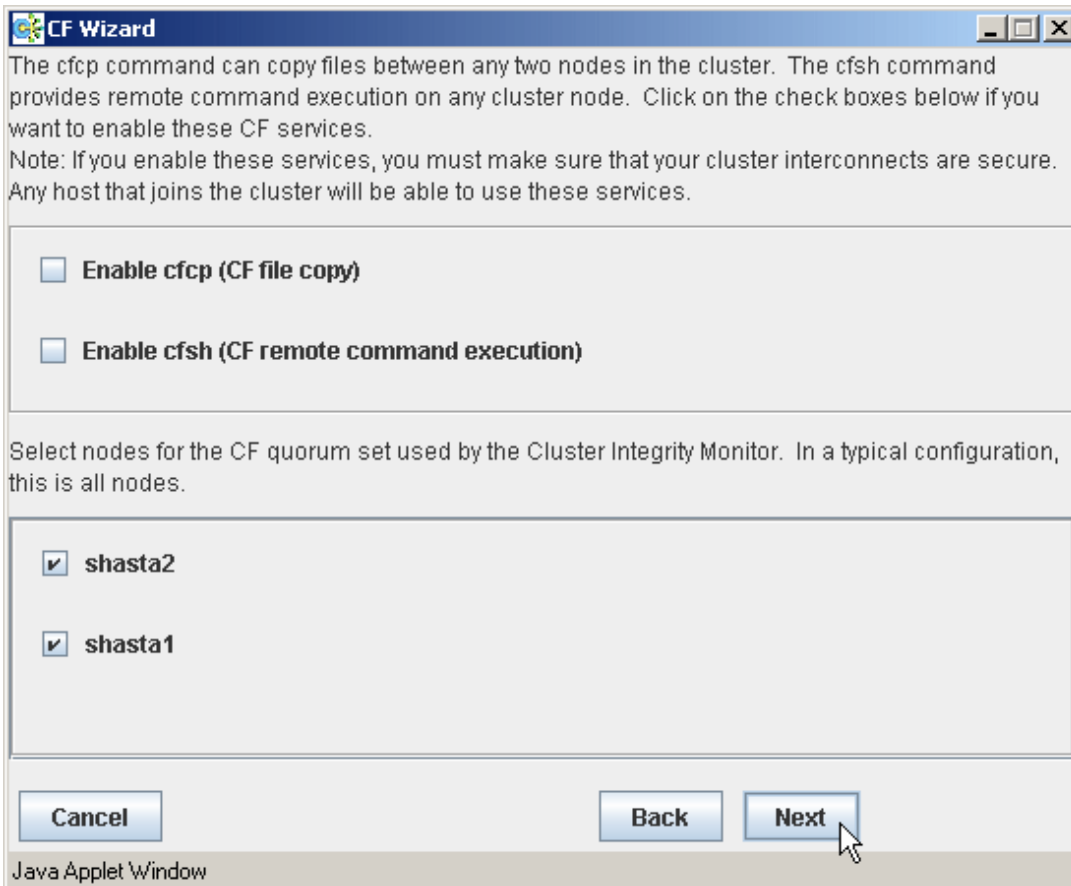
- On shasta2, a CIP interface will be configured with the following:

```
CIP nodename: shasta2RMS  
  
IPv6 address: FD00:0:0:1::2  
  
Prefix Length: 64
```

The CIP Wizard stores the configuration information in the file `/etc/cip.cf` on each node in the cluster. This is the default CIP configuration file. The Wizard will also update `/etc/hosts` on each node in the cluster to add the new IP node names.

When you click on the Next button, CIM configuration window appears.

Figure 1.21 CIM configuration window



The CIM configuration window in Figure 1.18 has the following parts:

- The upper portion allows you to enable *cfcp* and *cfsh*.

cfcp is a CF-based file copy program. It allows files to be copied among the cluster hosts. *cfsh* is a remote command execution program that, similar to *cfcp*, works between nodes in the cluster. The use of these programs is optional. In this example these items are not selected. If you enable these services, however, any node that has access to the cluster interconnects can copy files or execute commands on any node with root privileges.

- The lower portion allows you to determine which nodes should be monitored by CIM.

This window also lets you select which nodes should be part of the CF quorum set. The CF quorum set is used by the CIM to tell higher level services when it is safe to access shared resources.

Note

Do not change the default selection of the nodes

A checkbox next to a node means that node will be monitored by CIM. By default, all the nodes are checked. For almost all configurations, you will want to have all the nodes monitored by CIM.

This window will also allow you to configure CF Remote Services. You can enable either remote command execution, remote file copying, or both.

Note

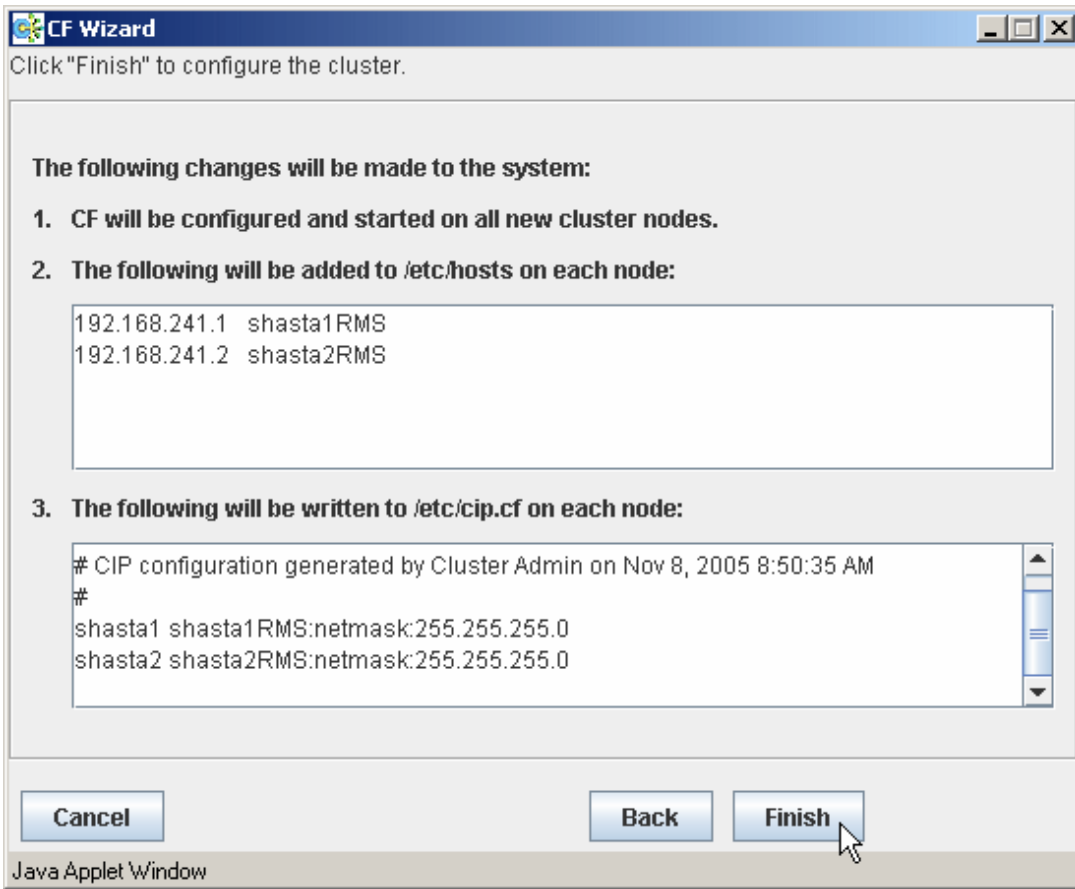
- Enabling either of these means that you must trust all the nodes on the CF interconnects and the CF interconnects must be secure. Otherwise any system able to connect to the CF interconnects will have access to these services.

- To use RMS, make sure to configure *cfcp* and *cfsh*.



Click on the Next button to go to the summary window.

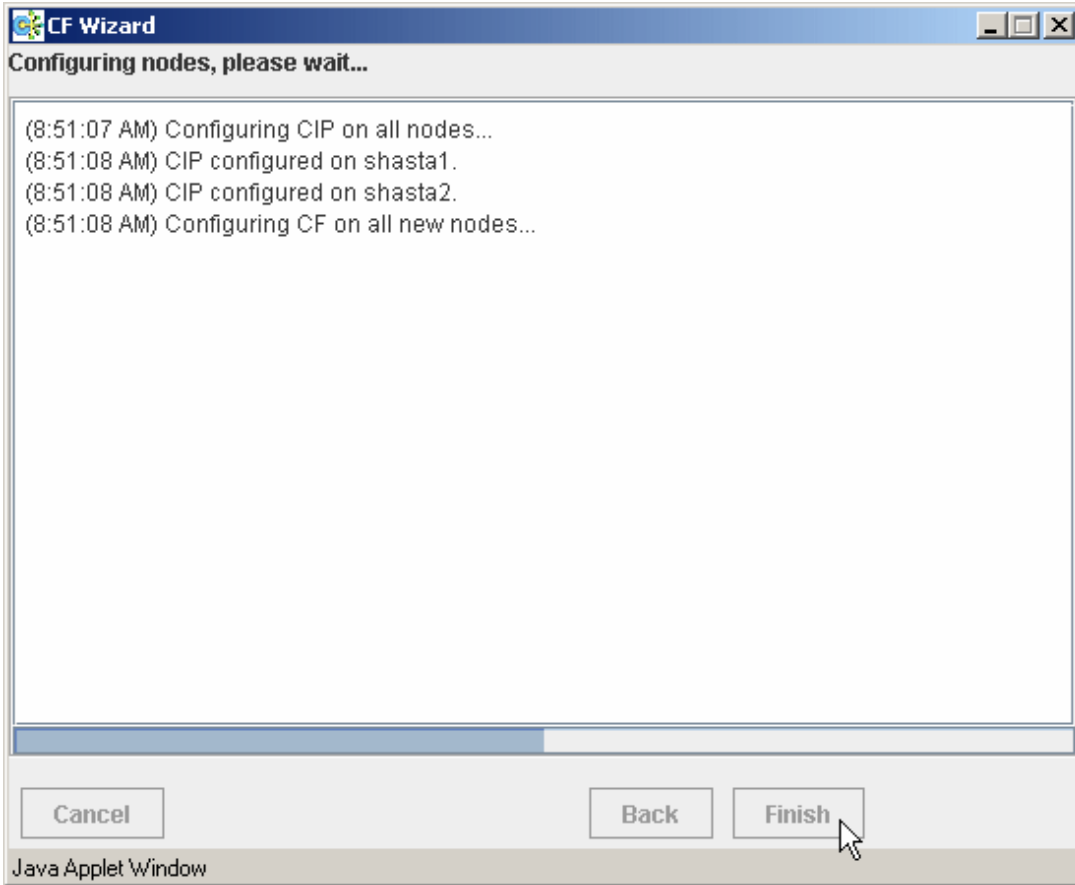
Figure 1.22 Summary window



This window summarizes the major changes that the CF, CIP, and CIM Wizards will perform. When you click on the Finish button, the CF Wizard performs the actual configuration on all the nodes.

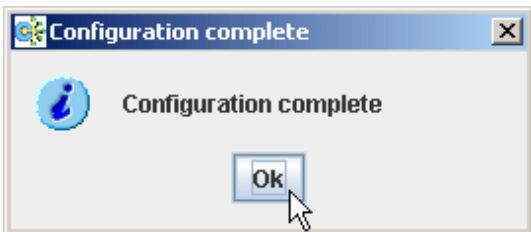
A window similar to the following figure is displayed while the configuration is being done.

Figure 1.23 Configuration processing window



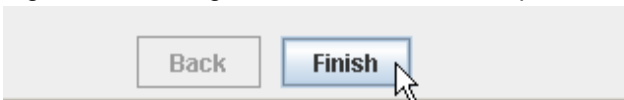
This window is updated after each configuration step. When the configuration successfully completes, a small completion pop-up window appears.

Figure 1.24 Configuration completion pop-up



Click on the OK button to close the pop-up window. The configuration processing window now has a Finish button.

Figure 1.25 Configuration window after completion

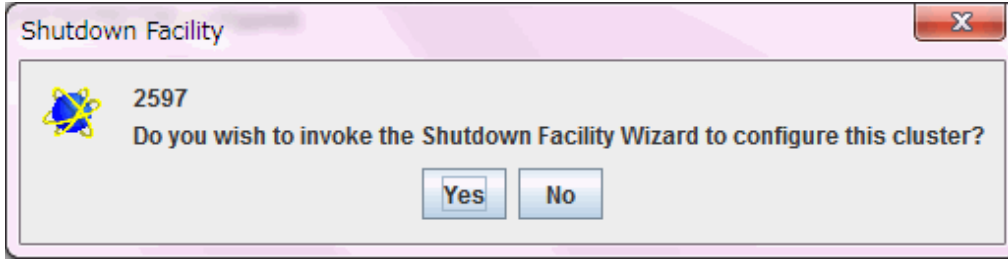


When the CF wizard is executed on the unconfigured node, it requests the CF driver to push CF modules to each Ethernet devices on the system. By this process, CF executes ping of each interface. This will enable the CF wizard to detect the network topology.

However, this unloading process may fail. To solve this problem, unload the driver on the error node, and then load the driver again. The unloading process is simplified by using GUI. See "4.6 Starting and stopping CF" for details.

When the window is closed by clicking the Finish button, following pop-up window is displayed.

Figure 1.26 SF wizard start notification pop-up



Click on the No button to close the pop-up window.

Point

For a cloud environment or an RHOSP environment, check the following:

- a. Execute the following command on either node to make sure that all the nodes have already joined in the cluster.

```
# cftool -n
```

Example:

```
# cftool -n
Node      Number State      Os          Cpu
shasta1   1      UP         Linux      EM64T
shasta2   2      UP         Linux      EM64T
```

In this example, make sure that both shasta1 and shasta2 are displayed in Node, and the State is in UP status.

- b. Execute the following command on all the nodes to make sure that the CF over IP configuration is enabled.

```
# cftool -d
```

Example:

```
# cftool -d
Number Device      Type Speed  Mtu    State Configured Address
4      /dev/ip0    6     n/a    1392   UP     YES     0a.00.00.c9.00.00
```

In this example, make sure that /dev/ip0 is only displayed in Device.

If the above procedure a or b is not performed correctly, make sure that the configurations of the security groups (or security rules) created with the following references are properly set.

- For an FJcloud-O environment, see "3.1.2.4 Creating the Security Group for the Cluster Interconnect" in "PRIMECLUSTER Installation and Administration Guide Cloud Services."
- For a NIFCLOUD environment, see "8.3.2.2 Rules Applied to the Cluster Interconnect" in "PRIMECLUSTER Installation and Administration Guide Cloud Services."
- For an AWS environment, see "20.3.2.2 Rules Applied to the Cluster Interconnect" in "PRIMECLUSTER Installation and Administration Guide Cloud Services."
- For an Azure environment, see "26.3.2.2 Security Rules Applied to the Cluster Interconnect" in "PRIMECLUSTER Installation and Administration Guide Cloud Services."
- For an RHOSP environment, see "H.2.2.2 Creating Virtual Network" in "PRIMECLUSTER Installation and Administration Guide."

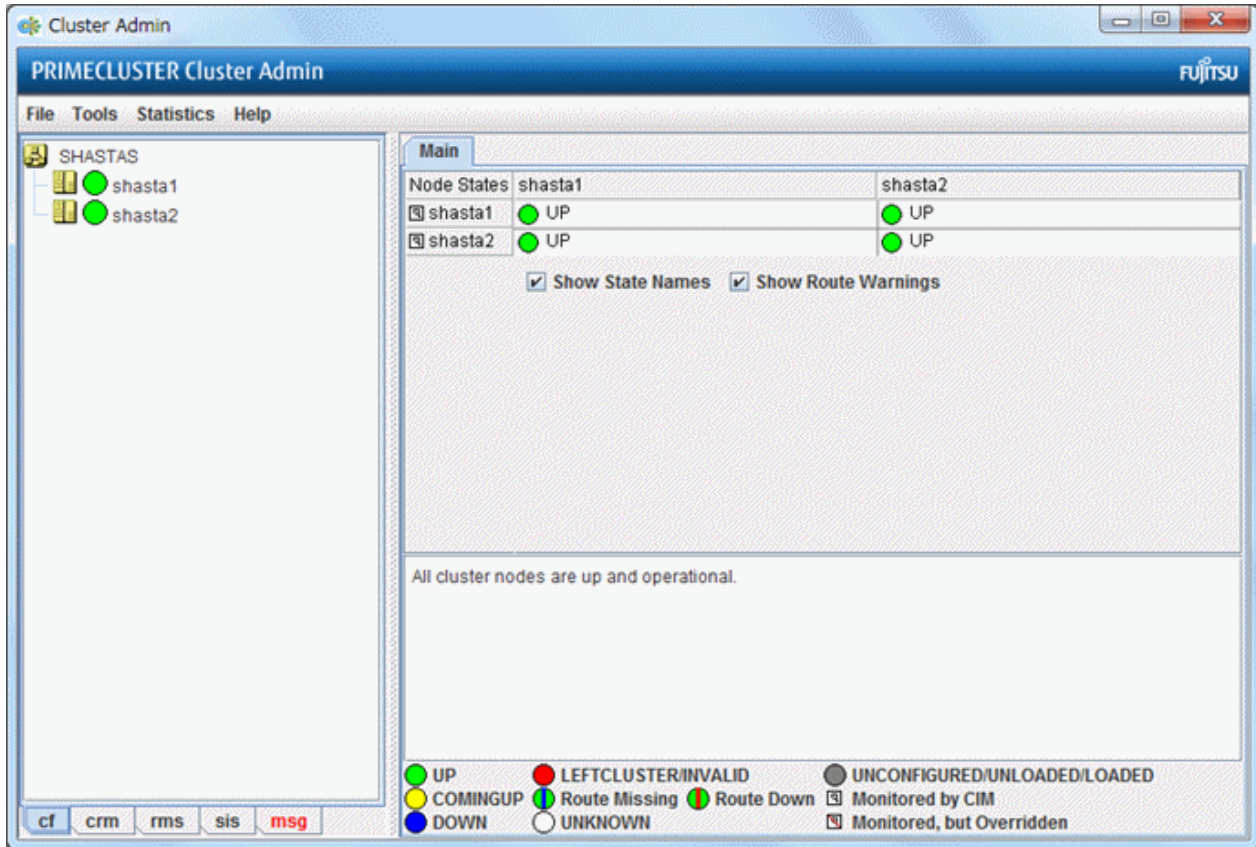
Also, when creating multiple virtual network interfaces and using a virtual network interface, for which a default gateway is not set, to communicate with interfaces on different subnets, see "21.1.3 Setting Instances" in "PRIMECLUSTER Installation and Administration Guide Cloud Services" and make sure that the configuration of the routing is set so that communication between interconnects can be performed.

If the configurations of the security groups (or security rules) and the configuration of the routing are properly set, execute the following procedure:

1. Stop CF. For procedures on stopping CF, see "4.6 Starting and stopping CF."
2. Unconfigure the CF configuration on all the nodes. For procedures on how to unconfigure CF, see "4.12 Unconfigure CF."
3. Re-execute "5.1.1 Setting Up CF and CIP" in "PRIMECLUSTER Installation and Administration Guide."

When the CF Wizard is completed, the main window of Cluster Admin will be displayed as below. After a few seconds, the latest information of the CF's configuration and status will be updated and displayed on it.

Figure 1.27 Main CF window



1.1.5 Adding a new node to CF

This section describes how to add a node to an existing CF cluster.

The first step is to make sure that Web-Based Admin View is properly configured on the new node. Refer to "PRIMECLUSTER Web-Based Admin View Operation Guide" for additional details on Web-Based Admin View configuration options.

After you have properly configured Web-Based Admin on the new node, you should start Cluster Admin. If you are already running the Cluster Admin GUI, exit it and then restart it.

The first window that Cluster Admin displays is the small "Figure 1.6 Initial connection pop-up" window. This window lists all of the nodes which are known to Web-Based Admin View. If the new node is not present in this list, then you should recheck your Web-Based Admin configuration and also verify that the new node is up.

To add the new node, select it in the initial connection pop-up. After making your selection, run the CF Wizard by clicking on the *Configure* button (see "Figure 1.8 CF loaded but not configured") The CF Wizard will appear, and you can use it to join the existing CF cluster.

The CF Wizard will allow you to configure CF, CIM, and CIP on the new node. After it is run, you should configure the Shutdown Facility on the new node.

You will also need to do additional configuration work for other PRIMECLUSTER products you might be using such as the Cluster Resource Manager (CRM), RMS, Global Disk Services (hereinafter GDS), GFS, and so forth.

1.1.6 Example of CF configuration by CLI

When configuring CF by CLI, perform the following steps.

In this section, the cluster system configured with two nodes where the CF node names are "shasta1" and "shasta2", is explained as an example.

1. For cluster nodes placed between different network segments, in a cloud environment, or in an RHOSP environment, make sure the IP address of cluster interconnect is set.

Example: When you use "eth1" as the NIC of the cluster interconnect.

Execute the ip command in order to make sure that the IP address (192.168.223.105) is set to "eth1."

```
# ip eth1
x: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu xxxx qdisc pfifo_fast state UP qlen xxxx
  link/ether xx:xx:xx:xx:xx:xx brd xx:xx:xx:xx:xx:xx
  inet 192.168.223.105/24 brd 192.168.223.255 scope global eth1
    valid_lft forever preferred_lft forever
  inet6 xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xx scope link
    valid_lft forever preferred_lft forever
```

2. Create CIP configuration files.

Specify /etc/cip.cf as below on all the nodes which configure the cluster system.

Example:

```
shasta1      shasta1RMS:netmask:255.255.255.0
shasta2      shasta2RMS:netmask:255.255.255.0
```

Note

If you manually create /etc/cip.cf, you cannot reconfigure CF by Cluster Admin. To reconfigure CF by Cluster Admin, delete the /etc/cip.cf file beforehand.

When reconfiguring CF, create the /etc/cip.cf file from Cluster Admin.

3. Set IP addresses.

Specify /etc/hosts as below on all the nodes which configure the cluster system.

Example:

```
<cip address1>      shasta1RMS
<cip address2>      shasta2RMS
```

4. Enable remote access by using cfcf/cfsh.

Specify /etc/default/cluster.config as below on all the nodes which configure the cluster system.

```
CFCF "cfcf"
CFSH "cfsh"
```

5. Edit /etc/default/cluster on all the nodes.

- a. Edit /etc/default/cluster with the following contents:

- For physical environment, KVM environment and VMware environment

```
nodename <CF node name>
clustername <Cluster name>
device <Cluster interconnect1>
device <Cluster interconnect2>
```

Example:

```
nodename shasta1
clustername PRIMECLUSTER1
```

```
device eth2
device eth3
```

Note

Make sure that the node name to be defined in `nodename` is the CF node name, not the node name of the OS.

- For cluster nodes placed between different network segments, in a cloud environment (other than a single-node cluster), or in an RHOSP environment

```
nodename <CF node name>
clustername <Cluster name>
device <IP device1> <Self IP address1> <Self broadcast address1> <IP address of other node1>
device <IP device2> <Self IP address2> <Self broadcast address2> <IP address of other node2>
```

Example: `/etc/default/cluster of shasta1`

```
nodename shasta1
clustername PRIMECLUSTER1
device /dev/ip0 192.168.223.105 192.168.223.255 192.168.123.112
device /dev/ip1 192.168.200.105 192.168.200.255 192.168.100.112
```

Note

- Make sure that the node name to be defined in `nodename` is the CF node name, not the node name of the OS.
- For 3 nodes or more, separate the IP address of all nodes for the IP address of another node with a space.
- A maximum of 4 cluster interconnects can be used. When using multiple number of cluster interconnects, list the multiple number of device line. Specify different `/dev/ipX` (X is 0 to 3) for each IP device of each device line.

- For a single-node cluster in a cloud environment

```
nodename <CF node name>
clustername <Cluster name>
device /dev/ip0 <IP address of dummy interface> <Broadcast address of dummy interface>
```

Example: `/etc/default/cluster of shasta1`

```
nodename shasta1
clustername PRIMECLUSTER1
device /dev/ip0 192.168.223.105 192.168.223.255
```

Note

Make sure that the node name to be defined in `nodename` is the CF node name, not the node name of the OS.

b. Set the owner, group, and access permission.

```
# chown root:root /etc/default/cluster
# chmod 600 /etc/default/cluster
```

c. Reboot the nodes.

6. Execute the following command with any node in the cluster system and set the Cluster Integrity Monitor (CIM).

```
# rcqconfig -a <nodename> ...
```

`nodename` : CF node name

Example:

```
# rcqconfig -a shasta1 shasta2
```

If this command fails, check again that CF node names and cluster names configured in /etc/default/cluster in step 5 are correct.

7. Check that the cluster system can be communicated with the RMS node name.

Example: When checking from shasta1

```
# ping shasta2RMS
```

If the cluster system cannot be communicated, check again that CF node names, RMS node names, and CIP addresses configured in /etc/cip.cf and /etc/hosts in step 2 and step 3 are correct.

For cluster nodes placed between different network segments, in a cloud environment, or in an RHOSP environment, check the CF configuration with the following procedure:

1. Execute the following command on any one node to make sure that all the nodes have already joined in the cluster.

```
# cftool -n
```

Example: When CF node names are shasta1, shasta2 and when there are 2 nodes

```
# cftool -n
Node      Number State      Os      Cpu
shasta1   1      UP        Linux   EM64T
shasta2   2      UP        Linux   EM64T
```

Make sure that both shasta1 and shasta2 are displayed on "Node" and "UP" is displayed on "State" for each node.

2. Execute the following command on all the nodes to make sure that the CF over IP configuration is enabled.

```
# cftool -d
```

Example: For 2 cluster interconnects

```
# cftool -d
Number Device  Type Speed  Mtu    State Configured Address
4      /dev/ip0 6      n/a    1392   UP     YES      0a.00.00.c9.00.00
5      /dev/ip1 6      n/a    1392   UP     YES      0a.00.00.ca.00.00
```

Make sure that /dev/ipX (X is 0 to 3. Only the number of cluster interconnect is displayed) is only displayed in Device.

If the above step 1 or step 2 is not performed correctly, check the following configurations again.

- For cluster nodes placed between different network segments, or in an FJcloud-Baremetal environment

1. See "Appendix J Using Firewall" in "PRIMECLUSTER Installation and Administration Guide" and check if the configuration necessary to allow communication of CF over IP is properly set.
2. Check again that the CF node name, RMS node name, CIP address, IP address of cluster interconnect, IP device, broadcast address, and cluster name set in /etc/cip.cf, /etc/default/cluster, and /etc/hosts are correct.

- In a cloud environment (other than an FJcloud-Baremetal environment) or an RHOSP environment

1. Make sure that the configurations of the security groups (or security rules) created with the following references are properly set.
 - For an FJcloud-O environment, see "3.1.2.4 Creating the Security Group for the Cluster Interconnect" in "PRIMECLUSTER Installation and Administration Guide Cloud Services."
 - For a NIFCLOUD environment, see "8.3.2.2 Rules Applied to the Cluster Interconnect" in "PRIMECLUSTER Installation and Administration Guide Cloud Services."
 - For an AWS environment, see "20.3.2.2 Rules Applied to the Cluster Interconnect" in "PRIMECLUSTER Installation and Administration Guide Cloud Services."
 - For an Azure environment, see "26.3.2.2 Security Rules Applied to the Cluster Interconnect" in "PRIMECLUSTER Installation and Administration Guide Cloud Services."

- For an RHOSP environment, see "H.2.2.2 Creating Virtual Network" in "PRIMECLUSTER Installation and Administration Guide."
2. When creating multiple virtual network interfaces and using a virtual network interface, for which a default gateway is not set, to communicate with interfaces on different subnets, see "21.1.3 Setting Instances" in "PRIMECLUSTER Installation and Administration Guide Cloud Services" and make sure that the configuration of the routing is set so that communication between interconnects can be performed.
 3. Check again that the CF node name, RMS node name, CIP address, IP address of cluster interconnect, IP device, broadcast address, and cluster name set in `/etc/cip.cf`, `/etc/default/cluster`, and `/etc/hosts` are correct.

1.2 CIP configuration file

The CIP configuration file is stored in `/etc/cip.cf` on each node in the cluster. Normally, you can use the GUI to create this file during cluster configuration time. However, there may be times when you wish to manually edit this file.

The format of a CIP configuration file entry is as follows:

```
cfname CIP_Interface_Info [ CIP_Interface_Info... ] [ IPv6 ]
```

- *cfname* tells what node the configuration information is for.
- *CIP_Interface_Info* gives information needed to configure a single CIP interface.

Normally, the configuration information of all the CIP interfaces on all the nodes is contained in the `cip.cf` configuration file.

- For IPv4, specify *CIP_Interface_Info* with the following format:

```
IPv4-Address[:netmask:<Netmask>]
```

Specify it without any spaces even around colons.

For IPv4-Address, specify as a number in Internet standard dotted-decimal notation or as the Host name. When specifying with the Host name, it needs to be defined in `/etc/hosts`.

For *<Netmask>*, specify the netmask value to be set to the IP address as a number in Internet standard dotted-decimal notation.

- For IPv6, specify *CIP_Interface_Info* with the following format:

```
Hostname: [" IPv6-Address/prefix_length ]"
```

Specify it without any spaces around colons, slashes, and inside of each brackets "[", "]"

For Hostname, describe the Host name to specify the cip address.

