QUANTUM

19 March 2026

# CLUSTERXL

## R81.10

Administration Guide

# Check Point Copyright Notice

© 2021 - 2026 Check Point Software Technologies Ltd.

All rights reserved. This product and related documentation are protected by copyright and distributed under licensing restricting their use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form or by any means without prior written authorization of Check Point. While every precaution has been taken in the preparation of this book, Check Point assumes no responsibility for errors or omissions. This publication and features described herein are subject to change without notice.

RESTRICTED RIGHTS LEGEND:

Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

TRADEMARKS:

Refer to the Copyright page for a list of our trademarks.

Refer to the Third Party copyright notices for a list of relevant copyrights and third-party licenses.

# Important Information

### Latest Software

We recommend that you install the most recent software release to stay up-to-date with the latest functional improvements, stability fixes, security enhancements and protection against new and evolving attacks.

### Certifications

For third party independent certification of Check Point products, see the Check Point Certifications page.

### Check Point R81.10

For more about this release, see the R81.10 home page.

### Latest Version of this Document in English

Open the latest version of this document in a Web browser.
Download the latest version of this document in PDF format.

### Feedback

Check Point is engaged in a continuous effort to improve its documentation.
Please help us by sending your comments.

## Revision History

| Date | Description |
|---|---|
| 13 January 2026 | Updated:<br>■ *"Viewing Critical Devices" on page 246* |
| 22 December 2025 | Updated:<br>■ *"ClusterXL Requirements and Compatibility" on page 17* |
| 25 June 2025 | Updated:<br>■ *"Viewing Cluster Correction Statistics" on page 285* |
| 23 June 2025 | Added:<br>■ *"ClusterXL Failover based on the Load on ClusterXL SND Instances" on page 204* |
| 11 June 2025 | Updated:<br>■ *"ClusterXL Configuration Commands" on page 304* |
| 24 January 2025 | Added:<br>■ *"Configuring Duration of Forced Failover in High Availability Mode" on page 233*<br>Updated:<br>■ *"Configuring ISP Redundancy on a Cluster" on page 193*<br>■ *"Configuring Policy Update Timeout" on page 163*<br>■ *"Dynamic Routing Protocols in a Cluster Deployment" on page 202*<br>■ *"ClusterXL Configuration Commands" on page 304* |
| 09 January 2025 | Updated:<br>■ *"Using the Classic Mode in SmartConsole" on page 87* |
| 05 April 2024 | Updated:<br>■ *"How State Synchronization Works" on page 62*<br>■ *"Example of Cluster IP Addresses on Different Subnets" on page 167* |

| Date | Description |
|---|---|
| 26 October 2023 | Updated:<br>■ *"IPv6 Support for ClusterXL" on page 24*<br>■ *"Configuring Virtual MAC (VMAC)" on page 103*<br>■ *"ClusterXL Monitoring Commands" on page 308* - removed the "`cphaprob latency`" command as not supported<br>■ *"Viewing Critical Devices" on page 246* - description of "Local Probing"<br>■ *"Configuring Services to Synchronize After a Delay" on page 63* |
| 16 March 2023 | Updated:<br>■ *"Active-Active Mode in ClusterXL" on page 47* |
| 20 February 2023 | Updated:<br>■ *"Dynamic Routing Failover" on page 59* |
| 29 January 2023 | Updated:<br>■ *"Configuring Assigned Load in the Load Sharing Unicast Mode" on page 122* |
| 19 January 2023 | Updated:<br>■ *"Active-Active Mode in ClusterXL" on page 47* |
| 26 December 2022 | Added:<br>■ *"Configuring Virtual MAC (VMAC)" on page 103*<br>■ *"Configuring Assigned Load in the Load Sharing Unicast Mode" on page 122* |
| 23 November 2022 | Updated:<br>■ *"Active-Active Mode in ClusterXL" on page 47* |
| 14 June 2022 | In the HTML version, added glossary terms to the text. |
| 10 March 2022 | Updated:<br>■ *"Active-Active Mode in ClusterXL" on page 47*<br>■ *"Using the Wizard Mode in SmartConsole" on page 81* |
| 27 April 2022 | Updated:<br>■ *"Active-Active Mode in ClusterXL" on page 47* |

| Date | Description |
|---|---|
| 10 March 2022 | Updated:<br>■ *"Active-Active Mode in ClusterXL" on page 47* , ,<br>■ *"Using the Wizard Mode in SmartConsole" on page 81* |
| 15 July 2021 | First release of this document |

# Table of Contents

**Introduction to ClusterXL** ... *15*
The Need for Clusters ... *15*
ClusterXL Solution ... *15*
How ClusterXL Works ... *16*
The Cluster Control Protocol ... *16*
**ClusterXL Requirements and Compatibility** ... *17*
Check Point Appliances and Open Servers ... *17*
Supported Number of Cluster Members ... *18*
Hardware Requirements for Cluster Members ... *18*
Software Requirements for Cluster Members ... *18*
VMAC Mode ... *19*
Supported Topologies for Synchronization Network ... *21*
Clock Synchronization in ClusterXL ... *24*
IPv6 Support for ClusterXL ... *24*
Synchronized Cluster Restrictions ... *25*
**High Availability and Load Sharing Modes in ClusterXL** ... *28*
Introduction to High Availability and Load Sharing modes ... *28*
High Availability ... *28*
Load Sharing ... *29*
Example ClusterXL Topology ... *30*
Example Diagram ... *30*
Defining the Cluster Member IP Addresses ... *31*
Defining the Cluster Virtual IP Addresses ... *31*
Defining the Synchronization Network ... *32*
Configuring Cluster Addresses on Different Subnets ... *32*
ClusterXL Mode Considerations ... *33*
Choosing High Availability, Load Sharing, or Active-Active mode ... *33*

Considerations for the Load Sharing Mode ... 33
IP Address Migration ... 34
ClusterXL Modes ... 35
High Availability Mode ... 35
Load Sharing Modes ... 39
Load Sharing Multicast Mode ... 39
Load Sharing Unicast Mode ... 40
ClusterXL Mode Comparison ... 43
Cluster Failover ... 44
What is Failover? ... 44
When Does a Failover Occur? ... 45
What Happens When a Cluster Member Recovers? ... 45
How a Recovered Cluster Member Obtains the Security Policy ... 46
General Failover Limitations ... 46
**Active-Active Mode in ClusterXL ... 47**
Introduction ... 47
Limitations ... 48
Configuring a new ClusterXL in Active-Active Mode ... 49
Changing the ClusterXL Mode to Active-Active ... 55
Dynamic Routing Failover ... 59
**Synchronizing Connections in the Cluster ... 60**
The Synchronization Network ... 61
How State Synchronization Works ... 62
Configuring Services to Synchronize After a Delay ... 63
Configuring Services not to Synchronize ... 65
Sticky Connections ... 67
Introduction to Sticky Connections ... 67
VPN Tunnels with 3rd Party Peers and Load Sharing ... 67
Third-Party Gateways in Hub and Spoke VPN Deployments ... 68
Configuring a Third-Party Gateway in a Hub and Spoke VPN Deployment ... 69

Non-Sticky Connections ... 71
Non-Sticky Connection Example: TCP 3-Way Handshake ... 71
Synchronizing Non-Sticky Connections ... 72
Synchronizing Clusters on a Wide Area Network ... 74
Synchronized Cluster Restrictions ... 77
**Configuring ClusterXL** ... **79**
Installing Cluster Members ... 79
Configuring Routing for Client Computers ... 79
Configuring the Cluster Control Protocol (CCP) Settings ... 80
Configuring the Cluster Object and Members ... 81
Using the Wizard Mode in SmartConsole ... 81
Using the Classic Mode in SmartConsole ... 87
Changing the Settings of Cluster Object in SmartConsole ... 97
Configuring General Properties ... 97
Working with Cluster Topology ... 97
Changing the Synchronization Interface ... 99
Adding Another Member to an Existing Cluster ... 101
Removing a Member from an Existing Cluster ... 101
Configuring a ClusterXL in Bridge Mode ... 102
**Advanced Features and Procedures** ... **103**
Configuring Virtual MAC (VMAC) ... 103
Virtual MAC ... 103
Same VMAC ... 108
Working with VPN in Cluster ... 113
Configuring VPN in Clusters ... 113
Defining VPN Peer Clusters with Separate Management Servers ... 114
Working with NAT in Cluster ... 115
Cluster Fold and Cluster Hide ... 115
Configuring NAT in Cluster ... 115
Configuring NAT on a Cluster Member ... 116

Working with VLANs in Cluster ... 117
Configuring Link Monitoring on the Cluster Interfaces ... 119
Configuring Assigned Load in the Load Sharing Unicast Mode ... 122
Working with Bond Interfaces in Cluster ... 124
Bond Interfaces (Link Aggregation) ... 125
Bond High Availability Mode in Cluster ... 128
Simple Redundant Topology ... 129
Fully Meshed Redundancy ... 130
Bond Failover in High Availability Mode ... 131
Configuring a Bond Interface in High Availability Mode ... 132
Making Sure the Bond Interface is Functioning Properly ... 133
Failover Support for VLANs ... 134
Sync Redundancy ... 135
Configuring Bond High Availability in VRRP Cluster ... 137
Bond Load Sharing Mode in Cluster ... 139
Bond Failover in Load Sharing Mode ... 140
Configuring a Bond Interface in High Availability Mode ... 141
Configuring Minimal Number of Required Subordinate Interfaces ... 142
Making Sure the Bond Interface is Functioning Properly ... 143
Group of Bonds ... 144
Introduction ... 144
Creating a new Group of Bonds ... 146
Adding a Bond interface to the existing Group of Bonds ... 147
Removing a Bond interface from the existing Group of Bonds ... 149
Deleting a Group of Bonds ... 150
Monitoring ... 151
Logs ... 151
Limitations ... 152
Performance Guidelines for Bond Interfaces ... 154
Troubleshooting Issues with Bonded Interfaces ... 156

Troubleshooting Workflow ... 156
Connectivity Delays on Switches ... 156
Advanced Cluster Configuration ... 158
Controlling the Clustering and Synchronization Timers ... 159
Blocking New Connections Under Load ... 160
Defining Non-Monitored Interfaces ... 162
Configuring Policy Update Timeout ... 163
Enhanced 3-Way TCP Handshake Enforcement ... 164
Cluster IP Addresses on Different Subnets ... 166
Example of Cluster IP Addresses on Different Subnets ... 167
Limitations of Cluster Addresses on Different Subnets ... 174
Connectivity Between Cluster Members ... 174
Load Sharing Multicast Mode with "Semi-Supporting" Hardware ... 174
Manual Proxy ARP ... 174
Connecting to the Cluster Members from the Cluster Network ... 175
Adding Another Member to an Existing Cluster ... 176
Adding a New Cluster Member to the Cluster Object ... 176
Adding an Existing Security Gateway as a Cluster Member to the Cluster Object ... 179
Removing a Member from an Existing Cluster ... 183
ISP Redundancy on a Cluster ... 186
Introduction ... 186
ISP Redundancy Modes ... 189
Outgoing Connections ... 190
Incoming Connections ... 191
Configuring ISP Redundancy on a Cluster ... 193
ISP Redundancy and VPN ... 199
Controlling ISP Redundancy from CLI ... 201
Force ISP Link State ... 201
The ISP Redundancy Script ... 201
Dynamic Routing Protocols in a Cluster Deployment ... 202

Router IP Address ... 202
Routing Table Synchronization ... 202
Wait for Clustering ... 202
Failure Recovery ... 203
ClusterXL Failover based on the Load on ClusterXL SND Instances ... 204
Overview ... 204
Configuration ... 205
Configuration Kernel Parameters ... 206
Configuration Examples ... 207
**ClusterXL Configuration Commands ... 209**
Configuring the Cluster Member ID Mode in Local Logs ... 213
Registering a Critical Device ... 214
Unregistering a Critical Device ... 218
Reporting the State of a Critical Device ... 219
Registering Critical Devices Listed in a File ... 221
Unregistering All Critical Devices ... 223
Configuring the Cluster Control Protocol (CCP) Settings ... 224
Initiating Manual Cluster Failover ... 225
Configuring the Minimal Number of Required Subordinate Interfaces for Bond Load Sharing ... 229
Configuring the Multi-Version Cluster Mechanism ... 232
Configuring Duration of Forced Failover in High Availability Mode ... 233
**Monitoring and Troubleshooting Clusters ... 234**
ClusterXL Monitoring Commands ... 235
Viewing Cluster State ... 240
Viewing Critical Devices ... 246
Viewing Cluster Interfaces ... 258
Viewing Bond Interfaces ... 263
Viewing Cluster Failover Statistics ... 268
Viewing Software Versions on Cluster Members ... 270

Viewing Delta Synchronization ... 271
Viewing IGMP Status ... 278
Viewing Cluster Delta Sync Statistics for Connections Table ... 279
Viewing Cluster IP Addresses ... 280
Viewing the Cluster Member ID Mode in Local Logs ... 282
Viewing Interfaces Monitored by RouteD ... 283
Viewing Roles of RouteD Daemon on Cluster Members ... 284
Viewing Cluster Correction Statistics ... 285
Viewing the Cluster Control Protocol (CCP) Settings ... 289
Viewing the State of the Multi-Version Cluster Mechanism ... 290
Viewing Full Connectivity Upgrade Statistics ... 291
Monitoring Cluster Status in SmartConsole ... 292
Background ... 292
ClusterXL Log Messages ... 292
General Logs ... 293
State Logs ... 293
Critical Device Logs ... 294
Interface Logs ... 295
Reason Strings ... 296
Working with SNMP Traps ... 297
How to Initiate Cluster Failover ... 298
Troubleshooting Issues with the Critical Device "routed" ... 300
Background ... 300
Standard Behavior of the Critical Device "routed" ... 300
Basic Troubleshooting Steps ... 301
ClusterXL Error Messages ... 302
**Command Line Reference** ... **303**
ClusterXL Configuration Commands ... 304
ClusterXL Monitoring Commands ... 308
cpconfig ... 313

cphastart ... 317
cphastop ... 318
cp_conf fullha ... 319
cp_conf ha ... 320
fw hastat ... 322
fwboot ha_conf ... 323
**ClusterXL Scripts** ... **324**
The clusterXL_admin Script ... 325
The clusterXL_monitor_ips Script ... 329
The clusterXL_monitor_process Script ... 333
**Cluster Management APIs** ... **337**
Introduction ... 337
List of APIs ... 337
API Examples ... 338
Known Limitations ... 353
**Glossary** ... **354**

# Introduction to ClusterXL

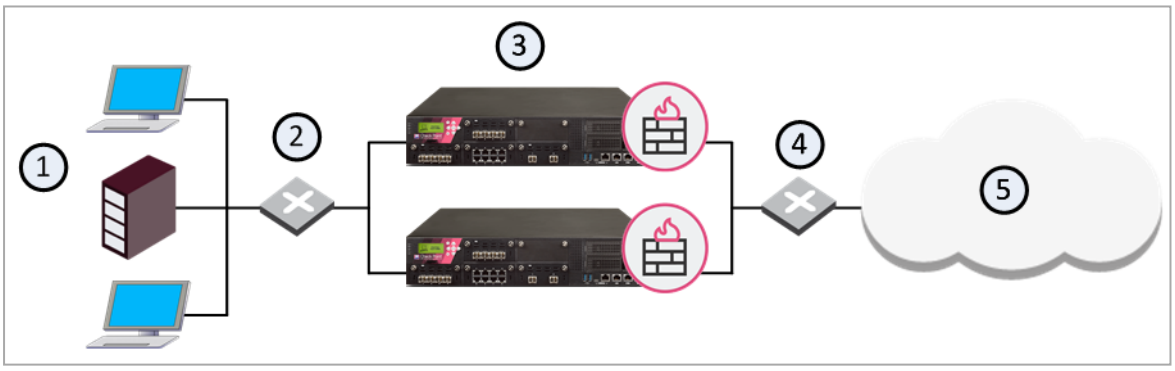
## The Need for Clusters

Security Gateways and VPN connections are business critical devices. The failure of a Security Gateway or VPN connection can result in the loss of active connections and access to critical data. The Security Gateway between the organization and the world must remain open under all circumstances.

## ClusterXL Solution

ClusterXL is a Check Point software-based cluster solution for Security Gateway redundancy and Load Sharing. A ClusterXL Security Cluster contains identical Check Point Security Gateways.

- A High Availability Security Cluster ensures Security Gateway and VPN connection redundancy by providing transparent failover to a backup Security Gateway in the event of failure.
- A Load Sharing Security Cluster provides reliability and also increases performance, as all members are active.




| Item | Description |
|---|---|
| 1 | Internal network |
| 2 | Switch for internal network |
| 3 | Security Gateways with ClusterXL Software Blade |
| 4 | Switch for external networks |
| 5 | Internet |

# How ClusterXL Works

ClusterXL uses *State Synchronization* to keep active connections alive and prevent data loss when a Cluster Member fails. With State Synchronization, each Cluster Member "knows" about connections that go through other Cluster Members.

ClusterXL uses virtual IP addresses for the cluster itself and unique physical IP and MAC addresses for the Cluster Members. Virtual IP addresses do not belong to physical interfaces.

**Note** - This guide contains information only for Security Gateway clusters. For additional information about the use of ClusterXL with VSX, see the *R81.10 VSX Administration Guide*.

# The Cluster Control Protocol

The Cluster Control Protocol (CCP) packets are the glue that links together the members in the Security Cluster.

CCP traffic is distinct from ordinary network traffic and can be viewed using any network sniffer.

CCP runs on UDP port 8116 between the Cluster Members, and has the following roles:

- It allows Cluster Members to report their own states and learn about the states of other members by sending keep-alive packets (this only applies to ClusterXL clusters).
- State Synchronization (Delta Sync).

The Check Point CCP is used by all ClusterXL modes.

**Important** - There is no need to add an explicit rule to the Security Policy Rule Base that accepts CCP packets.

For more information, see *"Configuring the Cluster Control Protocol (CCP) Settings" on page 224*.

# ClusterXL Requirements and Compatibility

*In This Section:*

Check Point Appliances and Open Servers ... 17
Supported Number of Cluster Members ... 18
Hardware Requirements for Cluster Members ... 18
Software Requirements for Cluster Members ... 18
VMAC Mode ... 19
Supported Topologies for Synchronization Network ... 21
Clock Synchronization in ClusterXL ... 24
IPv6 Support for ClusterXL ... 24
Synchronized Cluster Restrictions ... 25

## Check Point Appliances and Open Servers

You can install ClusterXL on Check Point appliances in one of these configurations:

- A Distributed configuration - the Cluster Members and the Security Management Server are installed on different computers.
- A Full High Availability configuration - the Cluster Members and the Security Management Servers are installed on the same computers (each computer runs a Standalone configuration).

You can install ClusterXL on Open Servers only in a distributed configuration - the Cluster Members and the Security Management Server are installed on different computers.

To see the ClusterXL supported platforms, see the *R81.10 Release Notes*.

For installation instructions, see the *R81.10 Installation and Upgrade Guide*.

# Supported Number of Cluster Members

- ClusterXL in High Availability mode supports up to 5 Cluster Members.
- ClusterXL in Load Sharing mode supports up to 5 Cluster Members.

  **Note** - Configuring more than 4 members significantly decreases cluster performance due to amount of Delta Sync.

- VRRP Cluster on Gaia OS supports only 2 Cluster Members (see sk105170).
- Virtual System Load Sharing (VSLS) mode supports up to 13 Cluster Members.

# Hardware Requirements for Cluster Members

ClusterXL operation completely relies on internal timers and calculation of internal timeouts, which are based on hardware clock ticks.

Therefore, in order to avoid unexpected behavior, ClusterXL is supported only between machines with identical CPU characteristics.

**Best Practice** - To avoid unexpected fail-overs due to issues with CCP packets on cluster interfaces, we strongly recommend to pair only identical physical interfaces as cluster interfaces - even when connecting the Cluster Members via a switch.
For example:

- Intel 82598EB on Member_A with Intel 82598EB on Member_B
- Broadcom NeXtreme on Member_A with Broadcom NeXtreme on Member_B

**Note** - There is no requirement for throughput of Sync interface to be identical to, or larger than throughput of traffic interfaces (although, to prevent a possible bottle neck, a good practice for throughput of Sync interface is to be at least identical to throughput of traffic interfaces).

# Software Requirements for Cluster Members

ClusterXL is supported only between identical operating systems - all Cluster Members must be installed on the same operating system).

ClusterXL is supported only between identical Check Point software versions - all Cluster Members must be installed with identical Check Point software, including OS build and hotfixes.

All Check Point software components must be the same on all Cluster Members. Meaning that the same Software Blades and features must be enabled on all Cluster Members:

- SecureXL status on all Cluster Members must be the same (either enabled, or disabled)
- Number of CoreXL Firewall instances on all Cluster Members must be the same

  **Notes:**
  - A Cluster Member with a greater number of CoreXL Firewall instances changes its state to DOWN.
  - Fail-over from a Cluster Member to a peer Cluster Member with a greater number of CoreXL Firewall instances keeps all connections.
  - Fail-over from a Cluster Member to a peer Cluster Member with a smaller number of CoreXL Firewall instances interrupts some connections. The connections that are interrupted are those that pass through CoreXL Firewall instances that do not exist on the peer Cluster Member.
- Advanced Dynamic Routing status and configuration on all Cluster Members must be the same

Otherwise, traffic might not be processed as expected and/or state of Cluster Members might change unexpectedly. In addition, Full Sync will fail.

# VMAC Mode

When ClusterXL is configured in High Availability mode or Load Sharing Unicast mode (not Multicast), a single Cluster Member is associated with the Cluster Virtual IP address. In a High Availability environment, the single member is the Active member. In a Load Sharing environment, the single member is the Pivot.

After fail-over, the new Active member (or Pivot member) broadcasts a series of Gratuitous ARP Requests (GARPs). The GARPs associate the Virtual IP address of the cluster with the physical MAC address of the new Active member or the new Pivot.

When this happens:

- **A member with a large number of Static NAT entries can transmit too many GARPs**

  Switches may not integrate these GARP updates quickly enough into their ARP tables. Switches continue to send traffic to the physical MAC address of the member that failed. This results in traffic outage until the switches have fully updated ARP cache tables.
- **Network components, such as VoIP phones, ignore GARPs**

  These components continue to send traffic to the MAC address of the failed member.

To minimize possible traffic outage during a fail-over, configure the cluster to use a Virtual MAC address (VMAC).

By enabling Virtual MAC in ClusterXL High Availability mode, or Load Sharing Unicast mode, all Cluster Members associate the same Virtual MAC address with all Cluster Virtual Interfaces and the Virtual IP address. In the Virtual MAC mode, the VMAC is advertised by the Cluster Members through GARP Requests.

For local connections and sync connections, the real physical MAC address of each Cluster Member is still associated with its real IP address.

**Note** - Traffic originating from the Cluster Members will be sent with the member's MAC Source address.

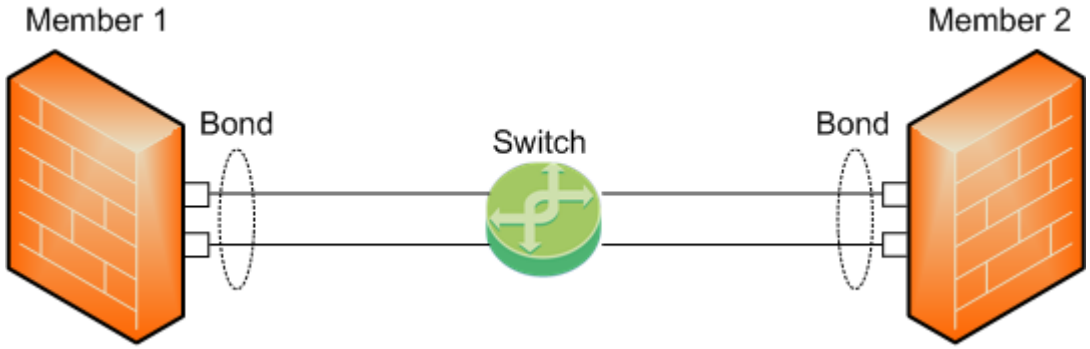
Failover time in a cluster with enabled VMAC mode is shorter than a failover in a cluster that uses a physical MAC addresses.

For configuration instructions, see *"Configuring Virtual MAC (VMAC)" on page 103*.

# Supported Topologies for Synchronization Network

**Note** - The topology numbers below are for convenience only.

### Topology 1



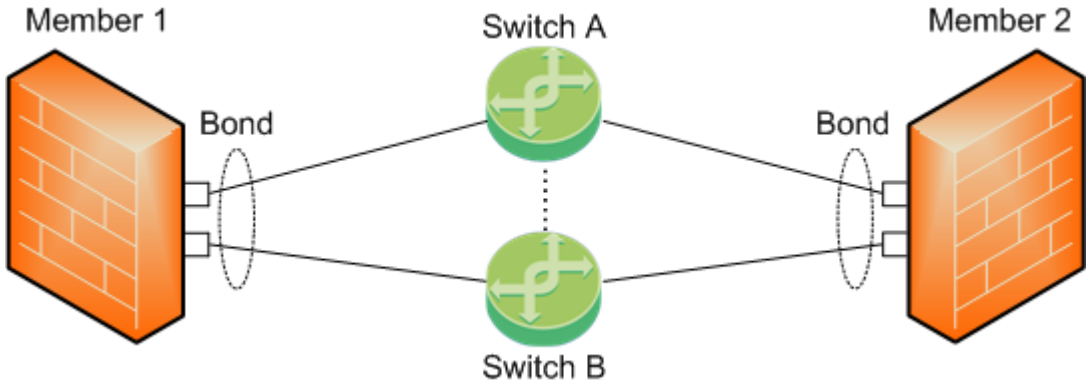

1. The Sync interface is a Bond of several physical subordinate interfaces.

   To work with this topology, you can configure the Bond interface in High Availability or Load Sharing mode.

2. All physical subordinate interfaces on all Cluster Members connect to the same switch.

3. In this topology, Cluster Members monitor only the link state of the subordinate interfaces.

### Topology 2

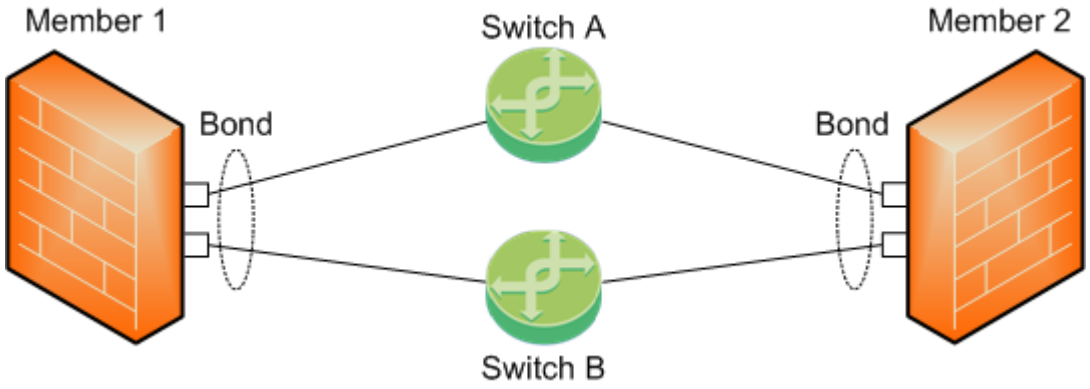



1. The Sync interface is a Bond of several physical subordinate interfaces.

   To work with this topology, you can configure the Bond interface in High Availability or Load Sharing mode.

2. On each Cluster Member, physical subordinate interfaces of the same Bond connect to different switches.
3. The switches must connect to each other through a cable.
4. In this topology, Cluster Members monitor only the link state of the subordinate interfaces.

**Topology 3 (Enhanced Active/Backup Bond)**




1. The Sync interface is a Bond of several physical subordinate interfaces.

   To work with this topology:

   a. You must configure the Bond interface in the **Active-Backup** mode.

   b. You must **not** configure the primary subordinate interface explicitly.
2. On each Cluster Member, physical subordinate interfaces of the same Bond connect to different switches.
3. The switches do **not** need to connect to each other through a cable.
4. In this topology, Cluster Members:

   - Monitor the link state of the subordinate interfaces.
   - Monitor the path between the Cluster Members, communicate about their connectivity, and agree on which bond subordinate interface to use.
   - Send Cluster Control Protocol (CCP) packets only on the active subordinate interface (there is no CCP broadcast on all the subordinate interfaces).

- Fail over only if **all** subordinate interfaces are down.

  **Additional information**

  If a Cluster Member does not receive CCP packets from other Cluster Members (for example, a cable is disconnected between switches), it sends a dedicated CCP packet to all other Cluster Members.

  This dedicated CCP packet includes a map of all available bond subordinate interfaces on that Cluster Member.

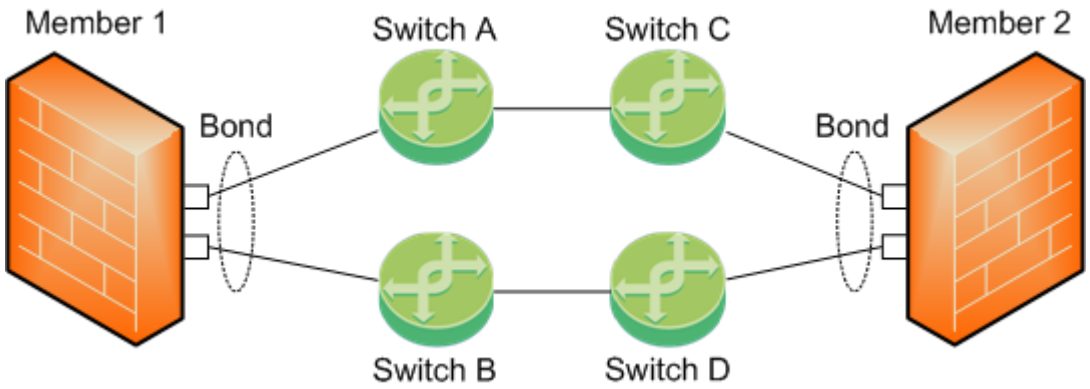
  When other Cluster Members receive this dedicated CCP packet, they:

  a. Compare the received map of available bond subordinate interfaces to the state of their own bond subordinate interfaces.

  b. Perform bond internal failover accordingly.

- Fail over to a specific selected subordinate interface instead of iterating over all available subordinate interfaces.

**Topology 4 (Enhanced Active/Backup Bond)**




1. The Sync interface is a Bond of several physical subordinate interfaces.

   To work with this topology:

   a. You must configure the Bond interface in the **Active-Backup** mode.

   b. You must **not** configure the primary subordinate interface explicitly.

2. On each Cluster Member, physical subordinate interfaces of the same Bond connect to different pairs of switches.

3. These pairs of switches connect to each other (chained together).

4. In this topology, Cluster Members:

- Monitor the link state of the subordinate interfaces.
- Monitor the path between the Cluster Members, communicate about their connectivity, and agree on which bond subordinate interface to use.
- Send Cluster Control Protocol (CCP) packets only on the active subordinate interface (there is no CCP broadcast on all the subordinate interfaces).
- Fail over only if **all** subordinate interfaces are down.

  **Additional information**

  If a Cluster Member does not receive CCP packets from other Cluster Members (for example, a cable is disconnected between switches), it sends a dedicated CCP packet to all other Cluster Members.

  This dedicated CCP packet includes a map of all available bond subordinate interfaces on that Cluster Member.

  When other Cluster Members receive this dedicated CCP packet, they:

  a. Compare the received map of available bond subordinate interfaces to the state of their own bond subordinate interfaces.

  b. Perform bond internal failover accordingly.
- Fail over to a specific selected subordinate interface instead of iterating over all available subordinate interfaces.

# Clock Synchronization in ClusterXL

When using ClusterXL, make sure to synchronize the clocks of all of the Cluster Members. You can synchronize the clocks manually, or using a protocol such as NTP. Features, such as VPN, only function properly when the clocks of all of the Cluster Members are synchronized.

# IPv6 Support for ClusterXL

R81.10 ClusterXL supports High Availability clusters for IPv6.

IPv6 status information is synchronized and the IPv6 clustering mechanism is activated during failover.

**Important** - You must configure IPv4 address on each interface that has the **Network Type "Cluster"**, **"Sync"**, or **"Cluster+Sync"** (in SmartConsole > the cluster object > **Network Management** page).

**You can define IPv6 addresses for:**

- Cluster virtual interfaces
- Member physical interfaces

**Limitations:**

- IPv6 is not supported for Load Sharing clusters.
- You cannot define IPv6 address for synchronization interfaces.

**IPv6 in ClusterXL High Availability:**

During failover, a cluster sends gratuitous ARP request packets to update hosts and routers connected to cluster interfaces. It does this by advertising the new MAC address for the virtual cluster IPv4 addresses.

ClusterXL updates the IPv6 network during failovers. ClusterXL sends Neighbor Advertisement messages to update the neighbor cache (which is equivalent to the ARP cache in IPv4) by advertising the new MAC address for the virtual cluster IPv6 address. In addition, ClusterXL will reply to any Neighbor Solicitation with a target address equal to the Virtual Cluster IPv6 address.

**Note** - ClusterXL failover event detection is based on IPv4 probing. During state transition the IPv4 driver instructs the IPv6 driver to reestablish IPv6 network connectivity to the HA cluster.

# Synchronized Cluster Restrictions

These restrictions apply when you synchronize Cluster Members:

- The use of more than one dedicated physical interface for synchronization redundancy is not supported.

  You can use Bonding for synchronization interface redundancy (see *"Sync Redundancy" on page 135*).

  Synchronization interface redundancy is not supported for VRRP Clusters. See sk92804.
- All Cluster Members must run on identically configured hardware platforms.
- If a Cluster Member goes down, user-authenticated connections through that member are lost.

  Other Cluster Members cannot restore the connection.

  Cluster Members maintain client-authenticated or session-authenticated connections.

The reason for this restriction is that a user space process on Cluster Members maintains the user authentication state.

Cluster Members cannot synchronize the user space information in the same way as they synchronize the kernel space information.

Cluster Members save the states of Session Authentication and Client Authentication in kernel tables, which they synchronize.

- Cluster Members cannot synchronize the connection statutes that use system resources. The reason is the same as for the user-authenticated connections.
- Accounting information for connections is accumulated on each Cluster Member, sent to the Management Server, and aggregated.

  In the event of a cluster failover, the accounting information that is not yet sent to the Management Server, is lost.

  To minimize this risk, you can reduce the time interval when accounting information is sent.

  To do this, in the cluster object > **Logs** > **Additional Logging** pane, set a lower value for the **Update Account Log every** attribute.
- If you change the IP address of the cluster synchronization network after the cluster is already up and running, you must restart the routing daemon on all cluster members.

  **Best Practice:**
  - In the High Availability mode, first perform these steps on all Standby cluster members.
  - In the Load Sharing Unicast mode, first perform these steps on all Non-Pivot cluster members.

  1. Connect to the command line on each cluster member.
  2. Log in to the Expert mode.
  3. Stop the RouteD daemon:

     ```
     tellpm process:routed
     ```
  4. Start the RouteD daemon:

     ```
     tellpm process:routed t
     ```
  5. Log in to Gaia Clish.
  6. Wait for several minutes.

7. Examine the peer neighborship in the configured dynamic routing protocols.

   See the *R81.10 Gaia Administration Guide*.

# High Availability and Load Sharing Modes in ClusterXL

## Introduction to High Availability and Load Sharing modes

ClusterXL is a software-based High Availability and Load Sharing solution that distributes network traffic between clusters of redundant Security Gateways.

ClusterXL has these High Availability features:

- Transparent failover in case of member failures
- Zero downtime for mission-critical environments (when using State Synchronization)
- Enhanced throughput (in Load Sharing modes)
- Transparent upgrades

All members in the cluster are aware of the connections passing through each of the other members. The Cluster Members synchronize their connection and status information across a secure synchronization network.

The glue that binds the members in a ClusterXL cluster is the Cluster Control Protocol (CCP), which is used to pass synchronization and other information between the Cluster Members.

## High Availability

In a High Availability cluster, only one member is active (Active/Standby operation). In the event that the active Cluster Member becomes unavailable, all connections are re-directed to a designated standby without interruption. In a synchronized cluster, the standby Cluster Members are updated with the state of the connections of the Active Cluster Member.

In a High Availability cluster, each member is assigned a priority. The highest priority member serves as the Security Gateway in normal circumstances. If this member fails, control is passed to the next highest priority member. If that member fails, control is passed to the next member, and so on.

Upon Security Gateway recovery, you can maintain the current Active Security Gateway (Active Up), or to change to the highest priority Security Gateway (Primary Up).

ClusterXL High Availability mode supports both IPv4 and IPv6.

# Load Sharing

ClusterXL Load Sharing distributes traffic within a cluster so that the total throughput of multiple members is increased. In Load Sharing configurations, all functioning members in the cluster are active, and handle network traffic (Active/Active operation).

If any member in a cluster becomes unreachable, transparent failover occurs to the remaining operational members in the cluster, thus providing High Availability. All connections are shared between the remaining Security Gateways without interruption.

ClusterXL Load Sharing modes do not support IPv6.

# Example ClusterXL Topology

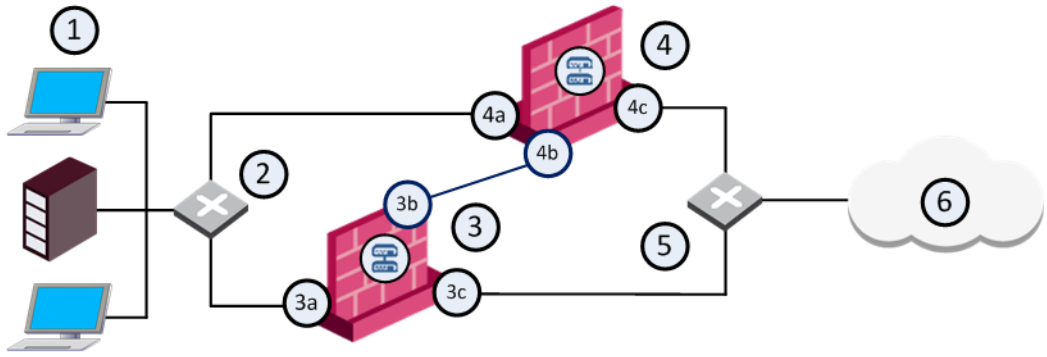
ClusterXL uses unique physical IP and MAC addresses for each **Cluster Member**, and a virtual IP addresses for the **cluster** itself.