For IPv6-Address and *prefix_length*, specify the IPv6 address and the prefix length denoted as a hexadecimal code which is separated by Internet standard colons.

- When using the IPv6 address, specify "IPv6" in the end of the line.

For example, the CIP configuration done in the section "[1.1.4 Example of creating a cluster](#)" would produce the following CIP configuration file:

```
shasta1 shasta1RMS:netmask:255.255.255.0
shasta2 shasta2RMS:netmask:255.255.255.0
```

Although not shown in this example, the CIP syntax does allow multiple CIP interfaces for a node to be defined on a single line. The `cip.cf` manual page has more details about the `cip.cf` file.

If you make changes to the `cip.cf` file by hand, you should be sure that the file exists on all the nodes, and all the nodes are specified in the file. Be sure to update all the nodes in the cluster with the new file. Changes to the CIP configuration file will not take effect until CIP is stopped and restarted.

After stopping all applications that use CIP, restart CIP by stopping and starting CF.

For instructions on starting and stopping CF, see the section "[4.6 Starting and stopping CF](#)".

Chapter 2 CF Registry and Integrity Monitor

This chapter discusses the CF registry (CFREG) and the Cluster Integrity Monitor (CIM).

2.1 CF Registry (CFREG)

The CFREG provides a set of CF base product services that allows cluster applications to maintain cluster global data that must be consistent on all of the nodes in the cluster and must live through a clusterwide reboot.

Typical applications include cluster-aware configuration utilities that require the same configuration data to be present and consistent on all of the nodes in a cluster (for example, cluster volume management configuration data).

The data is maintained as named registry entries residing in a data file where each node in the cluster has a copy of the data file. The services will maintain the consistency of the data file throughout the cluster.

A user-level daemon (cfregd), runs on each node in the cluster, and is responsible for keeping the data file on the node where it is running synchronized with the rest of the cluster. The cfregd process will be the only process that ever modifies the data file. Only one synchronization daemon process will be allowed to run at a time on a node. If a daemon is started with an existing daemon running on the node, the started daemon will log messages that state that a daemon is already running and terminate itself. In such a case, all execution arguments for the second daemon will be ignored.

2.2 Cluster Integrity Monitor(CIM)

The purpose of the CIM is to allow applications to determine when it is safe to perform operations on shared resources. It is safe to perform operations on shared resources when a node is a member of a cluster that is in a consistent state.

A consistent state is means that all the nodes of a cluster that are members of the CIM set are in a known and safe state. The nodes that are members of the CIM set are specified in the CIM configuration. Only these nodes are considered when the CIM determines the state of the cluster. When a node first joins or forms a cluster, the CIM indicates that the cluster is consistent only if it can determine the status of the other nodes that make up the CIM set and that those nodes are in a safe state.

As methods for guaranteeing a quorum, PRIMECLUSTER supports the following three CIM methods:

- Node State Management (NSM) method
- MMB method

The NSM method uses the state of the other node as reported by CF and periodically determines whether that node is in a known state. If the state is known, The CF states that are determined as known are UP and DOWN. If the node is being activated or is in LEFTCLUSTER state, that state is determined to be unknown (a quorum is not guaranteed).

The MMB method uses the PRIMEQUEST MMB interface and asynchronously determines the state of a cluster node.

In this way, the CIM provides applications with a set of functions that determine nodes are in a consistent state (whether or not there is a quorum).

PRIMECLUSTER uses CIM to determine whether a user application process that uses resources shared by multiple nodes in a cluster can be processed safely without triggering process contention. In other words, it is safe to perform operations on shared resources if the node executing the process is a member of a cluster system that is in a consistent state (quorum). In the PRIMECLUSTER system, the consistent state is set when all the nodes in the cluster system monitored by the CIM are either in operating (UP) or stopped (DOWN) state and are also in a safe state. The nodes monitored by the CIM to all the nodes that were set when the CIM was configured. Only these nodes are considered when the CIM checks the cluster status. When a node first joins or forms a cluster, the CIM indicates that the cluster is consistent only if it can determine the status of the other nodes that make up the CIM set and that those nodes are in a safe state.

CIM currently supports the Node State Management (NSM) method. The CIM reports on a cluster state that a node state is known and safe (True), or a node state is unknown (False) for the node. True and False are defined as follows:

True - All CIM nodes in the cluster are in a known and safe state.

False - One or more CIM nodes in the cluster are in an unknown or unsafe state.

2.2.1 Configuring CIM

You can perform CIM procedures through the following methods:

- Cluster Admin GUI

This is the preferred method of operation. Refer to the section "Adding and removing a node from CIM" for the GUI procedures.

- CLI

Refer to the manual pages for complete details on the CLI options and arguments. The commands can be found in the following directory:

```
/opt/SMAW/SMAWcf/bin
```

CLI

The CIM is configured using the command `rcqconfig(1M)` after CF starts. The `rcqconfig(1M)` command is used to set up or to change the CIM configuration. Make sure to run this command manually when the cluster is run on CLI.

When `rcqconfig(1M)` is invoked, it checks that the node is part of the cluster. When the `rcqconfig(1M)` command is invoked without any option, it checks if any configuration is present in the `CFReg.database` after the node joins the cluster. If there is none, it returns as error. When you are using the GUI, these actions are done as part of the configuration process.

`rcqconfig(1M)` configures a quorum set of nodes, among which CF decides the quorum state. `rcqconfig(1M)` is also used to show the current configuration. If `rcqconfig(1M)` is invoked without any configuration changes or with only the `-v` option, `rcqconfig(1M)` will apply any existing configuration to all the nodes in the cluster. It will then start or restart the quorum operation. `rcqconfig(1M)` can be invoked from the command line to configure or to start the quorum.

2.2.2 Query of the quorum state

CIM recalculates the quorum state when it is triggered by some node state change. However you can force the CIM to recalculate it by running `rcqquery(1M)` at any time. Refer to the manual pages for complete details on the CLI options and arguments.

`rcqquery(1M)` functions as follows:

- Queries the state of quorum and gives the result using the return code. It also gives you readable results if the verbose option is given.
- Returns True if the states of all the nodes in the quorum set are known. If the state of any node is unknown, then it returns False.
- Exits with a status of zero when a quorum exists, and it exits with a status of 1 when a quorum does not exist. If an error occurs during the operation, then it exits with any other non-zero value other than 1.

2.2.3 Reconfiguring quorum

Refer to the section "Adding and removing a node from CIM" for the GUI procedures.

CLI

The configuration can be changed at any time and is effective immediately. When a new node is added to the quorum set of nodes, the node being added must be part of the cluster so as to guarantee that the new node also has the same quorum configuration. Removing a node from the quorum set can be done without restriction.

When the configuration information is given to the command `rcqconfig(1M)` as arguments, it performs the transaction to CFREG to update the configuration information. Until CIM is successfully configured and gets the initial state of the quorum, CIM has to respond with the quorum state of False to all queries.



Example

In this example the cluster has formed but no quorum was established.

- Display the states of all the nodes in the cluster as follows:

```
shastal# cftool -n
Node   Number  State  Os      Cpu
shastal 1       UP     Linux   Pentium
shasta2 2       UP     Linux   Pentium
```

- Display the current quorum configuration as follows:

```
shastal# rcqconfig -g
```

Nothing is displayed, since no nodes have been added so far.

- Add new nodes in a quorum set of nodes as follows:

```
shastal# rcqconfig -a shastal shasta2
```

- Display the current quorum configuration parameters as follows:

```
shastal# rcqconfig -g
QUORUM_NODE_LIST= shastal shasta2
```

- Delete nodes from a quorum set of nodes as follows:

```
shastal# rcqconfig -d shastal
```

- Display the current quorum configuration parameters after one node is deleted as follows:

```
shastal# rcqconfig -g
QUORUM_NODE_LIST= shasta2
```

- Add a new node, shasta10 (which is not in the cluster), in a quorum set of nodes as follows:

```
shastal# rcqconfig -a shastal shasta2 shasta10
Cannot add node shasta10 that is not up.
```

- Since CF only configured the cluster to consist of shastal and shasta2, shasta10 does not exist. The quorum set remains unchanged.

```
shastal# rcqconfig -g
QUORUM_NODE_LIST= shasta2
```

Chapter 3 Cluster resource management

This chapter discusses the Resource Database, which is a synchronized clusterwide database.

The cluster Resource Database is intended to be used only by PRIMECLUSTER products. It is not a general purpose database which a customer could use for their own applications.

3.1 Resource Database configuration

This section discusses how to set up the Resource Database for the first time on a new cluster. The following procedure assumes that the Resource Database has not previously been configured on any of the nodes in the cluster.

Before you begin configuring the Resource Database, you must first make sure that CIP is properly configured on all the nodes. The Resource Database uses CIP for communicating between nodes, so it is essential that CIP is working.

The Resource Database also uses the CIP configuration file `/etc/cip.cf` to establish the mapping between the CF node name and the CIP name for a node. If a particular node has multiple CIP interfaces, then only the first one is used. This will correspond to the first CIP entry for a node in `/etc/cip.cf`. It will also correspond to `cip0` on the node itself.

Because the Resource Database uses `/etc/cip.cf` to map between CF and CIP names, it is critical that this file be the same on all the nodes. If you used the Cluster Admin CF Wizard to configure CIP, then this will already be the case. If you created some `/etc/cip.cf` files by hand, then you need to make sure that all the nodes are specified and they are the same across the cluster.

In general, the CIP configuration is fairly simple. You can use the Cluster Admin CF Wizard to configure a CIP subnet after you have configured CF. If you use the Wizard, then you will not need to do any additional CIP configuration. See the section "[1.1 CF, CIP, and CIM configuration](#)" for more details.

After CIP has been configured, you can configure the Resource Database on a new cluster by using the following procedure. This procedure must be done on all the nodes in the cluster.

1. Log in to the node with system administrator authority.
2. Verify that the node can communicate with other nodes in the cluster over CIP.
To test CIP network connectivity, execute the `ping(1M)` command or the `ping6(8)` command (when using the IPv6 address). The file `/etc/cip.cf` contains the CIP names that you should use in the `ping(1M)` command or the `ping6(8)` command.

If you are using RMS and you have only defined a single CIP subnetwork, then the CIP names will be of the following form:

```
cfnameRMS
```

For example, if you have two nodes in your cluster named `shasta1` and `shasta2`, then the CIP names for RMS would be `shasta1RMS` and `shasta2RMS`, respectively. You could then run the following commands:

```
shasta1# ping shasta2RMS
shasta2# ping shasta1RMS
```

This tests the CIP connectivity.

3. Execute the `clsetup` command. When used for the first time to set up the Resource Database on a node, it is called without any arguments as follows:

```
# /etc/opt/FJSVcluster/bin/clsetup
```

4. Execute the `clgettree` command to verify that the Resource Database was successfully configured on the node, as shown in the following:

```
# /etc/opt/FJSVcluster/bin/clgettree
```

The command should complete without producing any error messages, and you should see the Resource Database configuration displayed in a tree format.

For example, on a two-node cluster consisting of `shasta1` and `shasta2`, the `clgettree` command might produce output similar to the following:

```
Cluster 1 cluster
  Domain 2 Domain0
    Shared 7 SHD_Domain0
      Node 3 shasta1 UNKNOWN
      Node 5 shasta2 UNKNOWN
```

If you need to change the CIP configuration to fix the problem, you will also need to run the *clinitreset* command and start the information process over.

The format of *clgettree* is more fully described in its manual page. For the purpose of setting up the cluster, you need to check the following:

- Each node in the cluster should be referenced in a line that begins with the word Node.
- The *clgettree* output must be identical on all the nodes.

If either of the above conditions is not met, then it is possible that you may have an error in the CIP configuration. Double-check the CIP configuration using the methods described earlier in this section. The actual steps are as follows:

1. Make sure that CIP is properly configured and running.
2. Run *clinitreset* on all the nodes in the cluster.

```
# /etc/opt/FJSVcluster/bin/clinitreset
```

3. Reboot each node.
4. Rerun the *clsetup* command on each node.

```
# /etc/opt/FJSVcluster/bin/clsetup
```

5. Use the *clgettree* command to verify the configuration.

```
# /etc/opt/FJSVcluster/bin/clgettree
```

3.2 Startup synchronization

A copy of the Resource Database is stored locally on each node in the cluster. When the cluster is up and running, all of the local copies are kept in sync. However, if a node is taken down for maintenance, then its copy of the Resource Database may be out of date by the time it rejoins the cluster. Normally, this is not a problem. When a node joins a running cluster, then its copy of the Resource Database is automatically downloaded from the running cluster. Any stale data that it may have had is thus overwritten.

There is one potential problem. Suppose that the entire cluster is taken down before the node with the stale data had a chance to rejoin the cluster. Then suppose that all the nodes are brought back up again. If the node with the stale data comes up long before any of the other nodes, then its copy of the Resource Database will become the master copy used by all the nodes when they eventually join the cluster.

To avoid this situation, the Resource Database implements a startup synchronization procedure. If the Resource Database is not fully up and running anywhere in the cluster, then starting the Resource Database on a node will cause that node to enter into a synchronization phase. The node will wait up to `StartingWaitTime` seconds for other nodes to try to bring up their own copies of the Resource Database. During this period, the nodes will negotiate among themselves to see which one has the latest copy of the Resource Database. The synchronization phase ends when either all the nodes have been accounted for or `StartingWaitTime` seconds have passed. After the synchronization period ends, the latest copy of the Resource Database that was found during the negotiations will be used as the master copy for the entire cluster.

The default value for `StartingWaitTime` is 60 seconds.

This synchronization method is intended to cover the case where all the nodes in a cluster are down, and then they are all rebooted together. For example, some businesses require high availability during normal business hours, but power their nodes down at night to reduce their electric bill. The nodes are then powered up shortly before the start of the working day. Since the boot time for each node may vary slightly, the synchronization period of up to `StartingWaitTime` ensures that the latest copy of the Resource Database among all of the booting nodes is used.

Another important scenario in which all the nodes may be booted simultaneously involves the temporary loss and then restoration of power to the lab where the nodes are located.

However, for this scheme to work properly, you must verify that all the nodes in the cluster have boot times that differ by less than StartingWaitTime seconds. Furthermore, you might need to modify the value of StartingWaitTime to a value that is appropriate for your cluster.

Modify the value of StartingWaitTime as follows:

1. Start up all of the nodes in your cluster simultaneously. It is recommended that you start the nodes from a cold power on. Existing nodes are not required to reboot when a new node is added to the cluster.
2. After the each node has come up, look in /var/log/messages for message number 2200. This message is output by the Resource Database when it first starts. For example, enter the following command:

```
# grep 2200 /var/log/messages
Feb 23 19:00:41 shastal dcmond[407]: [ID 888197 daemon.notice] FJSVcluster: INFO: DCM: 2200:
Cluster configuration management facility initialization started.
```

Compare the timestamps for the messages on each node and calculate the difference between the fastest and the slowest nodes. This will tell you how long the fastest node has to wait for the slowest node.

3. Check the current value of StartingWaitTime by executing the clsetparam command on any of the nodes. For example, enter the following command:

```
# /etc/opt/FJSVcluster/bin/clsetparam -p StartingWaitTime
60
```

The output for our example shows that StartingWaitTime is set to 60 seconds.

4. If there is a difference in startup times found in Step 2, the StartingWaitTime, or if the two values are relatively close together, then you should increase the StartingWaitTime parameter. You can do this by running the clsetparam command on any one node in the cluster. For example, enter the following command:

```
# /etc/opt/FJSVcluster/bin/clsetparam -p StartingWaitTime 300
```

This sets the StartingWaitTime to 300 seconds.

3.2.1 Startup synchronization and the new node

After the Resource Database has successfully been brought up on the new node, then you need to check if the StartingWaitTime used for startup synchronization is still adequate. If the new node boots much faster or slower than the other nodes, then you may need to adjust the StartingWaitTime time.

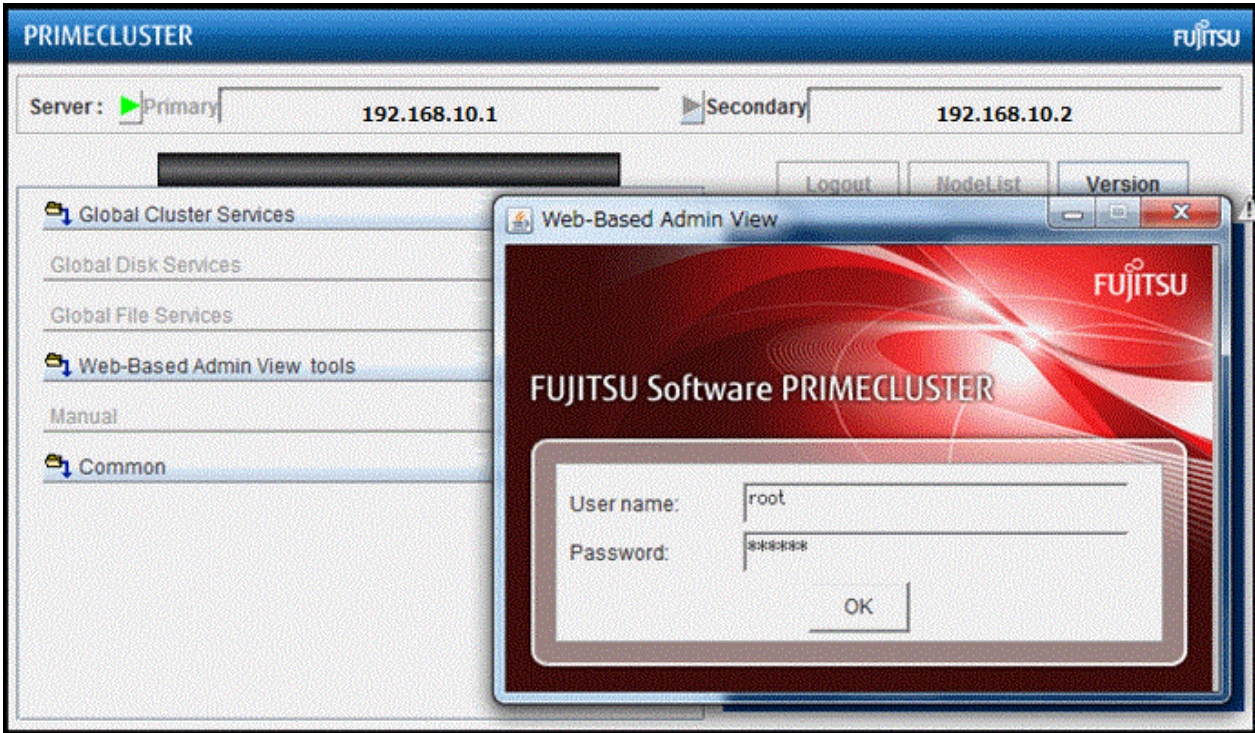
Chapter 4 GUI administration

This chapter covers the administration of features in the Cluster Foundation (CF) portion of Cluster Admin.

4.1 Starting Cluster Admin GUI and logging in

By starting the screen from the shortcut of the Java application (PRIMECLUSTER Web-Based Admin View Startup), the Web-Based Admin View main screen below is displayed.

Figure 4.1 Cluster Admin start-up window



Enter a user name in the User name field and the password and click on OK.

Use the appropriate privilege level while logging in. There are three privilege levels: root privileges, administrative privileges, and operator privileges.

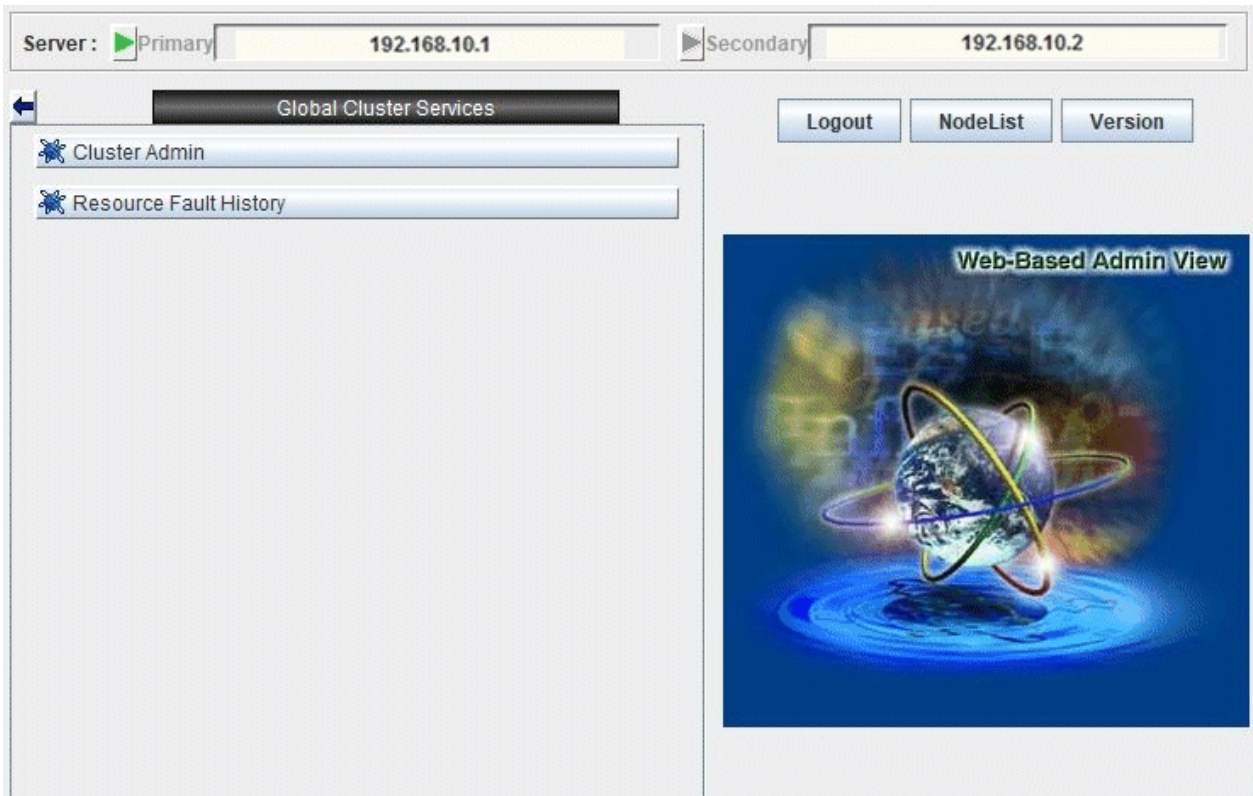
With the root privileges, you can perform all actions including configuration, administration and viewing tasks. With administrative privileges, you can view as well as execute commands but cannot make configuration changes. With the operator privileges, you can only perform viewing tasks.

Point

In this example we are using root and not creating user groups.

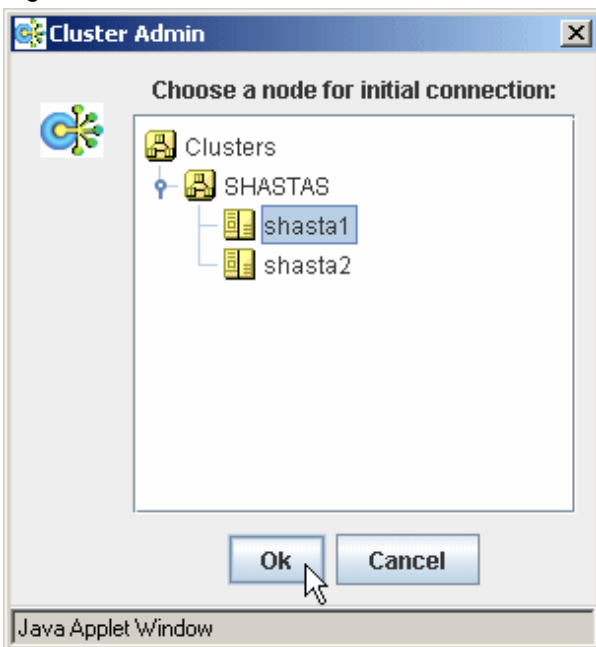
On the "Figure 4.1 Cluster Admin start-up window," click the Global Cluster Services button and the Cluster Admin button appears.

Figure 4.2 Top menu



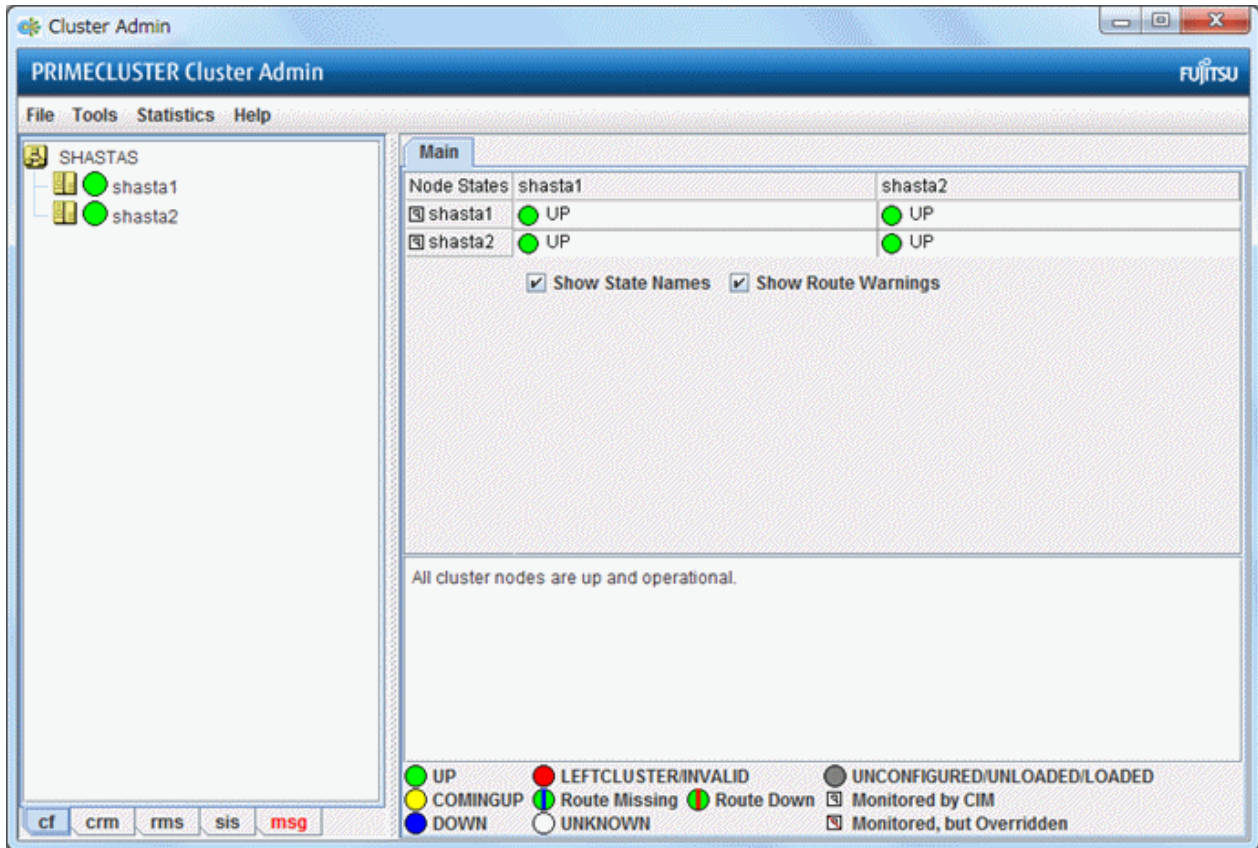
Click the Cluster Admin button. The Choose a node for initial connection window appears.

Figure 4.3 Initial connection choice window



Select a node and click on OK. The Cluster Admin main window appears.

Figure 4.4 Cluster Admin main window



By default, the cf tab is selected and the CF main window is presented. Use the appropriate privilege level while logging in.

4.2 Main CF table

When the GUI is first started, or after the successful completion of the configuration wizard, the main CF table will be displayed in the right panel. A tree showing the cluster nodes will be displayed in the left panel (see "Figure 4.4 Cluster Admin main window.")

The tree displays the local state of each node, but does not give information about how that node considers other nodes. If two or more nodes disagree about the state of a node, one or more colored exclamation marks appear next to the node. Each exclamation mark represents the node state of which another node considers that node to be.

The table in the right panel is called the main CF table. The column on the left of the table lists the CF states of each node of the cluster as seen by the other nodes in the cluster. For instance, the cell in the second row and first column is the state of shasta2 as seen by the node shasta1.

There is an option at the bottom of the table to toggle the display of the state names. The [Show State Names] checkbox is selected by default. If there are a large number of nodes in the cluster, once the [Show State Names] checkbox is unchecked, the table will display the node names vertically to allow a larger number of nodes to be displayed.

There are two types of CF states:

- Local states are the states a node can consider itself in.
- Remote states are the states a node can consider another node to be in.

The following table lists these various states.

Table 4.1 Local states

CF state		Description
UNLOADED	●	The node does not have a CF driver loaded.
LOADED	●	The node has a CF driver loaded, but is not running.

CF state		Description
COMINGUP	●	The node is in the process of starting and should be UP soon.
UP	●	The node is up and running normally.
INVALID	●	The node has an invalid configuration and must be reconfigured.
UNKNOWN	○	The GUI has no information from this node. This can be temporary, but if it persists, it probably means the GUI cannot contact that node.
UNCONFIGURED	●	The CF driver is loaded but the node is not yet configured to run CF.

The following table lists the remote states.

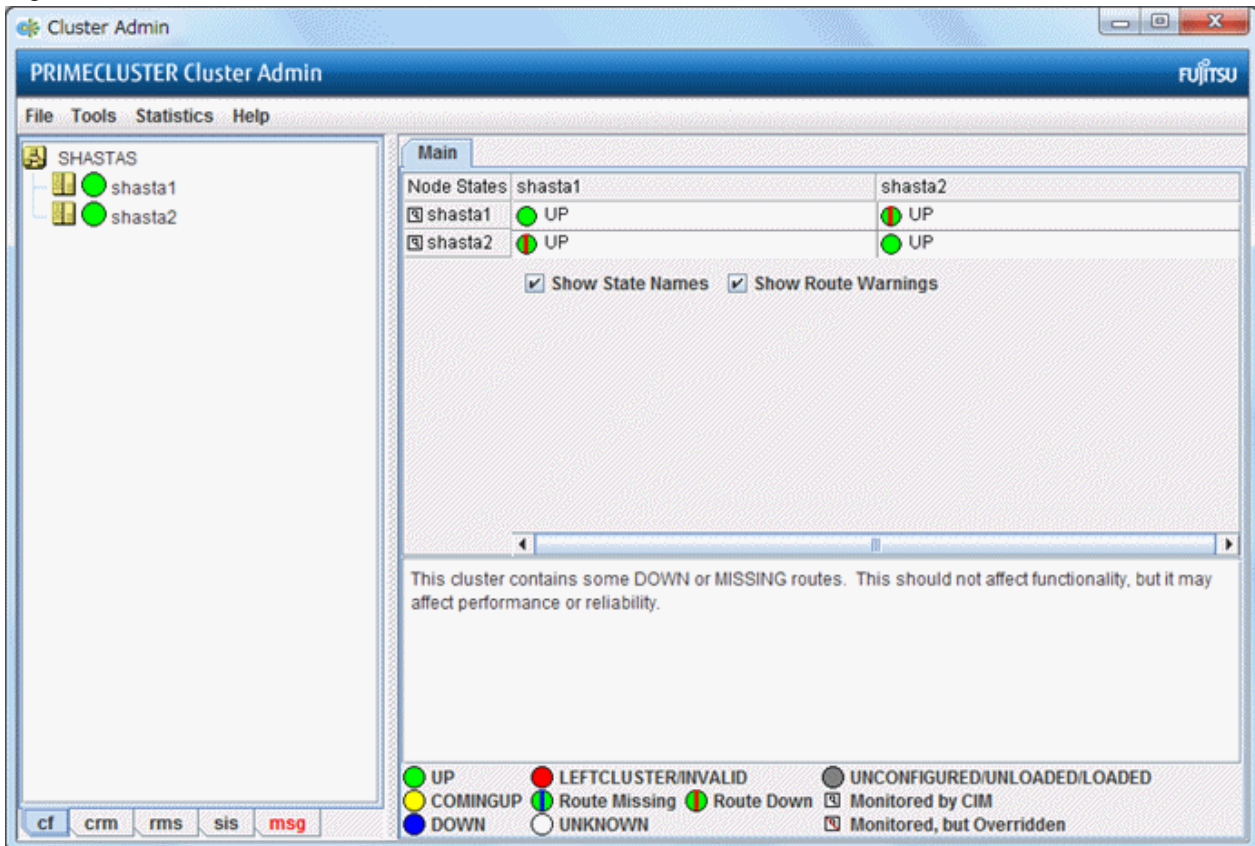
Table 4.2 Remote states

CF state		Description
UP	●	The node is up and part of this cluster.
DOWN	●	The node is down and not in the cluster.
UNKNOWN	○	The reporting node has no opinion on the reported node.
LEFTCLUSTER	●	The node has left the cluster unexpectedly, probably from a crash. To ensure cluster integrity, it will not be allowed to rejoin until marked DOWN.

4.3 CF route tracking

If a node is UP, but it has one or more DOWN routes, the green circle in the main CF table will have a red line through it (see the following.)

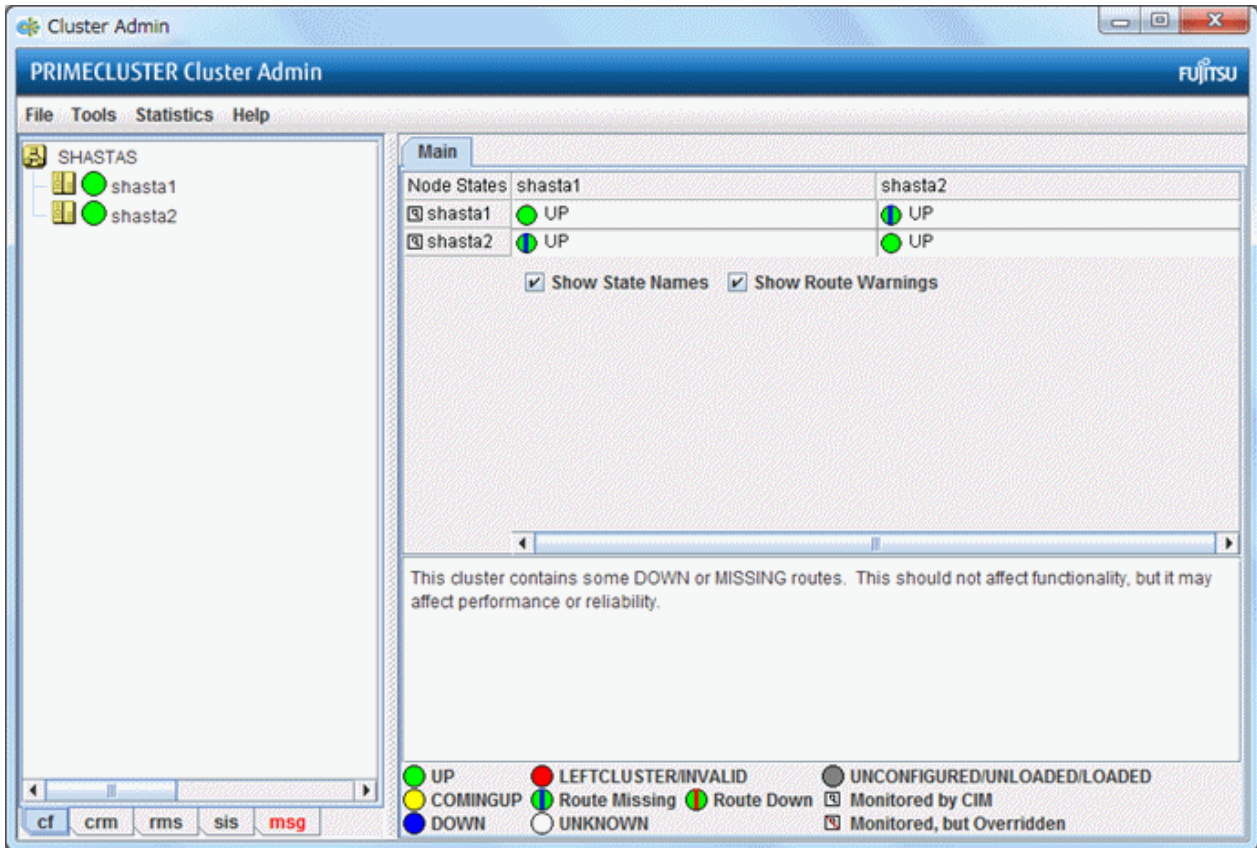
Figure 4.5 CF route DOWN



In this example, one of the network interfaces on shasta1 has been unplugged. Cluster Admin, therefore, shows that a route is DOWN. Since shasta2 cannot contact shasta1 over that interface, it also shows that there is a route down on shasta1. To see which routes are DOWN, click on the node in the left-panel tree and look at the route table.

If CF starts with one or more interfaces missing, then the green circle in the main CF table will have a blue line through it (see the following screen.)

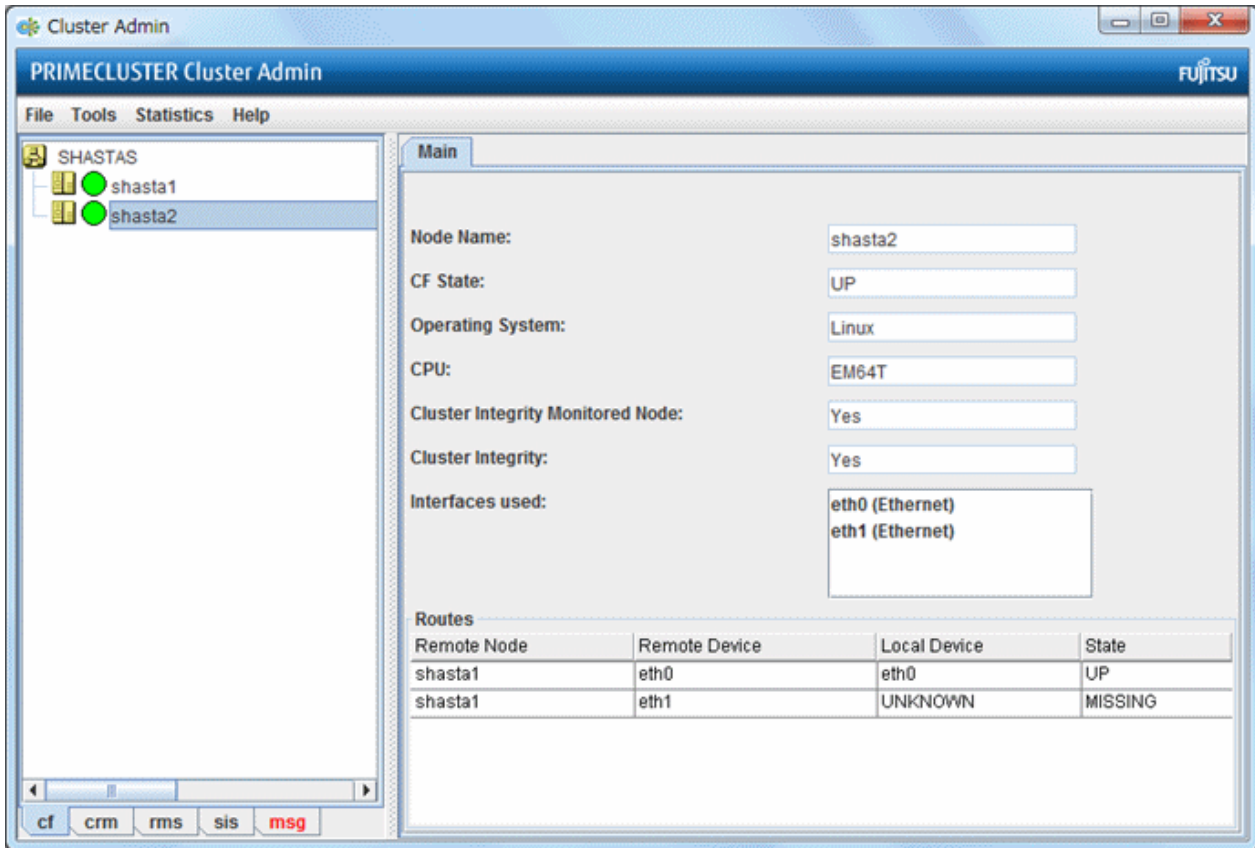
Figure 4.6 CF interface missing



In the above example, shasta2 has a broken connection to shasta1, and Cluster Admin indicates that a route is missing.

In our example, clicking on shasta1 in the left-panel tree shows that there is no route from shasta1 to the eth2 interface on shasta2 (see the following.)

Figure 4.7 CF route table



4.4 Node details

To get detailed information on a cluster node, left-click on the node in the left tree. This replaces the main table with a display of detailed information. (To bring the main table back, left-click on the cluster name in the tree.) This is the example of how the panel is displayed.

Figure 4.8 CF node information

Main

Node Name:

CF State:

Operating System:

CPU:

Cluster Integrity Monitored Node:

Cluster Integrity:

Interfaces used:

eth0 (Ethernet)
eth2 (Ethernet)

Routes

Remote Node	Remote Device	Local Device	State
shasta2	eth0	eth0	UP
shasta2	eth2	eth2	UP

Shown are the node's name, its CF state(s), operating system, platform, and the interfaces configured for use by CF. The states listed will be all of the states the node is considered to be in. For instance, if the node considers itself UNLOADED and other nodes consider it DOWN, DOWN/UNLOADED will be displayed.

The bottom part of the display is a table of all of the routes being used by CF on this node. It is possible for a node to have routes go down if a network interface or interconnect fails, while the node itself is still accessible.

4.5 Displaying the topology table

To examine and diagnose physical connectivity in the cluster, select Tools -> Topology. This menu option will produce a display of the physical connections in the cluster. This produces a table with the nodes shown along the left side and the interconnects of the cluster shown along the top. Each cell of the table lists the interfaces on that node connected to the interconnect. There is also a checkbox next to each interface showing if it is being used by CF. This table makes it easy to locate cabling errors or configuration problems at a glance.

This is the example of the topology table.

Figure 4.9 CF topology table

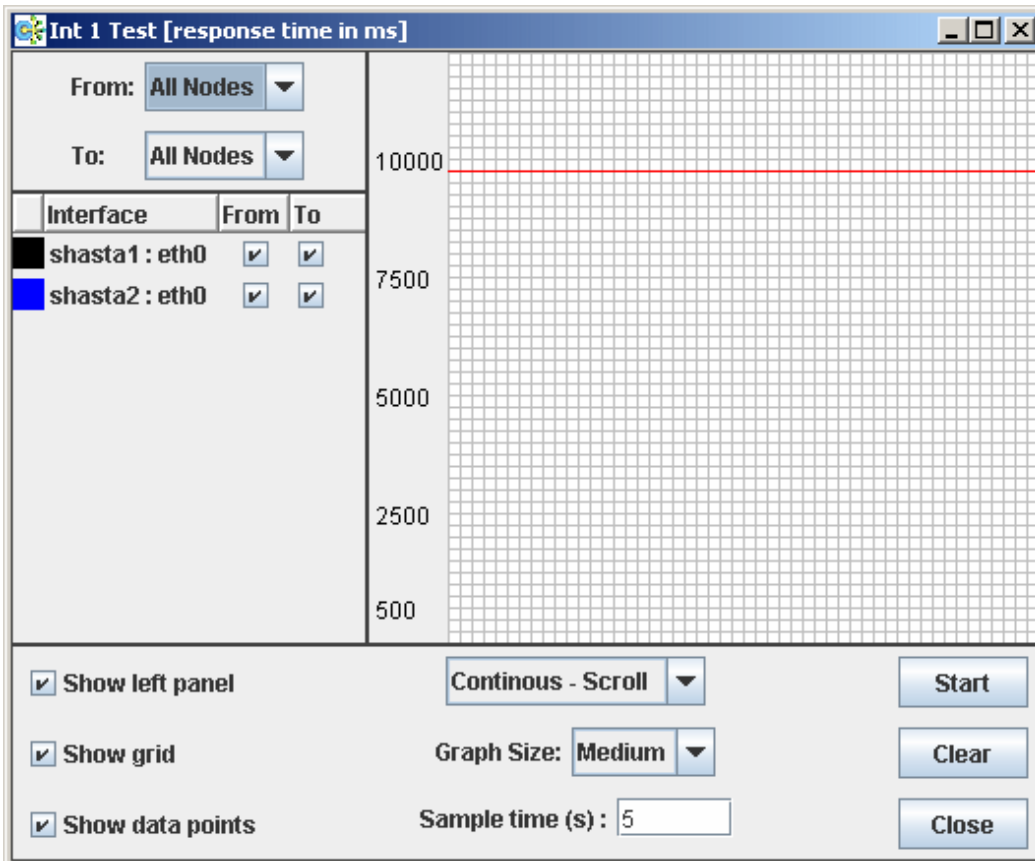
SHASTAS: Topology			
SHASTAS	Full Interconnects		
	<input checked="" type="checkbox"/> Int 1 <input type="button" value="Test"/>	<input checked="" type="checkbox"/> Int 2 <input type="button" value="Test"/>	<input checked="" type="checkbox"/> Int 3 <input type="button" value="Test"/>
shasta1	<input type="checkbox"/> eth0	<input checked="" type="checkbox"/> eth1	<input type="checkbox"/> eth2
shasta2	<input type="checkbox"/> eth0	<input checked="" type="checkbox"/> eth1	<input checked="" type="checkbox"/> eth2

This table displays the physical connectivity of the nodes in this cluster. This information is current as of (10:54:58 AM) and will not update. Nodes marked with a * will only show interfaces that are configured.

Pressing the Test button launches the Response Time monitor.

This tool allows you to see the response time for any combination of two nodes on that interconnect (see the following.).

Figure 4.10 Response Time monitor



The Y axis is the response time for CF pings in milliseconds and the X axis is a configurable period. The red line is the upper limit of the response time before CF will declare nodes to be in the LEFTCLUSTER state.

The controls to the left of the graph determine the nodes for which the graph displays data as follows:

- Set the selection boxes at the top to a specific node name, or to All Nodes.
- Select the check boxes next to the node names to specify specific nodes.

The controls on the left of the bottom panel control how the graphing and information collection is done as follows:

- Check the Show left panel check box to hide the left panel to provide more room for the graph.
- Check the Show grid check box to turn the grid on and off.
- Check the Show data points check box to display a simple line graph.

The controls in the middle of the bottom panel are as follows:

- The top drop-down menu controls how the graph is drawn. The following options are available:
 - Continuous-Scroll - Creates a continuous graph, so that when there are more data points than space, the graph scrolls.
 - Continuous-Clear - Graphs continuously until the graph is full, and then it starts a new graph.
 - Single Graph - Draws a single graph only.
- Graph size - Allows you to control how many data points are drawn.
- Sample time - Controls how often data points are taken.
- The buttons on the lower right control starting and stopping of the graph, clearing it, and closing the graph window.

The buttons on the right of the bottom panel are as follows:

- Start/Stop - Starts or stops the Response Time Monitor.

- Clear - Clears the data and starts a new graph.
- Close - Closes the Response Time Monitor and returns you to the CF Main screen.

Note

The Response Time Monitor is a tool for expert users such as consultants or skilled customers. Its output must be interpreted carefully. The Response Time Monitor uses user-space CF pings to collect its data. If the CF traffic between nodes in a cluster is heavy, then the Response Time Monitor may show slow response times, even if the cluster and the interconnects are working properly. Likewise, if a user does CF pings from the command line while the Response Time Monitor is running, then the data may be skewed.

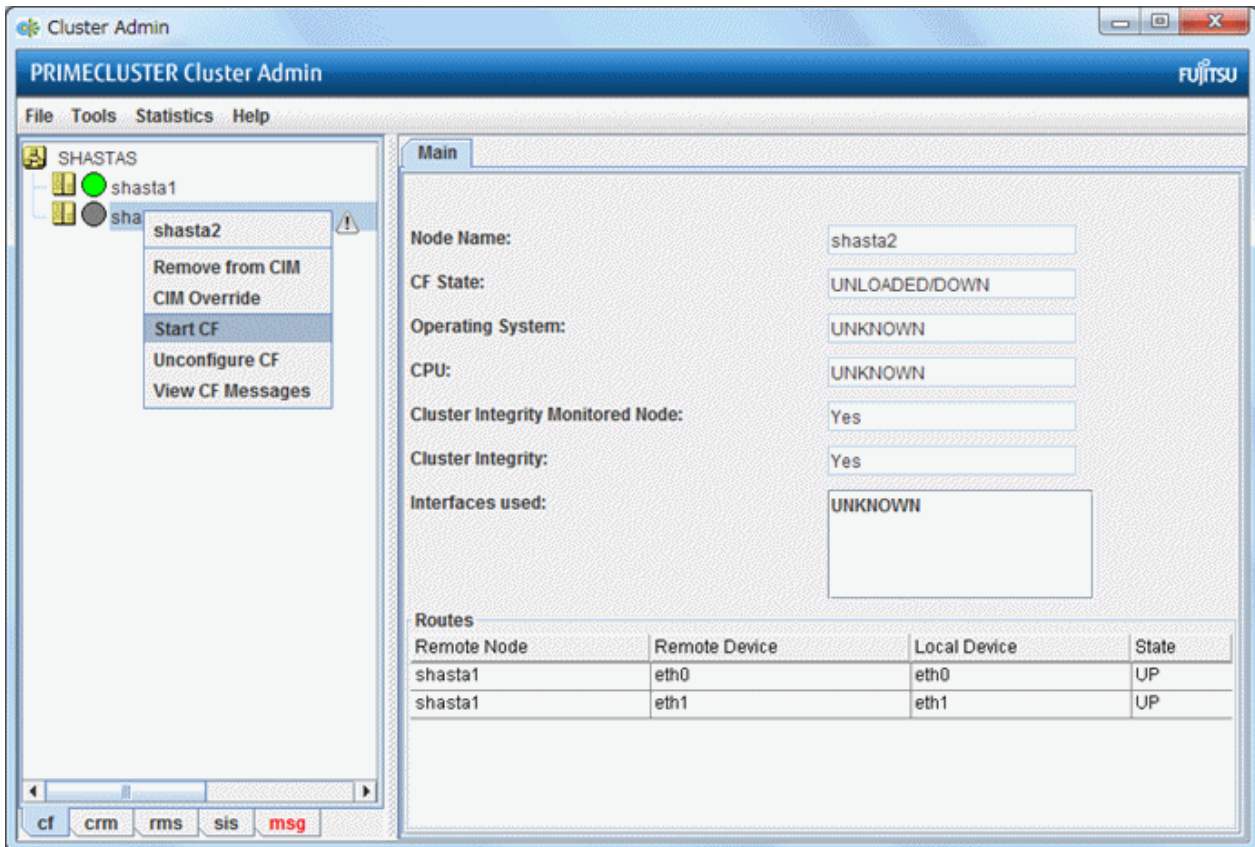
For best results, the Response Time Monitor should be run at times when CF traffic is relatively light, and the CF nodes are only lightly loaded.

4.6 Starting and stopping CF

There are two ways that you can start or stop CF from the GUI. The first is to simply right-click on a particular node in the tree in the left-hand panel. A state sensitive pop-up menu for that node will appear. If CF on the selected node is in a state where it can be started (or stopped), then the menu choice Start CF (or Stop CF) will be offered.

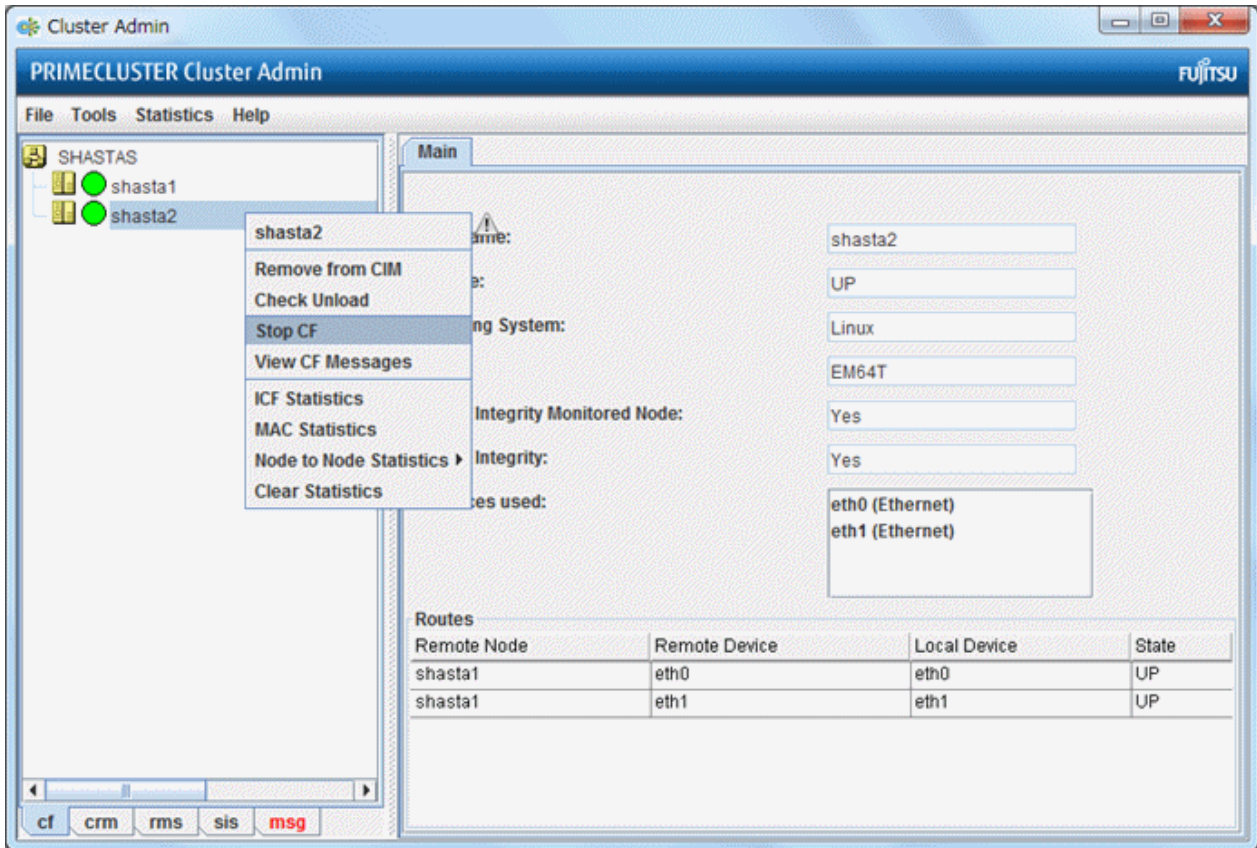
The following screen shows the content-sensitive menu pop-up when you select Start CF.

Figure 4.11 Starting CF



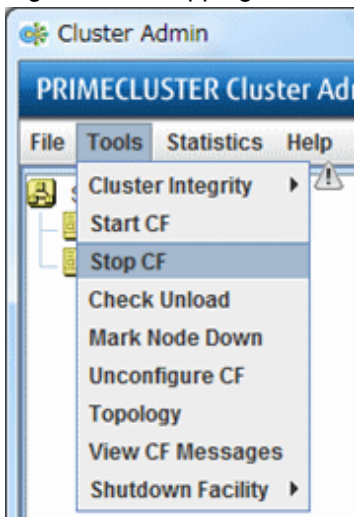
You can also go to the Tools pull-down menu and select either Start CF or Stop CF.

Figure 4.12 Starting and stopping CF from the Tools menu



A pop-up listing all the nodes where CF may be started or stopped will appear (see below.). You can then select the desired node to carry out the appropriate action.

Figure 4.13 Stopping CF and shutting down all the nodes



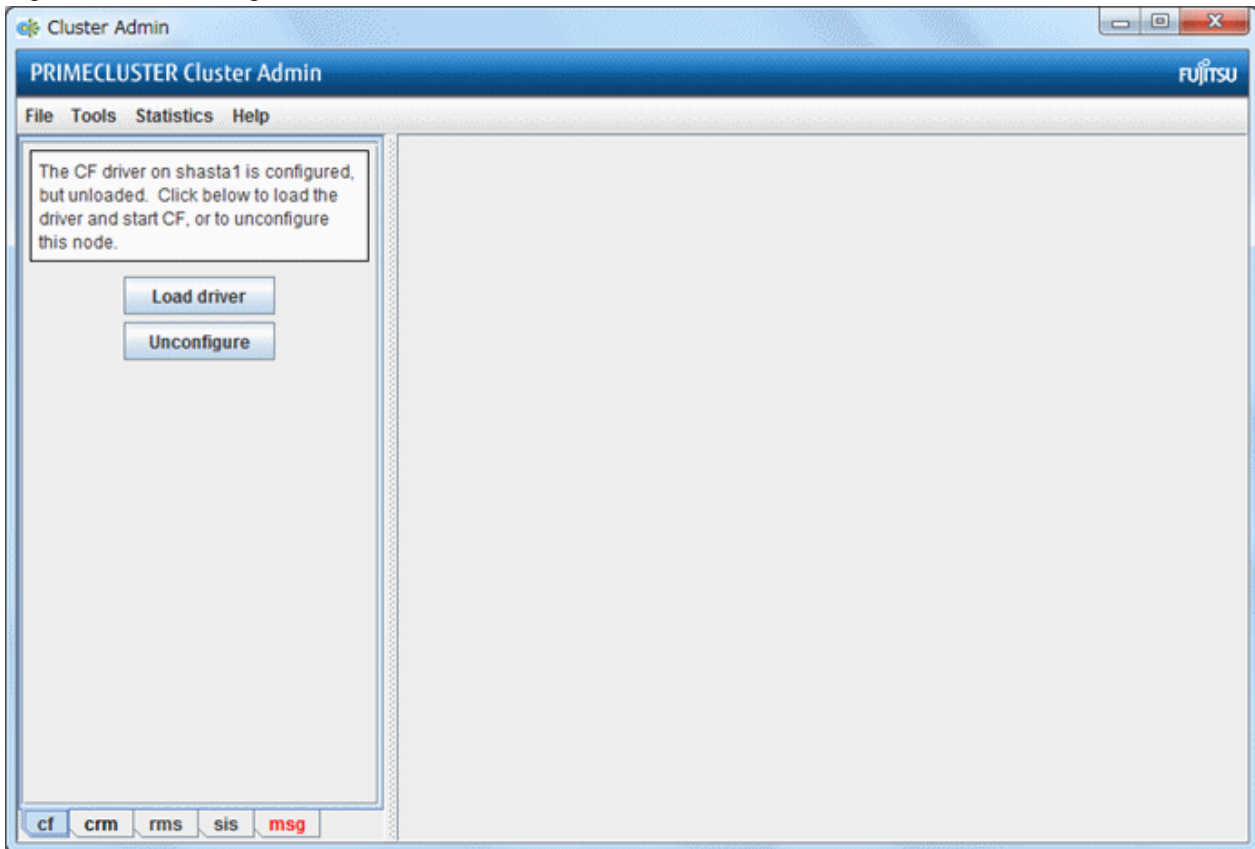
The CF GUI gets its list of CF nodes from the node you selected as the initial connection node that is shown in "[Figure 4.3 Initial connection choice window](#)". If CF is not up and running on the initial connection node, then the CF GUI will not display the list of nodes in the tree in the left panel.