Cluster interface virtual IP addresses do not belong to any real member interface.

## Example Diagram

The following diagram illustrates a two-member ClusterXL cluster, showing the cluster Virtual IP addresses and members physical IP addresses.

This sample deployment is used in many of the examples presented in this chapter.




| Item | Description |
|---|---|
| 1 | Internal network |
| 2 | Internal switch (internal cluster IP address 10.10.0.100) |
| 3 | Security Gateway - Cluster Member A |
| 3a | Virtual interface to the internal network (10.10.0.1) |
| 3b | Interface to the Cluster Sync network (10.0.10.1) |
| 3c | Virtual interface to the external network (192.168.10.1) |
| 4 | Security Gateway - Cluster Member B |
| 4a | Virtual interface to the internal network (10.10.0.2) |
| 4b | Interface to the Cluster Sync network (10.0.10.2) |
| 4c | Virtual interface to the external network (192.168.10.2) |
| 5 | External switch (external cluster IP address 192.168.10.100) |

| Item | Description |
|---|---|
| 6 | Internet |

Each Cluster Member has three interfaces: one external interface, one internal interface, and one for synchronization. Cluster Member interfaces facing in each direction are connected via a hub or switch.

All Cluster Member interfaces facing the same direction must be in the same network. For example, there must not be a router between Cluster Members.

The Management Server can be located anywhere, and connection should be established to either the internal or external cluster IP addresses.

These sections present ClusterXL configuration concepts shown in the example.

**Note** - In these examples, RFC 1918 private addresses in the range `192.168.0.0` to `192.168.255.255` are treated as public IP addresses.

# Defining the Cluster Member IP Addresses

The guidelines for configuring each Cluster Member are as follows:

All members within the cluster must have at least three interfaces:

- An interface facing the external network (that, for example, faces the Internet).
- An interface facing the internal network.
- An interface used for synchronization.

All interfaces pointing in a certain direction must be on the same network.

For example, in the previous illustration, there are two Cluster Members, Member_A and Member_B. Each has an interface with an IP address facing the Internet through a hub or a switch. This is the external interface with IP address 192.168.10.1 on Member_A and IP address 192.168.10.2 on Member_B.

**Note** - This release presents an option to use only two interfaces per member, one external and one internal, and to run synchronization over the internal interface. We do not recommend this configuration. It should be used for backup only. (See *"Synchronizing Connections in the Cluster" on page 60*.)

# Defining the Cluster Virtual IP Addresses

In the previous illustration, the IP address of the cluster is **192.168.10.100**.

The cluster has one external virtual IP address and one internal virtual IP address.

The external IP address is **192.168.10.100**, and the internal IP address is **10.10.0.100**.

## Defining the Synchronization Network

The previous illustration shows a synchronization interface with a unique IP address on each Cluster Member - IP **10.0.10.1** on Member_A and IP **10.0.10.2** on Member_B.

## Configuring Cluster Addresses on Different Subnets

Only one public IP address is required in a ClusterXL cluster, for the virtual cluster interface that faces the Internet. Physical IP addresses of all Cluster Members can be private.

Configuring different subnets for the cluster IP addresses and the members IP addresses (see *"Cluster IP Addresses on Different Subnets" on page 166*) is useful to:

- Configure a cluster to replace one Security Gateway in a pre-configured network, without the need to allocate new IP addresses to the Cluster Members.
- Allow organizations to use only one public IP address for the ClusterXL Cluster. This saves public IP addresses.

# ClusterXL Mode Considerations

## Choosing High Availability, Load Sharing, or Active-Active mode

Which cluster mode to choose depends on the need and requirements of the organization.

- A High Availability cluster mode ensures fail-safe connectivity for the organization.
- A Load Sharing cluster mode ensures fail-safe connectivity for the organization and provides the additional benefit of increased performance.
- An Active-Active cluster mode supports deployment of Cluster Members in different geographical areas (in different networks).

See *"ClusterXL Mode Comparison" on page 43*.

## Considerations for the Load Sharing Mode

Load Sharing Multicast mode is an efficient way to handle a high traffic load, because the load is distributed optimally between all Active Cluster Members.

However, not all switches can be used for Load Sharing Multicast mode. Load Sharing Multicast mode associates a multicast Cluster MAC addresses with a unicast Cluster Virtual IP addresses. This ensures that traffic destined for the cluster is received by all Cluster Members.

In response to ARP Request packets for Cluster Virtual IP address, Cluster Members send ARP Replies that contain a unicast Cluster Virtual IP address and a multicast MAC address. Some switches do not accept such ARP Replies. For some switches, adding a static ARP entry for the unicast Cluster Virtual IP address and the multicast MAC address will solve the issue. Other switches do not accept this type of static ARP entry.

Another consideration is whether your deployment includes networking devices with interfaces operating in a promiscuous mode. If on the same network segment there exist two such networking devices, and a ClusterXL in Load Sharing Multicast mode, traffic destined for the cluster that is generated by one of the networking device could also be processed by the other networking device.

For these cases, use Load Sharing Unicast mode, which does not require the use of multicast MAC address for the Cluster Virtual IP addresses.

In addition, see:

- sk101539 - ClusterXL Load Sharing mode limitations and important notes
- sk162637 - ClusterXL Load Sharing mode in R80.20 and higher

# IP Address Migration

If you wish to provide High Availability or Load Sharing to an existing Security Gateways configuration, we recommend taking the existing IP addresses from the Active Security Gateway, and make these the Cluster Virtual IP addresses, when feasible. Doing so lets you avoid altering of current IPsec endpoint identities, as well keep Hide NAT configurations the same in many cases.

# ClusterXL Modes

ClusterXL has several working modes.

This section briefly describes each mode and its relative advantages and disadvantages.

- **High Availability Mode**
- **Load Sharing Multicast Mode**
- **Load Sharing Unicast Mode**
- **Active-Active Mode** (see *"Active-Active Mode in ClusterXL" on page 47*)

**Note** - Many examples in the section refer to the sample deployment shown in the *"Example ClusterXL Topology" on page 30*.

## High Availability Mode

The ClusterXL High Availability mode provides basic High Availability capabilities in a cluster environment. This means that the cluster can provide Firewall services even when it encounters a problem, which on a regular Security Gateway results in a complete loss of connectivity. When combined with Check Point State Synchronization, ClusterXL High Availability can maintain connections through failover events, in a user-transparent manner, allowing a flawless connectivity experience. As a result, High Availability provides a backup mechanism, which organizations can use to reduce the risk of unexpected downtime, especially in a mission-critical environment (such as money transactions).

To achieve this, ClusterXL High Availability mode designates one of the Cluster Members as the Active member, while the other Cluster Members remain in Standby mode. The cluster associates the Virtual IP addresses with the physical MAC addresses of the physical interfaces on the Active member (by matching the cluster Virtual IP address with the unique MAC address of the appropriate physical interface). Therefore, all traffic directed at the cluster Virtual IP addresses, is actually routed (and filtered) by the Active Cluster Member.

The role of each Cluster Member is chosen according to its cluster state and Cluster Member priority. Cluster Member priorities correspond to the order, in which they appear in the **Cluster Members** page of the cluster object in SmartConsole. The top-most Cluster Member has the highest priority. You can modify this ranking at any time (requires policy installation and causes a failover).

In addition to its role as a Firewall, the Active Cluster Member is also responsible for informing the Standby Cluster Members of any changes to its cluster state and kernel tables. This keeps the peer Cluster Members up-to-date with the current traffic passing through the cluster.

Whenever the Active Cluster Member detects a problem that is severe enough to prevent this Cluster Member from working correctly, failover occurs in the cluster. One of the Standby Cluster Members (the Cluster Member with the next highest priority) assumes the role of the Active Cluster Member. If State Synchronization is enabled, the new Active Cluster Member recognizes any open connections and handles them according to their last known state.

Upon the recovery of a failed former Active Cluster Member with a higher priority, the role of the Active Cluster Member may or may not be switched back to that Cluster Member. This depends on the cluster object configuration - **Maintain current active Cluster Member**, or **Switch to higher priority Cluster**.

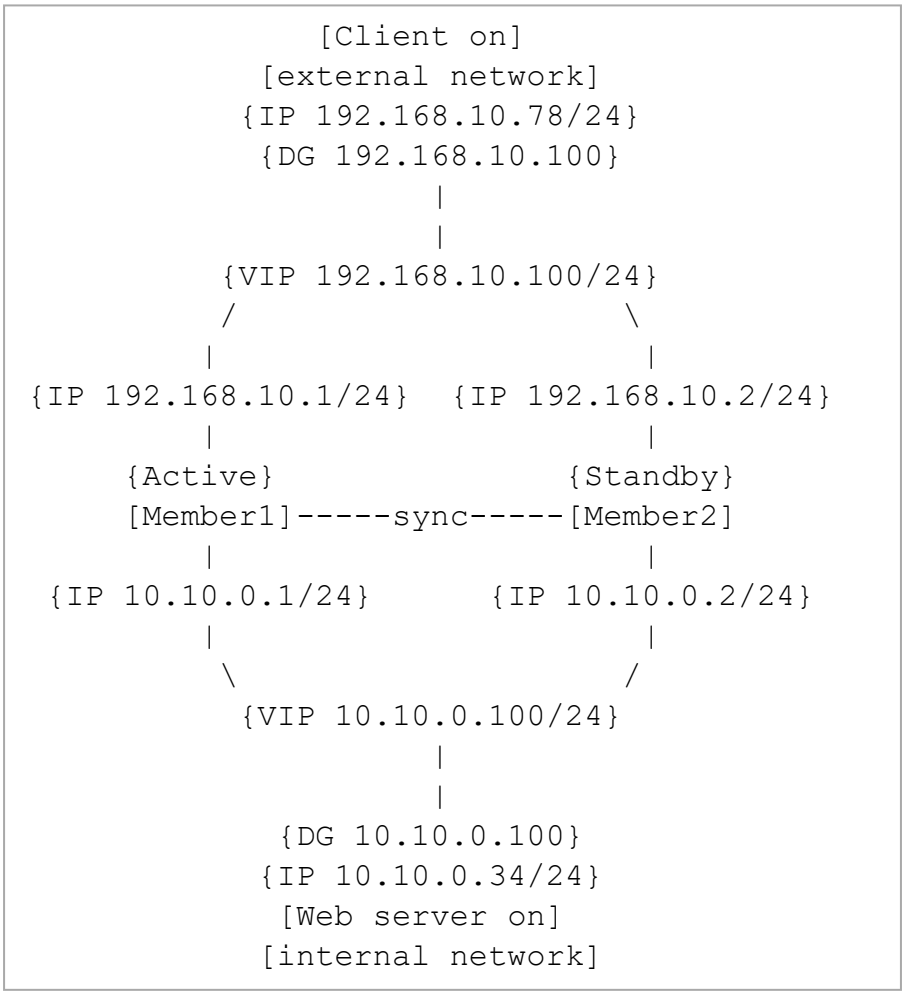
It is important to note that Standby Cluster Members may encounter problems as well. In this case, in the event of a cluster failover, the Standby Cluster Members are not considered for the role of a new Active Cluster Member.

**Example**

This scenario describes a connection from a Client computer on the external network to a Web server behind the Cluster (on the internal network).

The cluster of two members is configured in High Availability mode.

**Example topology:**



**Chain of events:**

1. The user tries to connect from his Client computer **192.168.10.78** to the Web server **10.10.0.34**.
2. The Default Gateway on Client computer is **192.168.10.100** (the cluster Virtual IP address).
3. The Client computer issues an ARP Request for IP **192.168.10.100**.
4. The Active Cluster Member (Member1) handles the ARP Request for IP **192.168.10.100**.
5. The Active Cluster Member sends the ARP Reply with the MAC address of the external interface, on which the IP address **192.168.10.1** is configured.
6. The Client computer sends the HTTP request packet to the Active Cluster Member - to the VIP address **192.168.10.100** and MAC address of the corresponding external interface.
7. The Active Cluster Member handles the HTTP request packet.
8. The Active Cluster Member sends the HTTP request packet to the Web server **10.10.0.34**.
9. The Web server handles the HTTP request packet.
10. The Web server generates the HTTP response packet.
11. The Default Gateway on Web server computer is **10.10.0.100** (the cluster Virtual IP address).
12. The Web server issues an ARP Request for IP **10.10.0.100**.
13. The Active Cluster Member handles the ARP Request for IP **10.10.0.100**.
14. The Active Cluster Member sends the ARP Reply with the MAC address of the internal interface, on which the IP address **10.10.0.1** is configured.
15. The Web server sends the HTTP response packet to the Active Cluster Member - to VIP address **10.10.0.100** and MAC address of the corresponding internal interface.
16. The Active Cluster Member handles the HTTP response packet.
17. The Active Cluster Member sends the HTTP response packet to the Client computer **192.168.10.78**.
18. From now, all traffic between the Client computer and the Web server is routed through the Active Cluster Member (Member1).
19. If a failure occurs on the current Active Cluster Member (Member1), the cluster fails over.

20. The Standby Cluster Member (Member2) assumes the role of the Active Cluster Member.
21. The Standby Cluster Member sends Gratuitous ARP Requests to both the **192.168.10.x** and the **10.10.0.x** networks.

    These GARP Requests associate the Cluster Virtual IP addresses with the MAC addresses of the physical interfaces on the *new* Active Cluster Member (former Standby Cluster Member):

    - Cluster VIP address **192.168.10.100** and MAC address of the corresponding external interface, on which the IP address **192.168.10.2** is configured.
    - Cluster VIP address **10.10.0.100** and MAC address of the corresponding internal interface, on which the IP address **10.10.0.2** is configured.
22. From now, all traffic between the Client computer and the Web server is routed through the *new* Active Cluster Member (**Member2**).
23. The former Active member (Member1) is now considered to be "down". Upon the recovery of a former Active Cluster Member, the role of the Active Cluster Member may or may not be switched back to that Cluster Member, depending on the cluster object configuration.

# Load Sharing Modes

See:

- sk101539 - ClusterXL Load Sharing mode limitations and important notes
- sk162637 - ClusterXL Load Sharing mode in R80.20 and higher

## Load Sharing Multicast Mode

Load Sharing lets you distribute network traffic between Cluster Members. In contrast to High Availability, where only a single member is active at any given time, all Cluster Members in a Load Sharing solution are active. The Cluster Members decide, which Cluster Member handles which packet. The cluster decision function performs this task - examining each packet going through the cluster, and determining which Cluster Member should handle it. As a result, a Load Sharing cluster utilizes all Cluster Members, which usually leads to an increase in its total throughput.

It is important to understand that ClusterXL Load Sharing, provides a full High Availability solution as well. When all Cluster Members are Active, traffic is evenly distributed between the Cluster Members. In case of a failover event, caused by a problem in one of the Cluster Members, the processing of all connections handled by the faulty member is immediately taken over by the other Cluster Members.

ClusterXL offers two separate Load Sharing solutions: Multicast and Unicast. The two modes differ in the way Cluster Members receive the packets sent to the cluster. This section describes the Load Sharing Multicast mode.

The Multicast mechanism, which is provided by the Ethernet network layer, allows several interfaces to be associated with a single physical (MAC) address. Unlike Broadcast, which binds all interfaces in the same subnet to a single MAC address, Multicast enables grouping within networks. This means that it is possible to select the interfaces within a single subnet that will receive packets sent to a given MAC address.

ClusterXL uses the Multicast mechanism to associate the cluster Virtual IP addresses with Multicast MAC addresses. This makes sure that all packets sent to the cluster, acting as a default gateway on the network, will reach all Cluster Members. Each Cluster Member then decides, whether it should process the packets or not. This decision is the core of the Load Sharing mechanism: it has to assure that at least one Cluster Member will process each packet (so that traffic is not blocked), and that no two Cluster Members will handle the same packets (so that traffic is not duplicated).

An additional requirement of the decision function, is to route each connection through a Cluster Member, to ensure that packets that belong to a single connection will be processed by the same Cluster Member. Unfortunately, this requirement cannot always be enforced, and in some cases, packets of the same connection will be handled by different Cluster Members. ClusterXL handles these situations using its State Synchronization mechanism, which mirrors connections on all Cluster Members.

**Example**

This scenario describes a user logging from the Internet to a Web server behind the cluster that is configured in Load Sharing Multicast mode.

1. The user requests a connection from **192.168.10.78** (his computer) to **10.10.0.34** (the Web server).
2. A router on the **192.168.10.x** network recognizes **192.168.10.100** (the cluster Virtual IP address) as the default gateway to the **10.10.0.x** network.
3. The router issues an ARP Request for IP **192.168.10.100**.
4. One of the Active Cluster Members processes the ARP Request, and responds with the Multicast MAC assigned to the cluster Virtual IP address of **192.168.10.100**.
5. When the Web server responds to the user requests, it recognizes **10.10.0.100** as its default gateway to the Internet.
6. The Web server issues an ARP Request for IP **10.10.0.100**.
7. One of the Active members processes the ARP Request, and responds with the Multicast MAC address assigned to the cluster Virtual IP address of **10.10.0.100**.
8. All packets sent between the user and the Web server reach every Cluster Member, which decide whether to process each packet.
9. When a cluster failover occurs, one of the Cluster Members goes down. However, traffic still reaches all of the Active Cluster Members. Therefore, there is no need to make changes in the network ARP routing. The only thing that changes, is the cluster decision function, which takes into account the new state of the Cluster Members.

## Load Sharing Unicast Mode

Load Sharing Unicast mode provides a Load Sharing solution adapted to environments, where Multicast Ethernet cannot operate. In this mode, a single Cluster Member, referred to as *Pivot*, is associated with the cluster Virtual IP addresses. This Pivot Cluster Member is the only Cluster Member to receive all packets sent to the cluster. The Pivot Cluster Member is then responsible for propagating the packets to other Cluster Members, creating a Load Sharing mechanism. Distribution of packets is performed by applying a decision function on each packet, the same way it is done in Load Sharing Multicast mode. The difference is that only one Cluster Member performs this selection: any non-Pivot member that receives a forwarded packet will handle it, without applying the decision function. Note that non-Pivot Cluster Members are still active, because they perform routing and Firewall tasks on a share of the traffic (although they do not perform decisions).

Even though the Pivot Cluster Member is responsible for the decision process, it still acts as a Security Gateway that processes packets (for example, the decision it makes, can be to handle a packet by itself). However, since its additional tasks can be time consuming, it is usually assigned a smaller share of the total load.

When a failure occurs in a non-Pivot member, its handled connections are redistributed between active non-Pivot Cluster Members, providing the same High Availability capabilities of High Availability and Load Sharing Multicast. When the Pivot Cluster Member encounters a problem, a regular failover event occurs, and, in addition, another active non-Pivot member assumes the role of the new Pivot. The Pivot member is always the active member with the highest priority. This means that when a former Pivot recuperates, it will retain its previous role.

**Example**

In this scenario, we use a Load Sharing Unicast cluster as the Security Gateway between the end user computer and the Web server.

1. The user requests a connection from **192.168.10.78** (his computer) to **10.10.0.34** (the Web server).
2. A router on the **192.168.10.x** network recognizes **192.168.10.100** (the cluster Virtual IP address) as the default gateway to the **10.10.0.x** network.
3. The router issues an ARP Request for IP **192.168.10.100**.
4. The Pivot Cluster Member handles the ARP Request, and responds with the MAC address that corresponds to its own unique IP address of **192.168.10.1**.
5. When the Web server responds to the user requests, it recognizes **10.10.0.100** as its default gateway to the Internet.
6. The Web server issues an ARP Request for IP **10.10.0.100**.
7. The Pivot Cluster Member handles the ARP Request, and responds with the MAC address that corresponds to its own unique IP address of **10.10.0.1**.
8. The user request packet reaches the Pivot Cluster Member on interface **192.168.10.1**.
9. The Pivot Cluster Member decides that the second non-Pivot Cluster Member should handle this packet, and forwards it to **192.168.10.2**.
10. The second Cluster Member recognizes the packet as a forwarded packet, and handles it.
11. Further packets are processed by either the Pivot Cluster Member, or forwarded and processed by the non-Pivot Cluster Member.
12. When a failover occurs from the current Pivot Cluster Member, the second Cluster Member assumes the role of Pivot.

13. The new Pivot member sends Gratuitous ARP Requests to both the **192.168.10.x** and the **10.10.0.x** networks. These GARP requests associate the cluster Virtual IP address of **192.168.10.100** with the MAC address that corresponds to the unique IP address of **192.168.10.2**, and the cluster Virtual IP address of **10.10.0.100** with the MAC address that correspond to the unique IP address of **10.10.0.2**.
14. Traffic sent to the cluster is now received by the new Pivot Cluster Member, and processed by it (as it is currently the only Active Cluster Member).
15. When the former Pivot Cluster Member recovers, it re-assumes the role of Pivot, by associating the cluster Virtual IP addresses with its own unique MAC addresses.

# ClusterXL Mode Comparison

This table summarizes the similarities and differences between the ClusterXL modes.

| Feature | High Availability | Load Sharing Multicast | Load Sharing Unicast | Active-Active |
|---|---|---|---|---|
| High Availability | Yes | Yes | Yes | No |
| Load Sharing | No | Yes | Yes | No |
| Performance | Good | Excellent | Very Good | Good |
| State Synchronization | Optional | Mandatory | Mandatory | Optional |
| Hardware Support | All routers | Not all routers are supported | All routers | All routers |
| Number of members that deal with network traffic | 1 | N | N | N |
| Number of members that receive packets from router | 1 | N | 1 | N |
| How cluster answers ARP requests for a MAC address | Unicast | Unicast | Unicast | N / A |
| VLAN Tagging Support | Yes | Yes | Yes | Yes |

# Cluster Failover

## What is Failover?

Failover is a cluster redundancy operation that automatically occurs if a Cluster Member is not functional. When this occurs, other Cluster Members take over for the failed Cluster Member.

**In the High Availability mode:**

- If the Active Cluster Member detects that it cannot function as a Cluster Member, it notifies the peer Standby Cluster Members that it must go down. One of the Standby Cluster Members (with the next highest priority) will promote itself to the Active state.
- If one of the Standby Cluster Members stops receiving Cluster Control Protocol (CCP) packets from the current Active Cluster Member, that Standby Cluster Member can assume that the current Active Cluster Member failed. As a result, one of the Standby Cluster Members (with the next highest priority) will promote itself to the Active state.
- If you do not use State Synchronization in the cluster, existing connections are interrupted when cluster failover occurs.

**In Load Sharing modes:**

- If a Cluster Member detects that it cannot function as a Cluster Member, it notifies the peer Cluster Members that it must go down. Traffic load will be redistributed between the working Cluster Members.
- If the Cluster Members stop receiving Cluster Control Protocol (CCP) packets from one of their peer Cluster Member, those working Cluster Members can assume that their peer Cluster Member failed. As a result, traffic load will be redistributed between the working Cluster Members.
- Because by design, all Cluster Members are always synchronized, current connections are not interrupted when cluster failover occurs.

To tell each Cluster Member that the other Cluster Members are alive and functioning, the ClusterXL Cluster Control Protocol (CCP) maintains a heartbeat between Cluster Members. If after a predefined time, no CCP packets are received from a Cluster Member, it is assumed that the Cluster Member is down. As a result, cluster failover can occur.

Note that more than one Cluster Member may encounter a problem that will result in a cluster failover event. In cases where all Cluster Members encounter such problems, ClusterXL will try to choose a single Cluster Member to continue operating. The state of the chosen member will be reported as *Active(!)*. This situation lasts until another Cluster Member fully recovers. For example, if a cross cable connecting the sync interfaces on Cluster Members malfunctions, both Cluster Members will detect an interface problem. One of them will change to the *Down* state, and the other to *Active (!)* state.

# When Does a Failover Occur?

A failover takes place when one of the following occurs in a cluster:

- Any Critical Device reports its state as "`problem`" (see *"Viewing Critical Devices" on page 246*).

  For example, the "`fwd`" process failed, or Security Policy is uninstalled on a Cluster Member.

- A Cluster Member does not receive Cluster Control Protocol (CCP) packets from its peer Cluster Member.

For more on failovers, see sk62570.

# What Happens When a Cluster Member Recovers?

**In the High Availability mode:**

- If cluster object is configured as **Maintain current active Cluster Member**, it means any Cluster Member that becomes Active, remains Active.

  If the Cluster Member with highest priority fails, cluster failover occurs. A Cluster Member with the next highest priority becomes Active.

  If the Cluster Member with highest priority recovers, cluster failover does not occurs again, and that Cluster Member becomes Standby.

- If cluster object is configured as **Switch to higher priority Cluster Member**, it means that Cluster Member with the highest priority always has to be Active.

  Cluster Member with the highest priority is the Cluster Member that appears at the top of the list in Cluster object > **Cluster Members** pane.

  If the Cluster Member with the highest priority fails, cluster failover occurs. A peer Cluster Member in Standby state, with the next highest priority, becomes Active.

  If the Cluster Member with the highest priority recovers, cluster failover occurs again. The Cluster Member with the highest priority becomes Active again. The Cluster Member with the next highest priority that was Active, returns to the Standby state.

**In the Load Sharing modes:**

- When the failed Cluster Member recovers, all connections are redistributed between all Active Cluster Members.

# How a Recovered Cluster Member Obtains the Security Policy

The Administrator installs the Security Policy on the cluster object, rather than separately on individual Cluster Members. The policy is automatically installed on all Cluster Members. The policy is sent to the IP addresses defined in the **General Properties** page of the cluster member object.

When a failed cluster member recovers, first it tries to fetch a policy from one of the peer Active Cluster Members. The assumption is that the other Cluster Members have a more up to date policy. If fetching a policy from peer cluster member fails, the recovered cluster member compares its own local policy to the policy on its Management Server. If the policy on the Management Server is more up to date than the one on the recovered cluster member, the policy is fetched from the Management Server. If the cluster member does not have a local policy, it retrieves one from the Management Server. This ensures that all Cluster Members use the same policy at any given moment.

# General Failover Limitations

Some connections may **not** survive cluster failover:

- Security Servers connections.
- Connections that are handled by the Check Point services, in which the option **Synchronize connections on cluster** is disabled.
- Connections initiated by the Cluster Member itself.
- TCP connections handled by the Check Point Active Streaming (CPAS) or Passive Streaming Layer (PSL) mechanism.
- Connections handled by Software Blades:
  - If the IPS Software Blade in the cluster object is configured to **Prefer connectivity**, and the Cluster Member that owns the connections is **Down**, then the connection is accepted without inspection.

    Otherwise, the Cluster Members drop the connection.
  - For all other Software Blades:
    - If the destination Cluster Member is available, the connection is forwarded to the Cluster Member that owns the connection.
    - If the destination Cluster Member is **not** available, the Cluster Members drop the connection.

# Active-Active Mode in ClusterXL

## Introduction

R80.40 introduced a new ClusterXL mode called **Active-Active**.

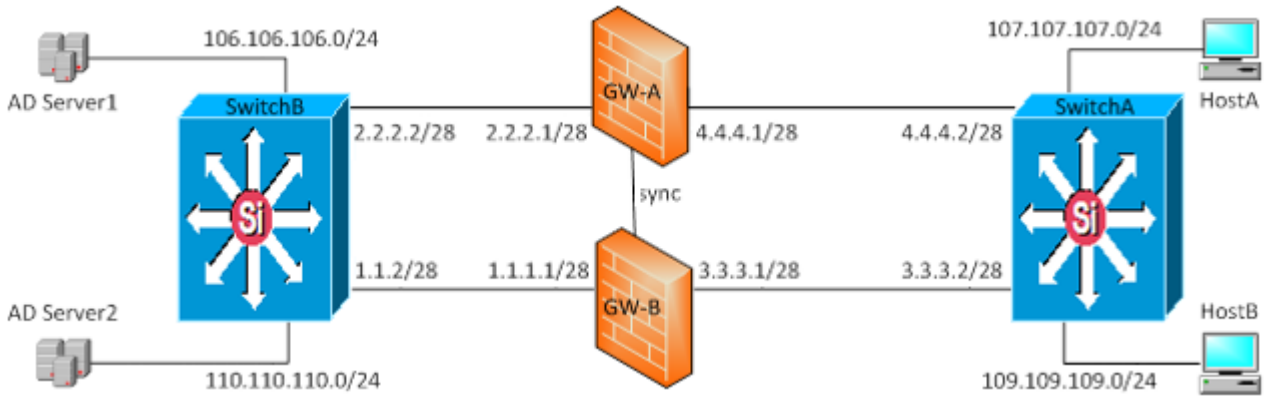
This mode is designed for a cluster, whose Cluster Members are located in different geographical areas (different sites, different cloud availability zones).

The IP addresses of the interfaces on each Cluster Member are on different networks (including the Sync interfaces).

Each Cluster Member inspects all traffic routed to it and synchronizes the recorded connections to its peer Cluster Members.

The traffic is **not** balanced between the members.

**Example Topology:**




**Important:**

- You can configure the Active-Active mode only on Management Server R80.40 (and higher) and only on ClusterXL R80.40 (and higher).
- The Active-Active mode does **not** provide Load Sharing of the traffic. Administrator must monitor the load on each Cluster Member (see *"Monitoring and Troubleshooting Clusters" on page 234*).
- Cluster Control Protocol (CCP) Encryption must be enabled, which is the default (see *"Configuring the Cluster Control Protocol (CCP) Settings" on page 224*).
- It is possible to configure the Cluster Control Protocol (CCP) to work on Layer 2.
- It is possible to enable Dynamic Routing on each Cluster Member , so they become routers in the applicable Area or Autonomous System on the site.
  In this case, you must enable the Bidirectional Forwarding Detection (BFD - *ip-reachability-detection*) on each interface that is configured for dynamic routing.

# Limitations

These limitations apply to Active-Active mode in ClusterXL:

- Only these Software Blades and features are supported:
  - Firewall
  - IPS
  - NAT
- Up to **four** Cluster Members are supported.
- VSX is **not** supported.
- All Multi-Portals are **not** supported (examples of Multi-Portals: Mobile Access Portal, Identity Awareness Captive Portal, Data Loss Prevention Portal).
- Names of interfaces that belong to the same "side" must be the same on all Cluster Members.

  Example: If you connected the interface *eth1* to Switch A on Cluster Member, then you must connect the interface *eth1* to Switch #A on all other Cluster Members.
- In the cluster object properties, go to the **Network Management** page, select a cluster interface and click **Edit**.

  In the **Network Type** field, it is not supported to select **Cluster + Sync** when you deploy a cluster on-premisesor in the cloud (examples: AWS, Azure).
- In the cluster object properties, go to the **Network Management** page, select a cluster interface, and click **Edit**.

  In the **Topology** section, only these options are supported for cluster interfaces:
  - **Override** > **Network defined by routes** (this is the default).
  - **Override** > **Specific** > select the applicable Network object or Network Group object.
- If you enable Dynamic Routing on Cluster Members, then you must enable the Bidirectional Forwarding Detection (BFD) - *ip-reachability-detection*) on each interface that is configured for dynamic routing.
- The option **Hide internal networks behind the Gateway's external IP** (in the ClusterXL object properties > **NAT** pane) is not supported.
- The option **Hide behind the gateway** (in the ClusterXL object properties > **NAT** pane) is not supported.

# Configuring a new ClusterXL in Active-Active Mode

**Note** - This procedure is to configure a new ClusterXL object.

**Procedure**

| Step | Instructions |
|---|---|
| 1 | Install ClusterXL Cluster Members.<br>See the *R81.10 Installation and Upgrade Guide*.<br>**Important:**<br>On all Cluster Members in Active-Active mode, names of interfaces that belong to the same "side" must be identical (Known Limitation PMTR-70256).<br>Examples:<br>■ If you connected the interface `eth1` to Switch #A on one Cluster Member, then you must connect the interface `eth1` to Switch #A on all other Cluster Members.<br>■ If you configured the interface `eth2` as a Sync interface on one Cluster Member, then you must configure the interface `eth2` as a Sync on all other Cluster Members. |
| 2 | **Optional:** On each Cluster Member, enable Dynamic Routing, so they become routers in the applicable Area or Autonomous System on the site.<br>**Important** - In this case, you must enable the Bidirectional Forwarding Detection (BFD - *ip-reachability-detection*) on each interface that is configured for dynamic routing.<br>See the *R81.10 Gaia Advanced Routing Administration Guide*. |
| 3 | Connect with SmartConsole to the Management Server. |
| 4 | From the left navigation panel, click **Gateways & Servers**. |

| Step | Instructions |
|---|---|
| 5 | Create a new Cluster object in one of these ways:<br>■ From the top toolbar, click **New (✳) > Cluster > Cluster**.<br>■ In the top left corner, click **Objects** menu **> More object types > Network Object > Gateways and Servers > Cluster > New Cluster**.<br>■ In the top right corner, click **Objects Pane > New > More > Network Object > Gateways and Servers > Cluster > Cluster**.<br>In the **Check Point Security Gateway Cluster Creation** window, you must click **Classic Mode**. |
| 6 | From the left tree, click **General Properties**.<br>a. In the **IPv4 Address** field, you must enter the **0.0.0.0** address.<br>b. On the **Network Security** tab, you must clear **IPsec VPN**.<br>c. For the list of supported Software Blades, see the Known Limitation PMTR-70255 in sk170418. |
| 7 | From the left tree, click **Cluster Members**.<br>a. Click **Add > New Cluster Member**.<br>The **Cluster Member Properties** window opens.<br>b. In the **Name** field, enter the applicable name for the first Cluster Member object.<br>c. Configure the main physical IP address(es) for this Cluster Member object.<br>In the **IPv4 Address** and **IPv6 Address** fields, configure the same IPv4 and IPv6 addresses that you configured on the **Management Connection** page of the Cluster Member's First Time Configuration Wizard.<br>Make sure the Security Management Server or Multi-Domain Server can connect to these IP addresses.<br>d. Click **Communication**.<br>e. In the **One-time password** and **Confirm one-time password** fields, enter the same Activation Key you entered during the Cluster Member's First Time Configuration Wizard.<br>f. Click **Initialize**.<br>g. Click **Close**.<br>h. Click **OK**.<br>i. Repeat Steps **a-h** to add the next Cluster Member. |

| Step | Instructions |
|---|---|
| | If the **Trust State** field does not show **Trust established**, follow these steps:<br>a. Connect to the command line on the Cluster Member.<br>b. Make sure there is a physical connectivity between the Cluster Member and the Management Server (for example, pings can pass).<br>c. Run:<br>`cpconfig`<br>d. Enter the number of this option:<br>`Secure Internal Communication`<br>e. Follow the instructions on the screen to change the Activation Key.<br>f. In SmartConsole, click **Reset**.<br>g. Enter the same Activation Key you entered in the `cpconfig` menu.<br>h. In SmartConsole, click **Initialize**. |

| Step | Instructions |
|---|---|
| 8 | From the left tree, click **ClusterXL and VRRP**.<br>a. In the **Select the cluster mode and configuration** section, select **Active-Active**.<br>b. In the **Tracking** section, select the applicable option.<br>c. **Optional:** Select **Use State Synchronization**.<br>This configures the Cluster Members to synchronize the information about the connections they inspect.<br>**Best Practice** - Enable this setting to prevent connection drops after a cluster failover.<br>d. **Optional:** Select **Start synchronizing [ ] seconds after connection initiation** and enter the applicable value.<br>This option is available only for clusters R80.20 and higher.<br>To prevent the synchronization of short-lived connections (which decreases the cluster performance), you can configure the Cluster Members to start the synchronization of **all** connections a number of seconds after they start.<br>**Range:** 2 - 60 seconds<br>**Default:** 3 seconds<br>**Notes:**<br>■ This setting in the cluster object applies to all connections that pass through the cluster.<br>You can override this global cluster synchronization delay in the properties of applicable services - see *"Configuring Services to Synchronize After a Delay" on page 63*.<br>■ The greater this value, the fewer short-lived connections the Cluster Members have to synchronize.<br>■ The connections that the Cluster Members did not synchronize, do not survive a cluster failover.<br>**Best Practice** - Enable and configure this setting to increase the cluster performance. |
| 9 | Click **OK** to update the cluster object properties with the new cluster mode. |
| 10 | Open the cluster object and continue the configuration. |

| Step | Instructions |
|---|---|
| 11 | From the left tree, click **Network Management**.<br>a. From the top, click the **Get Interfaces** > **Get Interfaces With Topology**.<br>**Important** - On all Cluster Members in Active-Active mode, names of interfaces that belong to the same "side" must be identical (Known Limitation PMTR-70256).<br>b. Select *each* interface and click **Edit**.<br>c. From the left tree, click the **General** page.<br>d. In the **General** section, in the **Network Type** field, select the applicable type:<br>■ For cluster traffic interfaces, select **Cluster**.<br>In the **IPv4** field, the dummy IP address **0.0.0.0 / 24** is configured automatically.<br>**Note** - SmartConsole requires an IP address configured for each interface. Cluster Members do **not** get and do **not** use this IP address.<br>■ For cluster synchronization interfaces, select **Sync**.<br>**Notes:**<br>• Check Point cluster supports only one synchronization network. If redundancy is required, configure a Bond interface.<br>• In the **Network Type** field, it is not supported to select "**Cluster+Sync**" (Known Limitation PMTR-70259) when you deploy a cluster in a cloud (for example: AWS, Azure).<br>■ For interfaces that do not pass the traffic between the connected networks, select **Private**.<br>e. In the **Member IPs** section, make sure the IPv4 address and its Net Mask are correct on each Cluster Member.<br>f. In the **Topology** section:<br>Make sure the settings are correct in the **Leads To** and **Security Zone** fields.<br>Only these options are supported on cluster interfaces (Known Limitation PMTR-70260):<br>■ **Override** > **Network defined by routes** (this is the default).<br>■ **Override** > **Specific** > select the applicable **Network** object or **Network Group** object. |
| 12 | Click **OK**. |
| 13 | Publish the SmartConsole session. |
| 14 | Configure and install the applicable Access Control Policy. |

| Step | Instructions |
|---|---|
| 15 | Configure and install the applicable Threat Prevention Policy. |
| 16 | Examine the cluster state.<br>See *"Viewing Cluster State" on page 240*.<br>**Example**<br>`Member1> show cluster state`<br><br>`Cluster Mode:  Active Active with IGMP Membership`<br><br>`ID  Unique Address  Assigned Load  State  Name`<br><br>`1 (local)  11.22.33.245  N/A  ACTIVE  Member1`<br>`2  11.22.33.246  N/A  ACTIVE  Member2`<br><br>`Active PNOTEs: None`<br><br>`... ... ...`<br><br>`Member1>` |

# Changing the ClusterXL Mode to Active-Active

**Procedure**

| Step | Instructions |
|---|---|
| 1 | Connect with SmartConsole to the Management Server. |
| 2 | From the left navigation panel, click **Gateways & Servers**. |
| 3 | Open the existing ClusterXL object. |
| 4 | From the left tree, click **ClusterXL and VRRP**. |
| 5 | In the **Select the cluster mode and configuration** section, select **Active-Active**. |
| 6 | **Mandatory:** Click **OK** to update the cluster object properties with the new cluster mode. |
| 7 | Open the cluster object and continue the configuration. |

| Step | Instructions |
|---|---|
| 8 | From the left tree, click **ClusterXL and VRRP**.<br>a. In the **Tracking** section, select the applicable option.<br>b. **Optional:** Select **Use State Synchronization**.<br>This configures the Cluster Members to synchronize the information about the connections they inspect.<br>**Best Practice** - Enable this setting to prevent connection drops after a cluster failover.<br>c. **Optional:** Select **Start synchronizing [] seconds after connection initiation** and enter the applicable value.<br>This option is available only for clusters with version R80.20 and higher.<br>To prevent the synchronization of short-lived connections (which hurts cluster performance), you can configure the Cluster Members to start the synchronization of all connections a number of seconds after they start.<br>**Range:** 2 - 60 seconds<br>**Default:** 3 seconds<br>**Notes:**<br>■ This setting in the cluster object applies to all connections that pass through the cluster.<br>■ You can override this global cluster synchronization delay in the properties of applicable services - see *"Configuring Services to Synchronize After a Delay" on page 63*.<br>■ The greater this value, the fewer short-lived connections the Cluster Members have to synchronize.<br>■ The connections that the Cluster Members did not synchronize do not survive a cluster failover.<br>**Best Practice** - Enable and configure this setting to improve the cluster performance. |

| Step | Instructions |
|---|---|
| 9 | From the left tree, click **Network Management**.<br>a. Select each interface and click **Edit**.<br>b. From the left tree, click the **General** page.<br>c. In the **General** section, in the **Network Type** field, select the applicable type:<br>■ For cluster traffic interfaces, select **Cluster**.<br>In the **IPv4** field, the dummy IP address **0.0.0.0 / 24** is configured automatically.<br>**Note** - SmartConsole requires an IP address configured for each interface. Cluster Members do **not** get and do **not** use this IP address.<br>■ For cluster synchronization interfaces, select **Sync**.<br>• Check Point cluster supports only one synchronization network. If redundancy is required, configure a Bond interface.<br>• In the **Network Type** field, it is **not** supported to select **Cluster + Sync** when you deploy a cluster in a cloud (for example: AWS, Azure).<br>• For interfaces that do not pass the traffic between the connected networks, select **Private**.<br>d. In the **Member IPs** section, make sure the IPv4 address and its Netmask are correct for each Cluster Member.<br>e. In the **Topology** section:<br>i. Make sure the settings are correct in the **Leads To** and **Security Zone** fields.<br>ii. Click **Modify**.<br>iii. In the **Leads To** section, if you select **Override**, only these options are supported on cluster interfaces (Known Limitation PMTR-70260):<br>■ **Override** > **Network defined by routes** (this is the default).<br>■ **Override** > **Specific** > select the applicable **Network** object or **Network Group** object.<br>iv. In the **Anti-Spoofing** section, make sure to clear the option **Perform Anti-Spoofing based on interface topology**.<br>v. Click **OK**. |
| 10 | Click **OK**. |
| 11 | Publish the SmartConsole session. |
| 12 | Configure and install the applicable Access Control Policy. |

| Step | Instructions |
|---|---|
| 13 | Configure and install the applicable Threat Prevention Policy. |
| 14 | Examine the cluster state.<br>See *"Viewing Cluster State" on page 240*.<br>**Example**<br>`Member1> show cluster state`<br><br>`Cluster Mode:  Active Active with IGMP Membership`<br><br>`ID  Unique Address  Assigned Load  State  Name`<br><br>`1 (local)  11.22.33.245  N/A  ACTIVE  Member1`<br>`2  11.22.33.246  N/A  ACTIVE  Member2`<br><br>`Active PNOTEs: None`<br><br>`... ... ...`<br><br>`Member1>` |

# Dynamic Routing Failover

- If there is an issue with the Sync interface (interface state or interface link):
  - The Critical Device **Interfaces Active Check** on the Cluster Member reports its state as "`problem`".
  - Each Cluster Member reports the state of its peer Cluster Member as **LOST**.
- If you run the command "`clusterXL_admin down`" (see *"The clusterXL_admin Script" on page 325*):
  - The Critical Device **admin_down** on the Cluster Member reports its state as "`problem`".
  - The Cluster Member, on which you ran this command, reports its state as **DOWN**.