Because of this, if you do not choose the all nodes option, and you want to stop CF on multiple nodes (including the initial node) by means of the GUI, ensure that the initial connection node is the last one on which you stop CF.

4.6.1 Starting CF

If CF is stopped on the initial connection node, the Cluster Admin main window appears with the CF options of Load driver or Unconfigure (see below.) The CF state must be UNLOADED or LOADED to start CF on a node.

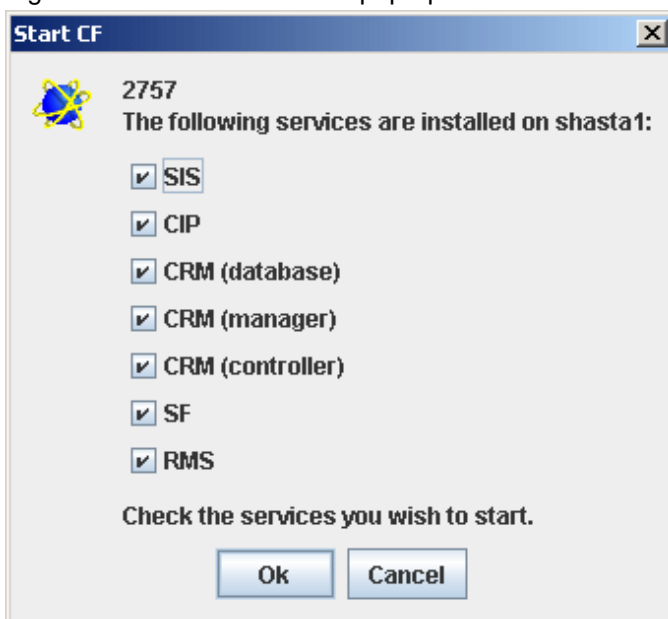
Figure 4.14 CF configured but not loaded



Click on the Load driver button to start the CF driver with the existing configuration.

The Start CF services popup appears (see below.) By default all CF services that have been installed on that node are selected to be started. The contents of this list may vary according to the installed products.

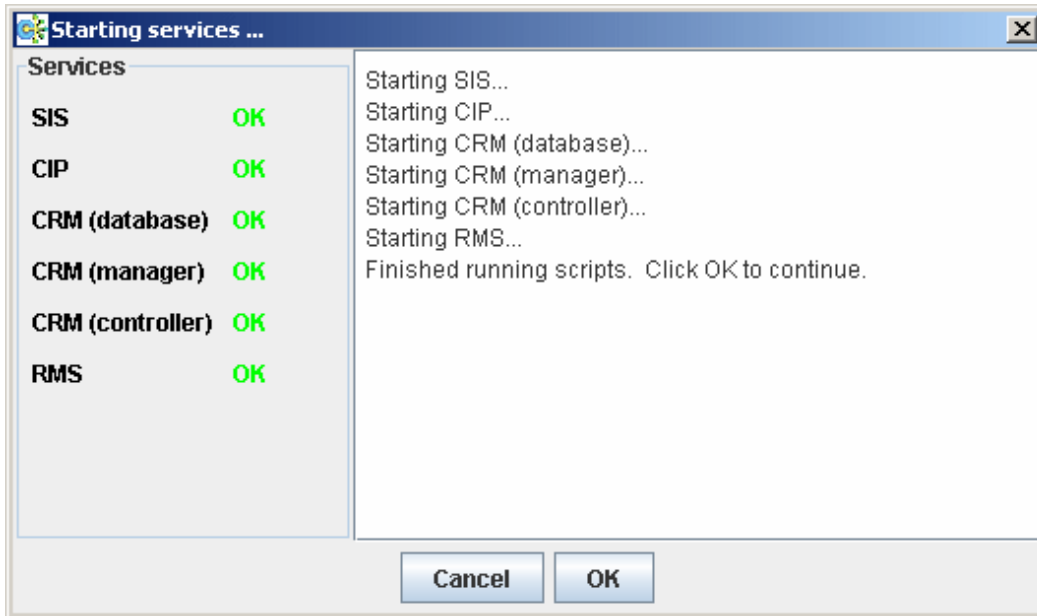
Figure 4.15 Start CF services pop-up



You may exclude CF services from startup by clicking on the selection check box for each service that you do not want to start. This should be done by experts only.

Click on the Ok button and a status popup appears with the results of each service start operation (see below.)

Figure 4.16 Start CF services status window



Click on [OK] to return to the Cluster Admin main window, after all the service startup has finished.

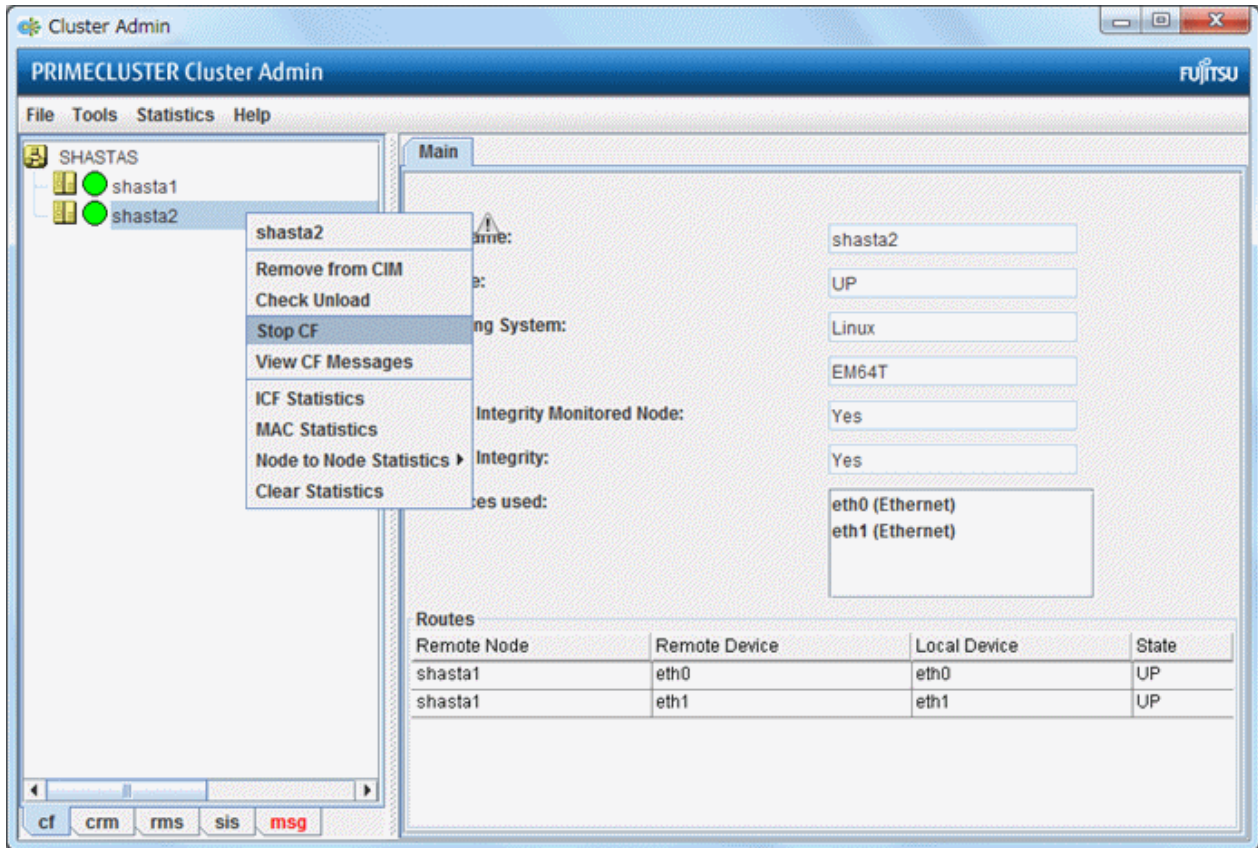
If [Cancel] is clicked, the startup process of the inactivated CF service is canceled.

The already activated CF service will remain in an activated state.

4.6.2 Stopping CF

If CF is already configured, right-click on a CF node name, and select [Stop CF] (see below).

Figure 4.17 Stop CF



A confirmation pop-up appears. Click <Yes> to continue.

Figure 4.18 Stopping CF



Before stopping CF, all services that use CF on that node should first be shut down. When you invoke [Stop CF] in the GUI, the GUI checks which services are still running. A list of these services is displayed in the pop-up to ask if you wish to continue the operation.

If you wish to continue the operation, shut down these services.

If the operation is canceled, the shutdown processing of a CF service that was not shut down will be canceled.

The already shut down CF service will remain shut down.

 **Note**

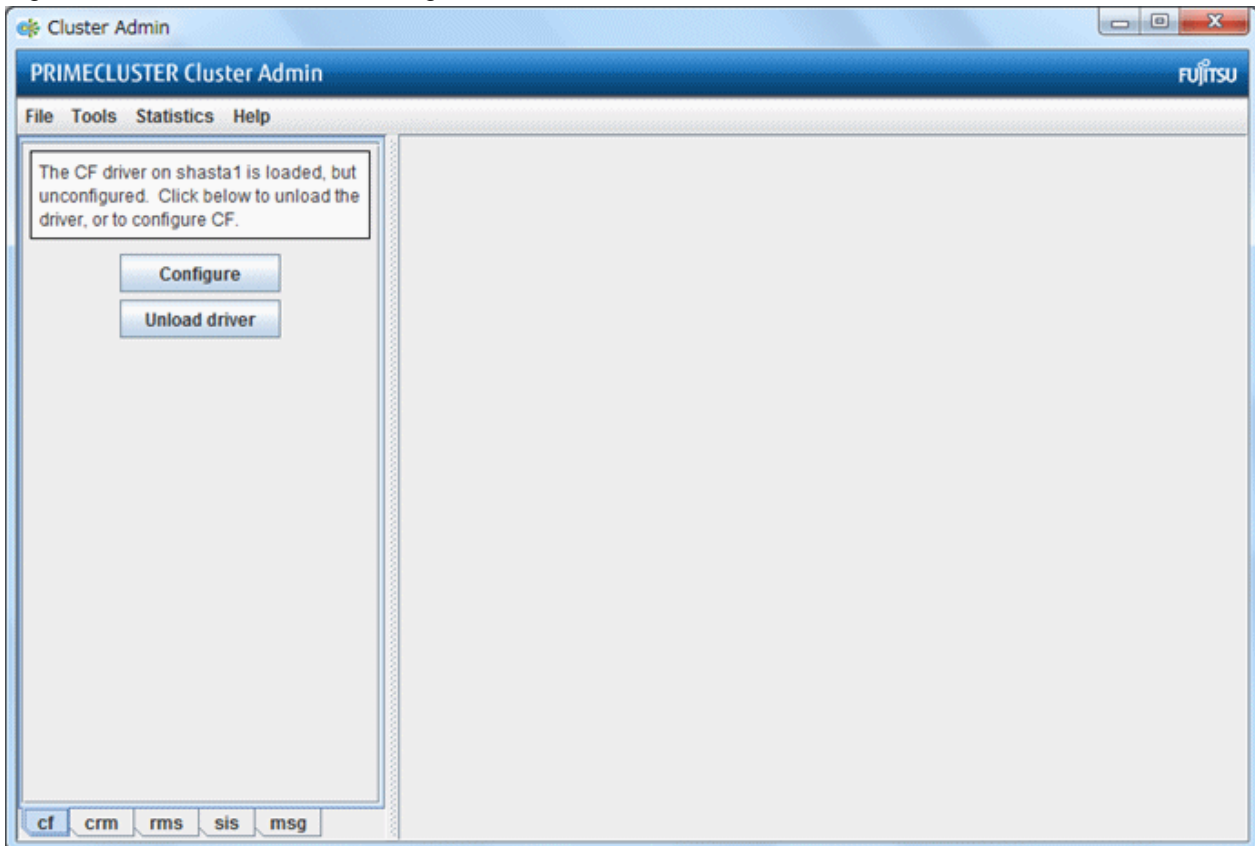
Only PRIMECLUSTER related services can be checked in the GUI as to which services are running. If a third-party product, for example Oracle RAC, is using a CF service, the product is not detected. In such cases, the third-party product should be shut down before you attempt to stop CF.

To stop CF on a node, the CF state must be UP, COMINGUP, or INVALID.

If CF is not configured, Cluster Admin displays the following screen.

Select [Unload driver].

Figure 4.19 CF loaded but not configured



4.7 Marking nodes DOWN

If a node is shut down normally, it is considered DOWN by the remaining nodes. If it leaves the cluster unexpectedly, it will be considered LEFTCLUSTER. It is important to mark a node DOWN as SOON as possible to allow normal cluster operation for the remaining nodes. The menu option Tools->Mark Node Down allows nodes to be marked as DOWN.

Note

Marking a node DOWN should be only done if the node is actually down (inoperable or inoperative); otherwise, this could cause data corruption.

To do this, select Tools->Mark Node Down. This displays a dialog of all of the nodes that consider another node to be LEFTCLUSTER. Clicking on one of them displays a list of all the nodes that node considered LEFTCLUSTER. Select one and then click OK. This clears the LEFTCLUSTER status on that node.

Refer to the Chapter "[Chapter 5 LEFTCLUSTER state](#)" for more information on the LEFTCLUSTER state.

4.8 Using PRIMECLUSTER log viewer

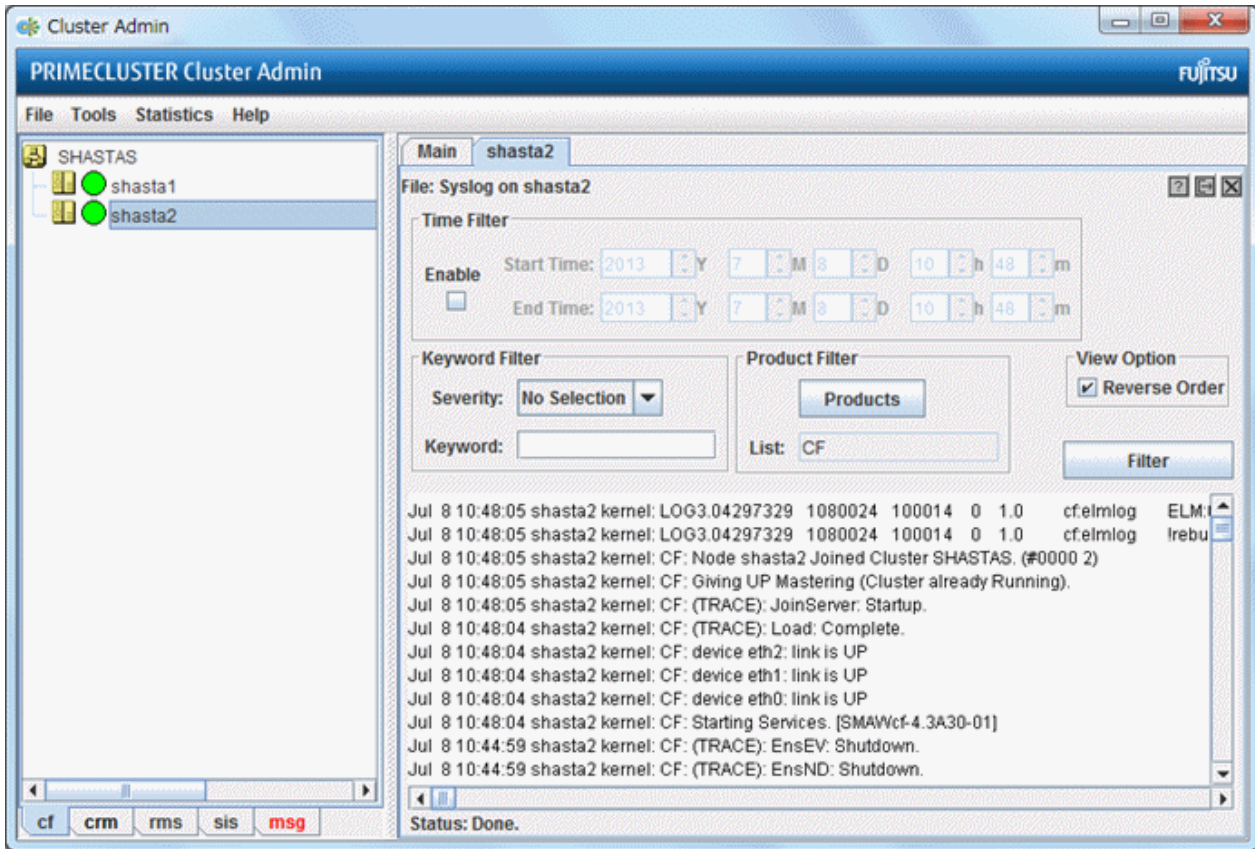
The CF log messages for a given node may be displayed by right-clicking on the node in the tree and selecting View CF Messages.

Alternately, you may go to the Tools menu and select View CF Messages. This brings up a pop-up where you can select the node whose syslog messages you would like to view.

When invoked from within CF, the PRIMECLUSTER log viewer only displays CF syslog messages. To view messages from other products, select the Products button on the Product Filter window pane on the PRIMECLUSTER log viewer screen.

Below is the example of the "Figure 4.20 PRIMECLUSTER log viewer."

Figure 4.20 PRIMECLUSTER log viewer



The messages appear in the right-hand panel. If you click on the Products button, then only the messages appear for the product that you select. To list the error messages for all of the installed PRIMECLUSTER products, choose All messages. Your choice is then listed in the List field.

The PRIMECLUSTER log viewer has search filters based on date/time/keyword and severity levels.

The Reverse Order checkbox is selected by default. This option reverses the order of the messages. To disable this feature, deselect the checkbox.

Note

If you have built a cluster system in the following environments, these logs cannot be displayed from Cluster Admin.

- Cloud environment
- Environment that uses Firewall
- Red Hat OpenStack Platform environment

Display CF messages (/var/log/messages) using a common UNIX text editor such as vi.

4.8.1 Search based on time filter

To perform a search based on a start and end time, click the check box for Enable, specify the start and end times for the search range, and click on the Filter button.

4.8.2 Search based on keyword

To perform a search based on a keyword, enter a keyword and click on the Filter button.

4.8.3 Search based on severity levels

To perform a search based severity levels, click on the Severity pull-down menu. You can choose from the severity levels shown in Table 3 and click on the Filter button.

Table 4.3 Table 3: PRIMECLUSTER log viewer severity levels

Severity level	Severity description
Emergency	Systems cannot be used.
Alert	Immediate action is necessary.
Critical	Error that makes it impossible for the associated PRIMECLUSTER product to continue running.
Error	Error condition that arises unexpectedly, causing the associated PRIMECLUSTER function to terminate abnormally.
Warning	Minor error that does not terminate the offending function.
Notice	Normal but important condition
Info	Provides information on the status of a PRIMECLUSTER operation.
Debug	Verbose message that provides more information on why an error condition occurred.

4.9 Displaying statistics

CF can display various statistics about its operation. There are three types of statistics available:

- ICF
- MAC
- Node to Node

To view the statistics for a particular node, right-click on that node in the tree and select the desired type of statistic.

Alternately, you can go to the Statistics menu and select the desired statistic. This will bring up a pop-up where you can select the node whose statistics you would like to view. The list of nodes presented in this pop-up will be all the nodes whose states are UP as viewed from the login node.

The figure below shows the display window for ICF Statistics.

Figure 4.21 ICF statistics

Counter	Value
ICF DATA packets xmit	367
ICF ENQ packets xmit	3
ICF ACK packets xmit	685
ICF NACK packets xmit	0
ICF HTBT_REQ packets xmit	9052
ICF HTBT_RPLY packets xmit	9052
ICF SYN packets xmit	1
ICF SYN_ACK packets xmit	1
ICF SQE packets xmit	0
ICF ECHO packets xmit	0
ICF NO_SVC packets xmit	0
ICF DATA packets rx	693
ICF ENQ packets rx	0
ICF ACK packets rx	359
ICF NACK packets rx	0
ICF HTBT_REQ packets rx	9052
ICF HTBT_RPLY packets rx	9052
ICF SYN packets rx	0

The figure below shows the display window for MAC Statistics.

Figure 4.22 MAC statistics

Counter	Value
Data packets sent	489
Control packets sent	21268
Packets received	21729
Packets dropped	0
Raw packets sent	12
Raw packets received	12
Raw packets dropped	0
Transmit errors	0
Receive errors	0

The figure below shows the display window for node to node statistics.

Figure 4.23 Node to Node statistics

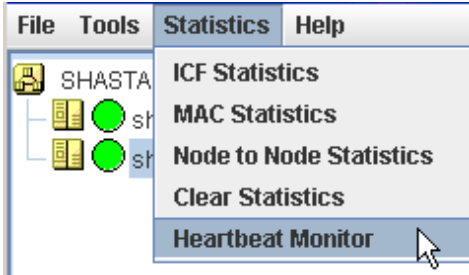
Counter	Value
ICF DATA packets xmit	414
ICF ENQ packets xmit	1
ICF ACK packets xmit	226
ICF NACK packets xmit	0
ICF HTBT_REQ packets xmit	7824
ICF HTBT_RPLY packets xmit	7821
ICF SYN packets xmit	1
ICF SYN_ACK packets xmit	1
ICF SQE packets xmit	0
ICF ECHO packets xmit	0
ICF NO_SVC packets xmit	0
ICF DATA packets rx	234
ICF ENQ packets rx	0
ICF ACK packets rx	405
ICF NACK packets rx	0
ICF HTBT_REQ packets rx	7821
ICF HTBT_RPLY packets rx	7824
ICF SYN packets rx	0

The statistics counters for a node can be cleared by right-clicking on a node and selecting Clear Statistics from the command pop-up. The Statistics menu also offers the same option.

4.10 Heartbeat monitor

To display the Heartbeat monitor, go to the Statistics menu and select Heartbeat Monitor (see below.)

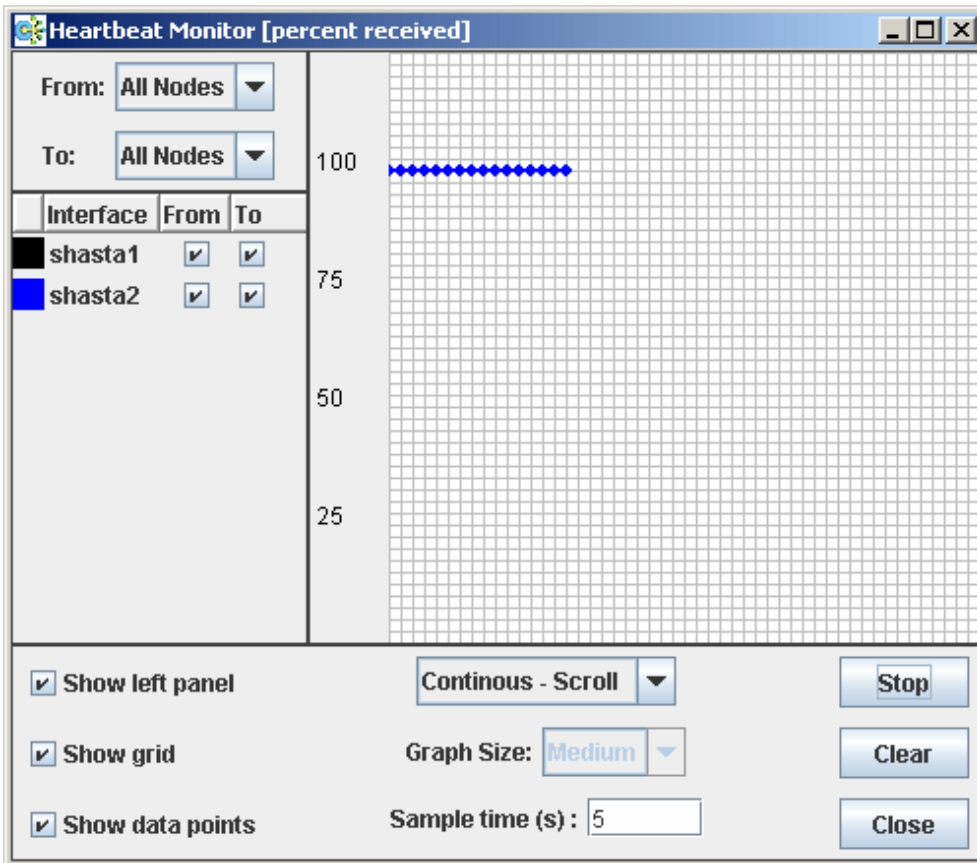
Figure 4.24 Selecting the Heartbeat monitor



The Heartbeat monitor allows you to monitor the percentage of heartbeats that are being received by CF over time. On a healthy cluster, this is normally close to 100 percent.

The Y axis is the percentage of heartbeats that have been successfully received and the X axis is a configurable time interval (see below.)

Figure 4.25 Heartbeat monitor



The controls on the left panel determine which data the graph shows as follows:

- The selection boxes at the top can be set to an individual node, or to All Nodes.
- The check boxes below the selection boxes allow the enabling and disabling of specific nodes.

The controls on the left of the bottom panel control how the graphing and information collection is done as follows:

- The Show left panel check box hides the left panel to provide more room for the graph.
- The Show grid check box turns the grid on and off.
- The Show data points check box can be turned off to display a simple line graph.

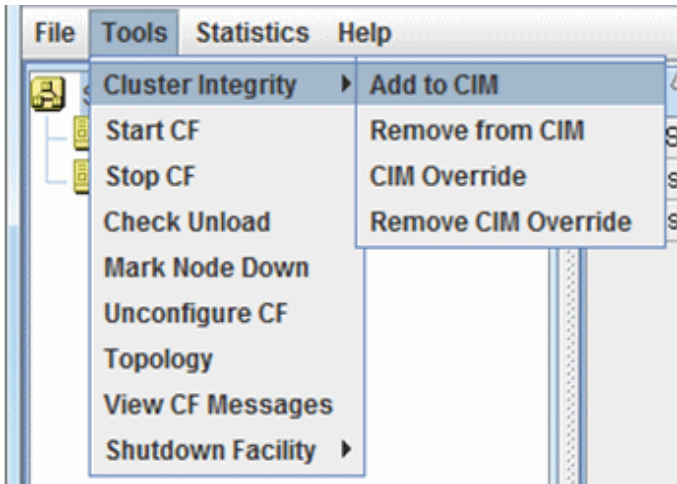
The controls in the bottom panel are as follows:

- The drop-down menu below the graph controls how the graph is drawn. The following options are available:
 - Continuous-Scroll - creates a continuous graph, so that when there are more data points than space, the graph scrolls.
 - Continuous-Clear - graphs continuously, but when the graph is full, clears it and starts a new graph.
 - Single Graph - creates a single graph only.
- Graph size - allows you to control how many data points are drawn.
- Sample time - controls how often data points are taken.
- The buttons on the lower right control starting and stopping of the graph, clearing it, and closing the graph window.

4.11 Adding and removing a node from CIM

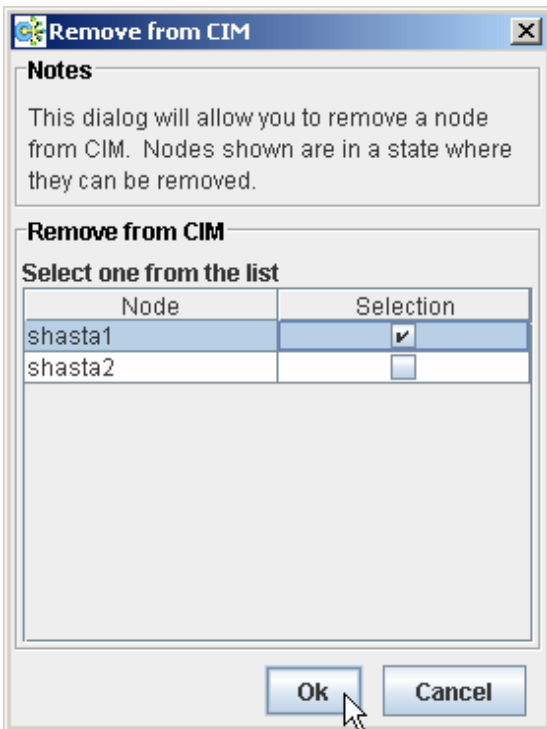
To add a node to CIM, click on the Tools pull-down menu. Select Cluster Integrity and Add to CIM from the expandable pull-down menu (see below.)

Figure 4.26 CIM options



The Add to CIM pop-up display appears. Choose the desired CF node and click on OK.