When the cluster state of a Cluster Member is **DOWN**, it stops processing the dynamic routing traffic to force the next hop router to update its routing tables. As a result, there may be a network outage, because it takes time for dynamic routing protocols to update their routing tables and propagate the changes.

# Synchronizing Connections in the Cluster

A failure of a Firewall results in an immediate loss of active connections in and out of the organization. Many of these connections, such as financial transactions, may be mission critical, and losing them will result in the loss of critical data. ClusterXL supplies an infrastructure that ensures that no data is lost in case of a failure, by making sure each Cluster Member is aware of the connections going through the other members. Passing information about connections and other Security Gateway states between the Cluster Members is called State Synchronization.

Every IP-based service (including TCP and UDP) recognized by the Security Gateway, is synchronized.

Members of a ClusterXL in Load Sharing mode must be synchronized.

Members of a ClusterXL in High Availability mode do not have to be synchronized. Although, if they are not synchronized, current connections are interrupted during cluster failover.

# The Synchronization Network

The Synchronization Network is used to transfer synchronization information about connections and other Security Gateway states between Cluster Members.

The synchronization network carries the most sensitive Security Policy information in the organization. Therefore, it is critical that you protect it against both malicious and unintentional threats.

We recommend that you secure the synchronization interfaces using one of these strategies:

- Enable the CCP Encryption (this is the default) on the Cluster Members (see *"Configuring the Cluster Control Protocol (CCP) Settings" on page 224*).
- Use a dedicated synchronization network.
- Connecting the physical network interfaces of the Cluster Members directly using a cross-cable. In a cluster with three or more members, use a dedicated hub or switch.

**Notes:**

- See *"Supported Topologies for Synchronization Network" on page 21*.
- You can synchronize members across a WAN. See *"Synchronizing Clusters on a Wide Area Network" on page 74*.
- In ClusterXL, the synchronization network is supported on the lowest VLAN tag of a VLAN interface.
  For example, if three VLANs with tags *10*, *20* and *30* are configured on interface *eth1*, only interface *eth1.10* may be used for synchronization.

# How State Synchronization Works

State Synchronization works in two modes:

- The *Full Sync* transfers all Security Gateway kernel table information from one Cluster Member to another.

  The Full Sync is used for initial transfers of state information, when a Cluster Member joins the cluster. If a Cluster Member is brought up after being down, it performs the Full Sync with the Active peer Cluster Member(s). After all Cluster Members are synchronized, only updates are transferred using the Delta Sync, because the Delta Sync is quicker than the Full Sync.

  1. To perform a Full Sync with a peer Cluster Member, the **cxld** daemon on a Cluster Member connects to the TCP port 263 on the peer Cluster Member.
  2. If this connection fails, then the Cluster Member falls back to the previous mechanism (as it worked in versions R80.40 and lower) - the **fwd** daemon connects to the TCP port 256 on the peer Cluster Member.

- The *Delta Sync* transfers the *changes* in the kernel tables between Cluster Members.

  After all Cluster Members complete a Full Sync, the Delta Sync is used for transfers of changes in state information of the connections.

  The Security Gateway kernel handles the Delta Sync using UDP connections on port 8116.

State Synchronization traffic typically makes up around 90% of all Cluster Control Protocol (CCP) traffic.

Cluster Members distinguish the State Synchronization packets from the rest of CCP traffic based on the opcode in the UDP data header.

# Configuring Services to Synchronize After a Delay

Some TCP services (for example, HTTP) are characterized by connections with a very short duration. There is no point to synchronize these connections, because every synchronized connection consumes resources on Cluster Members, and the connection is likely to have finished by the time a cluster failover occurs.

For short-lived services, you can use the *Delayed Notifications* feature to delay telling the Cluster Member about a connection, so that the connection is only synchronized, if it still exists X seconds after the connection was initiated. The Delayed Notifications feature requires SecureXL to be enabled on all Cluster Members (this is the default).

**Note** - In the Load Sharing Multicast mode, Cluster Members ignore this setting for asymmetric connections - when a response from a server to a client arrives to a different Cluster Member then the one that handled a request from a client to a server.

**Procedure:**

1. In SmartConsole, click **Objects > Object Explorer**.
2. In the left tree, click the small arrow on the left of the **Services** to expand this category.
3. In the left tree, select **TCP**.
4. Search for the applicable TCP service.
5. Double-click the applicable TCP service.
6. In the TCP service properties window, click **Advanced** page.
7. At the top, select **Override default settings**.

   On Domain Management Server: select **Override global domain settings**.
8. At the bottom, in the **Cluster and synchronization** section:
   a. Select **Synchronize connections on cluster if State Synchronization is enabled on the cluster**.
   b. Select **Start synchronizing**.
   c. Enter the applicable value.

   **Important** - This change applies to all policies that use this service.
9. Click **OK**.
10. Close the **Object Explorer**.

11. Publish the SmartConsole session.
12. Install the Access Control Policy on the cluster object.

**Note** - The Delayed Notifications setting in the service object is ignored, if Connection Templates are not offloaded by the Firewall to SecureXL. For additional information about the Connection Templates, see the *R81.10 Performance Tuning Administration Guide*.

# Configuring Services not to Synchronize

Synchronization of connections incurs a performance cost. Not all connections that go through a cluster must be synchronized:

- Protocols that run solely between Cluster Members need not be synchronized. Although you can synchronize them, you do not gain any benefit. This synchronization information does not help during a cluster failover.
- You can decide not to synchronize TCP, UDP and other service types. By default, Cluster Members synchronize all these services.
- The VRRP and the IGMP protocols are not synchronized by default (but you can choose to turn on synchronization for these protocols).
- Broadcast and multicast connections are not, and cannot be, synchronized.

You may choose not to synchronize a service if these conditions are true:

- A significant amount of traffic goes through the cluster. Not synchronizing the service reduces the amount of synchronization traffic, and so enhances cluster performance.
- The service typically opens short connections, whose loss may not be noticed. DNS (over UDP) and HTTP are typically responsible for most connections, frequently have short life, and inherent recoverability in the application level. Services that open long connections, such as FTP, should always be synchronized.
- Configurations that ensure bi-directional stickiness for all connections, do not require synchronization to operate (only to maintain High Availability). Such configurations include:
  - Any cluster in High Availability mode (for example, ClusterXL High Availability, or VRRP on Gaia).
  - ClusterXL in a Load Sharing mode with clear connections (no VPN, or Static NAT).
  - VPN and Static NAT connections passing through a ClusterXL cluster in a Load Sharing mode (either multicast, or unicast) may not maintain bi-directional stickiness. State Synchronization must be turned on for such environments.

You can have a synchronized service and a non-synchronized definition of a service, and use them selectively in the Rule Base. For more information, see the *R81.10 Security Management Administration Guide*.

**To configure a service not to synchronize in a cluster:**

1. In SmartConsole, click **Objects > Object Explorer**.
2. In the left tree, select **Services**.
3. Double-click the applicable existing synchronized service, for which you need to create a non-synchronized counterpart service.
4. Write down all the settings from both the **General** and **Advanced** pages.
5. Click **OK**.
6. Click **New > Service >** > select the *applicable service type*.
7. Enter the applicable name that distinguishes the new non-synchronized counterpart service from the existing synchronized service.
8. On the **General** page, configure the same settings as in the existing synchronized service.
9. On the **Advanced** page:
   a. Configure the same settings as in the existing synchronized service.
   b. In the **Cluster and synchronization** section, clear **Synchronize connections if State Synchronization is enabled on the cluster**.

   **Important** - This change applies to all policies that use this service.
10. Click **OK**.
11. Close the **Object Explorer**.
12. Use the synchronized service and the non-synchronized counterpart service in the applicable rules in the applicable Access Control Policies.
13. Publish the SmartConsole session.
14. Install the Access Control Policy on the cluster object.

# Sticky Connections

*In This Section:*


Introduction to Sticky Connections ..... 67

VPN Tunnels with 3rd Party Peers and Load Sharing ..... 67

Third-Party Gateways in Hub and Spoke VPN Deployments ..... 68

Configuring a Third-Party Gateway in a Hub and Spoke VPN Deployment ..... 69


## Introduction to Sticky Connections

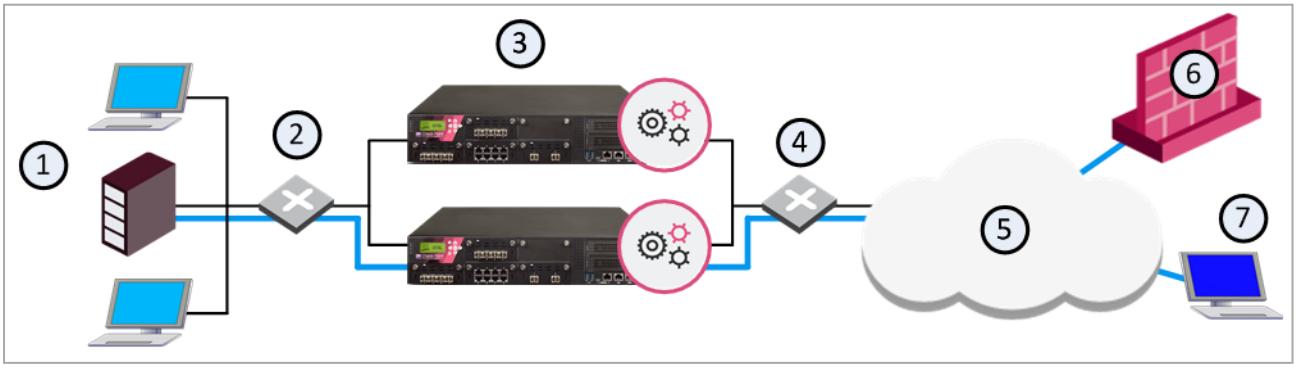
A connection is considered **sticky**, when all of its packets are handled, in either direction, by a single Cluster Member.
This is the case in High Availability mode, where all connections are routed through the same Cluster Member, and hence, are sticky.
This is also the case in Load Sharing mode, when there are no VPN peers, Static NAT rules, or SIP traffic.

In Load Sharing mode, there are cases, where it is necessary to ensure that a connection that starts on a specific Cluster Member will continue to be processed by the same Cluster Member in both directions. Certain connections can be made sticky by enabling the Sticky Decision Function in the cluster object in SmartConsole.

The Check Point *Cluster Correction Layer (CCL)* deals with asymmetric connections in Check Point cluster.

## VPN Tunnels with 3rd Party Peers and Load Sharing

Check Point provides interoperability with third-party vendor gateways by enabling them to peer with Check Point Security Gateways. A special case occurs when certain third-party peers (for example, Microsoft LT2P, Cisco gateways) attempt to establish VPN tunnels with ClusterXL in Load Sharing mode. These VPN peers are limited in their ability to store SAs, which means that a VPN session that begins on one Cluster Member and, due to Load Sharing, is routed on the return trip through another Cluster Member, is unrecognized and dropped.



| Item | Description |
|---|---|
| 1 | Internal network |
| 2 | Switch for internal network |
| 3 | ClusterXL in Load Sharing mode - Cluster Members "A" and "B" |
| 4 | Switch for external networks |
| 5 | Internet |
| 6 | 3rd party peer VPN Gateway |
| 7 | 3rd party peer laptop with VPN client |

**In this scenario:**

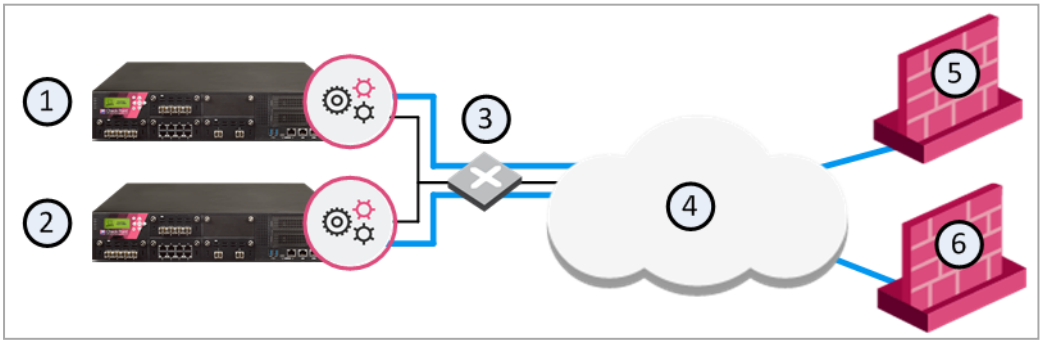
- A third-party peer (gateway or client) attempts to create a VPN tunnel.
- Cluster Members "A" and "B" belong to a ClusterXL in Load Sharing mode.

The third-party peers, lacking the ability to store more than one set of SAs, cannot negotiate a VPN tunnel with multiple Cluster Members, and therefore the Cluster Member cannot complete the routing transaction.

This issue is resolved for certain third-party peers or gateways that can save only one set of SAs by making the connection sticky. The Cluster Correction Layer (CCL) makes sure that a single Cluster Member processes all VPN sessions, initiated by the same third-party gateway.

# Third-Party Gateways in Hub and Spoke VPN Deployments

Another case, where Load Sharing mode requires the connection stickiness, which the Cluster Correction Layer (CCL) provides, is when integrating certain third-party gateways into a Hub and Spoke deployment. Without the ability to store more than one set of SAs, a third-party gateway must maintain its VPN tunnels on a single Cluster Member in order to avoid duplicate SAs.



| Item | Description |
|---|---|
| 1 | Security Gateway - Cluster Member A |
| 2 | Security Gateway - Cluster Member B |
| 3 | Switch for external networks |
| 4 | Internet |
| 5 | Gateway - Spoke A |
| 6 | Gateway - Spoke B |

**In this sample deployment:**

- The intent of this deployment is to enable hosts that reside behind Spoke A to communicate with hosts behind Spoke B.
- The ClusterXL in Load Sharing mode, is composed of Cluster Members "A" and "B", and serves as a VPN Hub.
- Spoke A is a third-party gateway, and is connected by a VPN tunnel that passes through the VPN Hub to Spoke B.
- Spoke B can be either another third-party gateway, or a Check Point Security Gateway.

Spokes A and B must be set to always communicate using the same Cluster Member. The Cluster Correction Layer (CCL) solves half of this problem, in that a single Cluster Member processes all VPN sessions initiated by either third-party gateway.

To make sure that all communications between Spokes A and B are always using the same Cluster Member, you must make some changes to the applicable **user.def** file (see sk98239). This second step ensures that both third-party gateways always connect to the same Cluster Member.

# Configuring a Third-Party Gateway in a Hub and Spoke VPN Deployment

To configure a third-party gateway as a spoke in a Hub and Spoke VPN deployment, perform the following on the Management Server:

1. Create a Tunnel Group to handle traffic from specific peers. Edit the applicable **user.def** file (see sk98239), and add a line similar to the following:

   ```
   all@{member1,member2} vpn_sticky_gws = {<10.10.10.1;1>,
   <20.20.20.1;1>};
   ```

   The elements of this configuration are as follows:

- **all** - All cluster interfaces
- **member1,member2** - Names of Cluster Members in SmartConsole
- **vpn_sticky_gws** - Table name
- **10.10.10.1** - IP address of Spoke A
- **20.20.20.1** - IP address of Spoke B
- **;1** - Tunnel Group Identifier, which indicates that the traffic from these IP addresses should be handled by the same Cluster Member

2. Other VPN peers can be added to the Tunnel Group by including their IP addresses in the same format as shown above. To continue with the example above, adding Spoke C would look like this:

```
all@{member1,member2} vpn_sticky_gws = {<10.10.10.1;1>,
<20.20.20.1;1>, <30.30.30.1;1>};
```

The Tunnel Group Identifier **;1** stays the same, which means that the listed peers will always connect through the same Cluster Member.

**Note** - More tunnel groups than Cluster Members may be defined.

This procedure turns off Load Sharing for the affected connections. If the implementation is to connect multiple sets of third-party gateways one to another, a form of Load Sharing can be accomplished by setting Security Gateway pairs to work in tandem with specific Cluster Members. For instance, to set up a connection between two other spokes (C and D), simply add their IP addresses to the line and replace the Tunnel Group Identifier **;1** with **;2**. The line would then look something like this:

```
all@{member1,member2} vpn_sticky_gws = {<10.10.10.1;1>,
<20.20.20.1;1>, <192.168.15.5;2>, <192.168.1.4;2>,};
```

Note that there are now two peer identifiers: **;1** and **;2**. Spokes A and B will now connect through one Cluster Member, and Spokes C and D through another.

**Note** - The tunnel groups are shared between active Cluster Members. In case of a change in cluster state (for example, failover or Cluster Member attach/detach), the reassignment is performed according to the new state.

# Non-Sticky Connections

*In This Section:*

Non-Sticky Connection Example: TCP 3-Way Handshake ..... 71
Synchronizing Non-Sticky Connections ..... 72

A connection is called **sticky** if a single Cluster Member handles all packets of that connection. In a **non-sticky** connection, the response packet of a connection returns through a different Cluster Member than the original request packet.

The cluster synchronization mechanism knows how to handle non-sticky connections properly. In a non-sticky connection, a Cluster Member can receive an out-of-state packet, which Firewall normally drops because it poses a security risk.

In Load Sharing configurations, all Cluster Members are active. In Static NAT and encrypted connections, the source and destination IP addresses change. Therefore, Static NAT and encrypted connections through a Load Sharing cluster may be non-sticky. Non-stickiness may also occur with Hide NAT, but ClusterXL has a mechanism to make it sticky.

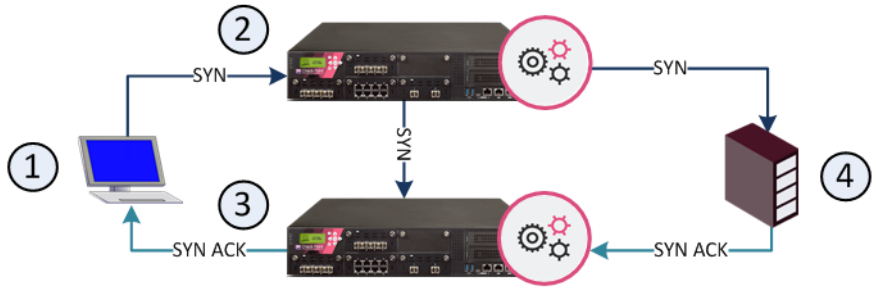
In High Availability configurations, all packets reach the Active member, so all connections are sticky.

If the other members do not know about a non-sticky connection, the packet will be out-of-state, and the connection will be dropped for security reasons. However, the Synchronization mechanism knows how to inform other members of the connection. The Synchronization mechanism thereby prevents out-of-state packets in valid, but non-sticky connections, so that these non-sticky connections are allowed.

Non-sticky connections will also occur if the network Administrator has configured asymmetric routing, where a reply packet returns through a different Security Gateway than the original packet.

## Non-Sticky Connection Example: TCP 3-Way Handshake

The 3-way handshake that initiates all TCP connections can very commonly lead to a non-sticky (often called asymmetric routing) connection. This diagram shows a sample scenario:



| Item | Description |
|---|---|
| 1 | Client |
| 2 | Security Gateway - Cluster Member A |
| 3 | Security Gateway - Cluster Member B |
| 4 | Server |

The client initiates a connection by sending a SYN packet to the server. The SYN passes through Cluster Member A, but the SYN-ACK reply returns through Cluster Member B. This is a non-sticky connection, because the reply packet returns through a different Security Gateway than the original packet.

The synchronization network notifies Cluster Member B. If Cluster Member B is updated before the SYN-ACK packet sent by the server reaches it, the connection is handled normally. If, however, synchronization is delayed, and the SYN-ACK packet is received on Cluster Member B before the SYN flag has been updated, then the Security Gateway treats the SYN-ACK packet as out-of-state, and drops the connection.

You can configure enhanced 3-Way TCP Handshake enforcement to address this issue (see *"Enhanced 3-Way TCP Handshake Enforcement" on page 164*).

## Synchronizing Non-Sticky Connections

The synchronization mechanism prevents out-of-state packets in valid, but non-sticky connections. The way it does this is best illustrated with reference to the 3-way handshake that initiates all TCP data connections. The 3-way handshake proceeds as follows:

1. SYN (client to server)
2. SYN-ACK (server to client)
3. ACK (client to server)
4. Data (client to server)

To prevent out-of-state packets, the following sequence (called "Flush and Ack") occurs

1. Cluster Member receives first packet (SYN) of a connection.
2. Suspects that it is non-sticky.
3. Hold the SYN packet.
4. Send the pending synchronization updates to all Cluster Members (including all changes relating to this packet).

5. Wait for all the other Cluster Members to acknowledge the information in the sync packet.
6. Release held SYN packet.
7. All Cluster Members are ready for the SYN-ACK.

# Synchronizing Clusters on a Wide Area Network

Organizations sometimes need to locate Cluster Members in geographical locations that are distant from each other.

A typical example is a replicated Data Center, whose locations are widely separated for disaster recovery purposes.

In such a configuration, it is clearly impractical to use a cross cable for the synchronization network.

The synchronization network can be spread over remote sites, which makes it easier to deploy geographically distributed clustering.

There are two limitations to this capability:

1. The synchronization network must guarantee no more than 100ms latency and no more than 5% packet loss.
2. The synchronization network may include only Layer 2 networking devices - switches and hubs.

   No Layer 3 routers are allowed on the synchronization network, because routers drop Cluster Control Protocol (CCP) packets.

   **Note** - The Active-Active mode also supports Layer 3 (see *"Active-Active Mode in ClusterXL" on page 47*).

You can monitor and troubleshoot geographically distributed clusters using the command line interface.

### Synchronized Cluster Restrictions

These restrictions apply when you synchronize Cluster Members:

- The use of more than one dedicated physical interface for synchronization redundancy is not supported.

  You can use Bonding for synchronization interface redundancy (see *"Sync Redundancy" on page 135*).

  Synchronization interface redundancy is not supported for VRRP Clusters. See sk92804.
- All Cluster Members must run on identically configured hardware platforms.
- If a Cluster Member goes down, user-authenticated connections through that member are lost.

  Other Cluster Members cannot restore the connection.

  Cluster Members maintain client-authenticated or session-authenticated connections.

The reason for this restriction is that a user space process on Cluster Members maintains the user authentication state.

Cluster Members cannot synchronize the user space information in the same way as they synchronize the kernel space information.

Cluster Members save the states of Session Authentication and Client Authentication in kernel tables, which they synchronize.

- Cluster Members cannot synchronize the connection statutes that use system resources. The reason is the same as for the user-authenticated connections.
- Accounting information for connections is accumulated on each Cluster Member, sent to the Management Server, and aggregated.

  In the event of a cluster failover, the accounting information that is not yet sent to the Management Server, is lost.

  To minimize this risk, you can reduce the time interval when accounting information is sent.

  To do this, in the cluster object > **Logs** > **Additional Logging** pane, set a lower value for the **Update Account Log every** attribute.
- If you change the IP address of the cluster synchronization network after the cluster is already up and running, you must restart the routing daemon on all cluster members.

  **Best Practice:**
  - In the High Availability mode, first perform these steps on all Standby cluster members.
  - In the Load Sharing Unicast mode, first perform these steps on all Non-Pivot cluster members.

  1. Connect to the command line on each cluster member.
  2. Log in to the Expert mode.
  3. Stop the RouteD daemon:

     ```
     tellpm process:routed
     ```
  4. Start the RouteD daemon:

     ```
     tellpm process:routed t
     ```
  5. Log in to Gaia Clish.
  6. Wait for several minutes.

7. Examine the peer neighborship in the configured dynamic routing protocols.

   See the *R81.10 Gaia Administration Guide*.

# Synchronized Cluster Restrictions

These restrictions apply when you synchronize Cluster Members:

- The use of more than one dedicated physical interface for synchronization redundancy is not supported.

  You can use Bonding for synchronization interface redundancy (see *"Sync Redundancy" on page 135*).

  Synchronization interface redundancy is not supported for VRRP Clusters. See sk92804.

- All Cluster Members must run on identically configured hardware platforms.
- If a Cluster Member goes down, user-authenticated connections through that member are lost.

  Other Cluster Members cannot restore the connection.

  Cluster Members maintain client-authenticated or session-authenticated connections.

  The reason for this restriction is that a user space process on Cluster Members maintains the user authentication state.

  Cluster Members cannot synchronize the user space information in the same way as they synchronize the kernel space information.

  Cluster Members save the states of Session Authentication and Client Authentication in kernel tables, which they synchronize.

- Cluster Members cannot synchronize the connection statutes that use system resources. The reason is the same as for the user-authenticated connections.
- Accounting information for connections is accumulated on each Cluster Member, sent to the Management Server, and aggregated.

  In the event of a cluster failover, the accounting information that is not yet sent to the Management Server, is lost.

  To minimize this risk, you can reduce the time interval when accounting information is sent.

  To do this, in the cluster object > **Logs** > **Additional Logging** pane, set a lower value for the **Update Account Log every** attribute.

- If you change the IP address of the cluster synchronization network after the cluster is already up and running, you must restart the routing daemon on all cluster members.

**Best Practice:**

- In the High Availability mode, first perform these steps on all Standby cluster members.
- In the Load Sharing Unicast mode, first perform these steps on all Non-Pivot cluster members.

1. Connect to the command line on each cluster member.
2. Log in to the Expert mode.
3. Stop the RouteD daemon:

   ```
   tellpm process:routed
   ```

4. Start the RouteD daemon:

   ```
   tellpm process:routed t
   ```

5. Log in to Gaia Clish.
6. Wait for several minutes.
7. Examine the peer neighborship in the configured dynamic routing protocols.

   See the *R81.10 Gaia Administration Guide*.

# Configuring ClusterXL

This procedure describes how to configure the Load Sharing Multicast, Load Sharing Unicast, and High Availability modes from scratch.

Their configuration is identical, apart from the mode selection in SmartConsole Cluster object or Cluster creation wizard.

## Installing Cluster Members

| Step | Instructions |
|---|---|
| 1 | See *"ClusterXL Requirements and Compatibility" on page 17*. |
| 2 | See *R81.10 Installation and Upgrade Guide*. |

## Configuring Routing for Client Computers

**Example topology:**

[internal network 10.10.2.0/24] --- (VIP 10.10.2.100/24) [Cluster] (VIP 192.168.2.100/24) --- [external network 192.168.2.0/24]

**To configure routing for client computers:**

1. Computers on the internal network 10.10.2.0/24 must be configured with Default Gateway IP 10.10.2.100
2. Computers on the external network 192.168.2.0/24 must be configured with Default Gateway IP 192.168.2.100
3. For Proxy ARP configuration, see sk30197.
4. In addition, see *"Cluster IP Addresses on Different Subnets" on page 166*.

# Configuring the Cluster Control Protocol (CCP) Settings

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

### Description

Cluster Members configure the Cluster Control Protocol (CCP) mode automatically.

**Important** - In R81.10, the CCP always runs in the unicast mode.

You can configure the Cluster Control Protocol (CCP) Encryption on the Cluster Members.

See *"Viewing the Cluster Control Protocol (CCP) Settings" on page 289*.

### Syntax for configuring the Cluster Control Protocol (CCP) Encryption

| Shell | Command |
|---|---|
| Gaia Clish | `set cluster member ccpenc {off | on}` |
| Expert mode | `cphaconf ccp_encrypt {off | on}`<br>`cphaconf ccp_encrypt_key <Key String>` |

# Configuring the Cluster Object and Members

You can use one of these procedures to define a cluster object and its members:

- **Simple Mode (Wizard)** - Lets you quickly create a new cluster and configure some basic cluster properties:
  - Cluster properties and Virtual IP addresses
  - Properties and Topology of Cluster Members
  - Synchronization interfaces and IP addresses

  See *"Using the Wizard Mode in SmartConsole" below*.
- **Classic Mode** - Opens the **Cluster Gateway Properties** window, where you manually create a cluster and configure its properties.

  See *"Using the Classic Mode in SmartConsole" on page 87*.

The **Cluster Gateway Properties** window lets you:

- Enable and configure Software Blades for the cluster
- Configure other cluster properties that you cannot configure with the wizards
- Change the properties of an existing cluster

See *"Changing the Settings of Cluster Object in SmartConsole" on page 97*.

## Using the Wizard Mode in SmartConsole

This version includes two wizards:

- Check Point Appliances and Open Servers
- Check Point Small Office Appliances

**Wizard for Check Point Appliances or Open Servers**

The **Cluster Wizard** is recommended for all Check Point Appliances (*except* models lower than 3000 series), and for Open Server platforms.

**To create a new cluster using Wizard Mode:**

| Step | Instructions |
|---|---|
| 1 | In SmartConsole, click **Objects** menu > **More object types > Network Object > Gateways and Servers > Cluster > New Cluster**. |

| Step | Instructions |
|---|---|
| 2 | In **Check Point Security Gateway Cluster Creation** window, click **Wizard Mode**. |
| 3 | In the **Cluster General Properties** window:<br>a. In the **Cluster Name** field, enter unique name for the cluster object.<br>b. In the **Cluster IPv4 Address**, enter the unique Cluster Virtual IPv4 addresses for this cluster.<br>This is the main IPv4 address of the cluster object.<br>c. In the **Cluster IPv6 Address**, enter the unique Cluster Virtual IPv6 addresses for this cluster.<br>This is the main IPv6 address of the cluster object.<br>**Important** - You must define a corresponding IPv4 address for every IPv6 address. This release does not support pure IPv6 addresses.<br>d. In the **Choose the Cluster's Solution** field, select the applicable option and click **Next**:<br>▪ **Check Point ClusterXL** and then select **High Availability** or **Load Sharing**<br>▪ **Gaia VRRP** |
| 4 | In the **Cluster member's properties** window perform these steps for *each* Cluster Member and click **Next**.<br>We assume you create a new cluster object from the scratch.<br>a. Click **Add > New Cluster Member** to configure each Cluster Member.<br>b. In the **Cluster Name** field, enter unique name for the Cluster Member object.<br>c. In the **Cluster IPv4 Address**, enter the unique Cluster Virtual IPv4 addresses for this Cluster Member.<br>This is the main IPv4 address of the Cluster Member object.<br>d. In the **Cluster IPv6 Address**, enter the unique Cluster Virtual IPv6 addresses for this Cluster Member.<br>This is the main IPv6 address of the Cluster Member object.<br>**Important** - You must define a corresponding IPv4 address for every IPv6 address. This release does not support pure IPv6 addresses.<br>e. In the **Activation Key** and **Confirm Activation Key** fields, enter a one-time password that you entered in First Time Configuration Wizard during the installation of this Cluster Member.<br>f. Click **Initialize**.<br>Management Server will try to establish SIC with each Cluster Member.<br>The **Trust State** field should show **Trust established**.<br>g. Click **OK**. |

<table>
<tr><th>Step</th><th>Instructions</th></tr>
<tr><td>5</td><td>In the Cluster Topology window, define a network type (network role) for each cluster interface and define the Cluster Virtual IP addresses. Click Next.<br>The wizard automatically calculates the subnet for each cluster network and assigns it to the applicable interface on each Cluster Member. The calculated subnet shows in the upper section of the window.<br>The available network objectives are:
<table>
<tr><th>Network Objective</th><th>Description</th></tr>
<tr><td>Cluster Interface</td><td>A cluster interface that connects to an internal or external network.<br>Enter the Cluster Virtual IP addresses for each network (internal or external).<br>In addition, see "Cluster IP Addresses on Different Subnets" on page 166.</td></tr>
</table>
</td></tr>
</table>

| Step | Instructions |
|---|---|

| Network Objective | Description |
|---|---|
| **Cluster Sync Interface** | A cluster Synchronization interface.<br>In Load Sharing mode, you must define a synchronization interface.<br>■ For Check Point Appliances or Open Servers: The Synchronization Network is supported only on the lowest VLAN tag of a VLAN interface.<br>■ For Small Office appliances (models lower than 3000 series): You can only select **1st sync** and only for the **LAN2/SYNC** interface. You cannot configure VLANs on the Synchronization interface.<br>**Important -**<br>■ Make sure that you do not define IPv6 address for Sync interfaces. The wizard does not let you define an interface with an IPv6 address as a Sync interface.<br>■ To configure two Sync interfaces in a cluster deployed in a cloud (**Sync** and **Cluster + Sync**):<br>1. Configure one group of interfaces as **Sync**.<br>2. Configure another group of interfaces as **Private**.<br>3. Complete the wizard.<br>4. Open the cluster object.<br>5. From the left tree, click the **Network Management** page.<br>6. Select the row of the interface configured as **Private** and click **Edit**.<br>7. From the left navigation tree, click the **General** page.<br>8. In the Network Type field, select **Cluster + Sync**.<br>9. Click **OK**.<br>10. Install the Access Control policy. |

| Step | Instructions |
|---|---|
| | <table><tr><th>Network Objective</th><th>Description</th></tr><tr><td>**Private**</td><td>An interface that is not part of the cluster.<br>Cluster does not monitor this interface.<br>Cluster failover does not occur if a fault occurs on this interface.<br>This option is recommended for the dedicated management interface.</td></tr></table> |
| 6 | In the **Cluster Definition Wizard Complete** window, click **Finish**. |

After you complete the wizard, we recommend that you open the cluster object and complete the configuration:

- Define Anti-Spoofing properties for each interface
- Change the Topology settings for each interface, if necessary
- Define the Network Type
- Configure other Software Blades, features and properties as necessary

### Wizard for Small Office Appliances

The **Small Office Cluster** wizard is recommended for Centrally Managed Check Point appliances - models lower than 3000 series.

**To create a new Small Office cluster using Wizard Mode:**

| Step | Instructions |
|---|---|
| 1 | In SmartConsole, click **Objects** menu > **More object types > Network Object > Gateways and Servers > Cluster > New Small Office Cluster**. |
| 2 | In **Check Point Security Gateway Cluster Creation** window, click **Wizard Mode**. |
| 3 | In the **Cluster General Properties** window:<br>a. Enter a unique name for the cluster object.<br>b. Select the correct hardware type.<br>c. Click **Next**. |

| Step | Instructions |
|---|---|
| 4 | In the **Cluster Members** window:<br>a. Enter the member name and IPv4 addresses for each Cluster Member.<br>b. Enter the one-time password for SIC trust.<br>c. Click **Next**.<br>d. Management Server will try to establish SIC with the Primary Cluster Member. |
| 5 | In the **Configure WAN Interface** page, configure the Cluster Virtual IPv4 address. |
| 6 | Define the Cluster Virtual IPv4 addresses for the other cluster interfaces. |
| 7 | Click **Next**, and then **Finish** to complete the wizard. |

After you complete the wizard, we recommend that you open the cluster object and complete the configuration:

- Define Anti-Spoofing properties for each interface
- Change the Topology settings for each interface, if necessary
- Define the Network Type
- Configure other Software Blades, features and properties as necessary

# Using the Classic Mode in SmartConsole

| Step | Instructions |
|---|---|
| 1 | Connect with SmartConsole to the Security Management Server or Domain Management Server that should manage this ClusterXL. |
| 2 | From the left navigation panel, click **Gateways & Servers**. |
| 3 | Create a new Cluster object in one of these ways:<br>■ From the top toolbar, click **New (✶) > Cluster > Cluster**.<br>■ In the top left corner, click **Objects** menu **> More object types > Network Object > Gateways and Servers > Cluster > New Cluster**.<br>■ In the top right corner, click **Objects Pane > New > More > Network Object > Gateways and Servers > Cluster > Cluster**. |
| 4 | In the **Check Point Security Gateway Creation** window, click **Classic Mode**.<br>The **Gateway Cluster Properties** window opens. |
| 5 | On the **General Properties** page > **Machine** section:<br>a. In the **Name** field, make sure you see the configured applicable name for this ClusterXL object.<br>b. In the **IPv4 Address** field, enter the unique cluster virtual IPv4 address, which is the main IPv4 address of the cluster object.<br>c. In the **IPv6 Address** field, enter the unique cluster virtual IPv6 address, which is the main IPv6 address of the cluster object.<br>ℹ **Important** - You must define a corresponding IPv4 address for every IPv6 address. This release does not support pure IPv6 addresses.<br>d. Make sure the Security Management Server or Multi-Domain Server can connect to these IP addresses. |
| 6 | On the **General Properties** page > **Platform** section, select the correct options:<br>a. In the **Hardware** field:<br>If you install the Cluster Members on Check Point Appliances, select the correct series of appliances.<br>If you install the Cluster Members on Open Servers, select **Open server**.<br>b. In the **Version** field, select **R81.10**.<br>c. In the **OS** field, select **Gaia**. |

| Step | Instructions |
|---|---|
| 7 | On the **General Properties** page:<br>a. On the **Network Security** tab, make sure the **ClusterXL** Software Blade is selected.<br>b. Enable the additional applicable Software Blades on the **Network Security** tab and on the **Threat Prevention** tab. |
| 8 | On the **Cluster Members** page:<br>a. Click **Add > New Cluster Member**.<br>The **Cluster Member Properties** window opens.<br>b. In the **Name** field, enter the applicable name for this Cluster Member object.<br>c. Configure the main physical IP address(es) for this Cluster Member object.<br>In the **IPv4 Address** and **IPv6 Address** fields, configure the same IPv4 and IPv6 addresses that you configured on the **Management Connection** page of the Cluster Member's First Time Configuration Wizard.<br>Make sure the Security Management Server or Multi-Domain Server can connect to these IP addresses.<br>**Note** - You can configure the Cluster Virtual IP address to be on a different network than the physical IP addresses of the Cluster Members. In this case, you must configure the required static routes on the Cluster Members.<br>d. Click **Communication**.<br>e. In the **One-time password** and **Confirm one-time password** fields, enter the same Activation Key you entered during the Cluster Member's First Time Configuration Wizard.<br>f. Click **Initialize**.<br>g. Click **Close**.<br>h. Click **OK**.<br>i. Repeat Steps **a-h** to add the next Cluster Member. |

| Step | Instructions |
|---|---|
| | If the **Trust State** field does not show **Trust established**, follow these steps:<br>a. Connect to the command line on the Cluster Member.<br>b. Make sure there is a physical connectivity between the Cluster Member and the Management Server (for example, pings can pass).<br>c. Run:<br>`cpconfig`<br>d. Enter the number of this option:<br>`Secure Internal Communication`<br>e. Follow the instructions on the screen to change the Activation Key.<br>f. In SmartConsole, click **Reset**.<br>g. Enter the same Activation Key you entered in the `cpconfig` menu.<br>h. In SmartConsole, click **Initialize**. |
| 9 | On the **ClusterXL and VRRP** page:<br>a. In the **Select the cluster mode and configuration** section, select the applicable mode:<br>▪ **High Availability** and **ClusterXL** (see *"High Availability and Load Sharing Modes in ClusterXL" on page 28* )<br>▪ **Load Sharing** and **Multicast** or **Unicast** (see *"High Availability and Load Sharing Modes in ClusterXL" on page 28* )<br>▪ **Active-Active** (see *"Active-Active Mode in ClusterXL" on page 47*)<br>b. In the **Tracking** section, select the applicable option.<br>c. In the **Advanced Settings** section: |

| Step | Instructions |
|---|---|
| | ■ If you selected the **High Availability** mode, then:<br>i. **Optional:** Select **Use State Synchronization**.<br>This configures the Cluster Members to synchronize the information about the connections they inspect.<br>**Best Practice** - Enable this setting to prevent connection drops after a cluster failover.<br>ii. **Optional:** Select **Start synchronizing [ ] seconds after connection initiation** and enter the applicable value.<br>This option is available only for clusters R80.20 and higher.<br>To prevent the synchronization of short-lived connections (which decreases the cluster performance), you can configure the Cluster Members to start the synchronization of **all** connections a number of seconds after they start.<br>**Range:** 2 - 60 seconds<br>**Default:** 3 seconds<br>**Notes:**<br>• This setting in the cluster object applies to all connections that pass through the cluster.<br>You can override this global cluster synchronization delay in the properties of applicable services - see *"Configuring Services to Synchronize After a Delay" on page 63* .<br>• The greater this value, the fewer short-lived connections the Cluster Members have to synchronize.<br>• The connections that the Cluster Members did not synchronize, do not survive a cluster failover.<br>**Best Practice** - Enable and configure this setting to increase the cluster performance.<br>iii. **Optional:** Select **Use Virtual MAC**.<br>This configures all Cluster Members to associate the same **virtual** MAC address with the Virtual IP address on the applicable interfaces (each Virtual IP address has its unique Virtual MAC address).<br>For more information, see sk50840. |

| Step | Instructions |
| --- | --- |
| | iv. Select the Cluster Member recovery method - which Cluster Member to select as **Active** during a fallback (return to normal operation after a cluster failover):<br>• **Maintain current active Cluster Member**<br>a. The Cluster Member that is currently in the **Active** state, remains in this state.<br>b. Other Cluster Members that return to normal operation, remain in the **Standby** state.<br>• **Switch to higher priority Cluster Member**<br>a. The Cluster Member that has the highest priority (appears at the top of the list on the **Cluster Members** page of the cluster object) becomes the new **Active**.<br>b. The state of the previously **Active** Cluster Member changes to **Standby**.<br>c. Other Cluster Members that return to normal operation remain in the **Standby** state. |

| Step | Instructions |
|---|---|
| | ■ If you selected the **Load Sharing > Multicast** mode, then:<br>i. **Optional:** Select **Use Sticky Decision Function**.<br>This option is available only for clusters R80.10 and lower.<br>For more information, click the **(?)** button in the top right corner.<br>ii. **Optional:** Select **Start synchronizing [ ] seconds after connection initiation** and enter the applicable value.<br>This option is available only for clusters R80.20 and higher.<br>To prevent the synchronization of short-lived connections (which decreases the cluster performance), you can configure the Cluster Members to start the synchronization of **all** connections a number of seconds after they start.<br>**Range:** 2 - 60 seconds<br>**Default:** 3 seconds<br>**Notes:**<br>• This setting in the cluster object applies to all connections that pass through the cluster.<br>You can override this global cluster synchronization delay in the properties of applicable services - see *"Configuring Services to Synchronize After a Delay" on page 63* .<br>• The greater this value, the fewer short-lived connections the Cluster Members have to synchronize.<br>• The connections that the Cluster Members did not synchronize, do not survive a cluster failover.<br>**Best Practice** - Enable and configure this setting to increase the cluster performance.<br>iii. Select the connection sharing method between the Cluster Members:<br>• **IPs, Ports, SPIs**<br>Configures each Cluster Member to inspect all connections with the same Source and Destination IP address, the same Source and Destination ports, and the same IPsec SPI numbers.<br>This is the least "sticky" sharing configuration that provides the best sharing distribution between Cluster Members.<br>This method decreases the probability that a certain connection passes through the same Cluster Member in both inbound and outbound directions<br>We recommend this method.<br>• **IPs, Ports**<br>Configures each Cluster Member to inspect all connections with the same Source and Destination IP address and the same Source and Destination ports, regardless of the IPsec SPI numbers.<br>Use this method only if there are problems when distributing IPsec packets between Cluster Members.<br>• **IPs**<br>Configures each Cluster Member to inspect all connections with the same Source and Destination IP address, regardless of the Source and Destination ports and IPsec SPI numbers.<br>This is the most "sticky" sharing configuration that provides the worst sharing distribution between Cluster Members. |

| Step | Instructions |
|---|---|
| | ■ If you selected the **Load Sharing > Unicast** mode, then:<br>i. **Optional:** Select **Use Sticky Decision Function**.<br>This option is available only for clusters R80.10 and lower.<br>For more information, click the **(?)** button in the top right corner.<br>ii. **Optional:** Select **Start synchronizing [ ] seconds after connection initiation** and enter the applicable value.<br>This option is available only for clusters R80.20 and higher.<br>To prevent the synchronization of short-lived connections (which decreases the cluster performance), you can configure the Cluster Members to start the synchronization of **all** connections a number of seconds after they start.<br>**Range:** 2 - 60 seconds<br>**Default:** 3 seconds<br>**Notes:**<br>• This setting in the cluster object applies to all connections that pass through the cluster.<br>You can override this global cluster synchronization delay in the properties of applicable services - see *"Configuring Services to Synchronize After a Delay" on page 63* .<br>• The greater this value, the fewer short-lived connections the Cluster Members have to synchronize.<br>• The connections that the Cluster Members did not synchronize, do not survive a cluster failover.<br>**Best Practice** - Enable and configure this setting to increase the cluster performance.<br>iii. **Optional:** Select **Use Virtual MAC**.<br>This configures all Cluster Members to associate the same **virtual** MAC address with the Virtual IP address on the applicable interfaces (each Virtual IP address has its unique Virtual MAC address).<br>For more information, see sk50840. |

| Step | Instructions |
| --- | --- |
| | iv. Select the connection sharing method between the Cluster Members:<br>• **IPs, Ports, SPIs**<br>Configures each Cluster Member to inspect all connections with the same Source and Destination IP address, the same Source and Destination ports, and the same IPsec SPI numbers.<br>This is the least "sticky" sharing configuration that provides the best sharing distribution between Cluster Members.<br>This method decreases the probability that a certain connection passes through the same Cluster Member in both inbound and outbound directions<br>We recommend this method.<br>• **IPs, Ports**<br>Configures each Cluster Member to inspect all connections with the same Source and Destination IP address and the same Source and Destination ports, regardless of the IPsec SPI numbers.<br>Use this method only if there are problems when distributing IPsec packets between Cluster Members.<br>• **IPs**<br>Configures each Cluster Member to inspect all connections with the same Source and Destination IP address, regardless of the Source and Destination ports and IPsec SPI numbers.<br>This is the most "sticky" sharing configuration that provides the worst sharing distribution between Cluster Members.<br>This method increases the probability that a certain connection passes through the same Cluster Member in both inbound and outbound directions<br>Use this method only if there are problems when distributing packets with different port numbers or distributing IPsec packets between Cluster Members. |

| Step | Instructions |
|---|---|
| 10 | On the **Network Management** page:<br>a. Select each interface and click **Edit**. The **Network: <Name of Interface>** window opens.<br>b. From the left tree, click the **General** page.<br>c. In the **General** section, in the **Network Type** field, select the applicable type:<br>■ For *cluster traffic interfaces*, select **Cluster**.<br>Make sure the Cluster Virtual IPv4 address and its Net Mask are correct.<br>■ For *cluster synchronization interfaces*, select **Sync** or **Cluster+Sync**.<br>**Notes:**<br>• We do not recommend the configuration **Cluster+Sync**.<br>• Check Point cluster supports only these settings:<br>◦ One **Sync** interface.<br>◦ One **Cluster+Sync** interface.<br>◦ One **Sync** interface and one **Cluster+Sync** interface.<br>• For Check Point Appliances or Open Servers:<br>The Synchronization Network is supported only on the lowest VLAN tag of a VLAN interface.<br>■ For *interfaces that do not pass the traffic* between the connected networks, select **Private**.<br>d. In the **Member IPs** section, make sure the IPv4 address and its Net Mask are correct on each Cluster Member.<br>**Notes:**<br>■ For a ClusterXL in High Availability mode that is deployed in a Cloud environment (Geo Cluster):<br>You can configure IP addresses that belong to different networks on *cluster synchronization interfaces* and on *cluster traffic interfaces*.<br>■ For *cluster traffic interfaces*, you can configure the Cluster Virtual IP address to be on a different network than the physical IP addresses of the Cluster Members. In this case, you must configure the required static routes on the Cluster Members. See *"Cluster IP Addresses on Different Subnets" on page 166*.<br>e. In the **Topology** section:<br>■ Make sure the settings are correct in the **Leads To** and **Security Zone** fields.<br>Only these options are supported on cluster interfaces (Known Limitation PMTR-70260):<br>• **Override** > **Network defined by routes** (this is the default).<br>• **Override** > **Specific** > select the applicable **Network** object or **Network Group** object. |

| Step | Instructions |
|---|---|
| | ■ To increase the security, enable the **Anti-Spoofing**. |
| 11 | Click **OK**. |
| 12 | Publish the SmartConsole session. |

# Changing the Settings of Cluster Object in SmartConsole

The **Cluster Gateway Properties** window in a cluster object contains many different ClusterXL properties, as well as other properties related to Security Gateway and Software Blades functionality.

This section includes only the properties and procedures directly related to ClusterXL.

## Configuring General Properties

1. In the **Name** field, enter a unique name for this cluster object.
2. In the **IPv4 Address** field, enter the unique Cluster Virtual IPv4 addresses for this cluster. This is the main IPv4 address of the cluster object.
3. In the **Cluster IPv6 Address** field, enter the unique Cluster Virtual IPv6 addresses for this cluster. This is the main IPv6 address of the cluster object.

   **Important** - You must define a corresponding IPv4 address for every IPv6 address. This release does not support pure IPv6 addresses.
4. In the **Hardware** field, select the correct hardware platform.
5. In the **Version** field, select the correct Check Point version.
6. In the **OS** field, select the correct operating system.
7. Configure the applicable cluster type:
   - To work with **ClusterXL** or with **VRRP on Gaia**, select **ClusterXL**.

     Go to the **ClusterXL and VRRP** pane and configure the applicable settings.
   - To work with any other cluster mode, clear **ClusterXL**.

     Go to the **3rd Party Configuration** pane and configure the applicable settings.
8. On the **Network Security** tab, enable other Software Blades as necessary.
9. Click **OK**.
10. Publish the SmartConsole session.

## Working with Cluster Topology

### IPv6 Considerations

To activate IPv6 functionality for an interface, define an IPv6 address for the applicable interface on each Cluster Member and in the cluster object. All interfaces configured with an IPv6 address must also have a corresponding IPv4 address. If an interface does not require IPv6, only the IPv4 definition address is necessary.

**Note** - You must configure synchronization interfaces with IPv4 addresses only. This is because the synchronization mechanism works using IPv4 only. All IPv6 information and states are synchronized using this interface.

1. Connect with SmartConsole to the Security Management Server or Domain Management Server that manages this cluster.
2. From the left navigation panel, click **Gateways & Servers**.
3. Open the cluster object.
4. From the left tree, click the **Network Management** page.
5. Select a cluster interface and click **Edit**.
6. From the left navigation tree, click **General** page:
   a. In the **General** section, configure these settings for Cluster Virtual Interface:
      - **Network Type** - one of these: **Cluster**, **Sync**, **Cluster + Sync**, **Private**

        The available network types (network objectives) are:

| Network Type | Description |
|---|---|
| **Cluster** | An interface that connects to an internal or external network. |
| **Cluster + Sync** | A cluster interface that also works as a Synchronization interface.<br>We do not recommend this configuration because it adds the Delta Sync traffic to the interface. |
| **Sync** | An interface used exclusively for cluster state synchronization. |
| **Private** | An interface that is not part of the cluster. ClusterXL does not monitor the state of this interface. As a result, there is no cluster failover if a fault occurs with this interface. This option is recommended for the management interface. |

      - **Virtual IPv4** - Virtual IPv4 address assigned to this Cluster Virtual Interface
      - **Virtual IPv6** - Virtual IPv6 address assigned to this Cluster Virtual Interface

      **Important** - You must define a corresponding IPv4 address for every IPv6 address. This release does not support the configuration of only IPv6 addresses.
7. In the **Member IPs** section, click **Modify** and configure these settings:

- Physical IPv4 address and Mask Length assigned to the applicable physical interface on each Cluster Member
- Physical IPv6 address and Mask Length assigned to the applicable physical interface on each Cluster Member

**Important** - You must define a corresponding IPv4 address for every IPv6 address. This release does not support the configuration of only IPv6 addresses.

In addition, see *"Cluster IP Addresses on Different Subnets" on page 166*.

8. In the **Topology** section, click **Modify** and configure these settings:
   - **Leads To** - one of these: **Internet (External)**, **This Network (Internal)**
   - **Security Zone** - one of these: **User defined**, **According to topology** (`ExternalZone`, `InternalZone`)
   - **Anti-Spoofing** - whether to perform the Anti-Spoofing, and how to do it (`Detect`, `Prevent`)
9. From the left navigation tree, click **QoS** page:
   a. In the **Bandwidth** section, configure these settings:
      - **Inbound Active** - rate limit for inbound traffic
      - **Outbound Active** - rate limit for outbound traffic
   b. In the **DiffServ and Low Latency classes** section, configure the applicable classes.
10. From the left navigation tree, click **Advanced** page:
    a. In the **Multicast Restrictions** section, configure the applicable settings for dropping multicast packets
    b. In the **Interfaces Names** section, configure the names of applicable interfaces
11. Click **OK**.
12. Publish the SmartConsole session.
13. Install the Access Control Policy on this cluster object.

## Changing the Synchronization Interface

**Important** - Schedule a maintenance window, because changing the synchronization interface can impact the traffic.

**To change the IPv4 address on the synchronization interface on Cluster Members:**

1. On each Cluster Member, change the IPv4 address on the Sync interface.

   Use Gaia Portal, or Gaia Clish.

   See the *R81.10 Gaia Administration Guide*.
2. Connect with SmartConsole to the Security Management Server or Domain Management Server that manages this cluster.
3. From the left navigation panel, click **Gateways & Servers**.
4. Open the cluster object.
5. In the **Gateway Cluster Properties** window, click **Network Management** page.
6. Click **Get Interfaces > Get Interfaces With Topology**.
7. Make sure the settings are correct.
8. Select the **Sync** interface and click **Edit**.
9. From the left navigation tree, click **General** page.
10. In the **General** section, in the **Network Type** field, select **Sync**.
11. Click **OK**.
12. Publish the SmartConsole session.
13. Install the Access Control Policy on this cluster object.

**To change the synchronization interface on Cluster Members to a new interface:**

1. On each Cluster Member:
   a. Configure a new interface that you will use as a new **Sync** interface.
   b. Delete the IPv4 address from the old **Sync** interface.

   Use Gaia Portal, or Gaia Clish.

   See the *R81.10 Gaia Administration Guide*.
2. Connect with SmartConsole to the Security Management Server or Domain Management Server that manages this cluster.
3. From the left navigation panel, click **Gateways & Servers**.
4. Open the cluster object.
5. In the **Gateway Cluster Properties** window, click **Network Management** page.
6. Click **Get Interfaces > Get Interfaces With Topology**.

7. Make sure the settings are correct.
8. Right-click on the old **Sync** interface and click **Delete Interface**.
9. Select the new interface and click **Edit**.
10. From the left navigation tree, click **General** page.
11. In the **General** section, in the **Network Type** field, select **Sync**.
12. Click **OK**.
13. In SmartConsole, install the Access Control Policy on this cluster object.
14. Publish the SmartConsole session.
15. Install the Access Control Policy on this cluster object.

## Adding Another Member to an Existing Cluster

See *"Adding Another Member to an Existing Cluster" on page 176*.

## Removing a Member from an Existing Cluster

See *"Removing a Member from an Existing Cluster" on page 183*.

# Configuring a ClusterXL in Bridge Mode

See the *R81.10 Installation and Upgrade Guide* - Chapter *Special Scenarios for Security Gateways* - Section *Deploying a Security Gateway or a ClusterXL in Bridge Mode*.

# Advanced Features and Procedures

This section describes advanced configuration in cluster.

# Configuring Virtual MAC (VMAC)

## Virtual MAC

### Overview

Most switches learn information about hosts connected to the network from ARP Requests and ARP Replies.

ClusterXL relies on this behavior and on its cluster interfaces it sends Gratuitous ARP Requests (G-ARP updates) to the connected networks so network devices (switches, hosts, routers) update their ARP tables with the applicable entries - the Cluster Virtual IP (VIP) and its corresponding MAC address.

**Note** - Cluster Members send these G-ARP every "N" seconds. This timeout is controlled by the value (in HTUs - HA Time Units) of the global kernel parameter `fwha_gratuitous_arp_timeout`.

#### G-ARP behavior in a Check Point cluster:

- In a ClusterXL High Availability mode:

  The current Active Cluster Member sends G-ARP updates for its Cluster Virtual IP (VIP) addresses and their corresponding MAC addresses - the real MAC addresses of the cluster interfaces on the Active Cluster Member.

  When a failover occurs, the new Active Cluster Member sends its G-ARP updates for its Cluster Virtual IP (VIP) addresses and their corresponding MAC addresses - the real MAC addresses of the cluster interfaces on the new Active Cluster Member.

- In a ClusterXL Load Sharing Unicast mode:

  The current Pivot Cluster Member sends G-ARP updates for its Cluster Virtual IP (VIP) addresses and their corresponding MAC addresses - the real MAC addresses of the cluster interfaces on the Pivot Cluster Member.

  When a failover occurs, the new Pivot Cluster Member sends its G-ARP updates for its Cluster Virtual IP (VIP) addresses and their corresponding MAC addresses - the real MAC addresses of the cluster interfaces on the new Pivot Cluster Member.

### G-ARP behavior on network switches and hosts:

Some Layer 3 switches do not integrate the information from G-ARP updates fast enough into their ARP cache table. As a result, such a switch continues to send traffic to the real MAC address of the previous Active member / Pivot member that failed. This causes a traffic outage on the network until the switch updates its ARP cache table.

When there is a large number of Static NAT entries on a Cluster Member, upon failover the ClusterXL mechanism sends many G-ARP updates. In some cases, the number of G-ARP updates is more than a switch can handle. As a result, some traffic is lost

In some other cases, different network devices (for example, VoIP phones) categorically ignore these G-ARP updates.

### Check Point solution:

To work with such switches, you can configure Cluster Members to use Virtual MAC (VMAC) addresses in G-ARP updates on the cluster interfaces instead of the real MAC addresses. In this scenario, such "smart" switches do need to update their ARP cache tables.

When you enable the Virtual MAC (VMAC) mode in a cluster object, all Cluster Members start to send their G-ARP updates for their Cluster Virtual IP (VIP) addresses with a Virtual MAC (VMAC) address. Cluster Members create a VMAC address for each VIP address. For details, see *"VMAC Value" on page 106*.

Example:

- Member_A has the interface `eth0` (with a real MAC "`A`") in the subnet 192.168.3.x/24
- Member_B has the interface `eth0` (with a real MAC "`B`") in the subnet 192.168.3.x/24
- Cluster VIP in this subnet is 192.168.3.1
- When you enable the VMAC mode, all Cluster Members start to send G-ARP packets for the VIP 192.168.3.1 and the **same** Source Virtual MAC address created for the cluster interfaces `eth0`.

A Cluster VMAC configuration makes G-ARP updates for NATed IP addresses unnecessary during a cluster failover. This decreases the likelihood of a traffic outage during the cluster failover.

**Source addresses for traffic originating from an Active / Pivot cluster member:**

**Important** - VMAC addresses apply only to cluster G-ARP updates. When Cluster Members forward traffic, they continue to use their real MAC addresses.

| Address | Value |
|---|---|
| Layer 2 Source (MAC) address | Real MAC address of corresponding Cluster Member's interface |
| Layer 3 Source (IP) address | Virtual IP address that was configured for corresponding cluster interfaces. |

**Source addresses for traffic originating from hosts on the network toward Cluster Members:**

| Address | Value |
|---|---|
| Layer 2 Source (MAC) address | Real MAC address of corresponding host's interface. |
| Layer 2 Destination (MAC) address | Virtual MAC address created for corresponding cluster interfaces. |
| Layer 3 Source (IP) address | IP address of corresponding host's interface. |
| Layer 3 Destination (IP) address | Virtual IP address that was configured for corresponding cluster interfaces. |

### Known Limitations

- VMAC mode is supported in ClusterXL only in the High Availability mode and in the Load Sharing Unicast mode.
- VMAC mode is supported in a VSX Cluster - in High Availability (default mode) and in Virtual System Load Sharing (VSLS) mode.
- In Gaia Clish, the "`show interface`" command does not show the VMAC addresses. Use "`show cluster members interfaces virtual`" command.
- In the Expert mode, the "`ifconfig`" command does not show the VMAC addresses. Use "`cphaprob -a if`" command.

### Configuration

**Best Practices:**

- We recommend to schedule a maintenance window.
- We recommend to enable the feature when all cluster members are in the state "UP".

You configure the VMAC mode in SmartConsole.

You can control the VMAC mode in the command line on each Cluster Member.

#### Configuration in SmartConsole:

1. In SmartConsole, open the cluster object.
2. From the left tree, select **ClusterXL and VRRP**.
3. Below **Select the cluster mode and configuration:** select **High Availability**.
4. In the drop-down menu to the right of **High Availability**, select **ClusterXL**.
5. Below **Advanced Settings**, select **Use Virtual MAC**.
6. Click **OK**.
7. Install the Access Control Policy on the cluster object.

#### Configuration in the command line of each Cluster Member (alternative method):

**Note** - To get the current value of the kernel parameter "`fwha_vmac_global_param_enabled`", run on each Cluster Member:

```
fw ctl get int fwha_vmac_global_param_enabled
```

1. To **disable** VMAC mode temporarily (does not survive reboot), on each Cluster Member configure the value of the kernel parameter "`fwha_vmac_global_param_enabled`" to `0`:

   ```
   fw ctl set int fwha_vmac_global_param_enabled 0
   ```

2. To **enable** the VMAC mode again, on each Cluster Member configure the value of the kernel parameter "`fwha_vmac_global_param_enabled`" to `1` (default value is `0`):

   ```
   fw ctl set int fwha_vmac_global_param_enabled 1
   ```

### VMAC Value

Total 48 bits (from left-to-right):

| Bits | Description | Value |
|---|---|---|
| First 24 bit | Unique constant value. | *00:1C:7F* |
| Next 8 bits | VSX Virtual System ID. | ■ In a VSX Cluster: Virtual System ID<br>■ In a non-VSX Cluster 000000000 |
| Last 16 bits | Unique value that the Management Server assigns to each cluster object.<br>This makes the VMAC value unique for each managed cluster. | Unique value for each cluster |

**Monitoring VMAC**

1. Connect to the command line on each Cluster Member.
2. To see the VMAC addresses, run this command and see the "`VMAC address`" values:
   - In Gaia Clish:

     ```
     show cluster members interfaces virtual
     ```

   - In the Expert mode:

     ```
     cphaprob -a if
     ```

3. Make sure that the ARP tables of the hosts connected to these Cluster Members now contain the VMAC value.

   Use the applicable command for the Operating System on the hosts.

   For example, on Linux, run "`arp -an`". On Windows, run "`arp-a`".

# Same VMAC

**Important** - This feature is available in R81.10 Jumbo Hotfix Accumulator Take 75 and higher.

### Overview

VMAC mode applies only to Gratuitous ARP packets. For more information about VMAC, see *"Virtual MAC" on page 103*.

When Cluster Members forward traffic, they continue to use their real MAC addresses.

Some Layer 3 switches learn about the hosts from the source IP addresses and source MAC addresses of the traffic that arrives at these switches.

If an IP-to-MAC address association on such a "smart" Layer 3 switch changes frequently, the switch may consider such an IP address as "misbehaving" or "rogue". Security features on such a "smart" switch can disable ARP table updates and freeze the current IP-to-MAC address association.

As a result, a traffic outage can occur because Cluster Members send traffic from the same source IP addresses, but with different source MAC addresses (VMAC in G-ARP and real MAC in regular traffic).

#### Check Point solution:

To work with such switches, you can configure Cluster Members to use Virtual MAC (VMAC) addresses on the cluster interfaces instead of the real MAC addresses.

Cluster interfaces that belong to the same subnet get the same VMAC address instead of their real MAC address. In this scenario, such "smart" switches always detect traffic from the same source IP addresses and the same source MAC addresses.

Example:

- eth1 - cluster interface on Member_1 has a VMAC address
- eth1 - cluster interface on Member_2 has the same VMAC address

### Known Limitations

- Only ClusterXL High Availability and VSX VSLS modes are supported.
- In the ClusterXL High Availability object, you must enable and configure the state synchronization:

1. On the **ClusterXL and VRRP** page, select **Use State Synchronization**.
2. On the **Network Management** page, you must configure a Sync interface.

   **Best Practice** - Configure a dedicated Sync interface on each Cluster Member, and in the interface properties, in the **Network Type** field, select **Sync**.

- Cluster Members monitor their cluster interfaces only with the Link Monitoring (and not with Cluster Control Protocol (CCP) packets).
- Physical cluster interfaces that have VLAN interfaces configured, must **not** have a Virtual IP Address:

  a. Open the Cluster Member object.

  b. In the left navigation tree, click **Network Management**.

  c. In the table, double-click applicable physical cluster interface for **each** Cluster Member.

     The **Network: <*Interface Name*>** window opens on the page **General**.

  d. In the **General** section, in the **Network Type** drop-down menu, select **Private**.

  e. Click **OK** to close the **Network** window.

  f. Click **OK** to close the **Gateway Cluster Properties** window.

  Example:

  - eth1 - physical interface with VLANs, must **not** have a VIP address
  - eth1.2 - VLAN interface, VIP is supported
  - eth1.3 - VLAN interface, VIP is supported
- The Management interface of Cluster Members does not use a VMAC address (continues to use only its real MAC address).
- To connect to Standby Cluster Members (for example an SSH connection from a Host), you must connect to their Management interfaces or Sync interfaces.
- Local connections from Standby Cluster Members to an IPv6 Host (for example, to an FTP server) are **not** supported.

### Configuring "Same VMAC" and "Silent Standby"

**Best Practices:**

- We recommend to schedule a maintenance window.
- We recommend to enable the feature when all cluster members are in the state "UP".

**Note** - A reboot is not required after you enable or disable the "Same VMAC" (and the "Silent Standby") features.

**To enable "Same VMAC" and "Silent Standby":**

1. Connect to the command line on each Cluster Member.
2. Log in to the Expert mode.
3. Enable the "Same VMAC" feature:

   ```
   fw ctl set -f int fwha_alter_vmac_param 1
   ```

4. Enable the "Silent Standby" feature:

   ```
   fw ctl set -f int fwha_silent_standby_mode 1
   ```

   **Important** - You must enable the "Silent Standby" feature if it is necessary to connect from Standby cluster members to a host / server on the network. The "Silent Standby" feature configures the Standby cluster member to communicate only through the Active cluster member (for more information, see sk169154 - section "(3.4) Standby member's connections").

5. Enable Link Monitoring of all cluster interfaces:
   a. Look for the required configuration file:

      ```
      ls -l $FWDIR/conf/cpha_link_monitoring.conf
      ```

      - If this file already exists, back it up:

        ```
        cp -v $FWDIR/conf/cpha_link_monitoring.conf{,_BKP}
        ```

      - If this file does not exist, create it:

        ```
        touch $FWDIR/conf/cpha_link_monitoring.conf
        ```

   b. Configure only the value "`all`" in the configuration file:

      ```
      echo all > $FWDIR/conf/cpha_link_monitoring.conf
      ```

6. Enable the VMAC on all cluster members:
   a. In SmartConsole, open the cluster object.
   b. From the left tree, select **ClusterXL and VRRP**.
   c. Below **Select the cluster mode and configuration:** select **High Availability**.

d. In the drop-down menu to the right of **High Availability**, select **ClusterXL**.

e. Below **Advanced Settings**, select **Use Virtual MAC**.

f. Click **OK**.

g. Install the Access Control Policy on the cluster object.

**To disable "Same VMAC" and "Silent Standby":**

1. Connect to the command line on each Cluster Member.
2. Log in to the Expert mode.
3. Disable the "Same VMAC" feature:

```
fw ctl set -f int fwha_alter_vmac_param 0
```

4. Disable the "Silent Standby" feature:

```
fw ctl set -f int fwha_silent_standby_mode 0
```

5. Disable Link Monitoring of all cluster interfaces:

```
echo '' > $FWDIR/conf/cpha_link_monitoring.conf
```

6. Disable the VMAC on all cluster members:

a. In SmartConsole, open the cluster object.

b. From the left tree, select **ClusterXL and VRRP**.

c. Below **Advanced Settings**, clear **Use Virtual MAC**.

d. Click **OK**.

e. Install the Access Control Policy on the cluster object.

**Monitoring "Same VMAC"**

1. Connect to the command line on the Cluster Member.
2. To see the VMAC addresses, run a command and see the "`VMAC address`" values:

- In Gaia Clish:

```
show cluster members interfaces virtual
```

- In the Expert mode:

```
cphaprob -a if
```

3. Make sure that the ARP tables of the hosts connected to these cluster members now contain the VMAC value.

   Use the applicable command for the Operating System on the hosts.

   For example, on Linux, run "`arp -an`". On Windows, run "`arp -a`".

4. To see the Same VMAC addresses, run the command and see the MAC addresses on the cluster interfaces:

   - In Gaia Clish:

```
show interfaces all
```

```
show interface <Name of Physical Interface><SPACE><TAB>
```

   - In the Expert mode:

```
ifconfig
```

# Working with VPN in Cluster

This section describes the configuration of VPN in cluster.

## Configuring VPN in Clusters

Configuring a cluster using SmartConsole is very similar to configuring a single Security Gateway.

All attributes of the VPN are defined in the Cluster object, except for two attributes that are defined for each Cluster Member object.

1. In SmartConsole, open the cluster object.
2. In the left navigation tree, go to **Cluster Members** page.
3. Select each Cluster Member and click **Edit**.

   The **Cluster Member Properties** window opens.
4. Go the **VPN** tab:
   - In the **Office Mode for Remote access** section:

     If you wish to use Office Mode for Remote Access, select **Offer Manual Office Mode** and define the IP pool allocated to each Cluster Member.
   - In the **Certificate List with keys stored on the Security Gateway** section:

     If your Cluster Member supports hardware storage for IKE certificates, define the certificate properties.

     In that case, Management Server directs the Cluster Member to create the keys and supply only the required material for creation of the certificate request.

     The certificate is downloaded to the Cluster Member during policy installation.
5. Click **OK** to close the **Cluster Member Properties** window.
6. In the left navigation tree, go to **ClusterXL and VRRP** page.
7. Make sure to select **Use State Synchronization**.

   This is required to synchronize IKE keys.
8. In the left navigation tree, go to **Network Management > VPN Domain** page.
9. Define the encryption domain of the cluster.

   Select one of the two possible settings:

- **All IP addresses behind Cluster Members based on Topology information**. This is the default option.
- **Manually defined**. Use this option if the cluster IP address is not on the member network, in other words, if the cluster virtual IP address is on a different subnet than the Cluster Member interfaces. In that case, select a network or group of networks, which must include the virtual IP address of the cluster, and the network or group of networks behind the cluster.

10. Click **OK** to close the **Gateway Cluster Properties** window.
11. Install the Access Control Policy on the cluster.

# Defining VPN Peer Clusters with Separate Management Servers

When working with a VPN peer that is a Check Point Cluster, and the VPN peer is managed by a different Management Server, do NOT define another cluster object. Instead, do the following:

1. In SmartConsole, go to **Objects** menu > **More object types > Network Object > Gateways and Servers > More > New Externally Managed VPN Gateway**.

   The **Externally Managed Check Point Gateway** window opens.
2. In the **General Properties** page, configure the name and the IP address.
3. In the **Topology** page, click **New** to add the external and internal *cluster* interfaces on the VPN peer.
4. In the **VPN Domain** section of the **Topology** page, define the encryption domain of the externally managed Security Gateway to be behind the internal Virtual IP address of the Security Gateway.