Figure 4.27 Add to CIM



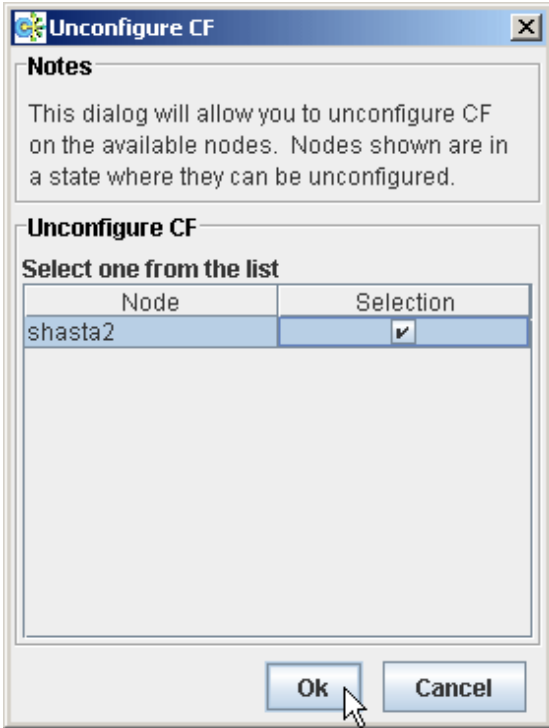
To remove a node from CIM by means of the Tools pull-down menu, select Cluster Integrity and Remove from CIM from the expandable pull-down menu. Choose the CF node to be removed from the pop-up and click on Ok. A node can be removed at any time.

Refer to the section "[2.2 Cluster Integrity Monitor\(CIM\)](#)" for more details on CIM.

4.12 Unconfigure CF

To unconfigure a CF node, first stop CF on that node. Then, from the Tools pull-down menu, click on Unconfigure CF. The Unconfigure CF pop-up display appears. Select the check box for the CF node to unconfigure, and click Ok (see below.)

Figure 4.28 Unconfigure CF

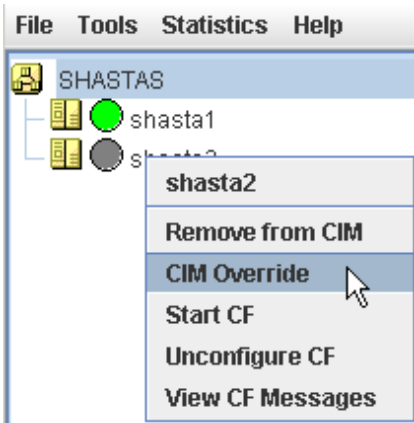


The unconfigured node will no longer be part of the cluster. However, other cluster nodes will still show that node as DOWN until they are rebooted.

4.13 CIM Override

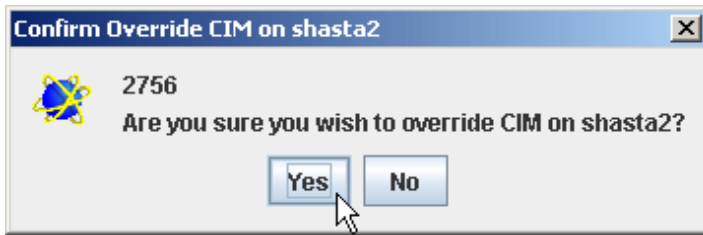
The CIM Override option causes a node to be ignored when determining a quorum. A node cannot be overridden if its CF state is UP. To select a node for CIM Override, right-click on a node and choose CIM Override (see below.)

Figure 4.29 CIM Override



A confirmation pop-up appears (see below.)

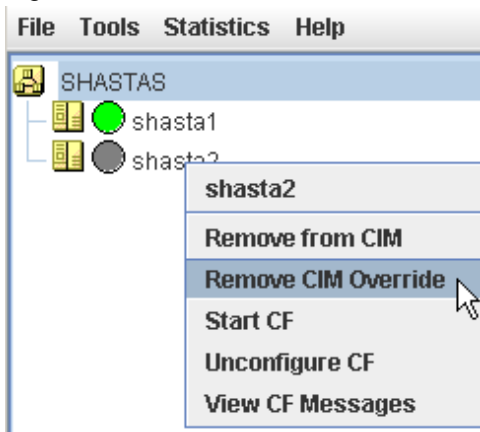
Figure 4.30 CIM Override confirmation



Click Yes to confirm.

Setting CIM override is a temporary action. It may be necessary to remove it manually again. This can be done by right-clicking on a node and selecting Remove CIM Override from the menu (see below.)

Figure 4.31 Remove CIM Override



CIM override is automatically removed when a node rejoins the cluster.

Chapter 5 LEFTCLUSTER state

This chapter describes the LEFTCLUSTER state and how to clear the state.

Occasionally, while CF is running, you may encounter the LEFTCLUSTER state, as shown by running the `cftool -n` command. A message will be printed to the console of the remaining nodes in the cluster. This can occur under the following circumstances:

- Broken interconnects - All cluster interconnects going to another node (or nodes) in the cluster are broken.
- Panicked nodes - A node panics.
- Reboot - Shutting down a node with the `reboot` command.

Note

Nodes running CF should normally be shut down with the `shutdown` command or with the `init` command. If the `reboot -f` command is executed, the node will be in the LEFTCLUSTER state.

The `shutdown` command or the `init` command will run the CF's Stop script that will allow CF to be cleanly shut down on that node. However, if you run the `reboot -f`, `halt -f`, or `poweroff -f` command, the CF's Stop script is not run, and the node will go down while CF is running. This will cause the node to be declared to be in the LEFTCLUSTER state by the other nodes.

If the Shutdown Facility (SF) is fully configured and running on all cluster nodes, it will try to resolve the LEFTCLUSTER state automatically. If SF is not configured and running, or the SF fails to clear the state, the state has to be cleared manually. This section explains the LEFTCLUSTER state and how to clear this state manually.

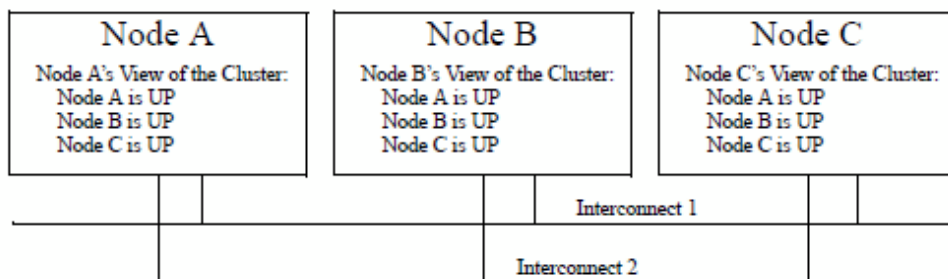
5.1 Description of the LEFTCLUSTER state

Each node in a CF cluster keeps track of the state of the other nodes in the cluster. For example, the other node's state may be UP, DOWN, or LEFTCLUSTER.

LEFTCLUSTER is an intermediate state between UP and DOWN, which means that the node cannot determine the state of another node in the cluster because of a break in communication.

For example, consider the three-node cluster shown in

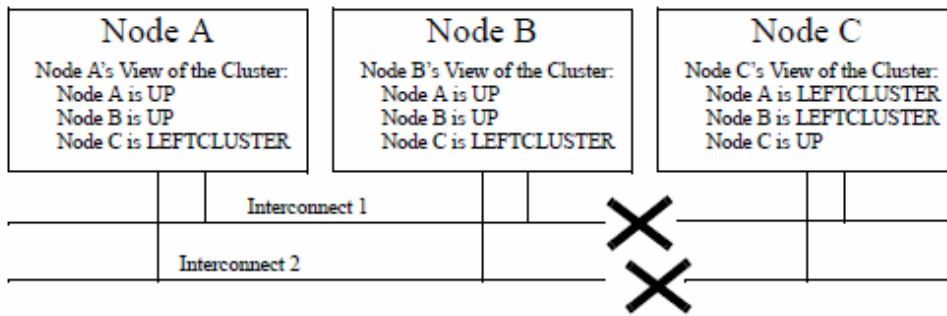
Figure 5.1 Three-node cluster with working connections



Each node maintains a table of what states it believes all the nodes in the cluster are in.

Now suppose that there is a cluster partition in which the connections to Node C are lost. The result is shown in the following figure.

Figure 5.2 Three-node cluster where connection is lost



Because of the break in network communications, Nodes A and B cannot be sure of Node C's true state. They therefore update their state tables to declare that Node C is in the LEFTCLUSTER state. Likewise, Node C cannot be sure of the true states of Nodes A and B, so it marks those nodes as being in the LEFTCLUSTER in its state table.

Note

LEFTCLUSTER is a state that a particular node believes other nodes are in. It is never a state that a node believes that it is in. For example, in, "Figure 5.2 Three-node cluster where connection is lost" each node believes that it is UP.

The purpose of the LEFTCLUSTER state is to warn applications which use CF that contact with another node has been lost and that the state of such a node is uncertain. This is very important for RMS.

For example, suppose that an application on Node C was configured under RMS to fail over to Node B if Node C failed. Suppose further that Nodes C and B had a shared disk to which this application wrote.

RMS needs to make sure that the application is, at any given time, running on either Node C or B but not both, since running it on both would corrupt the data on the shared disk.

Now suppose for the sake of argument that there was no LEFTCLUSTER state, but as soon as network communication was lost, each node marked the node it could not communicate with as DOWN. RMS on Node B would notice that Node C was DOWN. It would then start an instance of the application on Node C as part of its cluster partition processing. Unfortunately, Node C isn't really DOWN. Only communication with it has been lost. The application is still running on Node C. The applications, which assume that they have exclusive access to the shared disk, would then corrupt data as their updates interfered with each other.

The LEFTCLUSTER state avoids the above scenario. It allows RMS and other application using CF to distinguish between lost communications (implying an unknown state of nodes beyond the communications break) and a node that is genuinely down.

When SF notices that a node is in the LEFTCLUSTER state, it contacts the previously configured Shutdown Agent and requests that the node which is in the LEFTCLUSTER state be shut down. With PRIMECLUSTER, a weight calculation determines which node or nodes should survive and which ones should be shut down. SF has the capability to arbitrate among the shutdown requests and shut down a selected set of nodes in the cluster, such that the subcluster with the largest weight is left running and the remaining subclusters are shutdown.

In the example given, Node C would be shut down, leaving Nodes A and B running. After the SF software shuts down Node C, SF on Nodes A and B clear the LEFTCLUSTER state such that Nodes A and B see Node C as DOWN. Refer to the Chapter "Chapter 7 Shutdown Facility" for details on configuring SF and shutdown agents.

Note

Note that a node cannot join an existing cluster when the nodes in that cluster believe that the node is in the LEFTCLUSTER state. Therefore, any nodes in LEFTCLUSTER state have to be recovered before they can join an existing cluster.

5.2 Recovering from LEFTCLUSTER

When a node comes back up after being rebooted and attempts to rejoin the cluster, the join process automatically changes the node's state from LEFTCLUSTER to DOWN so that it can rejoin the cluster. When this occurs, the join server initially sees the node that is attempting to join the cluster as being in the LEFTCLUSTER state. The join server signals the joining node that it is busy because the joining node is

not in the DOWN state. It then notifies all of the remaining nodes in the cluster that the joining node is DOWN and to start the node-down processing, which must be completed before the node is allowed to rejoin the cluster. The joining node continues retrying to join the cluster until the node-down processing is completed on all of the cluster nodes at which time the joining node is allowed to rejoin the cluster.

If SF is not running on all the nodes, or if SF is unable to shut down the node which left the cluster, and the LEFTCLUSTER condition occurs, then the system administrator must manually clear the LEFTCLUSTER state. The procedure for doing this depends on how the LEFTCLUSTER condition occurred.

Restarting a node in the LEFTCLUSTER state can cause resource failures or a double fault. Therefore, do not restart the node and recover from the LEFTCLUSTER state by following the procedures in the following sections.

5.2.1 Caused by a panic/hung node

The LEFTCLUSTER state may occur because a particular node panicked or hung. In this case, the procedure to clear LEFTCLUSTER is as follows:

1. Make sure the node is really down. If the node panicked and came back up, proceed to Step 2. If the node is in the debugger, exit the debugger. The node will reboot if it panicked, otherwise shut down the node, called the offending node in the following discussion.
2. Use the Cluster Admin GUI to log into one of the running nodes in the cluster while the offending node is suspended. Go to the CF main window and select Mark Node Down from the Tools pull-down menu, then mark the offending node as DOWN. This may also be done from the command line by using the following command:

```
# cftool -k
```

3. Reboot the offending node, it should automatically join the cluster.

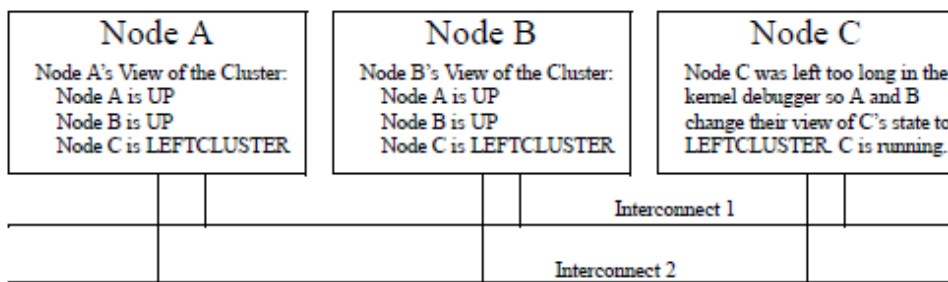


The state of the node is automatically changed from the state of LEFTCLUSTER in the state of DOWN so that the node may enter again when the offending node is reactivated.

5.2.2 Caused by staying in the kernel debugger too long

If the break point is set by the kernel debugger to suspend the operation of OS kernel, the heartbeat of CF will stop. In the following figure, Node C was placed in the kernel debugger too long so it appears as a hung node. Nodes A and B decided that Node C's state was LEFTCLUSTER.

Figure 5.3 Node C placed in the kernel debugger too long



To recover from this situation, you would need to do the following:

1. Shut down Node C, and bring it back up.
2. If Node C fails to join the cluster and remains in the LEFTCLUSTER state after being shut down and coming back up, start up the Cluster Admin on Node A or B. Use Mark Node Down from the Tools pull-down menu in the CF portion of the GUI to mark Node C DOWN.
3. The node should successfully join the cluster.

5.2.3 Caused by a cluster partition

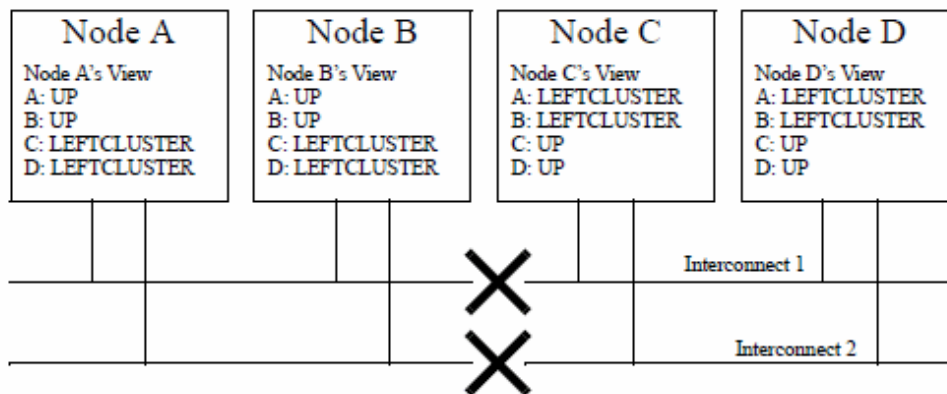
A cluster partition is a communications failure in which all CF communications between sets of nodes in the cluster are lost. In this case, the cluster itself is effectively partitioned into sub-clusters.

To manually recover from a cluster partition, you must do the following:

1. Decide which of the sub-clusters you want to survive. Typically, you will choose the sub-cluster that has the largest number of nodes in it or the one where the most important hardware is connected or the most important application is running.
2. Shut down all of the nodes in the sub-cluster which you don't want to survive.
3. Fix the network break so that connectivity is restored between all the nodes in the cluster.
4. Bring the nodes back up.
5. If the nodes fail to join the cluster and remain in the LEFTCLUSTER state after being shut down and coming back up, use the Cluster Admin GUI to log on to one of the surviving nodes and run the CF portion of the GUI. Select Mark Node Down from the Tools menu to mark all of the shutdown nodes as DOWN.
6. The nodes should successfully join the cluster.

For example, consider the following figure

Figure 5.4 Four-node cluster with cluster partition



In this figure, a four-node cluster has suffered a cluster partition. Both of its CF interconnects (Interconnect 1 and Interconnect 2) have been severed. The cluster is now split into two sub-clusters. Nodes A and B are in one sub-cluster while Nodes C and D are in the other.

To recover from this situation, in instances where SF fails to resolve the problem, you would need to do the following:

1. Decide which sub-cluster you want to survive. In this example, let us arbitrarily decide that Nodes A and B will survive.
2. Shut down all of the nodes in the other sub-cluster, here Nodes C and D.
3. Fix the interconnect break on Interconnect 1 and Interconnect 2 so that both sub-clusters will be able to communicate with each other again.
4. Bring Nodes C and D back up.
5. If the LEFTCLUSTER state persists on Nodes C or D, run the Cluster Admin GUI on either Node A or Node B. Start the CF portion of the GUI and go to Mark Node Down from the Tools pull-down menu. Mark any nodes still in the LEFTCLUSTER state as DOWN.

Chapter 6 CF topology table

This chapter discusses the CF topology table as it relates to the CF portion of the Cluster Admin GUI.

The CF topology table is part of the CF portion of the Cluster Admin GUI. The topology table may be invoked from the *Tools->Topology* menu item in the GUI (refer to the section "[4.5 Displaying the topology table](#)" in the Chapter "GUI administration"). It is also available during CF configuration in the CF Wizard in the GUI.

The topology table is designed to show the network configuration from perspective of CF. It shows what devices are on the same interconnects and can communicate with each other.

The topology table only considers Ethernet devices. It does not include any IP interconnects that might be used for CF, even if CF over IP is configured.

Displayed devices

The topology table is generated by doing CF pings on all the nodes in the cluster and then analyzing the results.

The rest of this chapter discusses the format of the topology table. The examples implicitly assume that all devices can be seen on each node. Again, this would be the case when first configuring a CF cluster.

6.1 Basic layout

The basic layout of the topology table is shown in the table below.

Table 6.1 Basic layout of the topology table

SHASTAS	Full interconnects		Partial interconnects		Unconnected devices
	Int 1	Int 2	Int 3	Int 4	
shasta1	eth0 eth2	eth1	eth3	eth5	eth4 eth6
shasta2	eth0	eth2	missing	eth1	
shasta3	eth1	eth2	eth3	missing	eth4

The upper-left-hand corner of the topology table gives the CF cluster name. Below it, the names of all of the nodes in the cluster are listed.

The CF devices are organized into three major categories:

- Full interconnects - Have working CF communications to each of the nodes in the cluster.
- Partial interconnects - Have working CF communications to at least two nodes in the cluster, but not to all of the nodes.
- Unconnected devices - Have no working CF communications to any node in the cluster.

If a particular category is not present, it will be omitted from the topology table. For example, if the cluster in "[Table 6.1 Basic layout of the topology table](#)" had no partial interconnects, then the table headings would list only full interconnects and unconnected devices (as well as the left-most column giving the clustername and node names).

Within the full interconnects and partial interconnects category, the devices are further sorted into separate interconnects. Each column under an Int number heading represents all the devices on an interconnect. (The column header Int is an abbreviation for Interconnect.) For example, in "[Table 6.1 Basic layout of the topology table](#)", there are two full interconnects listed under the column headings of Int 1 and Int 2.

Each row for a node represents possible CF devices for that node.

Thus, in "[Table 6.1 Basic layout of the topology table](#)", Interconnect 1 is a full interconnect. It is attached to eth0 and eth2 on shasta1. On shasta2, it is attached to eth0, and on shasta3, it is attached to eth1.

Since CF runs over Ethernet devices, the ethn devices in "[Table 6.1 Basic layout of the topology table](#)" represent the Ethernet devices found on the various systems. The actual names of these devices will vary depending on the type of Ethernet controllers on the system. For nodes whose CF driver was loaded with -L, only configured devices will be shown.

It should be noted that the numbering used for the interconnects is purely a convention used only in the topology table to make the display easier to read. The underlying CF product does not number its interconnects. CF itself only knows about CF devices and point-to-point routes.

If a node does not have a device on a particular partial interconnect, then the word missing will be printed in that node's cell in the partial interconnects column. For example, in "Table 6.1 Basic layout of the topology table", shasta2 does not have a device for the partial interconnect labeled Int 3.

6.2 Selecting devices

The basic layout of the topology table is shown in the following table. However, when the GUI actually draws the topology table, it puts check boxes next to all of the interconnects and CF devices as shown in the following table.

Table 6.2 Topology table with check boxes

SHASTAS	Full interconnects		Partial interconnects		Unconnected devices
	<input checked="" type="checkbox"/> Int1	<input checked="" type="checkbox"/> Int2	<input type="checkbox"/> Int3	<input type="checkbox"/> Int 4	
shasta1	<input checked="" type="checkbox"/> eth0 <input type="checkbox"/> eth2	<input checked="" type="checkbox"/> eth1	<input type="checkbox"/> eth3	<input type="checkbox"/> eth5	<input type="checkbox"/> eth4 <input type="checkbox"/> eth6
shasta2	<input checked="" type="checkbox"/> eth0	<input checked="" type="checkbox"/> eth2	missing	<input type="checkbox"/> eth1	
shasta3	<input checked="" type="checkbox"/> eth1	<input checked="" type="checkbox"/> eth2	<input type="checkbox"/> eth3	missing	<input type="checkbox"/> eth4

The check boxes show which of the devices were selected for use in the CF configuration. (In the actual topology table, check marks appear instead of x's.)

When the topology table is used outside of the CF Wizard, these check boxes are read-only. They show what devices were previously selected for the configuration. In addition, the unchecked boxes (representing devices which were not configured for CF) will not be seen for nodes where -L was used to load CF.

When the topology table is used within the CF Wizard, then the check boxes may be used to select which devices will be included in the CF configuration. Clicking on the check box in an Int number heading will automatically select all devices attached to that interconnect. However, if a node has multiple devices connected to a single interconnect, then only one of the devices will be selected.

For example, in Table 5, shasta1 has both eth0 and eth2 attached to Interconnect 1. A valid CF configuration allows a given node to have only one CF device configured per interconnect. Thus, in the CF Wizard, the topology table will only allow eth0 or eth2 to be selected for shasta1. In the above example, if eth2 were selected for shasta1, then eth0 would automatically be unchecked.

If the CF Wizard is used to add a new node to an existing cluster, then the devices already configured in the running cluster will be displayed as read-only in the topology table. These existing devices may not be changed without unconfiguring CF on their respective nodes.

For how to delete the CF configuration, see "4.12 Unconfigure CF."

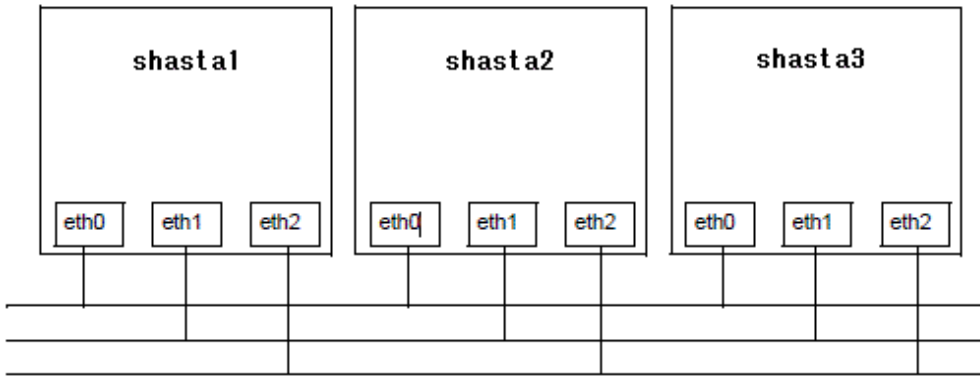
6.3 Examples of topology table

The following examples show various network configurations and what their topology tables would look like when the topology table is displayed in the CF Wizard on a totally unconfigured cluster. For simplicity, the check boxes are omitted.

Example 1

In this example, there is a three-node cluster with three full interconnects.

Figure 6.1 A three-node cluster with three full interconnects



The resulting topology table for the above figure is shown in the following table.

Table 6.3 Topology table for 3 full interconnects

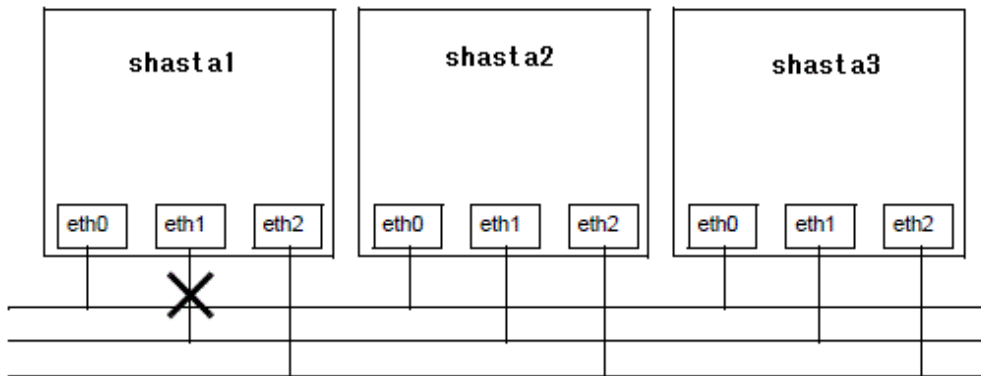
SHASTAS	Full interconnects		
	Int 1	Int 2	Int 3
shasta1	eth0	eth1	eth2
shasta2	eth0	eth1	eth2
shasta3	eth0	eth1	eth2

Since there are no partial interconnects or unconnected devices, those columns are omitted from the topology table.

Example 2

In this example, shasta1's Ethernet connection for eth1 has been broken.

Figure 6.2 Broken ethernet connection for eth1 on shasta1



The resulting topology table for Figure 6.3 is shown in Table 7.

Table 6.4 Topology table with broken Ethernet connection

SHASTAS	Full interconnects		Partial interconnects	Unconnected devices
	Int 1	Int 2	Int 3	
shasta1	eth0	eth2	missing	eth1
shasta2	eth0	eth2	eth1	
shasta3	eth0	eth2	eth1	

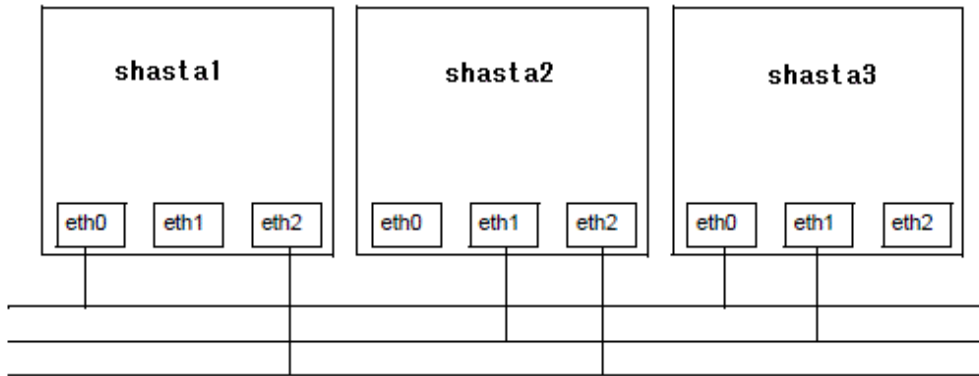
In the "Table 6.4 Topology table with broken Ethernet connection," eth1 for shasta1 now shows up as an unconnected device. Since one of the interconnects is missing a device for shasta1, the Partial Interconnect column now shows up. Note that the relationship between

interconnect numbering and the devices has changed between the "Table 6.3 Topology table for 3 full interconnects" and the "Table 6.4 Topology table with broken Ethernet connection". In "Table 6.3 Topology table for 3 full interconnects," for example, all eth1 devices were on Int 2. In the "Table 6.4 Topology table with broken Ethernet connection," the eth1 devices for Nodes B and C are now on the partial interconnect Int 3. This change in numbering illustrates the fact that the numbers have no real significance beyond the topology.

Example 3

This example shows a cluster with severe networking or cabling problems in which no full interconnects are found.

Figure 6.3 Cluster with no full interconnects



The resulting topology table for the above figure is shown in the blow table.

Table 6.5 Topology table with no full interconnects

SHASTAS	Partial interconnects			Unconnected devices
	Int 1	Int 2	Int 3	
shasta1	eth0	missing	eth2	eth1
shasta2	missing	eth1	eth2	eth0
shasta3	eth0	eth1	missing	eth2

In this table, the full interconnects column is omitted since there are none. Note that if this configuration were present in the CF Wizard, the wizard would not allow you to do configuration. The wizard requires that at least one full interconnect must be present.

Chapter 7 Shutdown Facility

This chapter describes the components and advantages of PRIMECLUSTER Shutdown Facility (SF) and provides administration information.



Note

Certain product options are region-specific. For information on the availability a specific Shutdown Agent (SA), contact field engineers.

7.1 Overview

The SF provides the interface for managing the shutdown of cluster nodes when error conditions occur. The SF also advises other PRIMECLUSTER products of the successful completion of node shutdown so that recovery operations can begin.

The SF is made up of the following major components:

- The Shutdown Daemon (SD)
- One or more Shutdown Agents (SA)
- *sdtool(1M)* command

Shutdown Daemon (SD)

The SD is started at system boot time and is responsible for the following:

- Monitoring the state of all cluster nodes
- Monitoring the state of all registered SAs
- Reacting to indications of cluster node failure and eliminating the nodes forcibly
- Resolving split-brain conditions
- Notifying other PRIMECLUSTER products that nodes were forcibly eliminated
- Checking the connection to the option hardware or the virtual machine function that forcibly eliminates cluster nodes periodically (in 10-minute intervals)

The SD uses SAs to perform most of its work with regard to cluster node monitoring and forced node elimination. In addition to SA's, the SD interfaces with the Cluster Foundation layer's ENS system to receive node failure indications and to advertise node elimination completion.