   If the encryption domain is just one subnet, select **All IP addresses behind Gateway based on Topology information**.

   If the encryption domain includes more than one subnet, select **Manually defined**.
5. Click **OK**.
6. Install the Access Control Policy on the cluster.

# Working with NAT in Cluster

This section describes the configuration of NAT in cluster.

## Cluster Fold and Cluster Hide

Network Address Translation (NAT) is a fundamental aspect of the way ClusterXL works.

- When a Cluster Member establishes an *outgoing* connection towards the Internet, the source address in the outgoing packets, is the physical IP address of the Cluster Member interface.

  The source IP address is changed using NAT to that of the external Virtual IP address of the cluster.

  This address translation is called "Cluster Hide".

  - When working with **VRRP on Gaia** cluster, this corresponds to the default setting in the **ClusterXL and VRRP** page of the cluster object of **Hide Cluster Members outgoing traffic behind the Cluster IP address** being selected.
  - When working with **VRRP on IPSO** cluster, this corresponds to the default setting in the **3rd Party Configuration** page of the cluster object of **Hide Cluster Members' outgoing traffic behind the Cluster's IP address** being selected.
- When a client establishes an *incoming* connection to external (virtual) address of the cluster, ClusterXL changes the destination IP address using NAT to that of the physical external address of one of the Cluster Members. This address translation is called "Cluster Fold".
  - When working with **VRRP on Gaia** cluster, this corresponds to the default setting in the **ClusterXL and VRRP** page of the cluster object of **Forward Cluster incoming traffic to Cluster Members IP address** being selected.
  - When working with **IPSO IP Clustering** cluster, this corresponds to the default setting in the **3rd Party Configuration** page of the cluster object of **Forward Cluster incoming traffic to Cluster Members' IP addresses** being selected.

## Configuring NAT in Cluster

Network Address Translation (NAT) can be performed on a Cluster, in the same way as it is performed on a Security Gateway.

This NAT is in addition to the automatic "Cluster Fold" and "Cluster Hide" address translations.

To configure NAT, edit the Cluster object, and in the **Cluster Properties** window, click the **NAT** page. Do NOT configure the **NAT** tab of the Cluster Member object.

# Configuring NAT on a Cluster Member

It is possible to perform Network Address Translation (NAT) on a non-cluster interface of a Cluster Member.

A possible scenario for this is if the non-Cluster interface of the Cluster Member is connected to another (non-cluster) internal Security Gateway, and you wish to hide the address of the non-Cluster interface of the Cluster Member.

Performing this NAT means that when a packet originates behind or on the non-Cluster interface of the Cluster Member, and is sent to a host on the other side of the internal Security Gateway, the source address of the packet will be translated.

**To configure NAT on a non-cluster interface of a Cluster Member:**

1. Edit the Cluster object.
2. In the **Cluster Member** page, edit the Cluster Member object.
3. In the **Cluster Member Properties** window, click the **NAT** tab.
4. Configure Static or Hide NAT as applicable.
5. Install the Access Control Policy on the cluster.

# Working with VLANs in Cluster

A VLAN switch tags packets that originate in a VLAN with a four-byte header that specifies, which switch port it came from

No packet is allowed to go from a switch port in one VLAN to a switch port in another VLAN, apart from ports ("global" ports) that are defined so that they belong to all the VLANs.

The Cluster Member is connected to the global port of the VLAN switch, and this logically divides a single physical port into many VLAN ports each associated with a VLAN tagged interface (VLAN interface) on the Cluster Member.

When defining VLAN tags on an interface, cluster IP addresses can be defined only on the VLAN interfaces (the tagged interfaces).

Defining a cluster IP address on a physical interface that has VLANs is not supported.

This physical interface has to be defined with the Network Type **Private**.

ClusterXL (including VSX) supports the Synchronization Network (CCP packets that carry Delta Sync information) only on the lowest VLAN ID (VLAN tag).

For example, if three VLANs with IDs `10`, `20` and `30` are configured on interface `eth1`, then you can use only the VLAN interface `eth1.10` for the State Synchronization.

This is the default interface monitoring in Check Point cluster:

| Interface type | Monitoring in ClusterXL (non-VSX) | Monitoring in VSX Cluster |
|---|---|---|
| Physical interfaces | Monitors all cluster interfaces. | Monitors all cluster interfaces. |
| VLAN interfaces | Monitors only lowest VLAN ID configured on a physical interface. | VSX High Availability (non-VSLS):<br>■ Monitors only lowest and highest VLAN IDs configured on a physical interface.<br>■ Monitors only lowest VLAN ID, if both VLAN IDs reside on the same Virtual System. |
| | Monitors only lowest and highest VLAN IDs configured on a physical interface. | Virtual System Load Sharing:<br>■ Monitors all VLAN IDs configured on a physical interface on each Virtual System.<br>■ When a Virtual System is connected to a Virtual Switch with the same physical interface and a lower VLAN ID, the `wrp` interface that leads to the Virtual Switch is considered the lowest VLAN ID for the physical interface. |

You can customize the default monitoring of VLAN IDs:

| Need to monitor VLAN | Monitoring in ClusterXL (non-VSX) | Monitoring in VSX Cluster |
|---|---|---|
| Only the lowest VLAN ID | Enabled by default. | Must disable the monitoring of all VLAN IDs - set the value of the kernel parameter `fwha_monitor_all_vlan` to 0.<br>See sk92826. |
| Only the lowest and highest VLAN IDs | Enabled by default.<br>Controlled by the kernel parameter `fwha_monitor_low_high_vlans`.<br>See sk92826. | VSX High Availability (non-VSLS): Enabled by default.<br>Controlled by the kernel parameter `fwha_monitor_low_high_vlans`.<br>See sk92826. |
| All VLAN IDs | Disabled by default.<br>Controlled by the kernel parameter `fwha_monitor_all_vlan`.<br>See sk92826. | Virtual System Load Sharing: Disabled by default.<br>Controlled by the kernel parameter `fwha_monitor_all_vlan`.<br>See sk92826. |
| Only specific VLAN IDs | Disabled by default.<br>Controlled by the kernel parameter `fwha_monitor_specific_vlan`.<br>See sk92784. | Disabled by default.<br>Controlled by the kernel parameter `fwha_monitor_specific_vlan`.<br>See sk92784. |

# Configuring Link Monitoring on the Cluster Interfaces

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

### Description

This procedure configures the Cluster Member to monitor only the physical link on the cluster interfaces (instead of monitoring the Cluster Control Protocol (CCP) packets):

- If a link disappears on the configured interface, the Cluster Member changes the interface's state to **DOWN**.

  This causes the Cluster Member to change its state to **DOWN**.

- If a link appears again on the configured interface, the Cluster Member changes the interface's state back to **UP**.

  This causes the Cluster Member to change its state back to **ACTIVE** or **STANDBY**.

See *"Viewing Cluster State" on page 240*.

### Procedure

| Step | Instructions |
|---|---|
| 1 | Connect to the command line on the Cluster Member. |
| 2 | Log in to the Expert mode. |
| 3 | See if the `$FWDIR/conf/cpha_link_monitoring.conf` file already exists:<br>`stat $FWDIR/conf/cpha_link_monitoring.conf` |
| 4 | If the `$FWDIR/conf/cpha_link_monitoring.conf` file already exists, create a backup copy:<br>`cp -v $FWDIR/conf/cpha_link_monitoring.conf{,_BKP}`<br>If the `$FWDIR/conf/cpha_link_monitoring.conf` file does not exist, create it:<br>`touch $FWDIR/conf/cpha_link_monitoring.conf` |
| 5 | Edit the `$FWDIR/conf/cpha_link_monitoring.conf` file:<br>`vi $FWDIR/conf/cpha_link_monitoring.conf` |
| 6 | ■ To monitor the link only on specific interfaces:<br>Enter the names of the applicable interfaces - each name on a new separate line.<br>Example:<br>`eth2`<br>`eth4`<br>■ To monitor the link on all interfaces:<br>Enter only this word:<br>`all` |
| 7 | Save the changes in the file and exit the editor. |

| Step | Instructions |
|---|---|
| 8 | Reboot the Cluster Member.<br><br>**Important** - This can cause a failover.<br><br>**Best Practices:**<br><br>■ **In High Availability cluster**<br>1. Perform the configuration steps on all Cluster Members<br>2. Reboot all the **Standby** Cluster Members<br>3. Initiate a manual failover on the **Active** Cluster Member<br>4. Reboot the former **Active** Cluster Member<br><br>■ **In Load Sharing Unicast cluster**<br>1. Perform the configuration steps on all Cluster Members<br>2. Reboot all the **non-Pivot** Cluster Members<br>3. Initiate a manual failover on the **Pivot** Cluster Member<br>4. Reboot the former **Pivot** Cluster Member<br><br>■ **In Load Sharing Multicast cluster**<br>1. Perform the configuration steps on all Cluster Members<br>2. Reboot all Cluster Members **except one**<br>3. Initiate a manual failover on the **remaining** Cluster Member<br>4. Reboot the **remaining** Cluster Member<br><br>**Note** - See *"Initiating Manual Cluster Failover" on page 225*. |

# Configuring Assigned Load in the Load Sharing Unicast Mode

By default, this is how a Cluster in the Load Sharing Unicast Mode assigns traffic to Cluster Members:

| Total number of Cluster Members | % of traffic inspected by the Pivot member | % of traffic inspected by each of the non-Pivot members |
|---|---|---|
| 1 | 100 | N / A |
| 2 | 30 | (100% - (% of traffic inspected by Pivot member)) / (# of non-Pivot members) = (100% - 30%) / 1 = 70% |
| 3 | 20 | (100% - (% of traffic inspected by Pivot member)) / (# of non-Pivot members) = (100% - 20%) / 2 = 40% |
| 4 | 10 | (100% - (% of traffic inspected by Pivot member)) / (# of non-Pivot members) = (100% - 10%) / 3 = 30% |
| 5 | 0 | (100% - (% of traffic inspected by Pivot member)) / (# of non-Pivot members) = (100% - 0%) / 4 = 25% |

**Note** - See the maximum supported number of Cluster Members in the *R81.10 Release Notes*.

**To change the default assigned traffic load:**

1. Back up the Security Management Server / applicable Domain Management Server.

   Refer to:

   - sk108902 - Best Practices - Backup on Gaia OS.
   - sk91400 - System Backup and Restore feature in Gaia.

2. Close **all** SmartConsole windows.

   **Note** - To make sure there are no active sessions, run the "`cpstat mg`" command in the Expert mode on the Security Management Server / in the context of *each* Domain Management Server.

3. Connect with *Database Tool (GuiDBEdit Tool)* to the Security Management Server / applicable Domain Management Server.
4. In the top left pane, go to **Table** > **Network Object** > **network_objects**.
5. In the top right pane, select the applicable Security Gateway / Cluster object.

   **Note** - In the column "`Class Name`":
   - "`gateway_cluster`" indicates a non-VSX Cluster
   - "`vs_cluster_netobj`" indicates a VSX Cluster
6. Press the **CTRL+F** keys (or go to the **Search** menu > click **Find**) > paste **Pivot_overhead**> click **Find Next**.
7. Change the value of the **Pivot_overhead** attribute for your needs:
   - To make the Pivot member inspect **more** traffic - **decrease** the value of "**Pivot_overhead**".
   - To make the Pivot member inspect **less** traffic - **increase** the value of "**Pivot_overhead**".
8. Save the changes: go to the **File** menu > click **Save All**.
9. Close the Database Tool (GuiDBEdit Tool).
10. Connect with SmartConsole to the Security Management Server / applicable Domain Management Server.
11. Install the Access Control Policy on the applicable Cluster object.

# Working with Bond Interfaces in Cluster

This section describes the configuration of Bond Interfaces and Group of Bonds in cluster.

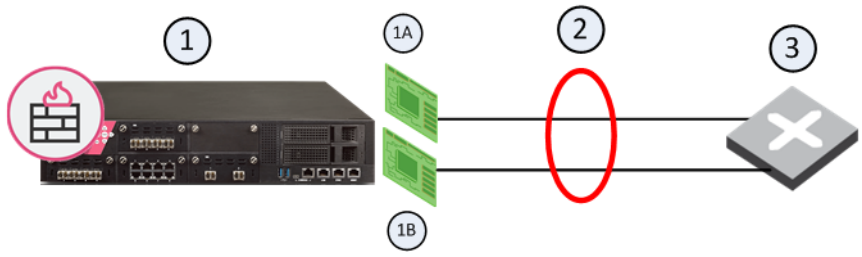
### Bonding (Link Aggregation) Terminology

- **Link Aggregation (Interface Bonding)**: Networking technology that binds multiple physical interfaces together into one virtual interface.
- **Bond**: A group of physical interfaces that operate together as one virtual interface and share an IP address and MAC address. A bond is identified by the cluster by its **Bond ID** (for example: `bond0`).
- **Bond Interface**: The logical representation of the bond.
- **Subordinate interface**: A physical interface that is a member of a bond. Subordinate interfaces do not have an IP Address and in some cases share the same MAC address.

# Bond Interfaces (Link Aggregation)

Check Point security devices support **Link Aggregation**, a technology that joins multiple physical interfaces into one virtual interface, known as a **bond interface**.

The bond interface share the load among many interfaces, which gives fault tolerance and increases throughput. Check Point devices with the Gaia OS kernel 3.10 and higher support the IEEE 802.3ad and IEEE 802.1ax Link Aggregation Control Protocol (LACP) for dynamic link aggregation.




| Item | Description |
|---|---|
| 1 | Security Gateway |
| 1A | Interface 1 |
| 1B | Interface 2 |
| 2 | Bond Interface |
| 3 | Router |

A **bond interface** (also known as a **bonding group** or **bond**) is identified by its **Bond ID** (for example: *bond1*) and is assigned an IP address. The physical interfaces included in the bond are called **subordinate interfaces** and do not have IP addresses.

You can configure a bond interface to use one of these functional strategies:

**High Availability (Active/Backup)**

Gives redundancy when there is an interface or a link failure. This strategy also supports switch redundancy.

Bond High Availability works in **Active/Backup** mode - interface Active/Standby mode. When an Active subordinate interface is down, the connection automatically fails over to the primary subordinate interface. If the primary subordinate interface is not available, the connection fails over to a different subordinate interface.

**Load Sharing (Active/Active)**

All subordinate interfaces in the UP state are used simultaneously.

Traffic is distributed among the subordinate interfaces to maximize throughput. Bond Load Sharing does not support switch redundancy.

**Note** - Bonding Load Sharing mode requires SecureXL to be enabled on Security Gateway or each Cluster Member.

You can configure Bond Load Sharing to use one of these modes:

| Mode | Description |
|---|---|
| **Round Robin** | Selects the Active subordinate interfaces sequentially.<br>**Note** - Scalable Platform Security Groups do **not** support this feature (Known Limitation MBS-4080). |
| **802.3ad** | Dynamically uses Active subordinate interfaces to share the traffic load. This mode uses the LACP protocol, which fully monitors the interface link between the Check Point Security Gateway and a switch.<br>Check Point devices with the Gaia OS kernel 3.10 and higher support the IEEE 802.3ad and IEEE 802.1ax Link Aggregation Control Protocol (LACP) for dynamic link aggregation. |
| **XOR** | All subordinate interfaces in the UP state are Active for Load Sharing. Traffic is assigned to Active subordinate interfaces based on one of these transmit hash policies:<br>■ Layer 2 information (XOR of hardware MAC addresses)<br>■ Layer 3+4 information (IP addresses and Ports) |

| Mode | Description |
|---|---|
| **ABXOR** | Subordinate interfaces in the UP state are assigned to sub-groups called *bundles*.<br>Only one bundle is Active at a time.<br>All subordinate interfaces in the Active bundle share the traffic load.<br>The system assigns traffic to all interfaces in the Active bundle based on the defined transmit hash policy.<br>**Note** - Scalable Platform Security Groups do **not** support this feature (Known Limitation MBS-1520). |

For Bonding High Availability mode and for Bonding Load Sharing mode:

- The number of bond interfaces that can be defined is limited by the maximal number of interfaces supported by each platform.

  See the *R81.10 Release Notes*.

- Up to 8 physical subordinate interfaces can be configured in a single bond interface.

# Bond High Availability Mode in Cluster

When dealing with mission-critical applications, an enterprise requires its network to be highly available.

Clustering provides redundancy at the Security Gateway level.

Without Bonding, redundancy of Network Interface Cards (NICs) or of the switches on either side of the Security Gateway are only possible in a cluster, and only by failover from one Cluster Member to another Cluster Member.

## Simple Redundant Topology

You can have redundancy of clustering without Bonding.

If a switch or Cluster Member fails, a High Availability cluster solution provides system redundancy.

For example, you can have a redundant system with two synchronized Cluster Members deployed in a simple redundant topology.

| Item | Description |
|---|---|
| 1 | Cluster Member GW1 with interfaces connected to the external switches (**5** and **6**) |
| 2 | Cluster Member GW2 with interfaces connected to the external switches (**5** and **6**) |
| 3 | Interconnecting network C1 |
| 4 | Interconnecting network C2 |
| 5 | Switch S1 |
| 6 | Switch S2 |

If Cluster Member GW1 (**1**), its NIC, or switch S1 (**5**) fails, Cluster Member GW2 (**2**) becomes the only Active member, connecting to switch S2 (**6**) over network C2 (**4**).

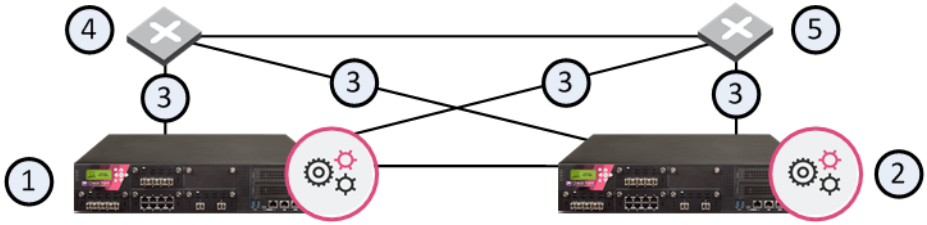
If any component fails (Cluster Member, NIC, or switch), the result of the failover is that no further redundancy exists.

A further failure of any active component completely stops network traffic through this cluster.

## Fully Meshed Redundancy

The Bonding High Availability mode, when deployed with ClusterXL, enables a higher level of reliability by providing granular redundancy in the network. This granular redundancy is achieved by using a fully meshed topology, which provides for independent backups for both NICs and switches.

A fully meshed topology further enhances the redundancy in the system by providing a backup to both the interface and the switch, essentially backing up the cable. Each Cluster Member has two external interfaces, one connected to each switch.




| Item | Description |
|---|---|
| 1 | Cluster Member GW1 with interfaces connected to the external switches (**4** and **5**) |
| 2 | Cluster Member GW2 with interfaces connected to the external switches (**4** and **5**) |
| 3 | Interconnecting network |
| 4 | Switch S1 |
| 5 | Switch S2 |

In this scenario:

- GW1 and GW2 are Cluster Members in the High Availability mode, each connected to the two external switches
- S1 and S2 are external switches
- Item 3 are the network connections

If any of the interfaces on a Cluster Member that connect to an external switch fails, the other interface continues to provide the connectivity.

If any Cluster Member, its NIC, or switch fails, the other Cluster Member, connecting to switch S2 over network C2. If any component fails (Cluster Member , NIC, or switch), the result of the failover is that no further redundancy exists. A further failure of any active component completely stops network traffic.

Bonding provides High Availability of NICs. If one fails, the other can function in its place.

## Bond Failover in High Availability Mode

In Bond High Availability mode, configured on Cluster Members, bond internal failover can occur in one of these cases:

- An active interface detects a failure in the link state, in a monitored interface.
- ClusterXL detects a failure in sending or receiving Cluster Control Protocol (CCP) packets.

Either of these failures causes a failover within the interface bond, or between Cluster Members, depending on the circumstances.

When a failure is detected, a log is recorded:

1. Connect with SmartConsole to the Management Server.
2. From the left navigation panel, click **Logs & Monitor** > **Logs**.

## Configuring a Bond Interface in High Availability Mode

On each Cluster Member, follow the instructions in the *R81.10 Gaia Administration Guide* - Chapter *Network Management* - Section *Network Interfaces* - Section *Bond Interfaces (Link Aggregation)*.

## Making Sure the Bond Interface is Functioning Properly

Examine the bond information to make sure the bond interface is UP:

| Step | Instructions |
|---|---|
| 1 | Connect to the command line on each Cluster Member. |
| 2 | Examine the cluster interfaces in one of these ways:<br>■ In Gaia Clish:<br>`show cluster members interfaces all`<br>■ In the Expert mode:<br>`cphaprob -am if`<br>See *"Viewing Cluster Interfaces" on page 258*. |
| 3 | Examine the bond interfaces in one of these ways:<br>■ In Gaia Clish:<br>`show cluster bond {all \| name <Name of Bond>}`<br>`show bonding groups`<br>■ In the Expert mode:<br>`cphaprob show_bond <Name of Bond>`<br>See *"Viewing Bond Interfaces" on page 263*. |

## Failover Support for VLANs

In Bonding High Availability mode, ClusterXL monitors VLAN IDs for connectivity failure or miscommunication, and initiate a failover when a failure is detected.

In a VLAN-enabled switched environment, ClusterXL monitors the VLAN with the lowest ID number. The monitoring is conducted by sending ClusterXL Control Protocol (CCP) packets on round-trip paths at a set interval.

The lowest VLAN ID indicates the status of the physical connection. This VLAN ID is always monitored, and a connectivity failure causes ClusterXL to initiate a failover.

ClusterXL does not detect a VLAN configuration problems on switches.

For more information, see *"Working with VLANs in Cluster" on page 117*.

## Sync Redundancy

The use of more than one physical synchronization interface (**1st sync, 2nd sync, 3rd sync**) for synchronization redundancy is **not** supported. For synchronization redundancy, you can use bond interfaces.

**Requirements and Limitations:**

- The bond subordinate interfaces on each Cluster Member must connect to the same switch or VLAN (for example, physical interface *eth1* on all Cluster Members must connect to the same switch).
- We recommend that interfaces and other network hardware support the IEEE 802.3 bond mode.
- If you use a Bond in High Availability mode, you must add subordinate interfaces to the bonding group in the same order on all Cluster Members.

**Important** - See *"Supported Topologies for Synchronization Network" on page 21*.

**To configure bond interfaces for Sync High Availability:**

1. Configure a bond interface on each Cluster Member with unused subordinate interfaces. See *"Configuring a Bond Interface in High Availability Mode" on page 141*.
2. Connect with SmartConsole to the Management Server.
3. From the left navigation panel, click **Gateways & Servers**.
4. Open the cluster object.
5. From the left tree, click **Network Management**.
6. At the top, click **Get Interfaces > Get Interfaces With Topology**.
7. Select the applicable interface and click **Edit**.
8. In the **General** section, in the **Network Type** field, select **Sync**.
9. Click **OK**.
10. Install the Access Control Policy on this cluster object.
11. On each Cluster Member, make sure that the Sync interfaces are in the bond.

    Examine the cluster interfaces in one of these ways:

- In Gaia Clish:

```
show cluster members interfaces all
```

- In the Expert mode:

```
cphaprob -am if
```

See *"Viewing Cluster Interfaces" on page 258*.

## Configuring Bond High Availability in VRRP Cluster

The R80.20 version introduced an improved Active/Backup Bond mechanism (Enhanced Bond) when working in ClusterXL.

If you work with ClusterXL, the Enhanced Bond feature is enabled by default, and no additional configuration is required.

If you change your cluster configuration from ClusterXL to VRRP (MCVR & VRRP), or configure the VRRP (MCVR & VRRP) cluster from scratch, the Enhanced Bond feature is disabled by default.

If you change your cluster configuration from VRRP to ClusterXL, you must manually enable the Enhanced Bond feature.

To enable the Enhanced Bond feature in VRRP Cluster, set the value of the kernel parameter `fwha_bond_enhanced_enable` to 1 on *each* VRRP Cluster Member. You can set the value of the kernel parameter temporarily, or permanently.

**Setting the value of the kernel parameter temporarily**

**Important** - This change does not survive reboot.

| Step | Instructions |
|---|---|
| 1 | Connect to the command line on *each* VRRP Cluster Member. |
| 2 | Log in to the Expert mode. |
| 3 | Set the value of the kernel parameter `fwha_bond_enhanced_enable` to 1:<br>`fw ctl set int fwha_bond_enhanced_enable 1` |
| 4 | Make sure the value of the kernel parameter `fwha_bond_enhanced_enable` was set to 1:<br>`fw ctl get int fwha_bond_enhanced_enable` |

**Setting the value of the kernel parameter permanently**

| Step | Instructions |
|---|---|
| 1 | Connect to the command line on *each* Cluster Member. |
| 2 | Log in to the Expert mode. |
| 3 | Back up the current `$FWDIR/boot/modules/fwkern.conf` file:<br>`cp -v $FWDIR/boot/modules/fwkern.conf{,_BKP}` |
| 4 | Edit the current `$FWDIR/boot/modules/fwkern.conf` file:<br>`vi $FWDIR/boot/modules/fwkern.conf` |
| 5 | Add this line to the file (spaces and comments are not allowed):<br>`fwha_bond_enhanced_enable=1` |
| 6 | Save the changes in the file and exit the editor. |
| 7 | Reboot the Cluster Member. |
| 8 | Make sure the value of the kernel parameter `fwha_bond_enhanced_enable` was set to 1:<br>`fw ctl get int fwha_bond_enhanced_enable` |

**Important** - If you change your cluster configuration from VRRP to ClusterXL, you must remove the kernel parameter configuration from each Cluster Member.

# Bond Load Sharing Mode in Cluster

In Bond Load Sharing mode:

- All subordinate interfaces are active, and connections are balanced between the bond subordinate interfaces, similar to the way ClusterXL Load Sharing balances connections between Cluster Members.
- Each connection is assigned to a specific subordinate interface. For the individual connection, only one subordinate interface is active. On failure of that interface, the bond fails over the connection to one of the other subordinate interfaces, which adds the failed interface connection to the connections it is already handling.
- All the subordinate interfaces of a bond must be connected to the same switch. The switch itself must support and be configured for Bonding, by the same standard (for example, 802.3ad, or XOR) as the Security Gateway bond.

**Important** - Bond Load Sharing mode requires SecureXL to be enabled on each Cluster Member (this is the default).

## Bond Failover in Load Sharing Mode

In Bond Load Sharing mode, configured on Cluster Members, bond internal failover can occur in one of these cases:

- An active interface detects a failure in the link state, in a monitored interface.
- ClusterXL detects a failure in sending or receiving Cluster Control Protocol (CCP) packets.

Either of these failures will induce a failover within the interface bond, or between Cluster Members, depending on the circumstances.

When a failure is detected, a log is recorded:

1. Connect with SmartConsole to the Management Server.
2. From the left navigation panel, click **Logs & Monitor** > **Logs**.

## Configuring a Bond Interface in High Availability Mode

On each Cluster Member, follow the instructions in the *R81.10 Gaia Administration Guide* - Chapter *Network Management* - Section *Network Interfaces* - Section *Bond Interfaces (Link Aggregation)*.

## Configuring Minimal Number of Required Subordinate Interfaces

ClusterXL considers a bond in Load Sharing mode to be in the "down" state when fewer than a minimal number of required subordinate interfaces stay in the "up" state.

By default, the minimal number of required subordinate interfaces, which must stay in the "up" state in a bond of **n** subordinate interfaces is **n-1**.

If one more subordinate interface fails (when **n-2** subordinate interfaces stay in the "up" state), ClusterXL considers the bond interface to be in the "down" state, even if the bond contains more than two subordinate interfaces.

If a smaller number of subordinate interfaces can pass the expected traffic, you can configure explicitly the minimal number of required subordinate interfaces.

Follow *"Configuring the Minimal Number of Required Subordinate Interfaces for Bond Load Sharing" on page 229*.

## Making Sure the Bond Interface is Functioning Properly

Examine the bond information to make sure the bond interface is UP:

| Step | Instructions |
|---|---|
| 1 | Connect to the command line on each Cluster Member. |
| 2 | Examine the cluster interfaces in one of these ways:<br>■ In Gaia Clish:<br>`show cluster members interfaces all`<br>■ In the Expert mode:<br>`cphaprob -am if`<br>See *"Viewing Cluster Interfaces" on page 258*. |
| 3 | Examine the bond interfaces in one of these ways:<br>■ In Gaia Clish:<br>`show cluster bond {all \| name <Name of Bond>}`<br>`show bonding groups`<br>■ In the Expert mode:<br>`cphaprob show_bond <Name of Bond>`<br>See *"Viewing Bond Interfaces" on page 263*. |

# Group of Bonds

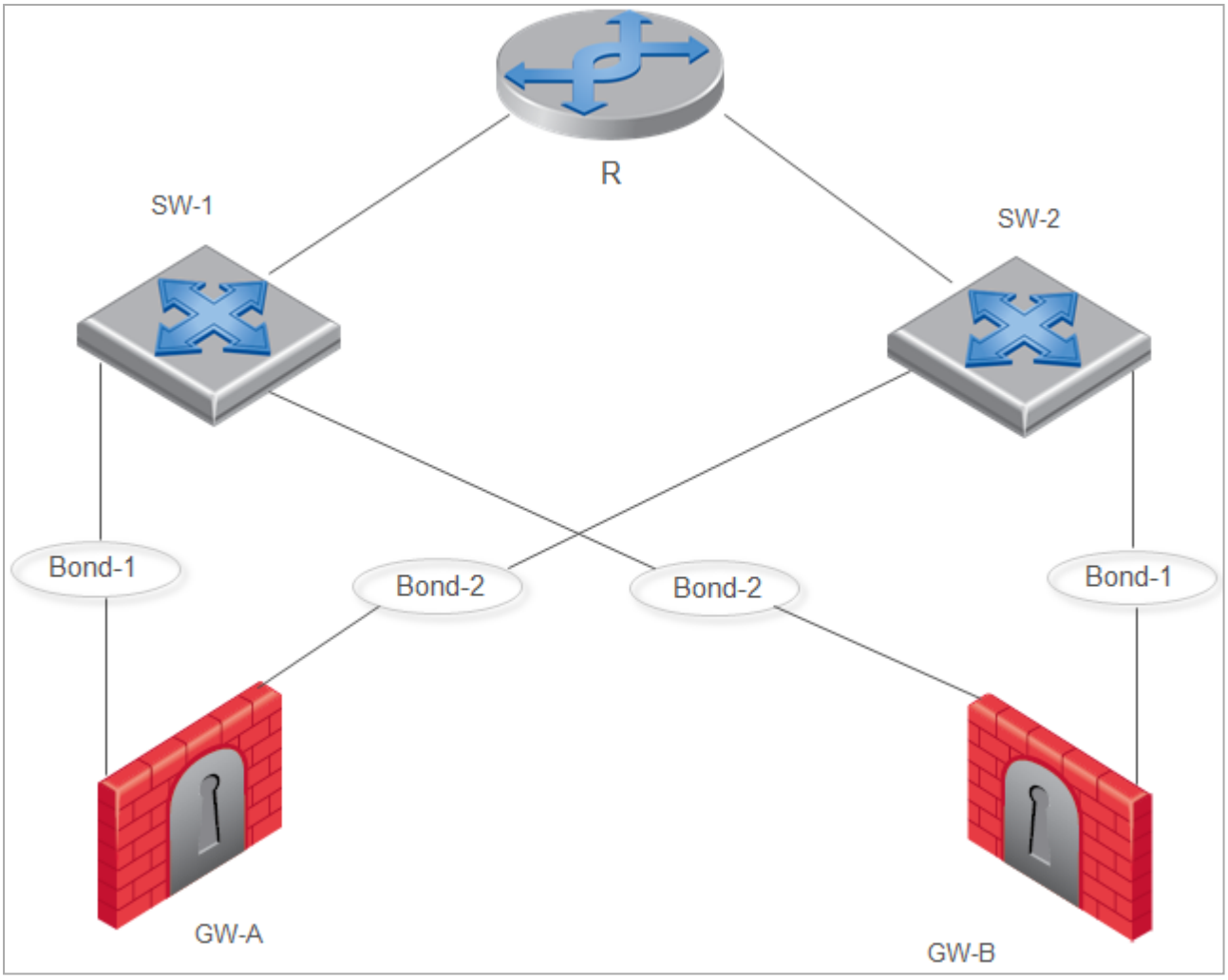
*In This Section:*

## Introduction

Group of Bonds, which is a logical group of existing Bond interfaces, provides additional link redundancy.

**Example topology with Group of Bonds**

- There is one router - **R**
- There are two switches that connect to the router **R**: **SW-1** and **SW-2**
- There are two Cluster Members **GW-A** (Active) and **GW-B** (Standby)
- There are two Bond interfaces on each Cluster Member: **Bond-1** and **Bond-2**
- On the Cluster Member **GW-A**:
  - **Bond-1** interface connects to the switch **SW-1**
  - **Bond-2** interface connects to the switch **SW-2**
- On the Cluster Member **GW-B**:
  - **Bond-1** interface connects to the switch **SW-2**
  - **Bond-2** interface connects to the switch **SW-1**




**Chain of events without Group of Bonds**

1. The Cluster Member **GW-A** is the Active and the Cluster Member **GW-B** is the Standby.
2. On the Cluster Member **GW-A**, the **Bond-1** interface fails.
3. On the Cluster Member **GW-A**, the Critical Device **Interface Active Check** reports its state as "`problem`".
4. The Cluster Member **GW-A** changes its cluster state from Active to Down.
5. The cluster fails over - the Cluster Member **GW-B** changes its cluster state from Standby to Active.

This is not the desired behavior, because the Cluster Member **GW-A** connects not only to the switch **SW-1**, but also to the switch **SW-2**. In our example topology, there is no actual reason to fail-over from the Cluster Member **GW-A** to the Cluster Member **GW-B**.

In order to overcome this problem, Cluster Members use the Group of Bonds consisting of **Bond-1** and **Bond-2**. The Group of Bonds fails only when both Bond interfaces fail on the Cluster Member. Only then the cluster fails over.

**Chain of events with configured Group of Bonds**

1. The Cluster Member **GW-A** is the Active and the Cluster Member **GW-B** is the Standby.
2. On the Cluster Member **GW-A**, the **Bond-1** interface fails.
3. On the Cluster Member **GW-A**, the Critical Device **Interface Active Check** reports its state as "`problem`".
4. The Cluster Member **GW-A** does **not** change its cluster state from Active to Down.
5. On the Cluster Member **GW-A**, the **Bond-2** interface fails as well.
6. The Cluster Member **GW-A** changes its cluster state from Active to Down.
7. The cluster fails over - the Cluster Member **GW-B** changes its cluster state from Standby to Active.

## Creating a new Group of Bonds

This procedure lets you create a new Group of Bonds.

**Procedure**

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

1. Connect to the command line on the Cluster Member.
2. Log in to the Expert mode.
3. In VSX Cluster, switch to the context of the applicable Virtual System:

   ```
   vsenv <VSID>
   ```

4. Modify the current `$FWDIR/boot/modules/fwkern.conf` file:
   a. Backup the current file:

      ```
      cp -v $FWDIR/boot/modules/fwkern.conf{,_BKP}
      ```

   b. Edit the current file:

      ```
      vi $FWDIR/boot/modules/fwkern.conf
      ```

c. Add these two lines at the bottom of the file (spaces or comments are not allowed):

```
fwha_group_of_bonds_str=<Name for Group of Bonds>:<List of all Bonds in this Group separated by comma>
fwha_arp_probe_method=1
```

Example:

```
fwha_group_of_bonds_
str=GoB0:bond0,bond1;GoB1:bond2,bond3
fwha_arp_probe_method=1
```

**Note** - The kernel parameter "`fwha_arp_probe_method`" configures the Cluster Member to use the Virtual IP address as the Source IP address in the ARP Requests during the probing of the local network.

d. Save the changes in the file and exit the editor.

5. Change the value of the kernel parameter `fwha_group_of_bonds_str` to add the Group of Bonds on-the-fly:

```
fw ctl set str fwha_group_of_bonds_str '<Name for Group of Bonds>:<List of all Bonds in this Group separated by comma>'
```

Example:

```
fw ctl set str fwha_group_of_bonds_str
'GoB0:bond0,bond1;GoB1:bond2,bond3'
```

**Notes:**

- The apostrophe characters are mandatory part of the syntax.
- Spaces are not allowed in the value of the kernel parameter `fwha_group_of_bonds_str`.

6. Change the value of the kernel parameter `fwha_arp_probe_method` on-the-fly:

```
fw ctl set int fwha_arp_probe_method 1
```

7. Make sure the Cluster Member accepted the new configuration:

```
fw ctl get str fwha_group_of_bonds_str
fw ctl get int fwha_arp_probe_method
```

8. In SmartConsole, install the Access Control Policy on the cluster object.

## Adding a Bond interface to the existing Group of Bonds

This procedure lets you add an additional bond interface to the existing Group of Bonds.

**Procedure**

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

1. Connect to the command line on the Cluster Member.
2. In VSX Cluster, switch to the context of the applicable Virtual System:

   ```
   vsenv <VSID>
   ```

3. Log in to the Expert mode.
4. Modify the current `$FWDIR/boot/modules/fwkern.conf` file:
   a. Backup the current file:

      ```
      cp -v $FWDIR/boot/modules/fwkern.conf{,_BKP}
      ```

   b. Edit the current file:

      ```
      vi $FWDIR/boot/modules/fwkern.conf
      ```

   c. Edit the value of the kernel parameter `fwha_group_of_bonds_str` to add the Bond interface to the existing Group of Bonds.

      Example:

      ```
      fwha_group_of_bonds_
      str=GoB0:bond0,bond1;GoB1:bond2,bond3,bond4
      ```

   d. Save the changes in the file and exit the editor.
5. Get the current value of the kernel parameter `fwha_group_of_bonds_str` and copy it:

   ```
   fw ctl get str fwha_group_of_bonds_str
   ```

6. Reset the current value of the kernel parameter `fwha_group_of_bonds_str`:

   ```
   fw ctl set str fwha_group_of_bonds_str ''
   ```

7. Make sure the Cluster Member reset the value of the kernel parameter `fwha_group_of_bonds_str`:

   ```
   fw ctl get str fwha_group_of_bonds_str
   ```

8. Change the value of the kernel parameter `fwha_group_of_bonds_str` to add the Bond interface to the existing Group of Bonds on-the-fly:

```
fw ctl set str fwha_group_of_bonds_str '<Name for Group of
Bonds>:<List of all Bonds in this Group separated by comma>'
```

Example:

```
fw ctl set str fwha_group_of_bonds_str
'GoB0:bond0,bond1;GoB1:bond2,bond3,bond4'
```

**Notes:**

- The apostrophe characters are mandatory part of the syntax.
- Spaces are not allowed in the value of the kernel parameter `fwha_group_of_bonds_str`.

9. Make sure the Cluster Member accepted the new configuration:

```
fw ctl get str fwha_group_of_bonds_str
```

10. In SmartConsole, install the Access Control Policy on the cluster object.

## Removing a Bond interface from the existing Group of Bonds

This procedure lets you remove a bond interface from an existing Group of Bonds.

**Procedure**

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

1. Connect to the command line on the Cluster Member.
2. Log in to the Expert mode.
3. In VSX Cluster, switch to the context of the applicable Virtual System:

```
vsenv <VSID>
```

4. Modify the current `$FWDIR/boot/modules/fwkern.conf` file:
   a. Backup the current file:

```
cp -v $FWDIR/boot/modules/fwkern.conf{,_BKP}
```

   b. Edit the current file:

```
vi $FWDIR/boot/modules/fwkern.conf
```

c. Edit the value of the kernel parameter `fwha_group_of_bonds_str` to remove the Bond interface from the existing Group of Bonds.

   Example:

   ```
   fwha_group_of_bonds_
   str=GoB0:bond0,bond1;GoB1:bond2,bond3
   ```

d. Save the changes in the file and exit the editor.

5. Get the current value of the kernel parameter `fwha_group_of_bonds_str` and copy it:

   ```
   fw ctl get str fwha_group_of_bonds_str
   ```

6. Reset the current value of the kernel parameter `fwha_group_of_bonds_str`:

   ```
   fw ctl set str fwha_group_of_bonds_str ''
   ```

7. Make sure the Cluster Member reset the value of the kernel parameter `fwha_group_of_bonds_str`:

   ```
   fw ctl get str fwha_group_of_bonds_str
   ```

8. Change the value of the kernel parameter `fwha_group_of_bonds_str` to remove the Bond interface from the existing Group of Bonds on-the-fly:

   ```
   fw ctl set str fwha_group_of_bonds_str '<Name for Group of
   Bonds>:<List of all Bonds in this Group separated by comma>'
   ```

   Example:

   ```
   fw ctl set str fwha_group_of_bonds_str
   'GoB0:bond0,bond1;GoB1:bond2,bond3
   ```

   **Notes:**
   - The apostrophe characters are mandatory part of the syntax.
   - Spaces are not allowed in the value of the kernel parameter `fwha_group_of_bonds_str`.

9. Make sure the Cluster Member accepted the new configuration:

   ```
   fw ctl get str fwha_group_of_bonds_str
   ```

10. In SmartConsole, install the Access Control Policy on the cluster object.

## Deleting a Group of Bonds

This procedure lets you delete an existing Group of Bonds.

**Procedure**

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

1. Connect to the command line on the Cluster Member.
2. Log in to the Expert mode.
3. In VSX Cluster, switch to the context of the applicable Virtual System:

   ```
   vsenv <VSID>
   ```

4. Modify the current `$FWDIR/boot/modules/fwkern.conf` file:
   a. Backup the current file:

      ```
      cp -v $FWDIR/boot/modules/fwkern.conf{,_BKP}
      ```

   b. Edit the current file:

      ```
      vi $FWDIR/boot/modules/fwkern.conf
      ```

   c. Delete these two lines in the file:

      ```
      fwha_group_of_bonds_str=<Name for Group of Bonds>:<List of all Bonds in this Group separated by comma>
      fwha_arp_probe_method=1
      ```

   d. Save the changes in the file and exit the editor.
5. Reset the current value of the kernel parameter `fwha_group_of_bonds_str`:

   ```
   fw ctl set str fwha_group_of_bonds_str ''
   ```

6. Make sure the Cluster Member reset the value of the kernel parameter `fwha_group_of_bonds_str`:

   ```
   fw ctl get str fwha_group_of_bonds_str
   ```

7. In SmartConsole, install the Access Control Policy on the cluster object.

## Monitoring

To see the configured Groups of Bonds, run the "`cphaprob show_bond_groups`" command. See *"Viewing Bond Interfaces" on page 263*.

## Logs

Cluster Members generate some applicable logs.

**Applicable log files**

1. User Space logs:
   - In non-VSX Cluster:

     In the `/var/log/messages` files.

     Output of the `dmesg` command.
   - In VSX Cluster:

     In the `$FWDIR/log/fwk.elg` file in the context of the applicable Virtual System.
2. Kernel Space logs:
   - In the kernel module `fw`, enable the debug flags `error` and `ioctl`
   - Kernel module `cluster`, enable the debug flag `if`

   See the *R81.10 Quantum Security Gateway Guide* - Chapter *Kernel Debug on Security Gateway*.

   The kernel debug shows:
   - A physical subordinate interface goes down/up.
   - A Bond interface goes down/up (regardless of a change in the Cluster Member's state).
   - A Group of Bonds goes down/up.

   **Note** - These logs are generated only once per event.

## Limitations

Specific limitations apply to a Group of Bonds.

**List of Limitations**

- The maximal length on the text string "*<Name for Group of Bonds>*" is 16 characters.
- The maximal length on this text string is 1024 characters:

  ```
  <Name for Group of Bonds>:<List of all Bonds in this Group separated by comma>
  ```
- You can configure the maximum of five Groups of Bonds on a Cluster Member or Virtual System.

- You can configure the maximum of five Bond interfaces in each Groups of Bonds.
- Group of Bonds feature does support Virtual Switches and Virtual Routers. Meaning, do not configure Groups of Bonds in the context of these Virtual Devices.
- Group of Bonds feature supports only Bond interfaces that belong to the same Virtual System.

  You cannot configure bonds that belong to different Virtual Systems into the same Group of Bonds.

  You must perform all configuration in the context of the applicable Virtual System.
- Group of Bonds feature does support Sync interfaces (an interface on a Cluster Member, whose Network Type was set as `Sync` or `Cluster+Sync` in SmartConsole in cluster object).
- Group of Bonds feature does support Bridge interfaces.
- If a Bond interface goes down on one Cluster Member, the "`cphaprob show_bond_groups`" command (see *"Viewing Bond Interfaces" on page 263*) on the peer Cluster Members also shows the same Bond interface as DOWN.

  This is because the peer Cluster Members stop receiving the CCP packets on that Bond interface and cannot probe the local network to determine that their Bond interface is really working.
- After you add a Bond interface to the existing Group of Bonds, you must install the Access Control Policy on the cluster object.
- After you remove a Bond interface from the existing Group of Bonds, you must install the Access Control Policy on the cluster object.

# Performance Guidelines for Bond Interfaces

**For optimal performance, follow these guidelines:**

1. Configure static affinities of bond subordinate interfaces to CPU Cores.
2. Whenever possible, dedicate one processing core to each interface.
3. If there are more physical interfaces than CPU cores, then some CPU cores handle two or more interfaces.

   Use pairs of subordinate interface of the same position with internal and external bonds.

   a. To view positions of subordinate interface in a bond, run in the Expert mode:

   ```
   cat /proc/net/bonding/<Name of Bond Interface>
   ```

   b. Note the sequence of the interfaces in the output.

   Compare this sequence for the two bonds (external bond and its respective internal bond).

   Subordinate interfaces that appear in the same position in the two bonds are interface pairs.

   Set these pairs to be handled by one processing CPU core.

**Example configuration**

An appliance has:

- Four processing CPU cores:

  core 0, core 1, core 2, and core 3

- Two bond interfaces:

  `bond0` with subordinate interfaces `eth0`, `eth1`, and `eth2`

  `bond1` with subordinate interfaces `eth3`, `eth4`, and `eth5`

In such case, two of the CPU cores need to handle two subordinate interfaces each.

An optimal configuration can be:

| CPU core | bond0 | bond1 |
|---|---|---|
| 0 | eth0 | eth3 |
| 1 | eth1 | eth4 |

| CPU core | bond0 | bond1 |
|---|---|---|
| 2 | eth2 | |
| 3 | | eth5 |

For more information, see the *R81.10 Performance Tuning Administration Guide*:

- Chapter *CoreXL* > Section *Configuring Affinity Settings*.
- Chapter *CoreXL* > Section *Affinity Settings for 16000 and 26000 Appliances*.

# Troubleshooting Issues with Bonded Interfaces

## Troubleshooting Workflow

1. View the logs from this cluster in **SmartConsole > Logs & Monitor > Logs**.
2. On the Cluster Members, examine the status of the bond interface in one of these ways:
   - In Gaia Clish:

     ```
     show cluster bond name <Name of Bond>
     ```

   - In the Expert mode:

     ```
     cphaprob show_bond <Name of Bond>
     ```

   See *"Viewing Bond Interfaces" on page 263*.
3. If there is a problem, see if the physical link is down:
   a. Look for a subordinate interface that reports the status of the link as "`no`".
   b. Examine the cable connections and other hardware.
   c. Examine the port configuration on the switch, to which this subordinate interface connects.

On a VSX Cluster Member, reboot is needed after these actions on a bond interface:

1. Changing a bond mode.
2. Adding a subordinate interface into an existing bond.

   **Note** - Removing a subordinate interface from an existing bond, does **not** require a reboot.

## Connectivity Delays on Switches

Connectivity delays may occur in switches during some internal bond failovers. With the various features that are now included on some switches, it can take close to a minute for a switch to begin servicing a newly connected interface.

These are suggestions for reducing the startup time after link failure.

1. Disable auto-negotiation on the relevant interface.
2. On Cisco switches, enable the PortFast feature (see the applicable Cisco

documentation).

**Warning** - The PortFast feature should never be used on ports that connect to switches or hubs. It is important that the Spanning Tree complete the initialization procedure in these situations. Otherwise, these connections may cause physical loops where packets are continuously forwarded (or even multiply) in such a way that can cause the network to fail.

3. Disable STP on the switch ports (see the applicable switch vendor documentation).

# Advanced Cluster Configuration

A number of synchronization and ClusterXL capabilities are controlled by kernel parameters.

**Important:** - In a Cluster, you must configure all the Cluster Members in the same way.

See the *R81.10 Quantum Security Gateway Guide* - Chapter *Working with Kernel Parameters on Security Gateway*.

# Controlling the Clustering and Synchronization Timers

**Best Practice** - Do **not** change the default values.

These kernel parameters control the clustering and synchronization timers.

| Parameter | Description | Default Value |
|---|---|---|
| `fwha_timer_sync_res` | The frequency of sync flush operations on the cluster.<br>Operations occur every number of milliseconds:<br>`10 x (fwha_timer_sync_res) x (fwha_timer_base_res)` | 1 |
| `fwha_timer_base_res` | Must be divisible by 10 with no remainders. | 10 |

# Blocking New Connections Under Load

**Important** - This section applies only to ClusterXL Load Sharing modes.

The reason for blocking new connections is that new connections are the main source of new Delta Synchronization traffic. Delta Synchronization may be at risk, if new traffic continues to be processed at high rate.

A related error message in cluster logs and in the `/var/log/messages` file is:

```
State synchronization is in risk
```

Reducing the amount of traffic passing through the Cluster Member protects the Delta Synchronization mechanism. See sk43896: Blocking New Connections Under Load in ClusterXL.

These kernel parameters let you control how Cluster Member behave:

| Kernel Parameter | Description |
|---|---|
| `fw_sync_block_new_conns` | Controls how Cluster Member detect heavy loads and whether they start blocking new connections.<br>Load is considered heavy when the synchronization transmit queue of the Cluster Member starts to fill beyond the value of the kernel parameter "`fw_sync_buffer_threshold`".<br>■ To enable blocking new connections under load, set the value of the "`fw_sync_block_new_conns`" to 0.<br>■ To disable blocking new connections under load, set the value of the "`fw_sync_block_new_conns`" to -1 (must use the hex value 0xFFFFFFFF). This is the default.<br>**Note** - Blocking new connections when sync is busy is only recommended for ClusterXL Load Sharing deployments. While it is possible to block new connections in ClusterXL High Availability mode, doing so does not solve inconsistencies in sync, because the High Availability mode prevents that from happening. |
| `fw_sync_buffer_threshold` | Configures the maximum percentage of the buffer that may be filled before new connections are blocked (see the parameter "`fw_sync_block_new_conns`" above).<br>The default percentage value is 80, with a buffer size of 512.<br>By default, if more than 410 consecutive packets are sent without getting an ACK on any one of them, new connections are dropped. |

<table>
<tr><th>Kernel Parameter</th><th>Description</th></tr>
<tr><td>fw_sync_<br>allowed_<br>protocols</td><td>Determines the type of connections that can be opened while the system is in a blocking state.<br>Thus, the user can have better control over the system behavior in cases of unusual load.<br>The value of this kernel parameter is a combination of flags, each specifying a different type of connection. The required value is the result of adding the separate values of these flags.<br>Summary table:
<table>
<tr><th>Flag</th><th>Value</th></tr>
<tr><td>ICMP_CONN_<br>ALLOWED</td><td>1</td></tr>
<tr><td>TCP_CONN_ALLOWED</td><td>2 (except for data connections)</td></tr>
<tr><td>UDP_CONN_ALLOWED</td><td>4 (except for data connections)</td></tr>
<tr><td>TCP_DATA_CONN_<br>ALLOWED</td><td>8 (the control connection should be established or allowed)</td></tr>
<tr><td>UDP_DATA_CONN_<br>ALLOWED</td><td>16 (the control connection should be established or allowed)</td></tr>
</table>
The default value is 24, which is the sum of "TCP_DATA_CONN_ALLOWED" (value 8) and UDP_DATA_CONN_ALLOWED (value 16). This means that the default allows only TCP and UDP data connections to be opened under load.</td></tr>
</table>

# Defining Non-Monitored Interfaces

Non-Monitored interfaces are Cluster Member interfaces that are not monitored by the ClusterXL mechanism.

You may wish to define an interface as non-monitored, if the interface is down for a long time, and you wish the Cluster Member to continue to be Active.

1. Connect with SmartConsole to the Management Server.
2. From the left navigation panel, click **Gateways & Servers**.
3. Open the cluster object.
4. From the left tree, click **Network Management**.
5. Select the interface, which is down for a long time and click **Edit**.
6. In the **Network Type** section, select **Private**.
7. Click **OK**.
8. Install the Access Control Policy on this cluster object.

# Configuring Policy Update Timeout

When policy is installed on a cluster, the Cluster Members undertake a negotiation process to make sure all of them have received the same policy before they actually apply it.

This negotiation process has a timeout mechanism, which makes sure a Cluster Member does not wait indefinitely for responses from other Cluster Members, which is useful in cases when another Cluster Member goes down when policy is being installed (for example).

In configurations, on which policy installation takes a long time (usually caused by a policy with a large number of rules), a cluster with more than two members, and slow members, this timeout mechanism may expire prematurely.

It is possible to tune the timeout with the kernel parameter **fwha_policy_update_timeout_factor**.

For more information about this kernel parameter, refer to sk92723 - Cluster flapping prevention.

# Enhanced 3-Way TCP Handshake Enforcement

The standard enforcement for a 3-way handshake that initiates a TCP connection provides adequate security by guaranteeing one-directional stickiness.

This means that it ensures that the SYN-ACK will always arrive after the SYN. However, it does not guarantee that the ACK will always arrive after the SYN-ACK, or that the first data packet will arrive after the ACK.

If you wish to have stricter policy that denies all out-of-state packets, you can configure the synchronization mechanism so that all the TCP connection initiation packets arrive in the right sequence (SYN, SYN-ACK, ACK, followed by the data).

**Warning** - The price for this extra security is a considerable delay in TCP connection establishment.

**Procedure to enable the enhanced TCP Handshake enforcement**

1. Close all SmartConsole windows connected to the Management Server.
2. Connect with Database Tool (GuiDBEdit Tool) (see sk13009) to the Security Management Server or Domain Management Server that manages this cluster.
3. In the left upper pane, go to **Table > Network Objects > network_objects**.
4. In the right upper pane, select the cluster object (the **Class Name** column shows **gateway_cluster**).
5. Press the **CTRL+F** keys (or go to **Search** menu > **Find**).
6. In the **Find** window, paste this string and click **Find Next**:

   ```
   sync_tcp_handshake_mode
   ```

7. In the lower pane, right-click on the **sync_tcp_handshake_mode** property and select **Edit**.
8. Choose **complete_sync** and click **OK**.

   For more information, see the section "*Synchronization modes for TCP 3-way handshake*" below.
9. To save the changes, from the **File** menu select **Save All**.
10. Close Database Tool (GuiDBEdit Tool).
11. Connect with SmartConsole to the Security Management Server or Domain Management Server that manages this cluster.
12. In SmartConsole, install the Access Control Policy onto the Cluster object.

**Synchronization modes for TCP 3-way handshake**

| Mode | Instructions |
|---|---|
| **Minimal sync** | This is the default 3-way handshake synchronization mode.<br>The 3-way handshake is not enforced.<br>This mode offers the best connectivity for users who are willing to compromise on security is this case. |
| **Complete sync** | All 3-way handshake packets are Sync-and-ACK'ed, and the 3-way handshake is enforced.<br>This mode slows down connection establishment considerably.<br>It may be used when there is no way to know where the next packet goes (for example, in 3rd party clusters). |
| **Smart sync** | In most cases, we can assume that if SYN and SYN-ACK were encountered by the same cluster member, then the connection is “sticky”.<br>ClusterXL uses one additional flag in Connections Table record that says, “If this member encounters a 3-way handshake packet, it should sync all other cluster members”.<br>When a SYN packet arrives, the member that encountered it, records the connection and turns off its flag. All other members are synchronized, and by using a post-sync handler, their flag is turned on (in their Connections Tables).<br>If the same member encounters the SYN-ACK packet, the connection is sticky, thus other cluster members are not informed.<br>Otherwise, the relevant member will inform all other member (since its flag is turned on).<br>The original member (that encountered the SYN) will now turn on its flag, thus all members will have their flag on.<br>In this case, the third packet of the 3-way handshake is also synchronized.<br>If for some reason, our previous assumption is not true (i.e., one cluster member encountered both SYN and SYN-ACK packets, and other members encountered the third ACK), then the “third” ACK will be dropped by the other cluster members, and we rely on the periodic sync and TCP retransmission scheme to complete the 3-way handshake.<br>This 3-way handshake synchronization mode is a good solution for ClusterXL Load Sharing users that want to enforce 3-way handshake verification with the minimal performance cost.<br>This 3-way handshake synchronization mode is also recommended for ClusterXL High Availability. |

# Cluster IP Addresses on Different Subnets

You can configure cluster Virtual IP addresses in different subnets than the physical IP addresses of the Cluster Members.

The network "sees" the cluster as one Security Gateway that operates as a network router. The network is not aware of the internal cluster structure and physical IP addresses of Cluster Members.

**Advantages of using different subnets:**

- You can create a cluster in an existing subnet that has a shortage of available IP addresses.
- You use only one Virtual IP address for the cluster. All other IP addresses can be on other subnets.
- You can "hide" physical Cluster Members' IP addresses behind the cluster Virtual IP address. This security practice is almost the same as NAT.

  **Note** - This capability is available only for ClusterXL clusters.

Traffic sent from Cluster Members to internal or external networks is hidden behind the cluster Virtual IP addresses and cluster MAC addresses. The cluster MAC address assigned to cluster interfaces is:

| Cluster Mode | MAC Address |
|---|---|
| High Availability | MAC address of the Active Cluster Member's interface |
| Load Sharing Multicast | Multicast MAC address of the cluster Virtual IP Address |
| Load Sharing Unicast | MAC address of the Pivot Cluster Member's interface |

The use of different subnets with cluster objects has some limitations - see *"Limitations of Cluster Addresses on Different Subnets" on page 174*.

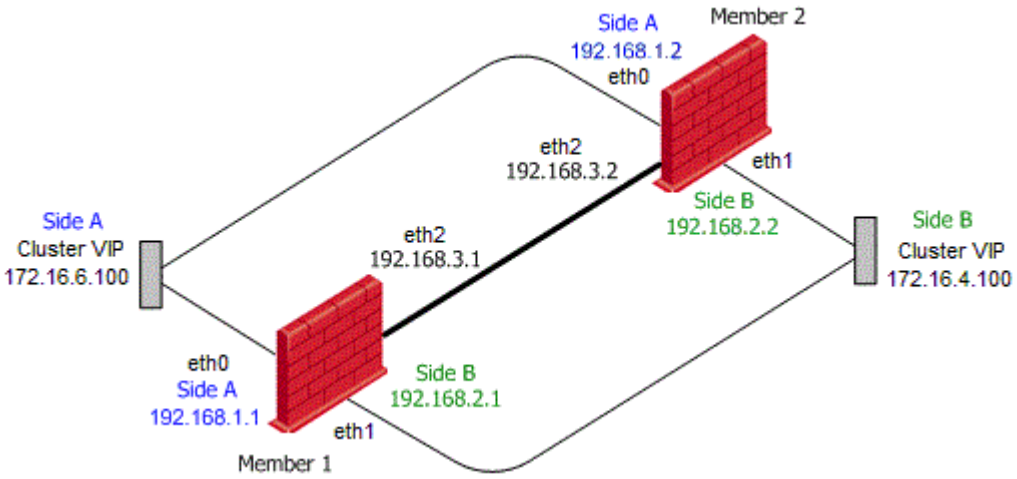
**Configuration:**

Follow the steps in *"Example of Cluster IP Addresses on Different Subnets" on page 167*.

# Example of Cluster IP Addresses on Different Subnets

In this example, a cluster separates the network **172.16.6.0 / 24** (Side "A", External) from the network **172.16.4.0 / 24** (Side "B", Internal).

The Cluster Members use these IP addresses:




| Side / Interface | Network IP Address | IP Addresses on Member_1 | IP Addresses on Member_2 |
|---|---|---|---|
| Side "A" `eth0` (External) | 192.168.1.0 / 24 | Real IP address on `eth0`: 192.168.1.1 / 24 | Real IP address on `eth0`: 192.168.1.2 / 24 |
| | 172.16.6.0 / 24 | Cluster VIP address on `eth0`: 172.16.6.100 / 24 | Cluster VIP address on `eth0`: 172.16.6.100 / 24 |
| Side "B" `eth1` (Internal) | 192.168.2.0 / 24 | Real IP address on `eth1`: 192.168.2.1 / 24 | Real IP address on `eth1`: 192.168.2.2 / 24 |
| | 172.16.4.0 / 24 | Cluster VIP address on `eth1`: 172.16.4.100 / 24 | Cluster VIP address on `eth1`: 172.16.4.100 / 24 |
| Cluster Sync `eth2` | 192.168.3.0 / 24 | Real IP address on `eth2`: 192.168.3.1 / 24 | Real IP address on `eth2`: 192.168.3.2 / 24 |

**Procedure:**

1. **On the Cluster Members, configure interfaces and static routes**

   On each Cluster Member, configure interfaces and these static routes:

- Next hop gateway for the external network **172.16.6.0** is the local interface **eth0** with the IP address **192.168.1.x**
- Next hop gateway for the internal network **172.16.4.0** is the local interface **eth1** with the IP address **192.168.2.x**

**Important** - You must enable the `scopelocal` attribute on these static routes.

See the *R81.10 Gaia Administration Guide* > Chapter *Network Management* > Section *IPv4 Static Routes*.

**Note** - In VSX Cluster, you must configure all routes only in SmartConsole in the VSX Cluster object.

Example commands for Gaia Clish:

- **Member_1:**

  (The IP address on the interface `eth0` is already configured.)

```
set static-route 172.16.6.0/24 nexthop gateway logical
eth0 on
set static-route 172.16.6.0/24 scopelocal on

set interface eth1 ipv4-address 192.168.2.1 mask-length
24
set interface eth1 state on
set static-route 172.16.4.0/24 nexthop gateway logical
eth1 on
set static-route 172.16.4.0/24 scopelocal on

set interface eth2 ipv4-address 192.168.3.1 mask-length
24
set interface eth2 state on

save config
```

- **Member_2:**

```
set static-route 172.16.6.0/24 nexthop gateway logical
eth0 on
set static-route 172.16.6.0/24 scopelocal on

set interface eth1 ipv4-address 192.168.2.2 mask-length
24
set interface eth1 state on
set static-route 172.16.4.0/24 nexthop gateway logical
eth1 on
set static-route 172.16.4.0/24 scopelocal on

set interface eth2 ipv4-address 192.168.3.2 mask-length
24
set interface eth2 state on

save config
```

2. **In SmartConsole, open the cluster object**

   a. Connect with SmartConsole to the Management Server.

   b. From the left navigation panel, click **Gateways & Servers**.

   c. Open the cluster object.

d. From the left tree, click **Network Management**.

**Important** - In the configuration of Cluster IP addresses on different subnets, the operation **Get Interfaces** > **Get Interfaces with Topology** does not detect the interface topology (External, Internal). You must configure the required topology manually.

3. **Configure the external cluster interface 'eth0'**

a. Select the **external** cluster interface `eth0` and click **Edit**.

b. In the **General** section, configure these settings:

| Field | Value |
|---|---|
| **Network Type** | Cluster |
| **IPv4** | 172.16.6.100 / 24 |

c. In the **Member IPs** section, click **Modify** and configure these settings:

| Field | Value |
|---|---|
| **'Member_1' IPv4** | 192.168.1.1 / 24 |
| **'Member_2' IPv4** | 192.168.1.2 / 24 |

d. In the **Topology** section, click **Modify**.

e. Select **Override**.

f. Select **This Network (Internal)**.

g. Select **Specific**.

h. Click in the drop-down field > in the top right corner, click **New** > click **Network Group**.

i. Configure a new **Network Group** object that contains the networks of Side "A":

i. Enter the object name. For example: **SideA_All_Networks_eth0**

ii. Configure a **Network** object for the real IP address of interfaces connected to Side "A" - **192.168.1.0 / 24**

Click **[+]** > at the top right corner, click ✳ (**New**) > click **Network**.

Enter an object name. For example: **SideA_Real_Network_eth0**

In the **Network address** field, enter **192.168.1.0**

In the **Net mask** field, enter **255.255.255.0**

Click **OK** to close the **New Network** window.

iii. Configure a **Network** object for the IP address of the Cluster VIP on Side "A" - **172.16.6.0 / 24**

Click **[+]** > at the top right corner, click ✳ (**New**) > click **Network**.

Enter an object name. For example: **SideA_VIP_Network_eth0**

In the **Network address** field, enter **172.16.6.0**

In the **Net mask** field, enter **255.255.255.0**

Click **OK** to close the **New Network** window.

iv. Click **OK** to close the **New Network Group** window.

j. The field **Specific** now shows the object **SideA_All_Networks_eth0**.

k. Click **OK** to close the **Topology Settings** window for `eth0`.

l. Click **OK** to close the **Network: eth0** window.

4. **Configure the internal cluster interface 'eth1'**

a. Select the **internal** cluster interface `eth1` and click **Edit**.

b. In the **General** section, configure these settings:

| Field | Value |
|---|---|
| **Network Type** | Cluster |
| **IPv4** | 172.16.4.100 / 24 |

c. In the **Member IPs** section, click **Modify** and configure these settings:

| Field | Value |
|---|---|
| **'Member_1' IPv4** | 192.168.2.1 / 24 |
| **'Member_2' IPv4** | 192.168.2.2 / 24 |

d. In the **Topology** section, click **Modify**.

e. Select **Override**.

f. Select **This Network (Internal)**.

g. Select **Specific**.

h. Click in the drop-down field > in the top right corner, click **New** > click **Network Group**.

i. Configure a new **Network Group** object that contains the networks of Side "B":

   i. Enter the object name. For example: **SideB_All_Networks_eth1**

   ii. Configure a **Network** object for the real IP address of interfaces connected to Side "B" - **192.168.2.0 / 24**

   Click **[+]** > at the top right corner, click ✳ (**New**) > click **Network**.

   Enter an object name. For example: **SideB_Real_Network_eth1**

   In the **Network address** field, enter **192.168.2.0**

   In the **Net mask** field, enter **255.255.255.0**

   Click **OK** to close the **New Network** window.

   iii. Configure a **Network** object for the IP address of the Cluster VIP on Side "B" - **172.16.4.0 / 24**

   Click **[+]** > at the top right corner, click ✳ (**New**) > click **Network**.

   Enter an object name. For example: **SideB_VIP_Network_eth1**

   In the **Network address** field, enter **172.16.4.0**

   In the **Net mask** field, enter **255.255.255.0**

   Click **OK** to close the **New Network** window.

   iv. Click **OK** to close the **New Network Group** window.

j. The field **Specific** now shows the object **SideB_All_Networks_eth1**.

k. Click **OK** to close the **Topology Settings** window for `eth1`.

l. Click **OK** to close the **Network: eth1** window.

5. **Configure the sync interface 'eth2'**

   a. Select the **sync** cluster interface `eth2` and click **Edit**.

   b. In the **General** section, configure these settings:

| Field | Value |
|---|---|
| **Network Type** | Sync |

   c. In the **Member IPs** section, click **Modify** and configure these settings:

| Field | Value |
|---|---|
| **'Member_1' IPv4** | 192.168.3.1 / 24 |
| **'Member_2' IPv4** | 192.168.3.2 / 24 |

   d. Click **OK** to close the **Network: eth1** window.

   e. Click **OK** to close the **Gateway Cluster Properties** window.

6. **Install the Access Control Policy**

   a. Publish the SmartConsole session.

   b. Install the Access Control Policy on this cluster object.

7. **Configure the default gateway on hosts located on Side "A"**

   The default gateway on all hosts on Side "A" must be **172.16.6.100 / 24**.

8. **Configure thdefault gateway on hosts located on Side "B"**

   The default gateway on all hosts on Side "B" must be **172.16.4.100 / 24**.

# Limitations of Cluster Addresses on Different Subnets

This new feature does not yet support all the capabilities of ClusterXL.

Some features require additional configuration to work properly, while others are not supported.

## Connectivity Between Cluster Members

Since ARP requests issued by Cluster Members are hidden behind the cluster IP and MAC addresses, requests sent by one Cluster Member to the other may be ignored by the destination computer.

To allow Cluster Members to communicate with each other, a static ARP should be configured for each Cluster Member, stating the MAC addresses of all other Cluster Members. IP packets sent between Cluster Members are not altered, and therefore no changes should be made to the routing table.

**Note** - Static ARP is not required in order for the Cluster Members to work properly as a cluster, since the cluster synchronization protocol does not rely on ARP.

## Load Sharing Multicast Mode with "Semi-Supporting" Hardware

Although not all types of network hardware work with multicast MAC addresses, some routers can pass such packets, even though they are unable to handle ARP Replies containing a multicast MAC address. Where a router *semi-supports* Load Sharing Multicast mode, it is possible to configure the cluster MAC address as a static ARP entry in the router internal tables, and thus allow it to communicate with the cluster.

When different subnets are used for the cluster IP addresses, static ARP entries containing the router MAC address need to be configured on each Cluster Member. This is done because this kind of router will not respond to ARP Requests containing a multicast source MAC address. These special procedures are not required when using routers that fully support multicast MAC addresses.

## Manual Proxy ARP

When using Static NAT, the cluster can be configured to automatically recognize the hosts hidden behind it, and issue ARP replies with the cluster MAC address, on their behalf. This process is known as *Automatic Proxy ARP*.

However, if you use the ClusterXL VMAC mode or different subnets for the cluster IP addresses, this mechanism will not work, and you must configure the proxy ARP manually. To do so, in SmartConsole, click **Menu > Global properties > NAT Network Address Translation**, and disable **Automatic ARP Configuration**. Then create the **$FWDIR/conf/local.arp** file.

For instructions, see sk30197.

## Connecting to the Cluster Members from the Cluster Network

Because the unique IP addresses may be chosen arbitrarily, there is no guarantee that these addresses are accessible from the subnet of the cluster IP address.

To access the Cluster Members through their unique IP addresses, you must configure routes on the accessing Cluster Member, such that the cluster IP is the Default Gateway for the subnet of the unique IP addresses.

# Adding Another Member to an Existing Cluster

**Important** - Schedule a full maintenance window to perform this procedure.

**Best Practice** - Before you change the current configuration, export a complete management database with "`migrate_server`" command. See the *R81.10 CLI Reference Guide*.

## Adding a New Cluster Member to the Cluster Object

1. **Install a new Cluster Member**

   Install a new Cluster Member you plan to add to the existing cluster.

   See the *R81.10 Installation and Upgrade Guide* > Chapter *Installing a ClusterXL, VSX Cluster, VRRP Cluster*.

   Follow only the step "*Install the Cluster Members*".

   **Important** - The new Cluster Member must run the same version with the same Hotfixes as the existing Cluster Members.

2. **Configure the new Cluster Member**

   On the new Cluster Member you plan to add to the existing cluster:

   a. Configure or change the IP addresses on the applicable interfaces to match the current cluster topology.

      Use Gaia Portal or Gaia Clish.

      See *R81.10 Gaia Administration Guide*.

   b. Configure or change the applicable static routes to match the current cluster topology.

      Use Gaia Portal or Gaia Clish.

   c. Connect to the command line.

   d. Log in to Gaia Clish or the Expert mode.

   e. Start the Check Point Configuration Tool. Run:

      ```
      cpconfig
      ```

   f. Select the option **Enable cluster membership for this gateway** and enter **y** to

confirm.

g. Reboot the new Cluster Member.

3. **Configure the cluster object in SmartConsole**

   a. Connect with SmartConsole to the Security Management Server or Domain Management Server that manages this cluster.

   b. From the left navigation panel, click **Gateways & Servers**.

   c. Open the existing cluster object.

   d. In the **Cluster Members** page, click **Add > New Cluster Member**.

      The **Cluster Members Properties** window opens.

      Follow the instructions on the screen to configure this new Cluster Member.

   e. Click the **General** tab.

   f. In the **Name** field, enter a Cluster Member name.

   g. In the **IPv4 Address** field, enter a physical IPv4 addresses.

      The Management Server must be able to connect to the Cluster Member at this IPv4 address.

      This IPv4 address can be an internal, or external. You can use a dedicated management interface on the Cluster Member.

      **Important** - You must define a corresponding IPv4 address for every IPv6 address. This release does not support the configuration of only IPv6 addresses.

   h. In the **IPv6 Address** field, enter a physical IPv6 address, if you need to use IPv6.

      The Management Server must be able to connect to the Cluster Member at this IPv6 address. This IPv6 address can be an internal, or external. You can use a dedicated management interface on the Cluster Member.

      **Important** - You must define a corresponding IPv4 address for every IPv6 address. This release does not support the configuration of only IPv6 addresses.

   i. Click **Communication**, and initialize Secure Internal Communication (SIC) trust.

      Enter the same key you entered during First Time Configuration Wizard on the new Cluster Member.

   j. Click the **NAT** tab to configure the applicable NAT settings.

   k. Click the **VPN** tab to configure the applicable VPN settings.

l. Click **OK**.

m. From the left tree, click **Network Management**.

- Make sure all interfaces are defined correctly.
- Make sure all IP addresses are defined correctly.

n. Click **OK**.

o. Publish the SmartConsole session.

p. Install the Access Control Policy on this cluster object.

Policy installation must succeed on all Cluster Members.

q. Install the Threat Prevention Policy on this cluster object.

Policy installation must succeed on all Cluster Members.

4. **Examine the cluster state**

On *each* Cluster Member (existing and the newly added):

a. Connect to the command line.

b. Log in to Gaia Clish or the Expert mode.

c. Make sure all Cluster Members detect each other and agree on their cluster states. Run:

| Shell | Command |
|---|---|
| Gaia Clish | i. `set virtual-system <VSID>`<br>ii. `show cluster state` |
| Expert mode | `cphaprob [-vs <VSID>] state` |

For more information, see *"Viewing Cluster State" on page 240*.

If Cluster Members do **not** detect each other, or do **not** agree on their cluster states, then restart the clustering.

**Procedure**

a. Connect to the command line on *each* Cluster Member (existing and the newly added).

b. Log in to Gaia Clish or the Expert mode.

c. Restart the clustering on *each* Cluster Member.

Run:

```
cphastop
cphastart
```

**Important** - This temporarily causes the Cluster Member not to be a part of the cluster. As a result, cluster failover can occur.

d. Make sure all Cluster Members detect each other and agree on their cluster states. Run:

| Shell | Command |
|---|---|
| Gaia Clish | i. `set virtual-system <VSID>`<br>ii. `show cluster state` |
| Expert mode | `cphaprob [-vs <VSID>] state` |

# Adding an Existing Security Gateway as a Cluster Member to the Cluster Object

**Important** - The existing Security Gateway must run the same version with the same Hotfixes as the existing Cluster Members.

1. **Configure the existing Security Gateway**

On the existing Security Gateway you plan to add to the existing cluster:

a. Configure or change the IP addresses on the applicable interfaces to match the current cluster topology.

Use Gaia Portal or Gaia Clish.

See *R81.10 Gaia Administration Guide*.

b. Configure or change the applicable static routes to match the current cluster topology.

Use Gaia Portal or Gaia Clish.

c. Connect to the command line.

d. Log in to Gaia Clish or the Expert mode.

e. Start the Check Point Configuration Tool. Run:

```
cpconfig
```

f. Select the option **Enable cluster membership for this gateway** and enter **y** to confirm.

g. Reboot the Security Gateway.

2. **Configure the cluster object in SmartConsole**

a. Connect with SmartConsole to the Security Management Server or Domain Management Server that manages this cluster.

b. From the left navigation panel, click **Gateways & Servers**.

c. Open the existing cluster object.

d. From the left tree, click **Cluster Members**.

e. Click **Add > Add Existing Gateway**.

Follow the instructions on the screen to configure this new Cluster Member.