The SD starts the SA periodically (in 10-minute intervals) to check the route that forcibly eliminates cluster nodes. The SD reflects the checked route status to the test status of each SA (Test State) displayed by the *sdtool(1M)* command.

Shutdown Agents (SA)

The SA's role is to attempt to shut down a remote cluster node in a manner in which the shutdown can be guaranteed. Some of the SAs are shipped with the SF product, but may differ based on the architecture of the cluster node on which SF is installed. SF allows any PRIMECLUSTER service layer product to shut down a node whether RMS is running or not.

An SA is responsible for shutting down, and verifying the shutdown of a cluster node. Each SA uses a specific method for performing the node shutdown such as:

- SA_ipmi offers the shutdown agent for IPMI-based systems.
- SA_lkcd offers the shutdown agent that uses the kernel panic status of other nodes.
- SA_irmc offers the shutdown agent that uses iRMC/MMB on the node PRIMEQUEST 3000 series.
- SA_icmp offers the shutdown agent that checks whether a node to be stopped is in the active or inactive state by using a network route.
- SA_vmchghost offers the shutdown agent of the system which uses the KVM virtual machine function.
- SA_libvirtgp and SA_libvirtgr offer the shutdown agent of the system which uses the KVM virtual machine function.

- SA_vwvnr offers the shutdown agent that is cooperated with VMware vCenter Server.
- SA_vmk5r offers the shutdown agent using the FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API.
- SA_vmosp offers the shutdown agent using the OpenStack API.
- SA_vmawsAsyncReset offers the shutdown agent using the AWS CLI.
- SA_vmazureReset offers the shutdown agent using the Azure CLI.
- SA_vmnifclAsyncReset offers the shutdown agent using the NIFCLOUD API.

See "7.2 Available SAs" for more information on SA.

If more than one SA is used, the first SA in the configuration file is used as the primary SA. SD always uses the primary SA. The other secondary SAs are used as fall back SAs only if the primary SA fails for some reason.

sdtool command

The *sdtool(1M)* command is the command line interface for interacting with the SD. With it the administrator can:

- Start and stop the SD (although this is typically done with an RC script run at boot time)
- View the current state of the SA's
- Force the SD to reconfigure itself based on new contents of its configuration file
- Dump the contents of the current SF configuration
- Enable/disable SD debugging output
- Eliminate a cluster node



Note

Although the *sdtool(1M)* command provides a cluster node elimination capability, the preferred method for controlled shutdown of a cluster node is the `/sbin/shutdown` command.

7.2 Available SAs

This section describes the following set of supported SAs:

- IPMI (Intelligent Platform Management Interface)
- kdump
- iRMC (integrated Remote Management Controller)
- ICMP
- VMCHKHOST
- libvirt
- VMware vCenter Server Functional Cooperation
- FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API
- OpenStack API
- AWS CLI
- Azure CLI
- NIFCLOUD API

7.2.1 IPMI



See

For complete configuration details of iRMC or BMC, refer to the appropriate hardware manual that came with your system.

The IPMI shutdown agent SA_ipmi provides the node shutdown function that uses iRMC (Remote Management Controller) of PRIMERGY servers.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_ipmi.log
```

7.2.2 kdump

The kdump shutdown agent SA_lkcd is the SA that can be used when kdump is used. After another node panics, this SA executes high-speed switchover while a crash dump is being collected by kdump.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_lkcd.log
```

7.2.3 iRMC

The iRMC shutdown agent uses the iRMC/MMB of PRIMEQUEST 3000 series to provide shutdown mechanisms for nodes.

The iRMC shutdown agent provides the following 3 shutdown mechanisms:

- SA_irmcp Shutdown mechanism that triggers panic in nodes through the iRMC.
- SA_irmcr Shutdown mechanism that resets the nodes through the iRMC.
- SA_irmcf Shutdown mechanism that turns off the power of the node through the MMB.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_irmcp.log  
/var/opt/SMAWsf/log/SA_irmcr.log  
/var/opt/SMAWsf/log/SA_irmcf.log
```

7.2.4 ICMP

The ICMP shutdown agent, SA_icmp, checks whether a node to be stopped is in the active or inactive state by using a network route.

If there is no response from the node to be stopped in all specified network routes, it determines that the node to be stopped is in the inactive state and terminates normally. If any responses are returned from the node in one or more network routes, it determines that the node to be stopped is in the active state and terminates abnormally.

If all of the network interfaces used to check the state of the node to be stopped are stopped, SA_icmp cannot check the state of the node to be stopped and terminates abnormally.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_icmp.log
```

7.2.5 VMCHKHOST

The VMCHKHOST shutdown agent, SA_vmchghost, is the SA that can be used when the function, which switches the Host OS in the event of an error, is used on the KVM virtual machine function.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmchghost.log
```

7.2.6 libvirt

The libvirt shutdown agent provides the shutdown function for nodes (guest OS) in systems that are using the KVM virtual machine function.

There are two types of the libvirt shutdown agent as follows:

- SA_libvirtgp SF that panics nodes (guest OS)
- SA_libvirtgr SF that resets nodes (guest OS)

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_libvirtgp.log  
/var/opt/SMAWsf/log/SA_libvirtgr.log
```

7.2.7 VMware vCenter Server Functional Cooperation

In a VMware environment, the VMware vCenter Server functional cooperation shutdown agent SA_vwvmr provides the function to shut down a node (guest OS) that is cooperated with VMware vCenter Server.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vwvmr.log
```

7.2.8 FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API

In an FJcloud-O/FJcloud-Baremetal environment, the FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API shutdown agent SA_vmk5r provides the function to shut down a node (virtual server/Bare Metal server) using the FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmk5r.log
```

7.2.9 OpenStack API

In an RHOSP environment, the OpenStack API shutdown agent SA_vmosr provides the function to shut down a node (instance) using the OpenStack API.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmosr.log
```

7.2.10 AWS CLI

In an AWS environment, the AWS CLI shutdown agent SA_vmawsAsyncReset provides the function to shut down a node (instance) using the AWS CLI.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmawsAsyncReset.log
```

7.2.11 Azure CLI

In an Azure environment, the Azure CLI shutdown agent SA_vmazureReset provides the function to shut down a node (virtual machine) using the Azure CLI.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmazureReset.log
```


7.2.12 NIFCLOUD API

In a NIFCLOUD environment, the NIFCLOUD API shutdown agent SA_vmnifclAsyncReset provides the function to shut down a node (server) using the NIFCLOUD API.

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmnifclAsyncReset.log
```

7.3 SF split-brain handling

The PRIMECLUSTER product provides the ability to gracefully resolve split-brain state as described in this section.

7.3.1 Administrative LAN

The SF handles a split-brain by using the administrative LAN.

7.3.2 SF split-brain handling

A split-brain condition is one in which one or more cluster nodes have stopped receiving heartbeats from one or more other cluster nodes, yet those nodes have been determined to still be running. Each of these distinct sets of cluster nodes is called a sub-cluster, and when a split-brain condition occurs the Shutdown Facility has a choice to make as to which sub-cluster should remain running.

Only one of the sub-clusters in a split-brain condition can survive. The SF determines which sub-cluster is most important and allows only that sub-cluster to remain. SF determines the importance of each sub-cluster by calculating the total node weight and application weight of each sub-cluster. The sub-cluster with the greatest total weight survives.

Node weights are defined in the SF configuration file rcsd.cfg.

Application weights are defined in RMS. Each RMS userApplication object can have a ShutdownPriority defined for it. The value of the ShutdownPriority is that application's weight. RMS calculates the total application weight for a particular node by adding up the weights of all applications that are Online on that node. If an application is switched from one node to another, its weight will be transferred to the new node.

SF combines the values for the RMS ShutdownPriority attributes and the SF weight assignments to determine how to handle a split-brain condition.

7.3.2.1 RMS ShutdownPriority attribute

RMS supports the ability to set application importance in the form of a ShutdownPriority value for each userApplication object defined within the RMS configuration. These values are combined for all userApplication objects that are Online on a given cluster node to represent the total application weight of that node. When a userApplication object is switched from one node to another, the value of that userApplication objects ShutdownPriority is transferred to the new node.

The higher the value of the ShutdownPriority attribute, the more important the application.

7.3.2.2 Shutdown Facility weight assignment

The Shutdown Facility supports the ability to define node importance in the form of a weight setting in the configuration file. This value represents a node weight for the cluster node.

The higher the node weight value, the more important the node.



Although SF takes into consideration both SF node weights and RMS application weights while performing split-brain handling, it is recommended to use only one of the weights for simplicity and ease of use. When both weights are used, split-brain handling results are much more complex.

It is recommended that you follow the guidelines in "[7.3.4 Configuration notes](#)" for help you with the configuration.

7.3.3 Runtime processing

Split-brain handling may be performed by the following element of the Shutdown Facility:

- The Shutdown Facility internal algorithm

This method uses the node weight calculation to determine which sub-cluster is of greater importance. The total node weight is equal to the value of the defined Shutdown Facility node weight added to the total application weight of the Online applications for this node as calculated within RMS.

SF internal algorithm

When the SF is selected as the split-brain resolution manager, the SF uses the node weight internally.

The SF on each cluster node identifies which cluster nodes are outside its sub-cluster and adds each one of them to an internal shutdown list. This shutdown list, along with the local nodes node weight, is advertised to the SF instances running on all other cluster nodes (both in the local sub-cluster and outside the local sub-cluster) via the admIP network defined in the SF configuration file. After the SFs on each cluster node receive the advertisements, they each calculate the heaviest sub-cluster. The heaviest sub-cluster shuts down all lower weight sub-clusters.

In addition to handling well-coordinated shutdown activities defined by the contents of the advertisements, the SF internal algorithm will also resolve split-brain if the advertisements fail to be received. If the advertisements are not received then the split-brain will still be resolved, but it may take a bit more time as some amount of delay will have to be incurred.

The split-brain resolution done by the SF in situations where advertisements have failed depends on a variable delay based on the inverse of the percentage of the available cluster weight the local sub-cluster contains. The more weight it contains the less it delays. After the delay expires (assuming the sub-cluster has not been shut down by a higher-weight sub-cluster) the SF in the sub-cluster begins shutting down all other nodes in all other sub-clusters.

If a sub-cluster contains greater than 50 percent of the available cluster weight, then the SF in that sub-cluster will immediately start shutting down all other nodes in all other sub-clusters.

If a cluster node to be shut down is restarted and the CF is started before the shutdown processing starts, this node is excluded from the shutdown target nodes.

7.3.4 Configuration notes

When configuring the Shutdown Facility, RMS, and defining the various weights, the administrator should consider what the eventual goal of a split-brain situation should be.

Typical scenarios that are implemented are as follows:

- Largest Sub-cluster Survival
- Specific Hardware Survival
- Specific Application Survival

The weights applied to both cluster nodes and to defined applications allow considerable flexibility in defining what parts of a cluster configuration should survive a split-brain condition. Using the settings outlined below, administrators can advise the Shutdown Facility about what should be preserved during split-brain resolution.

Largest Sub-cluster Survival

In this scenario, the administrator does not care which physical nodes survive the split, just that the maximum number of nodes survive. If RMS is used to control applications, it will move the applications to the surviving cluster nodes after split-brain resolution has succeeded.

This scenario is achieved as follows:

- By means of Cluster Admin, set the SF node weight values to 1. 1 is the default value for this attribute, so new cluster installations may simply ignore it.
- By means of the RMS Wizard Tools, set the RMS attribute ShutdownPriority of all userApplications to 0. 0 is the default value for this attribute, so if you are creating new applications you may simply ignore this setting.

As can be seen from the default values of both the SF weight and the RMS ShutdownPriority, if no specific action is taken by the administrator to define a split-brain resolution outcome, Largest Sub-cluster Survival is selected by default.

Specific Hardware Survival

In this scenario, the administrator has determined that one or more nodes contain hardware that is critical to the successful functioning of the cluster as a whole.

This scenario is achieved as follows:

- Using Cluster Admin, set the SF node weight of the cluster nodes containing the critical hardware to values more than double the combined value of cluster nodes not containing the critical hardware.
- Using the RMS Wizard Tools, set the RMS attribute ShutdownPriority of all userApplications to 0. 0 is the default value for this attribute so if you are creating new applications you may simply ignore this setting.

As an example, in a four-node cluster in which two of the nodes contain critical hardware, set the SF weight of those critical nodes to 10 and set the SF weight of the non-critical nodes to 1. With these settings, the combined weights of both non-critical nodes will never exceed even a single critical node.

Specific Application Survival

In this scenario, the administrator has determined that application survival on the node where the application is currently Online is more important than node survival. This can only be implemented if RMS is used to control the application(s) under discussion. This can get complex if more than one application is deemed to be critical and those applications are running on different cluster nodes. In some split-brain situations, all applications will not survive and will need to be switched over by RMS after the split-brain has been resolved.

This scenario is achieved as follows:

- Using Cluster Admin, set the SF node weight values to 1. 1 is the default value for this attribute, so new cluster installations may simply ignore it.
- Using the RMS Wizard Tools, set the RMS attribute ShutdownPriority of the critical applications to more than double the combined values of all non-critical applications, plus any SF node weight.
- ShutdownPriority is set with a range from 1 to 20.

As an example, in a four-node cluster there are three applications. Set the SF weight of all the nodes to 1, and set the ShutdownPriority of the three applications to 20, 1, 1. This would define that the application with a ShutdownPriority of 20 would survive no matter what, and further that the sub-cluster containing the node on which this application was running would survive the split no matter what. To clarify this example, if the cluster nodes were A, B, C and D all with a weight of 1, and App1, App2 and App3 had ShutdownPriority of 20, 1 and 1 respectively, even in the worst-case split that node D with App1 was split from nodes A, B and C which had applications App2 and App3 the weights of the sub-clusters would be D with 21 and A,B,C with 5. The heaviest sub-cluster (D) would win.

7.4 Configuring the Shutdown Facility

This section describes how to use Command Line Interface (CLI) to configure the Shutdown Facility (SF).

7.4.1 Shutdown Daemon

To configure the Shutdown Daemon (SD), you will need to modify the file

`/etc/opt/SMAW/SMAWsf/rcsd.cfg` on every node in the cluster.

A file, `rcsd.cfg.template`, is provided under the `/etc/opt/SMAW/SMAWsf` directory, which is a sample configuration file for the Shutdown Daemon using fictitious nodes and agents.



See

.....
It is important that the `rcsd.cfg` file is identical on all cluster nodes; care should be taken in administration to ensure that this is true.
.....

Here is the example of `rcsd.cfg`, which is created by editing an `rcsd.cfg.template`.

```
shastal,weight=2,admIP=shastal:agent=SA_irmcp,timeout=20:agent=SA_irmcr,timeout=20:agent=SA_irmcf,ti  
meout=20
```

```
shasta2,weight=2,admIP=shasta2:agent=SA_irmcp,timeout=20:agent=SA_irmcr,timeout=20:agent=SA_irmcf,ti  
meout=20
```

The configuration file must be created in the `/etc/opt/SMAW/SMAWsf` directory and must use `rcsd.cfg` as the file name.

The format of the configuration file is as follows:

```
cluster-node1 [,weight=w1][,admIP=admIP1]:agent=SA1, timeout=t1[:agent=SA2, timeout=T2]...  
cluster-node2 [,weight=w2][,admIP=admIP2]:agent=SA1, timeout=t1[:agent=SA2, timeout=T2]...  
...
```

- *cluster-nodeN* is the *cname* of a node within the cluster.

- *agent* and *timeout* are reserved words.

- *SAN* is the command name of a SA.

- *tN* is the timeout duration (seconds) of the SA.

The SA runs for the following cases:

- When a node is forcibly eliminated.

- When checking the connection to the option hardware used when a node is forcibly eliminated.

When a timeout occurs at the forced node elimination, the processing of the SA is stopped, and then the next SA is started. If all SAs fail to perform their processing, the node is left in the LEFTCLUSTER state.

When a timeout occurs when checking the connection to the option hardware used at the forced node elimination, the processing of the SA is stopped, and then the node becomes the TestFailed state.

- *wN* is the node weight.

- *admIPN* is the admin interface on the Administrative LAN on this cluster node. Available IP addresses are IPv4 and IPv6 address. The link local address of IPv6 is not available. When specifying the IPv6 address, enclose it in brackets "[]".

(Example: [1080:2090:30a0:40b0:50c0:60d0:70e0:80f0])

The order of the SAs in the configuration file should be such that the first SA in the list is the preferred SA. If this preferred SA is issued a shutdown request and if its response indicates a failure to shut down, the secondary SA is issued the shutdown request. This request/response is repeated until either the shutdown agent responds with a successful shutdown, or all SAs have been tried. If no SA is able to successfully shut down a cluster node, then operator intervention is required and the node is left in the LEFTCLUSTER state.

The location of the log file will be `/var/opt/SMAWsf/log/rcsd.log`.

7.4.2 Shutdown Agents

This section contains information on how to configure the following SAs with CLI.

- IPMI
- ICMP
- VMCHKHOST
- libvirt
- VMware vCenter Server functional cooperation
- FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API
- OpenStack API
- AWS CLI
- Azure CLI
- NIFCLOUD API

IPMI

To configure the IPMI shutdown agent (SA_ipmi), you need to create or modify the following file:

```
/etc/opt/SMAW/SMAWsf/SA_ipmi.cfg
```

The template of SA_ipmi.cfg file can be found at the following location:

```
/etc/opt/SMAW/SMAWsf/SA_ipmi.cfg.template
```

SA_ipmi.cfg file contains lines with four fields (and some subfields) on each line. Each line defines a node in the cluster that can be powered off (leaving it off) or powered off and then on again (power cycle). The fields are as follows:

- *cfname*
The name of the node in the CF cluster.

Note

.....
cfname must correspond to the IP address of the IPMI compliant onboard LAN interface in these nodes.
.....

- *Access-Information*
The access information is of the following format:

```
ip-address-of-unit:user:password
```

For ip-address-of-unit, specify the IP address of iRMC or BMC. Available IP addresses are IPv4 and IPv6 address. The link local address of IPv6 is not available. When specifying the IPv6 address, enclose it in brackets "[]".
(Example: [1080:2090:30a0:40b0:50c0:60d0:70e0:80f0])

- *Action*
The action can either be cycle or leave-off. If it is cycle, the node will be powered on again after powering off. If it is leave-off, manual action is required to turn the system back on.

Note

.....
The permissions of the SA_ipmi.cfg file are read/write by root only. This is to protect the password to the iRMC/BMC unit.
.....

The following is an example of SA_ipmi.cfg file:

```
shasta1 192.168.200.1::root:ipmipwd cycle
shasta2 192.168.200.2::root:ipmipwd cycle
shasta3 192.168.200.3::root:ipmipwd leave-off
shasta4 192.168.200.4::root:ipmipwd leave-off
```

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_ipmi.log
```

ICMP

To configure the ICMP shutdown agent (SA_icmp), you need to create or modify the following file:

```
/etc/opt/SMAW/SMAWsf/SA_icmp.cfg
```

The template of SA_icmp.cfg file can be found at the following location:

```
/etc/opt/SMAW/SMAWsf/SA_icmp.cfg.template
```

The format of SA_icmp.cfg file is as follows:

```
TIME_OUT=value
cfname:ip-address-of-node:NIC-name1,NIC-name2
```

The editable fields are defined as follows:

- *value*
Specify the time in seconds to check for the existence of a node. The recommended value is 5 seconds.
- *cfname*
Specify a CF node name.
- *ip-address-of-node*
Specify any of the following IP addresses of cfname. Available IP addresses are IPv4 and IPv6 address. The link local address of IPv6 is not available. When specifying the IPv6 address, enclose it in brackets "[]".
(Example: [1080:2090:30a0:40b0:50c0:60d0:70e0:80f0])
 - Administrative LAN
 - Public LAN
 - Cluster interconnect

You must describe one or more IP addresses for all the nodes in the cluster. Add a new line to specify more than one LAN routes. It is recommended that you specify more than one routes to check the LAN routes.
- *NIC-nameX*
For ip-address-of-node, specify a network interface used to check the existence of the node. If there are more than one network interfaces, separate them with a comma.



Note

Note the following when you describe network interfaces:

- When duplicating the network by GLS, you need to describe all redundant network interfaces. (For example, eth0,eth1)
- When bonding NICs, you need to describe a bonding device after the IP address. (For example, bond0)
- When describing cluster interconnects, you need to describe all network interfaces used in all paths of the cluster interconnects. (For example, eth2,eth3)

The following is an example of SA_icmp.cfg file:

```
TIME_OUT=5
shasta1:10.20.30.100:eth0,eth1
shasta1:10.20.40.200:eth2
shasta2:10.20.30.101:eth0,eth1
shasta2:10.20.40.201:eth2
```

The log file is stored in the following:

```
/opt/SMAW/SMAwf/log/SA_icmp.log
```

VMCHKHOST

To configure the Shutdown Agent (SA_vmchkhos) to switch the HOST OS in the event of an error in the virtual machine environment, you must create or modify the following file:

```
/etc/opt/SMAW/SMAwf/SA_vmchkhos.cfg
```

The template of SA_vmchkhos.cfg file is stored in the following:

```
/etc/opt/SMAW/SMAwf/SA_vmchkhos.cfg.template
```

Each line of SA_vmchkhos.cfg file includes the following five fields:

- *guest-cfname*
CF node name of a guest OS.
- *host-cfname*
CF node name of the Host OS.

- *ip-address*
IP address of the Host OS. Available IP addresses are IPv4 and IPv6 address. The link local address of IPv6 is not available.
- *user*
Account of the Host OS.
For a KVM environment, specify the user name that was created when setting libvirt to the Shutdown Facility.
- *password*
Login password for the account that was specified with "user."
The encrypted password for the user with general privileges for the Shutdown Facility, which was used when setting libvirt to the Shutdown Facility, is used.

The following is an example of SA_vmchkhosst.cfg file:

```
shasta1 hostos2 192.168.200.1 FJSVvmSP 3CA1wxVXKD8a93077BaEkA==
shasta2 hostos3 192.168.200.2 FJSVvmSP 3CA1wxVXKD8a93077BaEkA==
```

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmchkhosst.log
```

libvirt

To configure the Shutdown Agent (SA_libvirtgp) for PANIC or Shutdown Agent (SA_libvirtgr) for RESET of libvirt, you must create or modify the following files respectively.

```
/etc/opt/SMAW/SMAWsf/SA_libvirtgp.cfg
/etc/opt/SMAW/SMAWsf/SA_libvirtgr.cfg
```

The templates of SA_libvirtgp.cfg file and SA_libvirtgr.cfg file are stored in the following:

```
/etc/opt/SMAW/SMAWsf/SA_libvirtgp.cfg.template
/etc/opt/SMAW/SMAWsf/SA_libvirtgr.cfg.template
```

Each line of SA_libvirtgp.cfg file and SA_libvirtgr.cfg file includes the following five fields:

- *cname*
Node name on the CF cluster.
- *domain*
Domain name of a guest OS.
- *ip-address*
IP address of the hypervisor. Available IP addresses are IPv4 and IPv6 address. The link local address of IPv6 is not available.
- *user*
Account of the hypervisor. The user for shutdown facility is specified.
- *password*
Login password for the account specified in "user" field. Describe the password encrypted by the encryption command, sfcipher(8). For details on the sfcipher(8) command, see the manual pages of "sfcipher(8)."

The following is an example of SA_libvirtgp.cfg file:

```
shasta1 domain2 192.168.200.1 user 3CA1wxVXKD8a93077BaEkA==
shasta2 domain3 192.168.200.2 user 3CA1wxVXKD8a93077BaEkA==
```

The following is an example of SA_libvirtgr.cfg file:

```
shasta1 domain2 192.168.200.1 user 3CA1wxVXKD8a93077BaEkA==
shasta2 domain3 192.168.200.2 user 3CA1wxVXKD8a93077BaEkA==
```

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_libvirtgp.log
/var/opt/SMAWsf/log/SA_libvirtgr.log
```


VMware vCenter Server functional cooperation

To configure the VMware vCenter Server functional cooperation shutdown agent (SA_vwvnr), you need to create or modify the following file:

```
/etc/opt/SMAW/SMAWsf/SA_vwvnr.cfg
```

The template of SA_vwvnr.cfg file can be found at the following location:

```
/etc/opt/SMAW/SMAWsf/SA_vwvnr.cfg.template
```

The format of SA_vwvnr.cfg file is as follows:

```
CFName: cfname
VMName: vmname
vCenter_IP: IP address
vCenter_Port: port
user: user
passwd: passwd
```

The editable fields are defined as follows:

- *cfname*

The name of the node in the CF cluster.

- *vmname*

VM name of the guest OS. Use the character code UTF-8 if the name is described in Japanese. Use the character code UTF-8 if the name is described in Japanese.

- *IP address*

The IP address of VMware vCenter Server. Available IP addresses are IPv4 and IPv6 address. The link local address of IPv6 is not available.

- *port*

Specify the port number of VMware vCenter Server.

When using the default value (443), describe it as "vCenter_Port:".

- *user*

The user to log in VMware vCenter Server. Specify the user for the shutdown agent.

- *password*

Login password for the user specified in "user" field. Describe the password encrypted by the encryption command, sfcipher(8). For details on the sfcipher(8) command, see the manual pages of "sfcipher(8)."

The following is an example of SA_vwvnr.cfg file:

Example: Port number of VMware vCenter Server is the default value (443)

```
CFName: shastal
VMName: shastalvm
vCenter_IP: 192.168.100.1
vCenter_Port:
user: PCL
passwd: 3CA1wxVXKD8a93077BaEka==
```

Example: Port number of VMware vCenter Server is 10443

```
CFName: shastal
VMName: shastalvm
vCenter_IP: 192.168.100.1
vCenter_Port: 10443
```

```
user: PCL
passwd: 3CA1wxVXKD8a93077BaEkA==
```

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vwvmr.log
```

FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API

To configure the FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API shutdown agent (SA_vmk5r), you must create or modify the following files:

```
/opt/SMAW/SMAWRrms/etc/k5_endpoint.cfg
/etc/opt/SMAW/SMAWsf/SA_vmk5r.cfg
```

- /opt/SMAW/SMAWRrms/etc/k5_endpoint.cfg

The format of k5_endpoint.cfg file is as follows:

```
DOMAIN_NAME=domainname
PROJECT_NAME=projectname
IDENTITY=identityurl
COMPUTE=computeurl
```

The editable fields are defined as follows:

- *domainname*
Domain name (contractor number) of an FJcloud-O/FJcloud-Baremetal environment.
- *projectname*
Project name that configures the cluster in an FJcloud-O/FJcloud-Baremetal environment.
- *identityurl*
Endpoint URL of the regional user management of a region used in an FJcloud-O/FJcloud-Baremetal environment.
- *computeurl*
Endpoint URL of the compute (standard service) of a region used in an FJcloud-O/FJcloud-Baremetal environment.

The following is an example of the k5_endpoint.cfg file:

```
DOMAIN_NAME=primecluster_domain
PROJECT_NAME=primecluster_project
IDENTITY=https://identity.jp-east-3.cloud.global.fujitsu.com
COMPUTE=https://compute.jp-east-3.cloud.global.fujitsu.com
```

- /etc/opt/SMAW/SMAWsf/SA_vmk5r.cfg

The template of SA_vmk5r.cfg file is in the following location:

```
/etc/opt/SMAW/SMAWsf/SA_vmk5r.cfg.template
```

The format of SA_vmk5r.cfg file is as follows:

```
cfname machinename user passwd Action
```

Each line of SA_vmk5r.cfg file includes five fields. Each line defines a node in the cluster that can be powered off (leaving it off) or powered off and then on again (power cycle). The fields are as follows:

- *cfname*
The name of the node in the CF cluster.
- *machinename*
The virtual machine name on which the cluster host is working.

- *user*

User name to control the virtual machine.

- *password*

Login password for the user specified in "user" field. Describe the password encrypted by the encryption command, sfcipher(8). For details on the sfcipher(8) command, see the manual pages of "sfcipher(8)."

- *Action*

The action can either be cycle or leave-off. If it is cycle, the node will be powered on again after powering off. If it is leave-off, manual action is required to turn the system back on.

The following is an example of SA_vmk5r.cfg file:

```
node1 vm1 pcl O/gm+AYuWwE7ow3dgVG/Nw== cycle
node2 vm2 pcl O/gm+AYuWwE7ow3dgVG/Nw== cycle
```

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmk5r.log
```

OpenStack API

To configure the OpenStack API shutdown agent (SA_vmosr), you must create or modify the following files:

```
/opt/SMAW/SMAWRrms/etc/os_endpoint.cfg
/etc/opt/SMAW/SMAWsf/SA_vmosr.cfg
```

- /opt/SMAW/SMAWRrms/etc/os_endpoint.cfg

The format of os_endpoint.cfg file is as follows:

```
DOMAIN_NAME=domainname
PROJECT_NAME=projectname
IDENTITY=identityurl
COMPUTE=computeurl
```

The editable fields are defined as follows:

- *domainname*

Domain name of an RHOSP environment.

- *projectname*

Project name that configures the cluster in an RHOSP environment.

- *identityurl*

Endpoint URL of the Identity service used in an RHOSP environment.

Note: Do not include any character strings from "/v.X.X" in URL.

- *computeurl*

Endpoint URL of the Compute service used in an RHOSP environment.

Note: Do not include any character strings from "/v.X.X" in URL.

The following is an example of the os_endpoint.cfg file:

```
DOMAIN_NAME=primecluster_domain
PROJECT_NAME=primecluster_project
IDENTITY=https://192.168.11.11:5000
COMPUTE=https://192.168.11.11:8774
```

- /etc/opt/SMAW/SMAWsf/SA_vmosr.cfg

The template of SA_vmosr.cfg file is in the following location:

```
/etc/opt/SMAW/SMAWsf/SA_vmosr.cfg.template
```

The format of SA_vmosr.cfg file is as follows:

```
cfname instancename user passwd
```

The editable fields are defined as follows:

- *cfname*

The name of the node in the CF cluster.

- *instancename*

The instance name on which the cluster host is working.

- *user*

User name to control the instance.

- *password*

Login password for the user specified in "user" field. Describe the password encrypted by the encryption command, sfcipher(8). For details on the sfcipher(8) command, see the manual pages of "sfcipher(8)."

The following is an example of SA_vmosr.cfg file:

```
node1 instance1 pcl O/gm+AYuWwE7ow3dgVG/Nw==
node2 instance2 pcl O/gm+AYuWwE7ow3dgVG/Nw==
```

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmosr.log
```

AWS CLI

To configure the AWS CLI shutdown agent (SA_vmawsAsyncReset), you must create or modify the following file:

```
/etc/opt/SMAW/SMAWsf/SA_vmawsAsyncReset.cfg
```

The template of the SA_vmawsAsyncReset.cfg file is in the following location:

```
/etc/opt/SMAW/SMAWsf/SA_vmawsAsyncReset.cfg.template
```

The format of the SA_vmawsAsyncReset.cfg file is as follows:

```
cfname InstanceID [ProfileName]
```

Each line of the SA_vmawsAsyncReset.cfg file includes three fields. Each line defines a node in the cluster. The fields are as follows:

- *cfname*

The CF node name of the cluster host.

- *InstanceID*

The instance ID of AWS on which the cluster host is working.

- *[ProfileName]*

Describe the profile name of the credentials used with the AWS Command Line Interface.

When this item is omitted, the shutdown agent works with the default profile.

The following is an example of the SA_vmawsAsyncReset.cfg file:

```
node1 i-abcdef0123456789a userprofile1
node2 i-abcdef0123456789b userprofile1
```

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmawsAsyncReset.log
```

Azure CLI

To configure the Azure CLI shutdown agent (SA_vmazureReset), you must create or modify the following file:

```
/etc/opt/SMAW/SMAWsf/SA_vmazureReset.cfg
```

The template of the SA_vmazureReset.cfg file is in the following location:

```
/etc/opt/SMAW/SMAWsf/SA_vmazureReset.cfg.template
```

The format of the SA_vmazureReset.cfg file is as follows:

```
cfname ResourceID AppID TenantID CertPath {cycle | leave-off}
```

Each line of the SA_vmazureReset.cfg file includes six fields. Each line defines a node in the cluster. The fields are as follows:

- *cfname*
The CF node name of the cluster host.
- *ResourceID*
The resource ID of the Azure virtual machine on which the cluster host is working.
- *AppID*
Specify the application ID when registering the service principal.
- *TenantID*
Specify the tenant ID when registering the service principal.
- *CertPath*
Specify the path of the certificate.
- *{cycle | leave-off}*
Describe the operation after forcibly stopping the node.
cycle: Restart the node after forcibly stopping the node.
leave-off: The node is not restarted after it is forcibly stopped.

The following is an example of the SA_vmazureReset.cfg file:

```
node1 /subscriptions/1e234d12-39b8-49db-881a-35aa03b402b9/resourceGroups/pcl-mw1/providers/Microsoft.Compute/virtualMachines/node1 d5b7dac1-718f-448b-8e11-4a8cca6d9004 8ff7ddfd-fbcb-4700-ae52-6d071ac8d1b4 /root/examplecert.pem cycle
node2 /subscriptions/1e234d12-39b8-49db-881a-35aa03b402b9/resourceGroups/pcl-mw1/providers/Microsoft.Compute/virtualMachines/node2 d5b7dac1-718f-448b-8e11-4a8cca6d9004 8ff7ddfd-fbcb-4700-ae52-6d071ac8d1b4 /root/examplecert.pem cycle
```

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmazureReset.log
```

NIFCLOUD API

To configure the NIFCLOUD API shutdown agent (SA_vmnifclAsyncReset), you must create or modify the following file:

```
/etc/opt/SMAW/SMAWsf/SA_vmnifclAsyncReset.cfg
```

The template of the SA_vmnifclAsyncReset.cfg file is in the following location:

```
/etc/opt/SMAW/SMAWsf/SA_vmnifclAsyncReset.cfg.template
```

The format of the SA_vmnifclAsyncReset.cfg file is as follows:

```
CFNameX ServerName Region Accesskey SecretAccesskey
```

Each line of the SA_vmnifclAsyncReset.cfg file includes five fields. Each line defines a node in the cluster. The fields are as follows:

- *CFNameX*
The CF node name of the cluster host.
- *ServerName*
The server name of NIFCLOUD on which the cluster host is working.
- *Region*
The region of the endpoint used in a NIFCLOUD environment.
Specify the "jp-east-X" format for the East Japan region and the "jp-west-X" format for the West Japan region.
- *Accesskey*
The AccessKey used for authenticating the NIFCLOUD API.
- *SecretAccesskey*
The SecretAccessKey used for authenticating the NIFCLOUD API. Describe the SecretAccessKey encrypted by the encryption command, sfcipher(8).
For details on the sfcipher(8) command, see the manual pages of "sfcipher(8)."

The following is an example of the SA_vmnifclAsyncReset.cfg file:

```
# cat /etc/opt/SMAW/SMAWsf/SA_vmnifclAsyncReset.cfg
node1 pclserver1 jp-east-1 123456789ABCDEFGHIJK
PIJgi9wm4sYJlfWShOxWP6fpz0Xb6t9KmRc4sPDsRBIGfm6gT5m0xH2H2xf1r38G
node2 pclserver2 jp-east-1 123456789ABCDEFGHIJK
PIJgi9wm4sYJlfWShOxWP6fpz0Xb6t9KmRc4sPDsRBIGfm6gT5m0xH2H2xf1r38G
```

The log file is stored in the following:

```
/var/opt/SMAWsf/log/SA_vmnifclAsyncReset.log
```

7.5 SF administration

This section provides information on administering SF. SF can be administered with the CLI or Cluster Admin. It is recommended to use Cluster Admin.

7.5.1 Starting and stopping SF

This section describes the following administrative procedures for starting and stopping SF:

- Manually via the CLI
- Automatically via the systemd service

7.5.1.1 Starting and stopping SF manually

SF may be manually started or stopped by using the sdttool(1M) command. The sdttool(1M) command has the following options:

```
sdtool [-bsSre] [-k CF-node-name] [-d off|on]
```

-b	Start
-s	State (Human-readable format)
-S	State (Easy-to-analyze format)
-r	Re-configuration
-e	End
-k	Stop
-d	Debug

See the sdttool manual pages for details.

7.5.1.2 Starting and stopping SF automatically

SF can be started automatically using the `smawsf.service` of a service of `systemd`. The start/stop script for SF is installed as `/etc/init.d/SMAWsf`.

7.5.2 Checking SA status

This section describes the procedures on how to check the current status of SA.

To check the current status of SA, execute the following command on the node where the status of SA is checked.

```
# sdtool -s
```

Example

```
# sdtool -s
Cluster Host Agent SA State Shut State Test State Init State
-----
shasta1      SA_xx Idle      Unknown   TestWorked InitWorked
shasta1      SA_xx Idle      Unknown   TestWorked InitWorked
shasta2      SA_xx Idle      Unknown   TestWorked InitWorked
shasta2      SA_xx Idle      Unknown   TestWorked InitWorked
```

The following items are displayed.

Cluster Host

Node name in the cluster

Agent

Name of the shutdown agent

SA State

Current status of SA.

The status are as follows.

Status	Description
Idle	No currently running SA
Init-ing	Initializing
InitWorked	Initialization successfully completed
InitFailed	Initialization failed
Testing	Testing (10-minute interval)
TestWorked	Test successfully completed
TestFailed	Test failed
Killing	Force stop node ongoing
KillWorked	Force stop node successfully completed
KillFailed	Force stop node failed
UnInit-ing	Uninitializing
UnInitWorked	Uninitialization successfully completed
UnInitFailed	Uninitialization failed

Shut State

Status of force node stop.

The status are as follows.

Status	Description
Unknown	Force stop node is not executed
Killing	Force stop node ongoing
KillWorked	Force stop node successfully completed
KillFailed	Force stop node failed

Test State

Status of test executed every 10 minutes.

The status are as follows.

Status	Description
Unknown	Currently untested
Testing	Testing
TestWorked	Test successfully completed
TestFailed	Test failed (force stop cannot be executed)

Init State

Status of initialization/uninitialization of SA.

The status are as follows.

Status	Description
Unknown	Initialization is not executed
Init-ing	Initializing
InitWorked	Initialization successfully completed
InitFailed	Initialization failed
UnInit-ing	Uninitializing
UnInit-Failed	Uninitialization failed

Note

- If Init State is displayed as InitFailed, it indicates that an error occurred in initializing the SA.
- If Test State is displayed as TestFailed, it indicates that an error occurred while the SA is testing if the node displayed as Cluster Host can be stopped, and the node cannot be stopped forcibly.
In this case, an error may be occurring in software, hardware, or network resources that are used by the SA. Solve the error immediately.
- If Test State or Init State is displayed as Unknown, it indicates that SF has not yet stopped the node, tested the route, or initialized SA . Test State and Init State are temporary displayed as Unknown until the actual status is confirmed.
- If TestFailed or InitFailed is displayed, check the SA log file or /var/log/messages. In the log file, you can find the cause of the error why the SA testing or SA initialization failed. Once the cause of the error is solved and SF is restarted, InitWorked or TestWorked will be displayed as the current status.

7.6 Debugging

Whenever there is a recurring problem where the cause cannot be easily detected, turn on the debugger with the following command:

```
# sdttool -d on
```

The debug information is output in the log file `/var/opt/SMAWsf/log/rcsd.log`, which will provide additional information to find the cause of the problem. You can also use the `sdtol -d off` command to turn off the debugger.

Note that the `rcsd` log file does not contain logging information from a shutdown agent. Refer to the SA specific log files for logging information from each shutdown agent.

Chapter 8 Diagnostics and troubleshooting

This chapter provides help for troubleshooting and problem resolution for PRIMECLUSTER Cluster Foundation. This chapter will help identify the causes of problems and possible solutions. If a problem is in another component of the PRIMECLUSTER suite, the reader will be referred to the appropriate manual. This chapter assumes that the installation and verification of the cluster have been completed.

8.1 Beginning the process

Start the troubleshooting process by gathering information to help identify the causes of problems. You can use the CF log viewer facility from the Cluster Admin GUI, look for messages on the console, or look for messages in the

`/var/log/messages` file. You can use the `cftool(1M)` command for checking states, configuration information. To use the CF log viewer click on the Tools pull-down menu and select View Syslog messages (refer to the section "4.8 Using PRIMECLUSTER log viewer" for more details). The log messages are displayed. You can search the logs using a date/time filter or scan for messages based on severity levels. To search based on date/time, use the date/time filter and press the *Filter* button. To search based on severity levels, click on the *Severity* button and select the desired severity level. You can use keyword also to search the log. To detach the CF log viewer window, click on the *Detach* button; click on the *Attach* button to attach it again.

Collect information as follows:

- Look for messages on the console that contain the identifier CF.
- Look for messages in `/var/log/messages`. You might have to look in multiple files (`/var/log/messages.N`).
- Use `cftool` as follows:
 - `cftool -l`: Check local node state
 - `cftool -d`: Check device configuration
 - `cftool -n`: Check cluster node states
 - `cftool -r`: Check the route status

Error log messages from CF are always placed in the `/var/log/messages` file; some messages may be replicated on the console. Other device drivers and system software may only print errors on the console. To have a complete understanding of the errors on a system, both console and error log messages should be examined. The section "4.5 Error Messages" in "PRIMECLUSTER Messages" contains messages that can be found in the `/var/log/messages` file. This list of messages gives a description of the cause of the error. This information is a good starting point for further diagnosis.

All of the parts of the system put error messages in this file or on the console and it is important to look at all of the messages, not just those from the PRIMECLUSTER suite. The following is an example of a CF error message from the `/var/log/messages` file:

```
Aug 26 13:31:05 shastal kernel: LOG3.0429320 1080024 100014 0 1.0 CF: Giving UP Mastering  
(Cluster already Running)
```

The parts of this message are as follows:

The first 80 bytes are the log3 prefix:

```
Aug 26 13:31:05 shastal kernel: LOG3. .0429320 1080024 100014 0 1.0 cf:elmlog
```

This parts of the message is a standard prefix on each CF message in the log file that gives the date and time, the node name, and log3 specific information. Only the date, time, and node name are important in this context. The remainder is the error message from CF as follows:

```
CF: Giving UP Mastering (Cluster already Running).
```

When the node detects a joined server, and it enters an existing cluster instead of making a new cluster, this message is output. Refer to "Chapter 5 CF Messages" in "PRIMECLUSTER Messages" for details of the message.

Several options for the command `cftool(1M)` are available as sources for information. The following is an example:

```
root@shastal> cftool -l  
Node      Number  State  Os      Cpu      Flags  
shastal   2       UP     Linux   Pentium  0
```

This shows that the local node has joined a cluster as node number 2 and is currently UP. This is the normal state when the cluster is operational. Another possible response is as follows:

```
root@shasta1> cftool -l
Node      Number  State      Os      Cpu      Flags
shasta1   --      COMINGUP   --      --      --
```

This indicates that the CF driver is loaded and that the node is attempting to join a cluster. If the node stays in this state for more than a few minutes, then something is wrong and we need to examine the /var/log/messages file. In this case, we see as follows:

```
root@shasta1> tail /var/log/messages
Aug 28 10:38:25 shasta1 kernel: CF: (TRACE): Load: Complete.
Aug 28 10:38:25 shasta1 kernel: CF: (TRACE): JoinServer: Startup.
Aug 28 10:38:25 shasta1 kernel: CF: Giving UP Mastering (Cluster already Running).
Aug 28 10:38:25 shasta1 kernel: CF: shasta1: busy: local node not DOWN: retrying.
```

We see that this node is in the LEFTCLUSTER state on another node (shasta3). To resolve this condition, see Chapter "5.1 Description of the LEFTCLUSTER state" for a description of the LEFTCLUSTER state and the instructions for resolving the state.

The next option to *cftool*(1M) shows the device states as follows:

```
root@shasta1> cftool -d
Number  Device  Type  Speed  Mtu  State  Configured  Address
1       eth0    4     100   1432  UP     YES         00.03.47.c2.a8.82
2       eth1    4     100   1432  UP     YES         00.02.b3.88.09.f1
3       eth2    4     100   1432  UP     NO          00.02.b3.88.09.ea
```

Here we can see that there are two interconnects configured for the cluster (the lines with YES in the Configured column). This information shows the names of the devices and the device numbers for use in further troubleshooting steps.

The *cftool -n* command displays the states of all the nodes in the cluster. The node must be a member of a cluster and UP in the *cftool -l* output before this command will succeed:

```
root@shasta1> cftool -n
Node      Number  State      Os      Cpu
shasta1   1       UP         Linux   Pentium
shasta2   2       UP         Linux   Pentium
```

This indicates that the cluster consists of two nodes shasta1 and shasta2, both of which are UP. If the node has not joined a cluster, the command will wait until the join succeeds.

cftool -r lists the routes and the current status of the routes as follows:

```
root@shasta1> cftool -r
Node      Number  Srcdev  Dstdev  Type  State  Destaddr
shasta1   1       1       4       4     UP     00.03.47.c2.a8.82
shasta1   1       1       5       5     UP     00.03.47.c2.a8.cc
shasta2   2       2       4       4     UP     00.03.47.d1.af.ec
shasta2   2       2       5       5     UP     00.03.47.d1.af.ef
```

This shows that all of the routes are UP. If a route shows a DOWN state, then the step above where we examined the error log should have found an error message associated with the device. At least the CF error noting the route is down should occur in the error log. If there is not an associated error from the device driver, then the diagnosis steps are covered below.

The last route to a node is never marked DOWN, it stays in the UP state so that the software can continue to try to access the node. If a node has left the cluster or gone down, there will still be an entry for the node in the route table and one of the routes will still show as UP. Only the *cftool -n* output shows the state of the nodes. The following example shows:

```
root@shasta1> cftool -r
Node      Number  Srcdev  Dstdev  Type  State  Destaddr
shasta2   2       3       2       4     UP     00.03.47.d1.af.ec
shasta1   1       3       3       4     UP     00.03.47.c2.a8.82
```

```
root@shasta1> cftool -n
Node      Number  State      Os      Cpu
```

shasta2	1	LEFTCLUSTER	Linux	Pentium
shastal	2	UP	Linux	Pentium

8.2 Symptoms and solutions

The previous section discussed the collection of data. This section discusses symptoms and gives guidance for troubleshooting and resolving the problems. The problems dealt with in this section are divided into two categories: problems with joining a cluster and problems with routes, either partial or complete loss of routes. The solutions given here are either to correct configuration problems or to correct interconnect problems. Problems outside of these categories or solutions to problems outside of this range of solutions are beyond the scope of this manual and are either covered in another product's manual or require technical support from field engineers. Samples from the error log (`/var/log/messages`) have the `log3` header stripped from them in this section.

8.2.1 Join-related problems

Join problems occur when a node is attempting to become a part of a cluster. This section describes the nodes that have successfully joined a cluster. For the problems of the initial startup when a node joins a cluster for the first time, see the operation check section. If this node has previously been a part of the cluster and is now failing to rejoin the cluster, here are some initial steps in identifying the problem.

8.2.1.1 Identifying join-related problems

First, look in the error log and at the console messages for any clue to the problem. Have the Ethernet drivers reported any errors? Any other unusual errors? If there are errors in other parts of the system, the first step is to correct those errors. Once the other errors are corrected, or if there were no errors in other parts of the system, proceed as follows.

Is the CF device driver loaded? The device driver puts a message in the log file when it loads and the `cftool -l` command will indicate the state of the driver. The logfile message looks as follows:

```
CF: (TRACE): JoinServer: Startup.
```

`cftool -l` prints the state of the node as in the following:

```
root@shastal> cftool -l
Node      Number  State      Os
shastal   --      COMINGUP   --
```

This indicates that the driver is loaded and that the node is trying to join a cluster. If the errorlog message above does not appear in the logfile or the `cftool -l` command fails, then the device driver is not loading. If there is no indication in the `/var/log/messages` file or on the console why the CF device driver is not loading, it could be that the CF kernel binaries or commands are corrupted, and you might need uninstall and reinstall CF. Before any further steps can be taken, the device driver must be loaded.

After the CF device driver is loaded, it attempts to join a cluster as indicated by the following message:

```
CF: (TRACE): JoinServer: Startup
```

The join server will attempt to contact another node on the configured interconnects. If one or more other nodes have already started a cluster, this node will attempt to join that cluster. The following message in the error log indicates that this has occurred:

```
CF: Giving UP Mastering (Cluster already Running).
```

If this message does not appear in the error log, then the node did not see any other node communicating on the configured interconnects and it will start a cluster of its own. The following two messages will indicate that a node has formed its own cluster as follows:

```
CF: Local Node shastal Created Cluster SHASTAS. (#0000 1)
CF: Node shastal Joined Cluster SHASTAS. (#0000 1)
```

At this point, we have verified that the CF device driver is loading and the node is attempting to join a cluster. In the following list, problems are described with corrective actions. Find the problem description that most closely matches the symptoms of the node being investigated and follow the steps outlined there.

8.2.1.2 Solving join-related problems

Problem

The following are typical join problems.

The node does not join an existing cluster; it forms a cluster of its own.

Diagnosis

The error log shows the following messages:

```
CF: (TRACE): JoinServer: Startup.
CF: Local Node shasta1 Created Cluster SHASTAS. (#0000 1)
CF: Node shasta1 Joined Cluster SHASTAS. (#0000 1)
```

This indicates that the CF devices are all operating normally and suggests that the problem is occurring some place in the interconnect. The first step is to determine if the node can see the other nodes in the cluster over the interconnect. Use `cftool(1M)` to send an echo request to all the nodes of the cluster:

```
root@shasta1> cftool -e
Localdev Srcdev Address Cluster Node Number Joinstate
3 2 00.03.47.c2.a8.82 SHASTAS shasta1 2 6
3 3 00.03.47.d1.af.ec SHASTAS shasta2 1 6
```

This shows that node shasta2 sees node shasta1 using interconnect device 3 (Localdev) on shasta2 and device 2 (Srcdev) on shasta1. If the `cftool -e` shows only the node itself continue on in this section." If some or all of the expected cluster nodes appear in the list, attempt to rejoin the cluster by unloading the CF driver and then reloading the driver as follows:

```
root@shasta1> cfconfig -u
root@shasta1> cfconfig -l
```



There is no output from either of these commands, only error messages in the error log.

Problem

The node does not join the cluster and some or all the nodes respond to `cftool -e`.

Diagnosis

At this point, we know that the CF device is loading properly and that this node can communicate with at least one other node in the cluster. We should suspect at this point that the interconnect is missing messages. One way to test this hypothesis is to repeatedly send echo requests and see if the result changes over time, for example:

```
root@shasta1> cftool -e
Localdev Srcdev Address Cluster Node Number Joinstate
3 2 00.03.47.c2.aa.f9 SHASTAS shasta1 3 6
3 2 00.03.47.c2.a8.82 SHASTAS shasta2 2 6
3 3 00.03.47.d1.af.ec SHASTAS shasta3 1 6
```

```
root@shasta1> cftool -e
Localdev Srcdev Address Cluster Node Number Joinstate
3 2 00.03.47.c2.aa.f9 SHASTAS shasta1 3 6
3 2 00.03.47.c2.a8.82 SHASTAS shasta2 2 6
3 3 00.03.47.d1.af.ec SHASTAS shasta3 1 6
3 3 00.03.47.d1.ae.f9 SHASTAS shasta4 1 6
```

```
root@shasta1> cftool -e
Localdev Srcdev Address Cluster Node Number Joinstate
3 2 00.03.47.c2.aa.f9 SHASTAS shasta1 3 6
```

3	2	00.03.47.c2.a8.82	SHASTAS	shasta2	2	6
3	3	00.03.47.d1.af.ec	SHASTAS	shasta3	1	6

```

root@shastal> cftool -e
Localdev Srcdev Address Cluster Node      Number Joystate
3 2 00.03.47.c2.aa.f9 SHASTAS shastal 3 6
3 2 00.03.47.c2.a8.82 SHASTAS shasta2 2 6
3 3 00.03.47.d1.af.ec SHASTAS shasta3 1 6
3 3 00.03.47.d1.ae.f9 SHASTAS shasta4 1 6

```

```

root@shastal> cftool -e
Localdev Srcdev Address Cluster Node      Number Joystate
3 2 00.03.47.c2.aa.f9 SHASTAS shastal 3 6
3 2 00.03.47.c2.a8.82 SHASTAS shasta2 2 6
3 3 00.03.47.d1.af.ec SHASTAS shasta3 1 6
3 3 00.03.47.d1.ae.f9 SHASTAS shasta4 1 6

```

```

root@shastal> cftool -e
Localdev Srcdev Address Cluster Node      Number Joystate
3 2 00.03.47.c2.aa.f9 SHASTAS shastal 3 6
3 2 00.03.47.c2.a8.82 SHASTAS shasta2 2 6
3 3 00.03.47.d1.af.ec SHASTAS shasta3 1 6
3 3 00.03.47.d1.ae.f9 SHASTAS shasta4 1 6

```

Notice that the node shasta4 does not show up in each of the echo requests. This indicates that the connection to the node shasta4 is having errors. Because only this node is exhibiting the symptoms, we focus on that node. First, we need to examine the node to see if the Ethernet commands on that node show any errors. We log on to shasta4 and use the netstat(8) or ip(8) command to find out the network interface information and errors.

The netstat(8) or ip(8) command in Linux reports information about the network interfaces.

Further resolution of the problem consists of trying each of the following steps:

- Ensure that the Ethernet cable is securely inserted at each end.
- Try repeated *cftool -e* and look at the netstat -I or ip -s link. If the results of the *cftool(1M)* are always the same and the input errors are gone or greatly reduced, the problem is solved.
- Replace the Ethernet cable.
- Try a different port in the Ethernet hub or switch or replace the hub or switch, or temporarily use a cross-connect cable.
- Replace the Ethernet adapter in the node.

If none of these steps resolves the problem, then field engineers will have to further diagnose the problem.

Problem

The following console message appears on node shasta2 while node shastal is trying to join the cluster with node shasta2:

```

Aug 30 21:31:35 shasta2 kernel: CF: Local node is missing a route from node: shastal.
Aug 30 21:31:35 shasta2 kernel: CF: missing route on local device: eth1.
Aug 30 21:31:35 shasta2 kernel: CF: Node shastal Joined Cluster SHASTAS. (#0000 3)

```

Diagnosis

Look in /var/log/messages on node shastal.

Same message as on console.

No console messages on node shasta2.

Look in /var/log/messages on node shasta2.

```

shasta2:cftool -d
Number Device Type Speed Mtu State Configured Address
1 eth0 4 100 1432 UP YES 00.03.47.c2.a8.82

```


2	eth1	4	100	1432	UP	YES	00.02.b3.88.09.f1
3	eth2	4	100	1432	UP	NO	00.02.b3.88.09.ea

```
shastal: cftool -d
Number Device Type Speed Mtu State Configured Address
1 eth0 4 100 1432 UP YES 00.03.47.c2.a8.3c
2 eth1 4 100 1432 UP NO 00.02.b3.88.b8.89
3 eth2 4 100 1432 UP NO 00.02.b3.88.b7.46
```

Problem

eth1 is not configured are on node shastal:

Diagnosis

Look in /var/log/messages on node shasta2.

```
Aug 27 16:05:59 shasta2 kernel: e100: eth1 NIC Link is Down
Aug 27 16:06:08 shasta2 kernel: CF: Icf Error: (service err_type route_src route_dst). (#0000 0 2 1 1)
Aug 27 16:06:08 shasta2 kernel: CF: (TRACE): CFSF failure detected: no SFopen: passed to ENS:
shastal. (#0000 1)
Aug 27 16:06:08 shasta2 kernel: CF: Node shastal Left Cluster SHASTAS. (#00001)
```

Problem

The eth1 device or interconnect temporarily failed. It could be the NIC on either of the cluster nodes or a cable or the following hub problem.

Node in LEFTCLUSTER state

Node shastal panicked and has rebooted. The following console message appears on node shastal:

```
Aug 28 10:38:25 shastal kernel: CF: shastal: busy: local node not DOWN: retrying
```

Diagnosis

Look in /var/log/messages on node shasta1.

```
Aug 28 10:38: shastal kernel: CF: (TRACE): JoinServer: Startup.
Aug 28 10:38:25 shastal kernel: CF: Giving UP Mastering (Cluster already Running).
Aug 28 10:38:25 shastal kernel: CF: shasta2: busy: local node not DOWN: retrying
```

Last message repeats.

No new messages on console or in /var/log/messages on shasta2.

```
shasta2: cftool -n
Node Number State Os Cpu
shastal 1 LEFTCLUSTER Linux Pentium
shasta2 2 UP Linux Pentium
```

Problem

Node shastal has left the cluster and has not been declared DOWN.

Fix

```
cftool -k
```

This option will declare a node down. Declaring an operational node down can result in catastrophic consequences, including loss of data in the worst case. If you do not wish to declare a node down, quit this program now.

```
Enter node number: 1
Enter name for node #1: shastal
```

```
cftool(down): declaring node #1 (shasta1) down  
cftool(down): node shasta1 is down
```

The following console messages then appear on node shasta2:

```
Aug 28 10:47:39 shasta4 kernel: CF: SHASTAS: shasta1 is Down. (#0000 2)  
Aug 28 10:49:09 shasta4 kernel: CF: Node shasta1 Joined Cluster SHASTAS. (#0000 2)
```

The following console message appears on node shasta1:

```
Aug 28 10:49:00 shasta1 kernel: CF: Local Node shasta1 Created Cluster SHASTAS. (#0000 1)
```

8.3 Collecting Troubleshooting Information

If an error occurs in the PRIMECLUSTER system, collect information required for an investigation from all the nodes that construct the cluster and the cluster management servers. For details on how to collect information, see "Appendix C Troubleshooting" in "PRIMECLUSTER Installation and Administration Guide." Then, contact field engineers.

Chapter 9 CF over IP

This chapter describes CF over IP and how it is configured.

This chapter discusses the following:

- The section "9.1 Overview" introduces CF over IP and describes its use.
- The section "9.2 Configuring CF over IP" details how to configure CF over IP.