Read the warning and click **Yes**:

```
If you add <Name_of_Security_Gateway_object> to the
cluster, it will be converted to a cluster member. Some
settings will be lost.
The following settings will still remain:
-SIC
-VPN
-NAT (except for IP Pools)
In order to revert the conversion, session must be
discarded.
Are you sure you want to continue?
```

f. In the list of Cluster Members, select the new Cluster Member and click **Edit**.

g. Click the **NAT** tab and configure the applicable NAT settings.

h. Click the **VPN** tab and configure the applicable VPN settings.

i. From the left tree, click **Network Management**.

- Make sure all interfaces are defined correctly.
- Make sure all IP addresses are defined correctly.

j. Click **OK**.

k. Install the Access Control Policy on this cluster object.

   Policy installation must succeed on all Cluster Members.

l. Install the Threat Prevention Policy on this cluster object.

   Policy installation must succeed on all Cluster Members.

3. **Examine the cluster state**

   On *each* Cluster Member (existing and the newly added):

   a. Connect to the command line.

   b. Log in to Gaia Clish or the Expert mode.

   c. Make sure all Cluster Members detect each other and agree on their cluster states. Run:

| Shell | Command |
|---|---|
| Gaia Clish | i. `set virtual-system <VSID>`<br>ii. `show cluster state` |
| Expert mode | `cphaprob [-vs <VSID>] state` |

   For more information, see *"Viewing Cluster State" on page 240*.

If Cluster Members do **not** detect each other, or do **not** agree on their cluster states, then restart the clustering.

**Procedure**

a. Connect to the command line on *each* Cluster Member (existing and the newly added).

b. Log in to Gaia Clish or the Expert mode.

c. Restart the clustering on *each* Cluster Member.

   Run:

```
cphastop
cphastart
```

**Important** - This temporarily causes the Cluster Member not to be a part of the cluster. As a result, cluster failover can occur.

d. Make sure all Cluster Members detect each other and agree on their cluster states. Run:

| Shell | Command |
|---|---|
| Gaia Clish | i. `set virtual-system <VSID>`<br>ii. `show cluster state` |
| Expert mode | `cphaprob [-vs <VSID>] state` |

# Removing a Member from an Existing Cluster

**Important** - Schedule a full maintenance window to perform this procedure.

**Best Practice** - Before you change the current configuration, export a complete management database with "`migrate_server`" command. See the *R81.10 CLI Reference Guide*.

1. **Configure the cluster object in SmartConsole**

    a. Connect with SmartConsole to the Security Management Server or Domain Management Server that manages this cluster.

    b. From the left navigation panel, click **Gateways & Servers**.

    c. Open the existing cluster object.

    d. From the left tree, click **Cluster Members** page.

    e. Click **Remove > Delete Cluster Member**.

    Confirm when prompted.

    **Important:**
    - This operation *deletes* the object.
    - There must be *at least two* Cluster Members in the cluster object.

    f. From the left tree, click **Network Management**.

    - Make sure all interfaces are defined correctly.
    - Make sure all IP addresses are defined correctly.

    g. Click **OK**.

    h. Install the Access Control Policy on the cluster object.

2. **Restart the clustering and examine the cluster state**

    On *each* existing Cluster Member:

    a. Connect to the command line.

    b. Log in to Gaia Clish or the Expert mode.

c. Restart the clustering.

Run:

```
cphastop
cphastart
```

**Important** - This temporarily causes the Cluster Member not to be a part of the cluster. As a result, cluster failover can occur.

d. Make sure all Cluster Members detect each other and agree on their cluster states. Run:

| Shell | Command |
|---|---|
| Gaia Clish | i. `set virtual-system <VSID>`<br>ii. `show cluster state` |
| Expert mode | `cphaprob [-vs <VSID>] state` |

3. **Configure the removed Cluster Member**

On the Security Gateway you removed from the existing cluster:

a. Connect to the command line.

b. Log in to Gaia Clish or the Expert mode.

c. Start the Check Point Configuration Tool. Run:

```
cpconfig
```

d. Select the option **Disable cluster membership for this gateway** and enter **y** to confirm.

e. Select the option **Secure Internal Communication** > enter **y** to confirm > enter the new Activation Key. Make sure to write it down.

f. Exit from the `cpconfig` menu.

g. Reboot the Security Gateway.

4. **Establish SIC with the removed Security Gateway**

If you need to use the Security Gateway you removed from the existing cluster, then establish Secure Internal Communication (SIC) with it.

a. Connect with SmartConsole to the Security Management Server or Domain Management Server that manages this Security Gateway.

b. From the left navigation panel, click **Gateways & Servers**.

c. Create a new Security Gateway object in one of these ways:

   - From the top toolbar, click the **New (✳) > Gateway**.
   - In the top left corner, click **Objects** menu **> More object types > Network Object > Gateways and Servers > New Gateway**.
   - In the top right corner, click **Objects Pane > New > More > Network Object > Gateways and Servers > Gateway**.

d. Follow the instructions on the screen.

   Enter the same Activation Key you entered earlier in the `cpconfig` menu.

e. Click **OK**.

f. Publish the SmartConsole session.

g. Install the Access Control Policy on the Security Gateway object.

h. Install the Threat Prevention Policy on the Security Gateway object.

# ISP Redundancy on a Cluster

*In This Section:*

Introduction ..... 186
ISP Redundancy Modes ..... 189
Outgoing Connections ..... 190
Incoming Connections ..... 191

**Important** - ISP Redundancy is **not** supported if Dynamic Routing is configured (Known Limitation PMTR-68991).

**Note** - For information about ISP Redundancy on a Security Gateway, see the *R81.10 Quantum Security Gateway Guide*.

## Introduction

ISP Redundancy connects Cluster Members to the Internet through redundant Internet Service Provider (ISP) links.

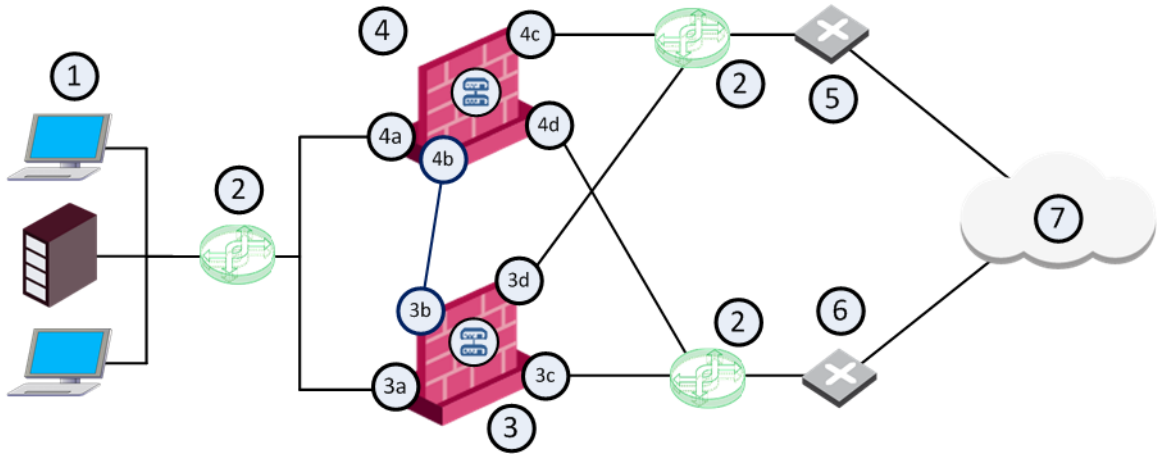
ISP Redundancy monitors the ISP links and chooses the best current link.

**Notes:**

- ISP Redundancy requires a minimum of two external interfaces and supports a maximum of ten interfaces.
  To configure more than two ISP links, the Management Server and the Cluster must run the version R81.10 and higher.
- ISP Redundancy is intended to traffic that originates on your internal networks and goes to the Internet.

**Important:**

- You must connect each Cluster Member with a dedicated physical interface to each of the ISPs.
- The IP addresses assigned to physical interfaces on each Cluster Member must be on the same subnet as the Cluster Virtual IP address.




IP addresses in the table below are only examples.

| Item | Description |
|---|---|
| 1 | Internal network |
| 2 | Switches |
| 3 | Cluster Member A |
| 3a | Cluster interface connected to the internal network (IP address 10.10.10.0/24)<br>■ Interface IP address 10.10.10.11<br>■ Virtual IP address 10.10.10.1 |
| 3b | Cluster interface (IP address 20.20.20.11) connected to the Sync network (IP address 20.20.20.0/24) |
| 3c | Cluster interface connected to a switch that connects to **ISP A**<br>■ Interface IP address 30.30.30.11<br>■ Virtual IP address 30.30.30.1 |
| 3d | Cluster interface connected to a switch that connects to **ISP B**<br>■ Interface IP address 40.40.40.11<br>■ Virtual IP address 40.40.40.1 |
| 4 | Cluster Member B |
| 4a | Cluster interface connected to the internal network (IP address 10.10.10.0/24)<br>■ Interface IP address 10.10.10.22<br>■ Virtual IP address 10.10.10.1 |
| 4b | Cluster interface (IP address 20.20.20.22) connected to the Sync network (IP address 20.20.20.0/24) |

| Item | Description |
|---|---|
| 4c | Cluster interface connected to a switch that connects to **ISP B**<br>■ Interface IP address 40.40.40.22<br>■ Virtual IP address 40.40.40.1 |
| 4d | Cluster interface connected to a switch that connects to **ISP A**<br>■ Interface IP address 30.30.30.22<br>■ Virtual IP address 30.30.30.1 |
| 5 | ISP B |
| 6 | ISP A |
| 7 | Internet |

# ISP Redundancy Modes

ISP Redundancy configuration modes control the behavior of outgoing connections from internal clients to the Internet:

| Mode | Description |
|---|---|
| **Load Sharing** | Uses all links to distribute the load of connections.<br>The incoming connections are alternated.<br>You can configure best relative loads for the links (set a faster link to handle more load).<br>New connections are randomly assigned to a link.<br>If one link fails, the other link takes the load.<br>In this mode, incoming connections can reach the application servers through any of ISP links because the Cluster can answer DNS requests for the IP address of internal servers with IP addresses from both ISPs by alternating their order. |
| **Primary/Backup** | Uses one link for connections.<br>It switches to the Backup link, if the Primary link fails.<br>When the Primary link is restored, new connections are assigned to it.<br>Existing connections continue on the Backup link until they are complete.<br>In this mode, incoming connections (from the Internet to application servers in the DMZ or internal networks) also benefit, because the Cluster returns packets using the same ISP Link, through which the connection was initiated. |

**Best Practice:**

- If all ISP links are basically the same, use the Load Sharing mode to ensure that you are making the best use of all ISP links.
- You may prefer to use one of your ISP links that is more cost-effective in terms of price and reliability.
  In that case, use the Primary/Backup mode and set the more cost-effective ISP as the Primary ISP link.

# Outgoing Connections

| Mode | Description |
|---|---|
| **Load Sharing** | Outgoing traffic that exits the Cluster on its way to the Internet is distributed between the ISP Links.<br>You can set a relative weight for how much you want each of the ISP Links to be used.<br>For example, if one link is faster, it can be configured to route more traffic across that ISP link than the other links. |
| **Primary/Backup** | Outgoing traffic uses an active primary link.<br>Hide NAT is used to change the source address of outgoing packets to the address of the interface, through which the packet leaves the Cluster.<br>This allows return packets to be automatically routed through the same ISP link, because their destination address is the address of the correct link.<br>Administrator configures the Hide NAT settings. |

# Incoming Connections

For external users to make incoming connections, the administrator must:

1. Give each application server one routable IP address for each ISP.
2. Configure Static NAT to translate the routable addresses to the real server address.

If the servers handle different services (for example, HTTP and FTP), you can use NAT to employ only routable IP addresses for all the publicly available servers.

External clients use one of the assigned IP addresses. In order to connect, the clients must be able to resolve the DNS name of the server to the correct IP address.

**Example**

**Note** - The example below is for two ISP links. The subnets **172.16.0.0/24** and **192.168.0.0/24** represent public routable IP addresses.

The Web server **www.example.com** is assigned an IP address from each ISP:

- 192.168.1.2 from ISP A
- 172.16.2.2 from ISP B

If the **ISP Link A** is down, then IP address **192.168.1.2** becomes unavailable, and the clients must be able to resolve the URL **www.example.com** to the IP address **172.16.2.2**.

An incoming connection is established, based on this example, in the following sequence:

1. When an external client on the Internet contacts **www.example.com**, the client sends a DNS query for the IP address of this URL.

   The DNS query reaches the Cluster.

   The Cluster has a built-in mini-DNS server that can be configured to intercept DNS queries (of Type A) for servers in its domain.
2. A DNS query arriving at an interface that belongs to one of the ISP links, is intercepted by the Cluster.
3. If the Cluster recognizes the name of the host, it sends one of the following replies:
   - In ISP Redundancy **Primary/Backup** mode, the Cluster replies only with the IP addresses associated with the Primary ISP link, as long as the Primary ISP link is active.
   - In ISP Redundancy **Load Sharing** mode, the Cluster replies with two IP addresses, alternating their order.
4. If the Cluster is unable to handle DNS requests (for example, it may not recognize the host name), it passes the DNS query to its original destination or the DNS server of the domain **example.com**.

5. When the external client receives the reply to its DNS query, it opens a connection. Once the packets reach the Cluster, the Cluster uses Static NAT to translate the destination IP address **192.168.1.2** or **172.16.2.2** to the real server IP address **10.0.0.2**.
6. The Cluster routes the reply packets from the server to the client through the same ISP link that was used to initiate the connection.

# Configuring ISP Redundancy on a Cluster

**Important:**

- ISP Redundancy requires a minimum of two interfaces.
- If you configure a Cloning Group and ISP Redundancy on a Cluster, then you must enable the Gaia feature "**Kernel Routes**". See the *R81.10 Gaia Advanced Routing Administration Guide*.

1. Connect with SmartConsole to the Security Management Server or Domain Management Server that manages this Cluster.
2. From the left navigation panel, click **Gateways & Servers**.
3. Open the applicable Cluster object.
4. Click **Other** > **ISP Redundancy**.
5. Select **Support ISP Redundancy**.
6. Select the redundancy mode:
   - **Load Sharing** - traffic is sent in a round-robin method over all configured ISP Links.
   - **Primary/Backup** - traffic is sent only over one ISP Link until it goes down (the order of arranged ISP Links determines in which order to use them).
7. Configure the ISP Links (at least two, at maximum ten).

   To configure more than two ISP links, the Management Server and a Cluster must run the version R81.10 and higher.

   **Procedure**

   Make sure you have the ISP data - the speed of the link and next hop IP address.

   - If the Cluster object has at least two interfaces with the Topology "**External**" in the **Network Management** page, you can configure the ISP links automatically.

     **Configuring ISP links automatically**

     a. Click **Other** > **ISP Redundancy**.

     b. Click **Set initial configuration**.

     The ISP Links are added automatically.

     c. If you selected the **Primary/Backup** mode, make sure the Primary interface is first in the list.

     Use the arrows on the right to change the order.

     d. Click **OK**.

- If the Cluster object has only one interface with the Topology "**External**" in the **Network Management** page, you must configure the ISP links manually.

  **Note** - We recommend to configure the Topology "**External**" for each interface the Cluster needs to use for ISP Redundancy.

  **Configuring ISP links manually**

  a. Click **Other** > **ISP Redundancy**.

  b. In the **IPS Links** section, click **Add**.

     The **ISP Link** window opens.

  c. Click the **General** tab.

  d. In the **Name** field, enter a name for this ISP link.

     The name you enter here is used in the ISP Redundancy commands (see *"Controlling ISP Redundancy from CLI" on page 201*).

  e. In the **Interface** field, select the correct interface of the Cluster for this ISP link.

     If one of the ISP links is the connection to a backup ISP, configure the ISP Redundancy Script (see *"Controlling ISP Redundancy from CLI" on page 201*).

  f. In the **Next Hop IP Address** field:

     - If the Cluster object has at least two interfaces with the Topology "**External**" in the **Network Management** page, leave this field empty and click **Get from routing table**. The next hop is the default gateway.
     - If the Cluster object has only one interface with the Topology "**External**" in the **Network Management** page, enter the corrent IP address of the next hop.

g. If earlier you selected the **Load Sharing** mode, then in the **Weight** field, enter the applicable value.

   For equal traffic distribution between the IPS links, enter the applicable ratio in each ISP link (100% / Number of ISP Links):

   - For two ISP links, enter **50** in each.
   - For three ISP links, enter **33** in each.
   - For four ISP links, enter **25** in each.
   - and so on.

   If one ISP link is faster, increase this value and decrease it for the other ISP links, so that the sum of these values is always equal 100.

h. **Optional:** Click the **Advanced** tab and configure hosts to be monitored, to make sure the link is working.

   Add the applicable host objects in the **Selected hosts** section.

i. Click **OK**.

8. Configure the Cluster to be the DNS server.

   **Procedure**

   The Cluster, or a DNS server behind it, must respond to DNS queries.

   It resolves IP addresses of servers in the DMZ (or another internal network).

   Get a public IP address from each ISP. If public IP addresses are not available, register the domain to make the DNS server accessible from the Internet.

   The Cluster intercepts DNS queries "Type A" for the web servers in its domain that come from external hosts.

   - If the Cluster recognizes the external host, it replies:
     - In ISP Redundancy **Load Sharing** mode, the Cluster replies with IP addresses of all ISP links, alternating their order.
     - In ISP Redundancy **Primary/Backup** mode, the Cluster replies with the IP addresses of the active ISP link.
   - If the Cluster does not recognize the host, it passes the DNS query on to the original destination, or to the domain DNS server.

**To enable the DNS server:**

a. Click **Other** > **ISP Redundancy**.

b. Select **Enable DNS Proxy**.

c. Click **Configure**.

d. Add your DMZ or Web servers.

Configure each server with a public IP address from each ISP.

e. In the **DNS TTL**, enter a number of seconds.

This sets a Time To Live for each DNS reply.

DNS servers in the Internet cannot cache your DNS data in the reply for longer than the TTL.

f. Click **OK**.

g. Configure Static NAT to translate the public IP addresses to the real server's IP address.

External clients use one of the configured IP addresses.

**Note** - If the servers use different services (for example, HTTP and FTP), you can use NAT for only the configured public IP addresses.

h. Define an Access Control Policy rule:

| Name | Source | Destination | VPN | Services & Applications | Action | Track | Install On |
|---|---|---|---|---|---|---|---|
| DNS Proxy | Applicable sources | Applicable DNS Servers | Any | domain_udp | Accept | None | Policy Targets |

**To register the domain and get IP addresses:**

a. Register your domain with each ISP.

b. Tell the ISP the configured IP addresses of the DNS server that respond to DNS queries for the domain.

c. For each server in the DMZ, get a public IP address from each ISP.

d. In SmartConsole, click **Menu** > **Global properties**.

e. From the left tree, click **NAT - Network Address Translation**.

f. In the **Manual NAT rules** section, select **Translate destination on client side**.

g. Click **OK**.

9. Configure the Access Control Policy for ISP Redundancy.

**Procedure**

The Access Control Policy must allow connections through the ISP links, with Automatic Hide NAT on network objects that start outgoing connections.

a. In the properties of the object for an internal network, select **NAT** > **Add Automatic Address Translation Rules**.

b. Select **Hide behind the gateway**.

c. Click **OK**.

d. Define rules for publicly reachable servers (Web servers, DNS servers, DMZ servers).

- If you have one public IP address from each ISP for the Cluster, define Static NAT.

  Allow specific services for specific servers.

  For example, configure NAT rules, so that incoming HTTP connections from your ISPs reach a Web server, and DNS connections from your ISPs reach the DNS server.

  **Example: Manual Static Rules for a Web Server and a DNS Server**

| Original Source | Original Destination | Original Services | Translated Source | Translated Destination | Translated Services | Install On | Comment |
|---|---|---|---|---|---|---|---|
| Any | Host object with IP address of Web Server | http | = Original | S 50.50.50.2 | = Original | Policy Targets | Incoming Web - ISP A |
| Any | Host object with IP address of Web Server | http | = Original | S 60.60.60.2 | = Original | Policy Targets | Incoming Web - ISP B |
| Any | Host object with IP address of DNS Server | domain_udp | = Original | S 50.50.50.3 | = Original | Policy Targets | Incoming DNS - ISP A |
| Any | Host object with IP address of DNS Server | domain_udp | = Original | S 60.60.60.3 | = Original | Policy Targets | Incoming DNS - ISP B |

- If you have a public IP address from each ISP for each publicly reachable server (in addition to the Cluster), configure the applicable NAT rules:

  i. Give each server a private IP address.

  ii. Use the public IP addresses in the **Original Destination**.

  iii. Use the private IP address in the **Translated Destination**.

  iv. Select **Any** as the **Original Service**.

**Note** - If you use Manual NAT, then automatic ARP does not work for the IP addresses behind NAT. You must configure the `local.arp` file as described in sk30197.

10. Install the Access Control Policy on this Cluster object.

# ISP Redundancy and VPN

**Note** - ISP Redundancy settings override the **VPN Link Selection** settings.

When ISP Redundancy is enabled, VPN encrypted connections survive a failure of an ISP link.

The settings in the ISP Redundancy page override settings in the **IPsec VPN > Link Selection** page.

**Configuring ISP Redundancy for VPN with a Check Point peer**

| Step | Instructions |
|---|---|
| 1 | Connect with SmartConsole to the Security Management Server or Domain Management Server that manages this Cluster. |
| 2 | From the left navigation panel, click **Gateways & Servers**. |
| 3 | Open the Cluster object. |
| 4 | In the left navigation tree, go to **Other** > **ISP Redundancy**. |
| 5 | Select **Apply settings to VPN traffic**. |
| 6 | In the left navigation tree, go to **IPsec VPN** > **Link Selection**. |
| 7 | Make sure that **Use ongoing probing. Link redundancy mode** shows the mode of the ISP Redundancy:<br>**High Availability** (for Primary/Backup) or **Load Sharing**.<br>The **VPN Link Selection** now only probes the ISP configured in ISP Redundancy. |

**Configuring ISP Redundancy for VPN with a third-party peer**

If the VPN peer is **not** a Check Point Security Gateway, the VPN may fail, or the third-party device may continue to encrypt traffic to a failed ISP link.

- Make sure the third-party VPN peer recognizes encrypted traffic from the secondary ISP link as coming from the Check Point cluster.
- Change the configuration of ISP Redundancy to **not** use these Check Point technologies:

- **Use Probing** - Makes sure that **Link Selection** uses another option.
- The options **Load Sharing**, **Service Based Link Selection**, and **Route based probing** work only on Check Point Security Gateways/ Clusters / Security Groups.

  If used, the Security Gateway / Cluster Members / Security Group uses one link to connect to the third-party VPN peer.

  The link with the highest prefix length and lowest metric is used.

# Controlling ISP Redundancy from CLI

You can control the ISP Redundancy behavior from CLI.

## Force ISP Link State

Use the "`fw isp_link`" command to force the ISP link state to Up or Down.

Use this to test installation and deployment, or to force the Cluster Members to recognize the true link state if it cannot (the ISP link is down but the gateway sees it as up).

- You can run this command on the Cluster Members:

```
fw isp_link <Name of ISP Link in SmartConsole> {up | down}
```

- You can run this command on the Security Management Server:

```
fw isp_link <Name of Cluster Member Object> <Name of ISP Link
in SmartConsole> {up | down}
```

For more information, see the *R81.10 CLI Reference Guide* > Chapter *Security Gateway Commands* - Section *fw* - Section *fw isp_link*.

## The ISP Redundancy Script

When the Cluster Members starts, or an ISP link state changes, the `$FWDIR/bin/cpisp_update` script runs on the Cluster Member.

This script changes the default route of the Cluster Members.

**Warning** - We do **not** recommend that you make any changes to this script.

# Dynamic Routing Protocols in a Cluster Deployment

ClusterXL supports Dynamic Routing (Unicast and Multicast) protocols as an integral part of Gaia Operating System.

As the network infrastructure views the clustered Security Gateway as a single logical entity, failure of a Cluster Member will be transparent to the network infrastructure and will not result in a ripple effect.

### Router IP Address

All Cluster Members use the Cluster Virtual IP address(es) as Router IP address(es).

### Routing Table Synchronization

Routing information is synchronized among the Cluster Members using the Forwarding Information Base (FIB) Manager process.

This is done to prevent traffic interruption in case of failover, and used for Load Sharing and High Availability modes.

The FIB Manager is the responsible for the routing information.

The FIB Manager is registered as a Critical Device called "**FIB**". If the routing database goes out of sync, this Critical Device reports its state as "`problem`". As a result, the Cluster Member changes its state to "DOWN" until the FIB Manager is synchronized.

### Wait for Clustering

- When Dynamic Routing protocols and/or DHCP Relay are configured on cluster, the "**Wait for Clustering**" option is **enabled automatically** in these cluster modes:
  - ClusterXL High Availability
  - ClusterXL Load Sharing Unicast
  - ClusterXL Load Sharing Multicast
  - VSX High Availability
  - VSX Load Sharing (VSLS)
- When Dynamic Routing protocols and/or DHCP Relay are configured on cluster, the "**Wait for Clustering**" is **disabled automatically** in these cluster modes:
  - VRRP Cluster on Gaia OS

### Failure Recovery

Dynamic Routing on ClusterXL avoids creating a ripple effect upon failover by informing the neighboring routers that the router has exited a maintenance mode.

The neighboring routers then reestablish their relationships to the cluster, without informing the other routers in the network.

These restart protocols are widely adopted by all major networking vendors.

This table lists the RFC and drafts compliant with Check Point Dynamic Routing:

| Protocol | RFC or Draft |
|---|---|
| OSPF Graceful restart | RFC 3623 |
| OSPF LLS | RFC 5613 |
| BGP Graceful restart | RFC 4724 |

# ClusterXL Failover based on the Load on ClusterXL SND Instances

**Note** - This feature is available starting from the R81.10 Jumbo Hotfix Accumulator Take 177.

## Overview

You can configure Cluster Members to monitor the CPU cores that run CoreXL SND instances separately from the CPU cores that run CoreXL Firewall instances.

As a result, Cluster Members can fail over based on these:

1. How many CPU cores that run CoreXL SND instances are under load as a specific number instead of a percentage of the total number of CPU cores.
2. The load threshold for CPU cores that run CoreXL SND instances.
3. The load duration for CPU cores that run CoreXL SND instances.

**Behavior without this feature (default):**

1. The Thread Blocker detects a soft lockup.
2. The Thread Blocker extracts the backtrace to this file: `/var/log/thread_blocker_device64.log`
3. CPU cores that run CoreXL SND instances are monitored by the same threshold as all other CPU cores (as defined in the kernel parameter "`fwha_cpu_utilization_threshold`").
4. The Critical Device "`cpu_utilization`" reports its state as "`problem`" when a certain percentage of all CPU cores are under high load (without specific control of the CoreXL SND instances).
5. ClusterXL uses the kernel parameter "`fwha_cpu_utilization_duration`" to calculate how long a load condition must continue before the Critical Device "`cpu_utilization`" reports its state as "`problem`", without specific control of CoreXL SND instances.
6. The command "`cphaprob list`" does not show which type of CPU core (CoreXL Firewall instance or CoreXL SND instance) caused the Critical Device "`cpu_utilization`" to report its state as "`problem`".
7. It is possible to decide whether to enable and monitor the Critical Device "`cpu_utilization`".

**Behavior when this feature is enabled and configured:**

1. The Thread Blocker detects a soft lockup.
2. The Thread Blocker extracts the backtrace to this file: `/var/log/thread_blocker_device64.log`
3. The Thread Blocker triggers a failover to the next available Cluster Member.
4. It is possible to configure a different threshold load for CPU cores that run CoreXL SND instances (as defined in the kernel parameter "`fwha_snd_utilization_threshold`").
5. The Critical Device "`cpu_utilization`" reports its state as "`problem`" when the specified number of CPU cores that run CoreXL SND instances are under load (as defined in the kernel parameter "f`wha_cpus_utilization_monitor_snds_units_count`").
6. It is possible to configure a specific load duration for CPU cores that run CoreXL SND instances (as defined in the kernel parameter "`fwha_snd_utilization_duration`").
7. The "`cphaprob list`" command shows which type of CPU core processor (CoreXL Firewall instance or CoreXL SND instance) caused the Critical Device "`cpu_utilization`" to report its state as "`problem`".
8. It is possible to decide whether to enable and monitor the Critical Device "`cpu_utilization`" and what to monitor - only the CoreXL SND instances, only the CoreXL Firewall instances, both CoreXL SND instances and the CoreXL Firewall instances.

# Configuration

1. Connect to the command line on each Cluster Member.
2. Run these commands:
   a.
   ```
   fw ctl set -f int fwk_thread_blocker_failover_on_timeout 1
   ```
   b.
   ```
   fw ctl set -f int fwha_cpu_utilization_is_blocking 1
   ```
   c.
   ```
   fw ctl set -f int fwha_cpu_utlization_monitor_enable
   ```
3. Run these commands to customize the monitoring of SND CPUs:
   a.
   ```
   fw ctl set -f int fwha_cpu_utlization_monitor_enable {1 | 2 | 3}
   ```
   b.
   ```
   fw ctl set -f int fwha_snd_utilization_threshold_enabled 1
   ```

c.
```
fw ctl set -f int fwha_cpus_utilization_monitor_snds_
units_enable 1
```

4. **Optional**: SND load threshold triggers a Critical Device (default is 90):

```
fw ctl set -f int fwha_snd_utilization_threshold <Percentage>
```

5. **Optional**::

```
fw ctl set -f int fwha_cpus_utilization_monitor_snds_units_
count <Units>
```

6. **Optional**: duration of high load on SNDs that triggers a Critical Device (default is 3):

```
fw ctl set -f int fwha_snd_utilization_duration <Minutes>
```

# Configuration Kernel Parameters

| Parameter | Description |
|---|---|
| `fwk_thread_blocker_failover_on_timeout {0 \| 1}` | Enables (`1`) or disables (`0`, this is the default) the cluster failover trigger when the Thread Blocker detects a soft lockup (freeze). |
| `fwha_snd_utilization_threshold_enabled {0 \| 1}` | Enables (`1`) or disables (`0`, this is the default) the monitoring of CPU cores that run CoreXL SND instances.<br>If you configure the value `0`, then the cluster Critical Device "`cpu_utilization`" reports its state as "`problem`" based on the high CPU utilization threshold for all CPU cores as configured in the kernel parameter "`fwha_cpu_utilization_threshold`". |
| `fwha_snd_utilization_threshold <Percentage>` | Configures the threshold for the CPU utilization (number of percent, default is 90) for CoreXL SND instances.<br>If the CPU utilization exceeds this threshold, the cluster Critical Device "`cpu_utilization`" reports its state as "`problem`". |
| `fwha_cpus_utilization_monitor_snds_units_enable {0 \| 1}` | If you configure the value `0` (this is the default), then the cluster Critical Device "`cpu_utilization`" reports its state as "`problem`" based on the high CPU utilization threshold for all CPU cores (SND and Firewall instances). |

| Parameter | Description |
|---|---|
| `fwha_snd_utilization_duration <Minutes>` | Configures the duration (default is `3`) of high CPU utilization on CPU cores that run CoreXL SND instances.<br>If the duration exceeds this threshold, the cluster Critical Device "`cpu_utilization`" reports its state as "`problem`". |
| `fwha_cpus_utilization_monitor_snds_units_count <Units>` | ■ Specifies the number of SND units whose load must exceed the critical threshold to trigger a Critical Device.<br>■ Defines how many CPU cores running CoreXL SND instances must experience high load to trigger the Critical Device. |
| `fwha_cpu_utlization_monitor_enable {1 \| 2 \| 3}` | Configures which CPU cores the cluster Critical Device "`cpu_utilization`" must monitor:<br>■ `0` - Disable the CPU utilization monitoring.<br>■ `1` - Monitor both CoreXL Firewall instances and SND instances (the cluster Critical Device "`cpu_utilization`" reports its state as "`problem`" when it detects high CPU utilization in either instance type).<br>■ `2` - Monitor only CoreXL SND instances.<br>■ `3` - Monitor only CoreXL Firewall instances. |
| `fwha_cpu_utilization_is_blocking {0 \| 1}` | ■ `0` - When CPU utilization is detected, it will be reported, and a non-blocking Critical Device will be triggered. However, no Cluster Member state will be affected.<br>■ `1` - A blocking Critical Device will be triggered, which will impact the Cluster Member's state and, if necessary, initiate a failover. |

# Configuration Examples

**Example 1**

```
fwha_cpu_utlization_monitor_enable 1

fw ctl set -f int fwha_snd_utilization_threshold_enabled 1

fw ctl set -f int fwha_snd_utilization_threshold 80

fw ctl set -f int fwha_cpu_utilization_threshold 90
```

The cluster Critical Device "`cpu_utilization`" reports its state as "`problem`" when one of these conditions occur:

- The CPU load exceeds 80% on SND CPU cores.
- The CPU load exceeds 90% on Firewall CPU cores.

### Example 2

```
fwha_cpu_utlization_monitor_enable 1

fw ctl set -f int fwha_snd_utilization_threshold_enabled 0

fw ctl set -f int fwha_cpu_utilization_threshold 90
```

The cluster Critical Device "`cpu_utilization`" reports its state as "`problem`" when one of these conditions occur:

- The CPU load exceeds 90% on SND CPU cores (because the default value of "`fwha_snd_utilization_threshold`" is 90).
- The CPU load exceeds 90% on Firewall CPU cores

### Example 3

```
fw ctl set -f int fwha_cpus_utilization_nums_percent 80

fw ctl set -f int fwha_cpus_utilization_monitor_snds_units_enable
1

fw ctl set -f int fwha_cpus_utilization_monitor_snds_units_count 5
```

The cluster Critical Device "`cpu_utilization`" reports its state as "`problem`" when both these conditions occur:

- The CPU load exceeds 80% on CPU cores.
- The CPU load exceeds the threshold on 5 CPU cores that run CoreXL SND instances.

# ClusterXL Configuration Commands

### Description

These commands let you configure internal behavior of the Clustering Mechanism.

**Important:**

- We do not recommend that you run these commands. These commands must be run automatically only by the Security Gateway or the Check Point Support.
- In a Cluster, you must configure all the Cluster Members in the same way.

### Syntax

**Notes:**

- In Gaia Clish:
  Enter the **`set cluster<ESC><ESC>`** to see all the available commands.
- In Expert mode:
  Run the **`cphaconf`** command see all the available commands.
  You can run the `cphaconf` commands only from the Expert mode.
- Syntax legend:
  1. Curly brackets or braces **{ }**:
     Enclose a list of available commands or parameters, separated by the vertical bar **|**, from which user can enter only one.
  2. Angle brackets **< >**:
     Enclose a variable - a supported value user needs to specify explicitly.
  3. Square brackets or brackets **[ ]**:
     Enclose an optional command or parameter, which user can also enter.
- You can include these commands in scripts to run them automatically.
  The meaning of each command is explained in the next sections.

Table: ClusterXL Configuration Commands

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Configure how to show the Cluster Member in local ClusterXL logs - by its Member ID or its Member Name (see *"Configuring the Cluster Member ID Mode in Local Logs" on page 213*) | `set cluster member idmode {id \| name}` | `cphaconf mem_id_mode {id \| name}` |

Table: ClusterXL Configuration Commands (continued)

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Register a single Critical Device (Pnote) on the Cluster Member (see *"Registering a Critical Device" on page 214*) | `N / A` | `cphaconf set_pnote -d <Name of Device> -t <Timeout in Sec> -s {ok\|init\|problem} [-p] [-g] register` |
| Unregister a single Critical Device (Pnote) on the Cluster Member (see *"Unregistering a Critical Device" on page 218*) | `N / A` | `cphaconf set_pnote -d <Name of Device> [-p] [-g] unregister` |
| Report (change) a state in a single Critical Device (Pnote) on the Cluster Member (see *"Reporting the State of a Critical Device" on page 219*) | `N / A` | `cphaconf set_pnote -d <Name of Device> -s {ok\|init\|problem} [-g] report` |
| Register several Critical Devices (Pnotes) from a file on the Cluster Member (see *"Registering Critical Devices Listed in a File" on page 221*) | `N / A` | `cphaconf set_pnote -f <Name of File> [-g] register` |
| Unregister all Critical Devices (Pnotes) on the Cluster Member (see *"Unregistering All Critical Devices" on page 223*) | `N / A` | `cphaconf set_pnote -a [-g] unregister` |
| Configure the Cluster Control Protocol (CCP) Encryption on the Cluster Member (see *"Configuring the Cluster Control Protocol (CCP) Settings" on page 224*) | `set cluster member ccpenc {off \| on}` | `cphaconf ccp_ encrypt {off \| on}`<br>`cphaconf ccp_ encrypt_key <Key String>` |
| Configure the Cluster Forwarding Layer on the Cluster Member (controls the forwarding of traffic between Cluster Members)<br>**Note** - For Check Point use only. | `set cluster member forwarding {off \| on}` | `cphaconf forward {off \| on}` |
| Print the current cluster configuration as loaded in the kernel on the Cluster Member (for details, see sk93306) | `N / A` | `cphaconf debug_data` |

Table: ClusterXL Configuration Commands (continued)

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Start internal failover between subordinate interfaces of specified bond interface - only in Bond High Availability mode (for details, see sk93306) | `N / A` | `cphaconf failover_bond <bond_name>` |
| Configure what happens during a failover after a Bond already failed over internally (for details, see sk93306) | `N / A` | `cphaconf enable_bond_failover <bond_name>` |
| Initiate manual cluster failover (see *"Initiating Manual Cluster Failover" on page 225*) | `set cluster member admin {down \| up}` | `clusterXL_admin {down \| up}` |
| Configure the minimal number of required subordinate interfaces for Bond Load Sharing (see *"Configuring the Minimal Number of Required Subordinate Interfaces for Bond Load Sharing" on page 229*) | `N / A` | `cphaconf bond_ls {set <Bond Name> <Value> \| remove <Bond Name>}` |
| Configuring Link Monitoring on the Cluster Interfaces (see *"Configuring Link Monitoring on the Cluster Interfaces" on page 119*) | `N / A` | `N / A` |
| Configuring the Multi-Version Cluster Mechanism (see *"Configuring the Multi-Version Cluster Mechanism" on page 232*) | `N / A` | `cphaconf mvc {off \| on}` |
| Configuring duration of forced cluster failover in the High Availability mode (see *"Configuring Duration of Forced Failover in High Availability Mode" on page 233*) | `set cluster force_failover member_id <ID> {time <Minutes> on \| off}` | `cphaconf force_failover member_id <ID> {time <Minutes> on \| off}` |

### List of the Gaia Clish "`set cluster`" commands

```
set cluster member admin {down | up} [permanent]
set cluster member ccpenc {off | on}
set cluster member forwarding {off | on}
set cluster member idmode {id | name}
set cluster member mvc {off | on}
set cluster force_failover member_id <ID> {time <Minutes> on |
off}
```

### List of the Expert mode "`cphaconf`" commands

**Note** - Some commands are not applicable to 3rd-party clusters.

```
cphaconf [-D] <options> start
cphaconf stop
cphaconf [-t <Sync IF 1>...] [-d <Non-Monitored IF 1>...] add
cphaconf clear-secured
cphaconf clear-non-monitored
cphaconf debug_data
cphaconf delete_link_local [-vs <VSID>] <IF name>
cphaconf set_link_local [-vs <VSID>] <IF name> <Cluster IP>
cphaconf mem_id_mode {id | name}
cphaconf failover_bond <bond_name>
cphaconf [-s] {set | unset | get} var <Kernel Parameter Name>
[<Value>]
cphaconf bond_ls {set <Bond Name> <Value> | remove <Bond Name>}
cphaconf set_pnote -d <Device> -t <Timeout in sec> -s {ok | init |
problem} [-p] [-g] register
cphaconf set_pnote -f <File> [-g] register
cphaconf set_pnote -d <Device> [-p] [-g] unregister
cphaconf set_pnote -a [-g] unregister
cphaconf set_pnote -d <Device> -s {ok | init | problem} [-g]
report
cphaconf ccp_encrypt {off | on}
cphaconf ccp_encrypt_key <Key String>
cphaconf mvc {on | off}
cphaconf force_failover member_id <ID> {time <Minutes> on | off}
```

**Note** - When a ClusterXL object has only one interface with the role "Cluster" or "Cluster+Sync", and this one interface is a Bond interface, then the Expert mode command "`cphaconf failover_bond <Name of Bond Interface>`" triggers the internal Bond failover only on the ClusterXL member on which you execute this command. The peer ClusterXL members do not automatically perform the interface failover in this Bond. See sk183496.