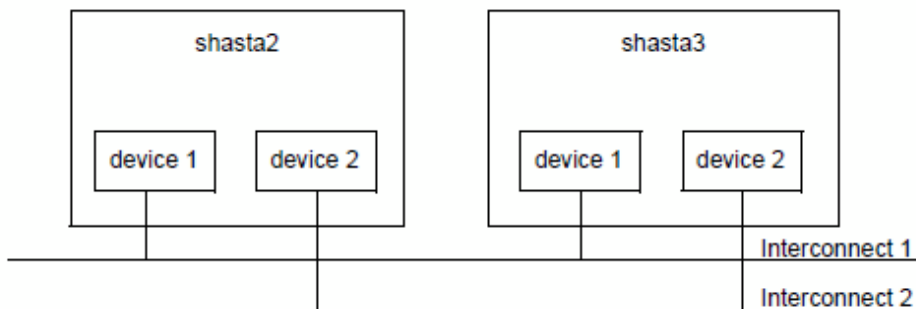
9.1 Overview

Note

- All IP configurations must be done prior to using CF over IP. The devices must be initialized with a unique IP address and a broadcast mask. IP must be configured to use these devices. If the configuration is not done, cfconfig(1M) will fail to load CF, and CF will not start.
- The devices used for CF over IP must not be controlled by an RMS userApplication that could unconfigure a device due to Offline processing.

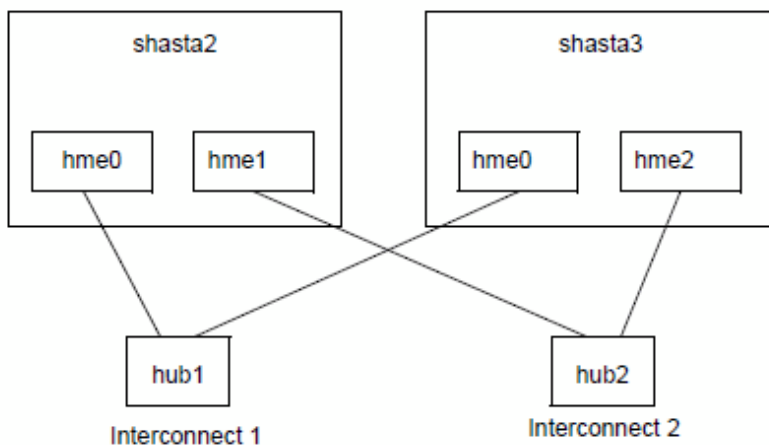
CF communications are based on the use of interconnects. A properly configured interconnect will have connections to all of the nodes in the cluster through some type of device. This is illustrated in the figure below.

Figure 9.1 Conceptual view of CF interconnects



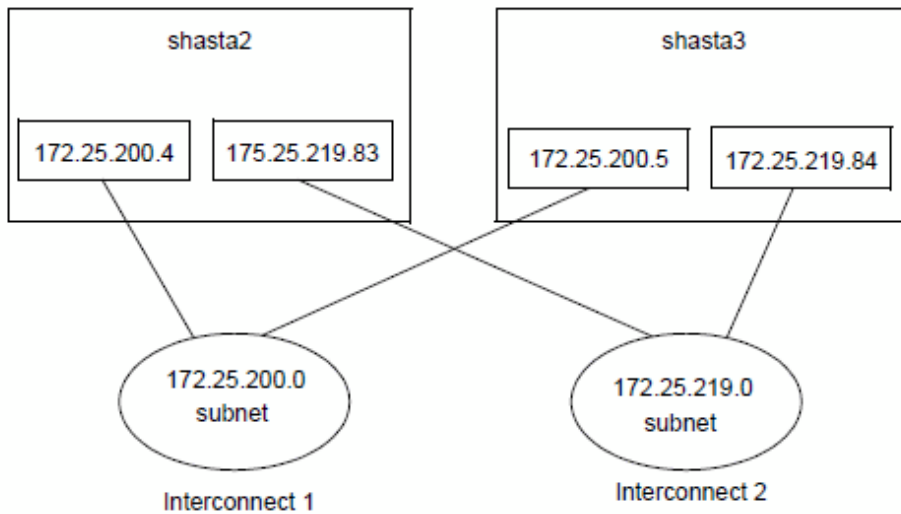
When CF is used over Ethernet, Ethernet devices are used as the interfaces to the interconnects. The interconnects themselves are typically Ethernet hubs or switches. An example of this is shown in the figure below.

Figure 9.2 CF with Ethernet interconnects



When CF is run over IP, IP interfaces are the devices used to connect to the interconnect. The interconnect is an IP subnetwork. Multiple IP subnetworks may be used for the sake of redundancy. The figure below shows a CF over IP configuration.

Figure 9.3 CF with IP interconnects



It is also possible to use mixed configurations in which CF is run over both Ethernet devices and IP subnetworks.

When using CF over IP, you should make sure that each node in the cluster has an IP interface on each subnet used as an interconnect.

Note

- IPv4 address is used for CF over IP.
- Use CF over IP only if one of the following applies:
 - For a cloud environment
 - For an RHOSP environment
 - When configuring an interconnect between cluster nodes with different network segments
 - When a device that only supports IP exists in the route tracking of the interconnect

Note

CF is not allowed to reach nodes that are on different subnets. Caution When selecting a subnetwork to use for CF, you should use a private subnetwork that only cluster nodes can access. CF security is based on access to its interconnects. Any node that can access an interconnect can join the cluster and acquire root privileges on any cluster node.

When CF over IP is used, this means that any node on the subnetworks used by CF must be trusted. You should not use the public interface to a cluster node for CF over IP traffic unless you trust every node on your public network.

9.2 Configuring CF over IP

To configure CF over IP, you should do the following:

- Designate which subnetworks you want to use for CF over IP. Up to four subnetworks can be used.
- Make sure that each node that is to be in the cluster has IP interfaces properly configured for each subnetwork. Make sure the IP broadcast and netmasks are correct and consistent on all nodes for the subnetworks.
- Make sure that all of these IP interfaces are up and running.
- Run the CF Wizard in Cluster Admin.

The CF Wizard has a window which allows CF over IP to be configured. The Wizard will probe all the nodes that will be in the cluster, find out what IP interfaces are available on each, and then offer them as choices in the CF over IP window. It will also try to group the choices for each node by subnetworks. See the section "[1.1 CF, CIP, and CIM configuration](#)" for details.

CF uses special IP devices to keep track of CF over IP configuration. There are four of these devices named as follows:

```
/dev/ip0  
/dev/ip1  
/dev/ip2  
/dev/ip3
```

These devices do not actually correspond to any device files in /dev. They are just place holders for CF over IP configuration information within the CF product. Any of these devices can have an IP address and broadcast address assigned by the cfconfig(1M) command (or by Cluster Admin which invokes the cfconfig(1M) command in the Wizard).

If you run cfconfig(1M) by hand, you may specify any of these devices to indicate you want to run CF over IP. The IP device should be followed by an IP address and broadcast address of an interface on the local node. The addresses must be in Internet dotted-decimal notation. For example, to configure CF on shasta2 in "[Figure 9.3 CF with IP interconnects](#)", the cfconfig(1M) command would be as follows:

```
# cfconfig -S A clustername \  
/dev/ip0 172.25.200.4 172.25.200.255 \  
/dev/ip1 172.25.219.83 172.25.219.255
```

For cluster nodes placed between different network segments, configure the following:

```
# cfconfig -S A clustername \  
/dev/ip0 192.168.223.105 192.168.223.255 192.168.123.112 \  
/dev/ip1 192.168.200.105 192.168.200.255 192.168.100.112
```

It does not matter which IP device you use. The above command could equally have used /dev/ip2 and /dev/ip3.

Note

The cfconfig(1M) command does not do any checks to make sure that the IP addresses are valid.

The IP devices chosen in the configuration will appear in other commands such as cftool -d and cftool -r.

IP interfaces will not show up in CF pings using cftool -p unless they are configured for use with CF and the CF driver is loaded.

Note

cftool -d shows a relative speed number for each device, which is used to establish priority for the message send. If the configured device is IP, the relative speed 100 is used. This is the desired priority for the logical IP device. If a Gigabit Ethernet hardware device is also configured, it will have priority.

Glossary

Items in this glossary that apply to specific Cluster Foundation components are indicated with the following notation:

- (CF)-Cluster Foundation
- (RMS)-Reliant Monitor Services
- (RCVM)-Volume Manager (not available in all markets)
- (SIS)-Scalable Internet Services

Some of these products may not be installed on your cluster. See field engineers for more information.

AC

See *Access Client*.

Access Client

GFS kernel module on each node that communicates with the Meta Data Server and provides simultaneous access to a shared file system.

administrative LAN

In PRIMECLUSTER configurations, an administrative LAN is a private local area network (LAN) on which machines such as the System Console and Cluster operation management PC reside. Because normal users do not have access to the administrative LAN, it provides an extra level of security. The use of an administrative LAN is optional.

See also *public LAN*.

API

See *Application Program Interface*.

application (RMS)

In the RMS context, an application object is a special resource used to group other resources into a logical collection. Typically, it is used to represent a real-world application or application suite in a high-availability configuration.

Application Program Interface

A shared boundary between a service provider and the application that uses that service.

application template (RMS)

A predefined group of object definition value choices used by the Wizard Tools or the RMS Wizard Kit to create object definitions for a specific type of application.

attribute (RMS)

The part of an object definition that specifies how the base monitor acts and reacts for a particular object type during normal operations.

automatic switchover (RMS)

The procedure by which RMS automatically switches control of a userApplication over to another node after specified conditions are detected.

See also *directed switchover (RMS)*, *failover (RMS, SIS)*, *switchover (RMS)*, *symmetrical switchover (RMS)*.

availability

Availability describes the need of most enterprises to operate applications via the Internet 24 hours a day, 7 days a week. The relationship of the actual to the planned usage time determines the availability of a system.

base cluster foundation (CF)

This Cluster Foundation module resides on top of the basic OS and provides internal interfaces for the CF (Cluster Foundation) functions that the Cluster Foundation services use in the layer above.

See also *Cluster Foundation (CF)*.

base monitor (RMS)

The RMS module that maintains the availability of resources. The base monitor is supported by daemons and detectors. Each node being monitored has its own copy of the base monitor.

Cache Fusion

The improved interprocess communication interface in Oracle 9i that allows logical disk blocks (buffers) to be cached in the local memory of each node. Thus, instead of having to flush a block to disk when an update is required, the block can be copied to another node by passing a message on the interconnect, thereby removing the physical I/O overhead.

CF (Cluster Foundation or Cluster Framework)

See *Cluster Foundation (CF)*.

CF node name (CF)

The CF cluster node name, which is configured when a CF cluster is created.

child (RMS)

A resource defined in the configuration file that has at least one parent. A child can have multiple parents, and can either have children itself (making it also a parent) or no children (making it a leaf object).

See also *resource (RMS)*, *object (RMS)*, *parent (RMS)*, *leaf object (RMS)*.

cluster

A set of computers that work together as a single computing source. Specifically, a cluster performs a distributed form of parallel computing.

See also *RMS configuration (RMS)*.

Cluster Admin

A Java-based, OS-independent management tool for Cluster Foundation products such as CF, SIS, and RMS. Cluster Admin is available from the Web-Based Admin View interface.

See also *Cluster Foundation (CF)*, *Scalable Internet Services (SIS)*, *Reliant Monitor Services (RMS)*, *Web-Based Admin View*.

Cluster Foundation (CF)

The set of Cluster Foundation modules that provides basic clustering communication services.

See also *base cluster foundation (CF)*.

cluster interconnect (CF)

The set of private network connections used exclusively for Cluster Foundation communications.

Cluster Interconnect Protocol

CIP is an interface such as eth0 except the physical layer is built on top of the cluster interconnect.

Cluster Join Services (CF)

This Cluster Foundation module handles the forming of a new cluster and the addition of nodes.

concatenated virtual disk (RCVM)

Concatenated virtual disks consist of two or more pieces on one or more disk drives. They correspond to the sum of their parts. Unlike simple virtual disks where the disk is subdivided into small pieces, the individual disks or partitions are combined to form a single large logical disk.

See also *mirror virtual disk (RCVM)*, *simple virtual disk*, *striped virtual disk*, *virtual disk*.

configuration file (RMS)

In the RMS context, the single file that defines the monitored resources and establishes the interdependencies between them. The default name of this file is config.us.

custom detector (RMS)

See *detector (RMS)*.

custom type (RMS)

See *generic type (RMS)*.

daemon

A continuous process that performs a specific function repeatedly.

database node (SIS)

Nodes that maintain the configuration, dynamic data, and statistics in a SIS configuration.

See also *gateway node (SIS)*, *service node (SIS)*, *Scalable Internet Services (SIS)*.

detector (RMS)

A process that monitors the state of a specific object type and reports a change in the resource state to the base monitor.

DHCP

Dynamic Host Control Protocol. A standard method of delivering information to a host at boot time. This is most often used to dynamically assign the host's IP address and netmask, but many other parameters are possible, including domain names, DNS servers, and time servers.

directed switchover (RMS)

The RMS procedure by which an administrator switches control of a userApplication over to another node.

See also *automatic switchover (RMS)*, *failover (RMS, SIS)*, *switchover (RMS)*, *symmetrical switchover (RMS)*.

DOWN (CF)

A node state that indicates that the node is unavailable (marked as down). A LEFTCLUSTER node must be marked as DOWN before it can rejoin a cluster.

See also *UP (CF)*, *LEFTCLUSTER (CF)*, *node state (CF)*.

Enhanced Lock Manager (ELM) (CF)

A light weight, high performance, highly responsive lock manger, specifically designed for providing a high reliability heartbeat messaging mechanism for Cluster Foundation modules.

ENS (CF)

See *Event Notification Services (CF)*.

environment variables

Variables or parameters that are defined globally.

error detection (RMS)

The process of detecting an error. For RMS, this includes initiating a log entry, sending a message to a log file, or making an appropriate recovery response.

Ethernet

LAN standard that is standardized by IEEE 802.3. Currently, except for special uses, nearly all LANs are Ethernets. Originally the expression Ethernet was a LAN standard name for a 10 megabyte per second type LAN, but now it also used as a general term that includes high-speed Ethernets and gigabyte Ethernets.

Event Notification Services (CF)

This Cluster Foundation module provides an atomic-broadcast facility for events.

failover (RMS, SIS)

With SIS, this process switches a failed node to a backup node. With RMS, this process is known as switchover.

See also *automatic switchover (RMS)*, *directed switchover (RMS)*, *switchover (RMS)*, *symmetrical switchover (RMS)*.

gateway node (SIS)

Gateway nodes have an external network interface. All incoming packets are received by this node and forwarded to the selected service node, depending on the scheduling algorithm for the service.

See also *service node (SIS)*, *database node (SIS)*, *Scalable Internet Services (SIS)*.

generic type (RMS)

An object type which has generic properties. A generic type is used to customize RMS for monitoring resources that cannot be assigned to one of the supplied object types.

See also *object type (RMS)*.

GFS shared file system

A shared file system that allows simultaneous access from multiple Linux(R) systems that are connected to shared disk units, while maintaining data consistency, and allows processing performed by a node to be continued by other nodes even if the first node fails.

A GFS shared file system can be mounted and used concurrently from multiple nodes.

Global Disk Services

This optional product provides volume management that improves the availability and manageability of information stored on the disk unit of the Storage Area Network (SAN).

Global File Services

This optional product provides direct, simultaneous accessing of the file system on the shared storage unit from two or more nodes within a cluster.

Global Link Services

This optional products provides network high availability solutions by multiplying a network route.

graph (RMS)

See *system graph (RMS)*.

graphical user interface

A computer interface with windows, icons, toolbars, and pull-down menus that is designed to be simpler to use than the command-line interface.

GUI

See *graphical user interface*.

HBA blockage function

Function that stops the HBA in the switchover source when node switching takes place. This function enables PRIMECLUSTER node switching to be executed at high speed. This function shortens the node switchover time by forcibly stopping all ongoing input/output processes in the HBA of the switchover source.

high availability

A system design philosophy in which redundant resources are employed to avoid single points of failure.

interconnect (CF)

See *cluster interconnect (CF)*.

Internet Protocol address

A numeric address that can be assigned to computers or applications.

See also *IP aliasing*.

Internode Communications facility

This module is the network transport layer for all Cluster Foundation internode communications. It interfaces by means of OS-dependent code to the network I/O subsystem and guarantees delivery of messages queued for transmission to the destination node in the same sequential order unless the destination node fails.

IP address

See *Internet Protocol address*.

IP aliasing

This enables several IP addresses (aliases) to be allocated to one physical network interface. With IP aliasing, the user can continue communicating with the same IP address, even though the application is now running on another node.

See also *Internet Protocol address*.

iRMC (integrated Remote Management Controller)

Abbreviation for integrated Remote Management Controller which is one of the hardware mounted in PRIMEQUEST/PRIMERGY.

JOIN (cluster join services module) (CF)

See *Cluster Join Services (CF)*.

keyword

A word that has special meaning in a programming language. For example, in an RMS configuration file, the keyword object identifies the kind of definition that follows.

leaf object (RMS)

A bottom object in a system graph. In the configuration file, this object definition is at the beginning of the file. A leaf object does not have children.

LEFTCLUSTER (CF)

A node state that indicates that the node cannot communicate with other nodes in the cluster. That is, the node has left the cluster. The reason for the intermediate LEFTCLUSTER state is to avoid the network partition problem.

See also *UP (CF)*, *DOWN (CF)*, *network partition (CF)*, *node state (CF)*.

link (RMS)

Designates a child or parent relationship between specific resources.

local area network

See *public LAN*.

local node

The node from which a command or process is initiated.

See also *remote node*.

log file

The file that contains a record of significant system events or messages. The RMS Wizard Tools, the RMS base monitor, and RMS detectors each maintain their own log files.

Management Board

One of the hardware units installed in PRIMEQUEST.

Management Information Base

A hierarchical database of information about the local network device. The database is maintained by network management software such as an SNMP agent.

See also *Simple Network Management Protocol*.

MDS

See *Meta Data Server*.

message

A set of data transmitted from one software process to another process, device, or file.

message queue

A designated memory area which acts as a holding place for messages so they can be processed in the same order they were received.

Meta Data Server

GFS daemon that centrally manages the control information, or meta-data, of a file system.

MIB

See *Management Information Base*.

MIPC

Mesh Interprocessor Communication

mirror virtual disk (RCVM)

Mirror virtual disks consist of two or more physical devices, and all output operations are performed simultaneously on all of the devices.

See also *concatenated virtual disk (RCVM)*, *simple virtual disk*, *striped virtual disk*.

mirrored disks (RCFS)

A set of disks that contain the same data. If one disk fails, the remaining disks of the set are still available, preventing an interruption in data availability.

See also *mirrored pieces (RCFS)*.

mirrored pieces (RCFS)

Physical pieces that together comprise a mirrored virtual disk. These pieces include mirrored disks and data disks.

See also *mirrored disks (RCFS)*.

MMB

Abbreviation for Management Board, which is one of the hardware units installed in PRIMEQUEST.

mount point

The point in the directory tree where a file system is attached.

multihosting

Multiple controllers simultaneously accessing a set of disk drives.

native operating system

The part of an operating system that is always active and translates system calls into activities.

network partition (CF)

This condition exists when two or more nodes in a cluster cannot communicate over the interconnect; however, with applications still running, the nodes can continue to read and write to a shared device, compromising data integrity.

node

A host that is a member of a cluster.

node state (CF)

Every node in a cluster maintains a local state for every other node in that cluster. The node state of every node in the cluster must be either UP, DOWN, or LEFTCLUSTER.

See also *UP (CF)*, *DOWN (CF)*, *LEFTCLUSTER (CF)*.

object (RMS)

A representation of a physical or virtual resource in the RMS configuration file or in a system graph.

See also *leaf object (RMS)*, *object definition (RMS)*, *object type (RMS)*.

object definition (RMS)

An entry in the configuration file that identifies a resource to be monitored by RMS. Attributes included in the definition specify properties of the corresponding resource.

See also *attribute (RMS)*, *object type (RMS)*.

object type (RMS)

A category of similar resources monitored as a group, such as disk drives. Each object type has specific properties, or attributes, which limit or define what monitoring or action can occur. When a resource is associated with a particular object type, attributes associated with that object type are applied to the resource.

See also *generic type (RMS)*.

online maintenance

The capability of adding, removing, replacing, or recovering devices without shutting or powering off the node.

operating system dependent (CF)

This module provides an interface between the native operating system and the abstract, OS-independent interface that all Cluster Foundation modules depend upon.

Oracle Real Application Clusters (RAC)

Oracle RAC allows access to all data in a database to users and applications in a clustered or MPP (massively parallel processing) platform. Formerly known as Oracle Parallel Server (OPS).

OSD (CF)

See *operating system dependent (CF)*.

parent (RMS)

An object in the RMS configuration file or system graph that has at least one child.

See also *child (RMS)*, *configuration file (RMS)*, *leaf object (RMS)*, *system graph (RMS)*.

physical IP address

IP address that is assigned directly to the interface (for example, hme0) of a network interface card.

primary node (RMS)

The default node on which a user application comes online when RMS is started. This is always the node name of the first child listed in the userApplication object definition.

PRIMECLUSTER services (CF)

Service modules that provide services and internal interfaces for clustered applications.

private network addresses

Private network addresses are a reserved range of IP addresses specified by the Internet Corporation for Assigned Names and Numbers (ICANN). Modern switches and routers prevent these addresses from being routed to the Internet, allowing two or more organizations to assign the same private addresses for internal use without causing conflicts or security risks.

private resource (RMS)

A resource accessible only by a single node and not accessible to other RMS nodes.

See also *resource (RMS)*, *shared resource*.

public LAN

The local area network (LAN) by which normal users access a machine.

See also *administrative LAN*.

queue

See *message queue*.

redundancy

The capability of one component to assume the resource load of another physically similar component in case the original component fails or is shut down. Common examples include RAID hardware and/or RAID software to replicate data stored on secondary storage devices, multiple network connections to provide alternate data paths, and multiple nodes that can be dynamically reprovisioned to maintain critical services in a cluster.

Reliant Monitor Services (RMS)

The package that maintains high availability of user-specified resources by providing monitoring and switchover capabilities.

remote node

A node that is accessed through a LAN or telecommunications line.

See also *local node*, *node*.

reporting message (RMS)

A message that a detector uses to report the state of a particular resource to the base monitor.

resource (RMS)

A hardware or software element (private or shared) that provides a function, such as a mirrored disk, mirrored disk pieces, or a database server. A local resource is monitored only by the local node.

See also *private resource (RMS)*, *shared resource*.

resource definition (RMS)

See *object definition (RMS)*.

resource label (RMS)

The name of the resource as displayed in a system graph.

resource state (RMS)

Current state of a resource.

RMS

See *Reliant Monitor Services (RMS)*.

RMS commands (RMS)

Commands that enable RMS resources to be administered from the command line.

RMS configuration (RMS)

A configuration made up of two or more nodes connected to shared resources. Each node has its own copy of operating system and RMS software, as well as its own applications.

RMS Wizard Kit

RMS configuration products that have been designed for specific applications. Each component of the Wizard Kit includes customized default settings, subapplications, detectors, and scripts.

See also *RMS Wizard Tools*, *Reliant Monitor Services (RMS)*.

RMS Wizard Tools

A software package composed of various configuration and administration tools used to create and manage applications in an RMS configuration.

See also *RMS Wizard Kit*, *Reliant Monitor Services (RMS)*.

route

In the PRIMECLUSTER Concepts Guide, this term refers to the individual network paths of the redundant cluster interfaces that connect the nodes to each other.

SAN

See *Storage Area Network*.

scalability

The ability of a computing system to efficiently handle any dynamic change in work load. Scalability is especially important for Internet-based applications where growth caused by Internet usage presents a scalable challenge.

Scalable Internet Services (SIS)

The package that dynamically balances network traffic loads across cluster nodes while maintaining normal client/server sessions for each connection.

script (RMS)

A shell program executed by the base monitor in response to a state transition in a resource. The script may cause the state of a resource to change.

service node (SIS)

Service nodes provide one or more TCP services (such as FTP, Telnet, and HTTP) and receive client requests forwarded by the gateway nodes.

See also *database node (SIS)*, *gateway node (SIS)*, *Scalable Internet Services (SIS)*.

SF

See *Shutdown Facility*.

shared resource

A resource, such as a disk drive, that is accessible to more than one node.

See also *private resource (RMS)*, *resource (RMS)*.

Shutdown Facility

The Cluster Foundation interface that manages the shutdown and startup of cluster nodes. The SF is automatically invoked during failover operations. It also notifies other Cluster Foundation products of the successful completion of node shutdown so that recovery operations can begin.

Simple Network Management Protocol

A set of protocols that facilitates the exchange of information between managed network devices. The protocols are implemented by software agents residing in the devices. Each agent can read and write data in the local Management Information Base (MIB) in response to SNMP requests from other devices on the network.

See also *Management Information Base*.

simple virtual disk

Simple virtual disks define either an area within a physical disk partition or an entire partition.

See also *concatenated virtual disk (RCVM)*, *striped virtual disk*, *virtual disk*.

SIS

See *Scalable Internet Services (SIS)*.

SNMP

See *Simple Network Management Protocol*.

state

See *resource state (RMS)*.

Storage Area Network

The high-speed network that connects multiple, external storage units and storage units with multiple computers. The connections are generally fiber channels.

striped virtual disk

Striped virtual disks consist of two or more pieces. These can be physical partitions or further virtual disks (typically a mirror disk). Sequential I/O operations on the virtual disk can be converted to I/O operations on two or more physical disks. This corresponds to RAID Level 0 (RAID0).

See also *concatenated virtual disk (RCVM)*, *mirror virtual disk (RCVM)*, *simple virtual disk*, *virtual disk*.

switchover (RMS)

The process by which RMS switches control of a userApplication over from one monitored node to another.

See also *automatic switchover (RMS)*, *directed switchover (RMS)*, *failover (RMS, SIS)*, *symmetrical switchover (RMS)*.

symmetrical switchover (RMS)

This means that every RMS node is able to take on resources from any other RMS node.

See also *automatic switchover (RMS)*, *directed switchover (RMS)*, *failover (RMS, SIS)*, *switchover (RMS)*.

system disk (GDS)

Disk on which the active operating system is installed. System disk refers to the entire disk that contains the slices that are currently operating as one of the following file systems (or the swap area):

/, /usr, /var, /boot, /boot/efi, or swap area

system graph (RMS)

A visual representation (a map) of monitored resources used to develop or interpret the RMS configuration file.

See also *configuration file (RMS)*.

template

See *application template (RMS)*.

type

See *object type (RMS)*.

UP (CF)

A node state that indicates that the node can communicate with other nodes in the cluster.

See also *DOWN (CF)*, *LEFTCLUSTER (CF)*, *node state (CF)*.

virtual disk

A pseudo-device that allows a portion or a combination of physical disks to be treated as a single logical disk. The virtual disk driver is inserted between the highest level of the OS logical input/output (I/O) system and the physical device driver(s), allowing all logical I/O requests to be mapped to the appropriate area on the physical disk(s).

See also *concatenated virtual disk (RCVM)*, *mirror virtual disk (RCVM)*, *simple virtual disk*, *striped virtual disk*.

Web-Based Admin View

A Java-based, OS-independent interface to Cluster Foundation management components.

wizard (RMS)

An interactive software tool that creates a specific type of application using pretested object definitions.

Wizard Kit

See *RMS Wizard Kit*.

Wizard Tools (RMS)

See *RMS Wizard Tools*.

Index

	[Numbers]	
Configuring CF over IP.....	100	
	[A]	
Adding and removing a node from CIM.....	60	
Add to CIM.....	61	
AWS CLI.....	75	
Azure CLI.....	75	
	[C]	
Caused by a cluster partition.....	67	
Caused by a panic/hung node.....	66	
Caused by staying in the kernel debugger too long.....	66	
CF Heartbeat monitor.....	59	
CF node information.....	47	
CF node name.....	1	
CF over IP.....	99	
CF route table.....	46	
CF route tracking.....	44	
CF topology table.....	48	
Checking SA status.....	89	
CIM options.....	61	
CIM Override.....	62	
CIP name.....	38	
clgettree command execution result.....	38	
clgettree command to verify configuration.....	39	
clinitreset.....	39	
clsetup.....	38,39	
cluster name.....	1	
Cluster resource management.....	38	
	[D]	
Default StartingWaitTime.....	39	
Description of the LEFTCLUSTER state.....	64	
Displaying statistics.....	57	
Displaying the topology table.....	47	
	[F]	
FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API	75	
	[G]	
GUI administration.....	41	
	[I]	
init command.....	64	
interconnect.....	1	
	[L]	
LEFTCLUSTER state.....	64	
	[M]	
Main CF table.....	43	
Marking nodes DOWN.....	55	
	[N]	
NIFCLOUD API.....	76	
Node details.....	46	
Node joins a running cluster.....	39	
	[O]	
OpenStack API.....	75	
Overview.....	99	
	[R]	
rcqconfig.....	36	
rcsd.cfg configuration file.....	79	
rcsd.cfg file.....	78	
rcsd.log.....	91	
rcsd log file.....	91	
reboot -f command.....	64	
Recovering from LEFTCLUSTER.....	65	
Resource Database configuration.....	38	
Response Time monitor.....	48	
RFC 1918.....	2	
RFC 4193.....	2	
	[S]	
SA specific log files.....	91	
sdttool command.....	90	
Search based on keyword.....	56	
Search based on severity levels.....	57	
Search based on time filter.....	56	
Shutdown Agent.....	65	
shutdown command.....	64	
SMAWsf directory.....	79	
Starting and stopping CF.....	50	
Starting CF.....	52	
Starting Cluster Admin GUI and logging in.....	41	
Startup synchronization.....	39	
Stopping CF.....	53	
Synchronization phase.....	39	
	[U]	
Unconfigure CF.....	61	
Using PRIMECLUSTER log viewer.....	55	
	[V]	
VMware vCenter Server Functional Cooperation.....	75	