# Configuring the Cluster Member ID Mode in Local Logs

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

### Description

This command configures how to show the Cluster Member in the local ClusterXL logs - by its Member ID (default), or its Member Name.

This configuration applies to these local logs:

- `/var/log/messages`
- `dmesg`
- `$FWDIR/log/fwd.elg`

See *"Viewing the Cluster Member ID Mode in Local Logs" on page 282*.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `set cluster member idmode {id \| name}` |
| Expert mode | `cphaconf mem_id_mode {id \| name}` |

### Example

```
[Expert@Member1:0]# cphaprob names

Current member print mode in local logs is set to: ID

[Expert@Member1:0]#
[Expert@Member1:0]# cphaconf mem_id_mode name

Member print mode in local logs: NAME

[Expert@Member1:0]#
[Expert@Member1:0]# cphaprob names

Current member print mode in local logs is set to: NAME

[Expert@Member1:0]#
```

# Registering a Critical Device

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

### Description

You can add a user-defined critical device to the default list of critical devices. Use this command to register <device> as a critical process, and add it to the list of devices that must run for the Cluster Member to be considered active. If <device> fails, then the Cluster Member is seen as failed.

If a Critical Device fails to report its state to the Cluster Member in the configured timeout, the Critical Device, and by design the Cluster Member, are seen as failed.

Define the status of the Critical Device that is reported to ClusterXL upon registration.

This initial status can be one of these:

- **ok** - Critical Device is alive.
- **init** - Critical Device is initializing. The Cluster Member is Down. In this state, the Cluster Member cannot become Active.
- **problem** - Critical Device failed. If this state is reported to ClusterXL, the Cluster Member immediately goes Down. This causes a failover.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | N / A |
| Expert mode | `cphaconf set_pnote -d <Name of Critical Device> -t <Timeout in Sec> -s {ok \| init \| problem} [-p] [-g] register` |

**Notes:**

- The "`-t`" flags specifies how frequently to expect the periodic reports from this Critical Device.
  If no periodic reports should be expected, then enter the value 0 (zero).
- The "`-p`" flag makes these changes permanent (survive reboot).
- The "`-g`" flag applies the command to all configured Virtual Systems.

### Restrictions

- Total number of critical devices (pnotes) on Cluster Member is limited to 16.
- Name of any critical device (pnote) on Cluster Member is limited to 15 characters, and must not include white spaces.

### Related topics

- *"Viewing Critical Devices" on page 246*
- *"Reporting the State of a Critical Device" on page 219*
- *"Registering Critical Devices Listed in a File" on page 221*
- *"Unregistering a Critical Device" on page 218*
- *"Unregistering All Critical Devices" on page 223*

### Example use case

This example assumes that all other Critical Devices report their statues as "`ok`".

Do these steps on each Cluster Member:

1. Create a shell script that does these actions:
   a. Examine the used space in the critical disk partitions.
   b. If the used space in these partitions:
      - Is above the threshold, then register a user-defined Critical Device and report its status as "`problem`".

        As a result, the Cluster Member changes its cluster state to "`Down`" (if there is at least one Cluster Member in the status "`Active`").
      - Is below the threshold, then register a user-defined Critical Device and report its status as "`ok`".

        As a result, the Cluster Member changes its cluster state to "`Standby`" or "`Active`".

   Example script syntax:

```
#!/bin/bash

# Get the used % in the partition lv_current
var_lv_current=$(df -h | grep lv_current | awk '{print $5}'|
sed 's/[%]//')
# Get the used % in the partition lv_log
var_lv_log=$(df -h | grep lv_log | awk '{print $5}'| sed 's/
[%]//')

if [ "$var_lv_current" -gt 90 -o "$var_lv_log" -gt 90 ];
    then
        # If the partition is full, then bring the cluster
member down
        $FWDIR/bin/cphaconf set_pnote -d Storage -s problem
report
    else
        # If the partition is free enough, then bring the
cluster member up
        $FWDIR/bin/cphaconf set_pnote -d Storage -s ok report
        $FWDIR/bin/cphaconf set_pnote -d Storage unregister
fi
```

2. Configure a scheduled job to run this shell script at the relevant time intervals.

   See the *R81.10 Gaia Administration Guide*.

# Unregistering a Critical Device

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

### Description

This command unregisters a user-defined Critical Device (Pnote). This means that this device is no longer considered critical.

If a Critical Device was registered with a state "`problem`", before you ran this command, then after you run this command, the status of the Cluster Member depends only on the states of the remaining Critical Devices.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `N/A` |
| Expert mode | `cphaconf set_pnote -d <Name of Critical Device> [-p] [-g] unregister` |

**Notes:**

- The "`-p`" flag makes these changes permanent.
  This means that after you reboot, these Critical Devices remain unregistered.
- The "`-g`" flag applies the command to all configured Virtual Systems.

### Related topics

- *"Viewing Critical Devices" on page 246*
- *"Reporting the State of a Critical Device" on page 219*
- *"Registering a Critical Device" on page 214*
- *"Registering Critical Devices Listed in a File" on page 221*
- *"Unregistering All Critical Devices" on page 223*

# Reporting the State of a Critical Device

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

### Description

This command manually reports (changes) the state of a Critical Device to ClusterXL.

The reported state can be one of these:

- **ok** - Critical Device is alive.
- **init** - Critical Device is initializing. The Cluster Member is Down. In this state, the Cluster Member cannot become Active.
- **problem** - Critical Device failed. If this state is reported to ClusterXL, the Cluster Member immediately goes Down. This causes a failover.

If a Critical Device fails to report its state to the Cluster Member within the defined timeout, the Critical Device, and by design the Cluster Member, are seen as failed. This is true only for Critical Devices with timeouts. If a Critical Device is registered with the "**-t 0**" parameter, there is no timeout. Until the Critical Device reports otherwise, the state of the Critical Device is considered to be the last reported state.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | N / A |
| Expert mode | `cphaconf set_pnote -d <Name of Critical Device> -s {ok \| init \| problem} [-g] report` |

**Notes:**

- The "`-g`" flag applies the command to all configured Virtual Systems.
- If the "`<Name of Critical Device>`" reports its state as "`problem`", then the Cluster Member reports its state as failed.

### Related topics

- *"Viewing Critical Devices" on page 246*
- *"Registering a Critical Device" on page 214*
- *"Registering Critical Devices Listed in a File" on page 221*
- *"Unregistering a Critical Device" on page 218*
- *"Unregistering All Critical Devices" on page 223*

# Registering Critical Devices Listed in a File

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

### Description

This command registers all the user-defined Critical Devices listed in the specified file.

This file must be a plain-text ASCII file, with each Critical Device defined on a separate line.

Each definition must contain three parameters, which must be separated by a space or a tab character:

```
<Name of Device> <Timeout> <Status>
```

Where:

| Parameter | Description |
|---|---|
| *<Name of Device>* | The name of the Critical Device.<br>■ Maximal name length is 15 characters<br>■ The name must not include white spaces (space or tab characters). |
| *<Timeout>* | If the Critical Device *<Name of Device>* fails to report its state to the Cluster Member within this specified number of seconds, the Critical Device (and by design the Cluster Member), are seen as failed.<br>For no timeout, use the value `0` (zero). |
| *<Status>* | The Critical Device *<Name of Device>* reports one of these statuses to the Cluster Member:<br>■ `ok` - Critical Device is alive.<br>■ `init`- Critical Device is initializing. The Cluster Member is Down. In this state, the Cluster Member cannot become Active.<br>■ `problem` - Critical Device failed. If this state is reported to ClusterXL, the Cluster Member immediately goes Down. This causes a failover. |

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | N / A |
| Expert mode | `cphaconf set_pnote -f /<Path>/<Name of File> [-g] register` |

**Note** - The "`-g`" flag applies the command to all configured Virtual Systems.

### Related topics

- *"Viewing Critical Devices" on page 246*
- *"Reporting the State of a Critical Device" on page 219*
- *"Registering a Critical Device" on page 214*
- *"Unregistering a Critical Device" on page 218*
- *"Unregistering All Critical Devices" on page 223*

# Unregistering All Critical Devices

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

### Description

This command unregisters all critical devices from the Cluster Member.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | N / A |
| Expert mode | `cphaconf set_pnote -a [-g] unregister` |

**Notes:**

- The "`-a`" flag specifies that all Pnotes must be unregistered
- The "`-g`" flag applies the command to all configured Virtual Systems

### Related topics

- *"Viewing Critical Devices" on page 246*
- *"Reporting the State of a Critical Device" on page 219*
- *"Registering a Critical Device" on page 214*
- *"Registering Critical Devices Listed in a File" on page 221*
- *"Unregistering a Critical Device" on page 218*

# Configuring the Cluster Control Protocol (CCP) Settings

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

### Description

Cluster Members configure the Cluster Control Protocol (CCP) mode automatically.

**Important** - In R81.10, the CCP always runs in the unicast mode.

You can configure the Cluster Control Protocol (CCP) Encryption on the Cluster Members.

See *"Viewing the Cluster Control Protocol (CCP) Settings" on page 289*.

### Syntax for configuring the Cluster Control Protocol (CCP) Encryption

| Shell | Command |
|---|---|
| Gaia Clish | `set cluster member ccpenc {off | on}` |
| Expert mode | `cphaconf ccp_encrypt {off | on}`<br>`cphaconf ccp_encrypt_key <Key String>` |

# Initiating Manual Cluster Failover

### Description

This command initiates a manual cluster failover (see sk55081 and *"How to Initiate Cluster Failover" on page 298*).

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `set cluster member admin {down \| up}` |
| Expert mode | `clusterXL_admin {down \| up}` |

### Example

```
[Expert@Member1:0]# cphaprob state

Cluster Mode:   High Availability (Active Up) with IGMP Membership

ID         Unique Address  Assigned Load   State          Name

1 (local)  11.22.33.245    100%            ACTIVE         Member1
2          11.22.33.246    0%              STANDBY        Member2


Active PNOTEs: None

... ...

[Expert@Member1:0]#

[Expert@Member1:0]# clusterXL_admin down
This command does not survive reboot. To make the change permanent, please run 'set cluster
member admin down/up permanent' in clish or add '-p' at the end of the command in expert mode
Setting member to administratively down state ...
Member current state is DOWN
[Expert@Member1:0]#

[Expert@Member1:0]# cphaprob state

Cluster Mode:   High Availability (Active Up) with IGMP Membership

ID         Unique Address  Assigned Load   State          Name

1 (local)  11.22.33.245    0%              DOWN           Member1
2          11.22.33.246    100%            ACTIVE         Member2


Active PNOTEs: ADMIN

Last member state change event:
   Event Code:                 CLUS-111400
   State change:               ACTIVE -> DOWN
   Reason for state change:    ADMIN_DOWN PNOTE
   Event time:                 Sun Sep  8 19:35:06 2019

Last cluster failover event:
   Transition to new ACTIVE:   Member 1 -> Member 2
   Reason:                     ADMIN_DOWN PNOTE
   Event time:                 Sun Sep  8 19:35:06 2019

Cluster failover count:
   Failover counter:           2
   Time of counter reset:      Sun Sep  8 16:08:34 2019 (reboot)

[Expert@Member1:0]#

[Expert@Member1:0]# clusterXL_admin up
This command does not survive reboot. To make the change permanent, please run 'set cluster
member admin down/up permanent' in clish or add '-p' at the end of the command in expert mode
Setting member to normal operation ...
Member current state is STANDBY
[Expert@Member1:0]#

[Expert@Member1:0]# cphaprob state

Cluster Mode:   High Availability (Active Up) with IGMP Membership

ID         Unique Address  Assigned Load   State          Name

1 (local)  11.22.33.245    0%              STANDBY        Member1
2          11.22.33.246    100%            ACTIVE         Member2


Active PNOTEs: None

Last member state change event:
```

```
   Event Code:                 CLUS-114802
   State change:               DOWN -> STANDBY
   Reason for state change:    There is already an ACTIVE member in the cluster (member 2)
   Event time:                 Sun Sep  8 19:37:03 2019

Last cluster failover event:
   Transition to new ACTIVE:   Member 1 -> Member 2
   Reason:                     ADMIN_DOWN PNOTE
   Event time:                 Sun Sep  8 19:35:06 2019

Cluster failover count:
   Failover counter:           2
   Time of counter reset:      Sun Sep  8 16:08:34 2019 (reboot)

[Expert@Member1:0]#
```

# Configuring the Minimal Number of Required Subordinate Interfaces for Bond Load Sharing

**Important** - In a Cluster, you must configure all the Cluster Members in the same way.

### Description

ClusterXL considers a bond in Load Sharing mode to be in the "down" state when fewer than a minimal number of required subordinate interfaces stay in the "up" state.

By default, the minimal number of required subordinate interfaces, which must stay in the "up" state in a bond of **n** subordinate interfaces is **n-1**.

If one more subordinate interface fails (when **n-2** subordinate interfaces stay in the "up" state), ClusterXL considers the bond interface to be in the "down" state, even if the bond contains more than two subordinate interfaces.

If a smaller number of subordinate interfaces can pass the expected traffic, you can configure explicitly the minimal number of required subordinate interfaces.

Divide your maximal expected traffic speed by the speed of your subordinate interfaces and round up the result to find an applicable minimal number of required subordinate interfaces.

**Notes**:

- Cluster Members save the configuration in the `$FWDIR/conf/cpha_bond_ls_config.conf` file.
- The commands below save the changes in this file.
- Each line in the file has this syntax:

  ```
  <Name of Bond Interface> <Minimal Number of Required Subordinate Interfaces>
  ```

In addition, see *"Viewing Bond Interfaces" on page 263*.

### Syntax to add the minimal number of required subordinate interfaces for a specific Bond interface

| Shell | Command |
|---|---|
| Gaia Clish | N / A |
| Expert mode | `cphaconf bond_ls set <Name of Bond Interface> <Minimal Number of Required Subordinate Interfaces>` |

### Syntax to remove the configured minimal number of required subordinate interfaces for a specific Bond interface

| Shell | Command |
|---|---|
| Gaia Clish | N / A |
| Expert mode | `cphaconf bond_ls remove <Name of Bond Interface>` |

### Syntax to see the current configuration of the minimal number of required subordinate interfaces

| Shell | Command |
|---|---|
| Gaia Clish | N / A |
| Expert mode | `cat $FWDIR/conf/cpha_bond_ls_config.conf` |

### Procedure

| Step | Instructions |
|---|---|
| 1 | Connect to the command line on each Cluster Member. |
| 2 | Log in to the Expert mode. |
| 3 | Add or remove the minimal number of required subordinate interfaces for a specific Bond interface:<br>`cphaconf bond_ls set <Bond> <Minimal Number of Subordinate Interfaces>`<br>`cphaconf bond_ls remove <Bond>` |
| 4 | Examine the configuration:<br>`cat $FWDIR/conf/cpha_bond_ls_config.conf` |
| 5 | In SmartConsole, install the Access Control policy on this cluster object. |

## Example

```
[Expert@Member1:0]# cat $FWDIR/conf/cpha_bond_ls_config.conf
# ... (truncated for brevity) ...
# Example:
# bond0 2

[Expert@Member1:0]#

[Expert@Member1:0]# cphaconf bond_ls set bond1 2
Set operation succeeded

[Expert@Member1:0]# cat $FWDIR/conf/cpha_bond_ls_config.conf
# ... (truncated for brevity) ...
# Example:
# bond0 2

bond1 2
[Expert@Member1:0]#

[Expert@Member1:0]# cphaconf bond_ls remove bond1
Remove operation succeeded

[Expert@Member1:0]#

[Expert@Member1:0]# cat $FWDIR/conf/cpha_bond_ls_config.conf
# ... (truncated for brevity) ...
# Example:
# bond0 2

[Expert@Member1:0]#
```

# Configuring the Multi-Version Cluster Mechanism

### Description

This command changes the state of the Multi-Version Cluster (MVC) Mechanism - enable or disable it.

**Important:**

- The MVC Mechanism is *disabled* by default.
- For limitations of the MVC Mechanism, see the *R81.10 Installation and Upgrade Guide* > Chapter *Upgrading Gateways and Clusters* > Section *Upgrading ClusterXL, VSX Cluster, VRRP Cluster* > Section *Multi-Version Cluster Upgrade*.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `set cluster member mvc {off | on}` |
| Expert mode | `cphaconf mvc {off | on}` |

### Parameters

| Parameter | Description |
|---|---|
| `off` | Disables the MVC Mechanism on this Cluster Member. |
| `on` | Enables the MVC Mechanism on this Cluster Member. |

**Notes:**

- This command does *not* provide an output. To view the current state of the MVC Mechanism, see *"Viewing the State of the Multi-Version Cluster Mechanism" on page 290*.
- The change made with this command survives reboot.
- If a specific scenario requires you to disable the MVC Mechanism before the first start of an R81.10 Cluster Member (for example, immediately after an upgrade to R81.10), then disable it before the first policy installation on this Cluster Member.

# Configuring Duration of Forced Failover in High Availability Mode

### Description

In the ClusterXL High Availability mode, you can force the current Active cluster member to fail over to the specified cluster member for the specified number of minutes.

After the specified duration ends, or after you manually disable the forced failover, the cluster recovers based on the mode you configured in the cluster object:

- **Primary Up** - The cluster fails over to the cluster member that is configured as primary (if it is not currently Active).
- **Active Up** - The current Active cluster member remains Active.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `set cluster force_failover member_id <ID> {time <Minutes> on \| off}` |
| Expert mode | `cphaconf force_failover member_id <ID> {time <Minutes> on \| off}` |

### Parameters

| Parameter | Description |
|---|---|
| *<ID>* | Specifies the ID of the cluster member, to which the current Active cluster member must fail over.<br>See *"Viewing Cluster State" on page 240*. |
| `off` | Disables the forced failover. |
| `on` | Enables the forced failover. |

# Monitoring and Troubleshooting Clusters

This section describes how to monitor and troubleshooting clusters.

# ClusterXL Monitoring Commands

### Description

Use the monitoring commands to make sure that the cluster and the Cluster Members work properly, and to define Critical Devices. A Critical Device (also known as a *Problem Notification*, or *pnote*) is a special software device on each Cluster Member, through which the critical aspects for cluster operation are monitored. When the critical monitored component on a Cluster Member fails to report its state on time, or when its state is reported as problematic, the state of that member is immediately changed to 'Down'.

### Syntax

**Notes:**

- In Gaia Clish:
  Enter the **`show cluster<ESC><ESC>`** to see all the available commands.
- In Expert mode:
  Run the **`cphaprob`** command see all the available commands.
  You can run the `cphaprob` commands from Gaia Clish as well.
- Syntax legend:
  1. Curly brackets or braces **{ }**:
     Enclose a list of available commands or parameters, separated by the vertical bar **|**, from which user can enter only one.
  2. Angle brackets **< >**:
     Enclose a variable - a supported value user needs to specify explicitly.
  3. Square brackets or brackets **[ ]**:
     Enclose an optional command or parameter, which user can also enter.
- You can include these commands in scripts to run them automatically.
  The meaning of each command is explained in the next sections.

Table: ClusterXL Monitoring Commands

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Show states of Cluster Members and their names (see *"Viewing Cluster State" on page 240*) | `show cluster state` | `cphaprob [-vs <VSID>] state` |
| Show Critical Devices (Pnotes) and their states on the Cluster Member (see *"Viewing Critical Devices" on page 246*) | `show cluster members pnotes {all \| problem}` | `cphaprob [-l] [-ia] [-e] list` |

Table: ClusterXL Monitoring Commands (continued)

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Show cluster interfaces on the cluster member (see *"Viewing Cluster Interfaces" on page 258*) | `show cluster members interfaces {all \| secured \| virtual \| vlans}` | `cphaprob [-vs all] [-a] [-m] if` |
| Show cluster bond configuration on the Cluster Member (see *"Viewing Bond Interfaces" on page 263*) | `show cluster bond {all \| name <bond_name>}` | `cphaprob show_bond [<bond_name>]` |
| Show groups of bonds on the Cluster Member (see *"Viewing Bond Interfaces" on page 263*) | `N / A` | `cphaprob show_bond_groups` |
| Show (and reset) cluster failover statistics on the Cluster Member (see *"Viewing Cluster Failover Statistics" on page 268*) | `show cluster failover [reset {count \| history}]` | `cphaprob [-reset {-c \| -h}] [-l <count>] show_failover` |
| Show information about the software version (including hotfixes) on the local Cluster Member and its matches/mismatches with other Cluster Members (see *"Viewing Software Versions on Cluster Members" on page 270*) | `show cluster release` | `cphaprob release` |
| Show Delta Sync statistics on the Cluster Member (see *"Viewing Delta Synchronization" on page 271*) | `show cluster statistics sync [reset]` | `cphaprob [-reset] syncstat` |
| Show Delta Sync statistics for the Connections table on the Cluster Member (see *"Viewing Cluster Delta Sync Statistics for Connections Table" on page 279*) | `show cluster statistics transport [reset]` | `cphaprob [-reset] ldstat` |
| Show the Cluster Control Protocol (CCP) mode on the Cluster Member (see *"Viewing Cluster Interfaces" on page 258*) | `show cluster members interfaces virtual` | `cphaprob [-vs all] -a if` |

Table: ClusterXL Monitoring Commands (continued)

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Show the IGMP membership of the Cluster Member (see *"Viewing IGMP Status" on page 278*) | `show cluster members igmp` | `cphaprob igmp` |
| Show cluster unique IP's table on the Cluster Member (see *"Viewing Cluster IP Addresses" on page 280*) | `show cluster members ips`<br>`show cluster members monitored` | `cphaprob tablestat`<br>`cphaprob -m tablestat` |
| Show the Cluster Member ID Mode in local logs - by Member ID (default) or Member Name (see *"Viewing the Cluster Member ID Mode in Local Logs" on page 282*) | `show cluster members idmode` | `cphaprob names` |
| Show interfaces, which the RouteD monitors on the Cluster Member when you configure OSPF (see *"Viewing Interfaces Monitored by RouteD" on page 283*) | `show ospf interfaces [detailed]` | `cphaprob routedifcs` |
| Show roles of RouteD daemon on Cluster Members (see *"Viewing Roles of RouteD Daemon on Cluster Members" on page 284*) | `show cluster roles` | `cphaprob roles` |
| Show Cluster Correction Statistics (see *"Viewing Cluster Correction Statistics" on page 285*) | `N / A` | `cphaprob [{-d \| -f \| -s}] corr` |
| Show the Cluster Control Protocol (CCP) mode (see *"Viewing the Cluster Control Protocol (CCP) Settings" on page 289*) | `show cluster members interfaces virtual` | `cphaprob -a if` |
| Show the Cluster Control Protocol (CCP) Encryption settings (see *"Viewing the Cluster Control Protocol (CCP) Settings" on page 289*) | `show cluster members ccpenc` | `cphaprob ccp_encrypt` |
| Shows the state of the Multi-Version Cluster (see *"Viewing the State of the Multi-Version Cluster Mechanism" on page 290*) | `show cluster members mvc` | `N / A` |
| Show Full Connectivity Upgrade statistics (see *"Viewing Full Connectivity Upgrade Statistics" on page 291*) | `N / A` | `cphaprob fcustat` |

### List of the Gaia Clish "`show cluster`" commands

```
show cluster
      bond
            all
            name <Name of Bond>
      failover
      members
            ccpenc
            idmode
            igmp
            interfaces
                  all
                  secured
                  virtual
                  vlans
            ips
            monitored
            mvc
            pnotes
                  all
                  problem
      release
      roles
      state
      statistics
            sync [reset]
            transport [reset]
```

### List of the Expert mode "cphaprob" commands

**Note** - Some commands are not applicable to 3rd party clusters.

```
cphaprob [-vs <VSID>] state
cphaprob [-reset {-c | -h}] [-l <count>] show_failover
cphaprob names
cphaprob [-reset] [-a] syncstat
cphaprob [-reset] ldstat
cphaprob [-l] [-i[a]] [-e] list
cphaprob [-vs all] [-a] [-m] if
cphaprob show_bond [<bond_name>]
cphaprob show_bond_groups
cphaprob igmp
cphaprob fcustat
cphaprob [-m] tablestat
cphaprob routedifcs
cphaprob roles
cphaprob release
cphaprob ccp_encrypt
cphaprob [{-d | -f | -s}] corr
```

# Viewing Cluster State

### Description

This command monitors the cluster status (after you set up the cluster).

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | 1. `set virtual-system <VSID>`<br>2. `show cluster state` |
| Expert mode | `cphaprob [-vs <VSID>] state` |

### Example

```
Member1> show cluster state

Cluster Mode:   High Availability (Active Up) with IGMP Membership

ID          Unique Address  Assigned Load   State          Name

1 (local)  11.22.33.245    100%            ACTIVE(!)      Member1
2          11.22.33.246    0%              DOWN           Member2


Active PNOTEs: COREXL

Last member state change event:
   Event Code:                 CLUS-116505
   State change:               INIT -> ACTIVE(!)
   Reason for state change:    All other machines are dead (timeout), FULLSYNC PNOTE
   Event time:                 Sun Sep  8 15:28:39 2019
v Cluster failover count:
   Failover counter:           0
   Time of counter reset:      Sun Sep  8 15:28:21 2019 (reboot)

Member1>
```

Description of the "`cphaprob state`" command output fields:

Table: Description of the output fields

| Field | Description |
|---|---|
| **Cluster Mode** | Can be one of these:<br>■ Load Sharing (Multicast).<br>■ Load Sharing (Unicast).<br>■ High Availability (Primary Up).<br>■ High Availability (Active Up).<br>■ Virtual System Load Sharing<br>■ For third-party clustering products: *Service*, refer to Clustering Definitions and Terms, for more information. |
| **ID** | ■ In the High Availability mode - indicates the Cluster Member priority, as configured in the cluster object in SmartConsole.<br>■ In Load Sharing mode - indicates the Cluster Member ID, as configured in the cluster object in SmartConsole. |
| **Unique Address** | Usually, shows the IP addresses of the Sync interfaces.<br>In some cases, can show IP addresses of other cluster interfaces. |
| **Assigned Load** | ■ In the ClusterXL High Availability mode - shows the *Active* Cluster Member with 100% load, and all other *Standby* Cluster Members with 0% load.<br>■ In ClusterXL Load Sharing modes (Unicast and Multicast) - shows all *Active* Cluster Members with 100% load. |
| **State** | ■ In the ClusterXL High Availability mode, only one Cluster Member in a fully-functioning cluster must be *ACTIVE*, and the other Cluster Members must be in the *STANDBY* state.<br>■ In the ClusterXL Load Sharing modes (Unicast and Multicast), all Cluster Members in a fully-functioning cluster must be *ACTIVE*.<br>■ In 3rd-party clustering configuration, all Cluster Members in a fully-functioning cluster must be *ACTIVE*. This is because this command only reports the status of the Full Synchronization process.<br>See the summary table below. |
| **Name** | Shows the names of Cluster Members' objects as configured in SmartConsole. |
| **Active PNOTEs** | Shows the Critical Devices that report theirs states as "`problem`" (see *"Viewing Critical Devices" on page 246*). |

Table: Description of the output fields (continued)

| Field | Description |
|---|---|
| **Last member state change event** | Shows information about the last time this Cluster Member changed its cluster state. |
| **Event Code** | Shows an event code.<br>For information, see sk125152. |
| **State change** | Shows the previous cluster state and the new cluster state of this Cluster Member. |
| **Reason for state change** | Shows the reason why this Cluster Member changed its cluster state. |
| **Event time** | Shows the date and the time when this Cluster Member changed its cluster state. |
| **Last cluster failover event** | Shows information about the last time a cluster failover occurred. |
| **Transition to new ACTIVE** | Shows which Cluster Member became the new *Active*. |
| **Reason** | Shows the reason for the last cluster failover. |
| **Event time** | Shows the date and the time of the last cluster failover. |
| **Cluster failover count** | Shows information about the cluster failovers. |
| **Failover counter** | Shows the number of cluster failovers since the boot.<br>Notes:<br>■ This value survives reboot.<br>■ This counter is synchronized between Cluster Members. |
| **Time of counter reset** | Shows the date and the time of the last counter reset, and the reset initiator. |

When you examine the state of the Cluster Member, consider whether it forwards packets, and whether it has a problem that prevents it from forwarding packets. Each state reflects the result of a test on critical devices. This table shows the possible cluster states, and whether or not they represent a problem.

Table: Description of the cluster states

| Cluster State | Description | Forwarding packets? | Is this state a problem? |
|---|---|---|---|
| **ACTIVE** | Everything is OK. | Yes | No |
| **ACTIVE(!)**<br>**ACTIVE(!F)**<br>**ACTIVE(!P)**<br>**ACTIVE (!FP)** | A problem was detected, but the Cluster Member still forwards packets, because it is the only member in the cluster, or because there are no other Active members in the cluster. In any other situation, the state of the member is *Down*.<br>■ `ACTIVE(!)` - See above.<br>■ `ACTIVE(!F)` - See above. Cluster Member is in the freeze state.<br>■ `ACTIVE(!P)` - See above. This is the Pivot Cluster Member in Load Sharing Unicast mode.<br>■ `ACTIVE(!FP)` - See above. This is the Pivot Cluster Member in Load Sharing Unicast mode and it is in the freeze state. | Yes | Yes |
| **DOWN** | One of the Critical Devices reports its state as "`problem`" (see *"Viewing Critical Devices" on page 246*). | No | Yes |
| **LOST** | The peer Cluster Member lost connectivity to this local Cluster Member (for example, while the peer Cluster Member is rebooted). | No | Yes |

Table: Description of the cluster states (continued)

| Cluster State | Description | Forwarding packets? | Is this state a problem? |
|---|---|---|---|
| **READY** | State *Ready* means that the Cluster Member recognizes itself as a part of the cluster and is literally ready to go into action, but, by design, something prevents it from taking action. Possible reasons that the Cluster Member is not yet Active include:<br>■ Not all required software components were loaded and initialized yet and/or not all configuration steps finished successfully yet. Before a Cluster Member becomes *Active*, it sends a message to the rest of the Cluster Members, to check if it can become *Active*. In High Availability mode it checks if there is already an Active member and in Load Sharing Unicast mode it checks if there is a Pivot member already. The member remains in the *Ready* state until it receives the response from the rest of the Cluster Members and decides which, which state to choose next (*Active, Standby, Pivot, or non-Pivot*).<br>■ Software installed on this Cluster Member has a higher version than all the other Cluster Members. For example, when a cluster is upgraded from one version of Check Point Security Gateway to another, and the Cluster Members have different versions of Check Point Security Gateway, the Cluster Members with the new version have the *Ready* state, and the Cluster Members with the previous version have the *Active/Active Attention* state. | No | No |

Table: Description of the cluster states (continued)

| Cluster State | Description | Forwarding packets? | Is this state a problem? |
|---|---|---|---|
| | This applies only when the Multi-Version Cluster Mechanism is disabled (see *"Viewing the State of the Multi-Version Cluster Mechanism" on page 290*).<br><br>See sk42096 for a solution. | | |
| **STANDBY** | Applies only to a High Availability mode. Means that the Cluster Member waits for an Active Cluster Member to fail in order to start packet forwarding. | No | No |
| **BACKUP** | Applies only to a VSX Cluster in Virtual System Load Sharing mode with three or more Cluster Members configured.<br>State of a Virtual System on a third (and so on) VSX Cluster Member. | No | No |
| **INIT** | The Cluster Member is in the phase after the boot and until the Full Sync completes. | No | No |

# Viewing Critical Devices

### Description

There are a number of built-in Critical Devices, and the Administrator can define additional Critical Devices.

When a Critical Device reports its state as a "`problem`", the Cluster Member reports its state as "`DOWN`".

To see the list of Critical Devices on a Cluster Member, and of all the other Cluster Members, run the commands listed below on the Cluster Member.

Table: Built-in Critical Devices

| Critical Device | Description | Meaning of the "OK" state | Meaning of the "problem" state |
|---|---|---|---|
| `Problem Notification` | Monitors all the Critical Devices. | None of the Critical Devices on this Cluster Member report its state as `problem`. | At least one of the Critical Devices on this Cluster Member reports its state as "`problem`". |
| `Init` | Monitors if "`HA module`" was initialized successfully. See sk36372. | This Cluster Member receives cluster state information from peer Cluster Members. | |
| `Interface Active Check` | Monitors the state of cluster interfaces. | All cluster interfaces on this Cluster Member are up (CCP packets are sent and received on all cluster interfaces). | At least one of the cluster interfaces on this Cluster Member is down (CCP packets are not sent and/or received on time). |

Table: Built-in Critical Devices (continued)

| Critical Device | Description | Meaning of the "OK" state | Meaning of the "problem" state |
|---|---|---|---|
| `Load Balancing Configuration` | Currently is not used (see sk36373). | | |
| `Recovery Delay` | Monitors the state of a Virtual System (see sk92353). | State of a Virtual System can be changed on this Cluster Member. | State of a Virtual System cannot be changed yet on this Cluster Member. |
| `CoreXL Configuration` | Monitors CoreXL configuration for inconsistencies on all Cluster Members. | Number of configured CoreXL Firewall instances on this Cluster Member is the same as on all peer Cluster Members. | Number of configured CoreXL Firewall instances on this Cluster Member is different from peer Cluster Members. **Important** - A Cluster Member with a greater number of CoreXL Firewall instances changes its state to DOWN. |
| `Fullsync` | Monitors if Full Sync on this Cluster Member completed successfully. | This Cluster Member completed Full Sync successfully. | This Cluster Member was not able to complete Full Sync. |
| `Policy` | Monitors if the Security Policy is installed. | This Cluster Member successfully installed Security Policy. | Security Policy is not currently installed on this Cluster Member. |

Table: Built-in Critical Devices (continued)

| Critical Device | Description | Meaning of the "OK" state | Meaning of the "problem" state |
|---|---|---|---|
| `fwd` | Monitors the Security Gateway process called `fwd`. | `fwd` daemon on this Cluster Member reported its state on time. | `fwd` daemon on this Cluster Member did not report its state on time. |
| `cphad` | Monitors the ClusterXL process called `cphamcset`. also see the `$FWDIR/log/cphamcset.elg` file. | `cphamcset` daemon on this Cluster Member reported its state on time. | `cphamcset` daemon on this Cluster Member did not report its state on time. |
| `routed` | Monitors the Gaia process called `routed`. | `routed` daemon on this Cluster Member reported its state on time. | `routed` daemon on this Cluster Member did not report its state on time. |
| `cvpnd` | Monitors the Mobile Access back-end process called `cvpnd`.<br>This pnote appears if Mobile Access Software Blade is enabled. | `cvpnd` daemon on this Cluster Member reported its state on time. | `cvpnd` daemon on this Cluster Member did not report its state on time. |
| `ted` | Monitors the Threat Emulation process called `ted`. | `ted` daemon on this Cluster Member reported its state on time. | `ted` daemon on this Cluster Member did not report its state on time. |

Table: Built-in Critical Devices (continued)

| Critical Device | Description | Meaning of the "OK" state | Meaning of the "problem" state |
|---|---|---|---|
| `VSX` | Monitors all Virtual Systems in VSX Cluster. | On VS0, means that states of all Virtual Systems are not `Down` on this Cluster Member. On other Virtual Systems, means that VS0 is alive on this Cluster Member. | Minimum of blocking states of all Virtual Systems is not "active" (the VSIDs will be printed on the line `Problematic VSIDs:`) on this Cluster Member. |
| `VSX Config` | Monitors the consistency and health of the VSX configuration. | All Virtual Systems (VS) and Virtual Routers are properly configured and synchronized. | There is a mismatch or inconsistency in the VSX configuration between cluster members. |

Table: Built-in Critical Devices (continued)

| Critical Device | Description | Meaning of the "OK" state | Meaning of the "problem" state |
|---|---|---|---|
| `Instances` | This Critical Device appears in VSX HA mode (not VSLS) cluster. | The number of CoreXL Firewall instances in the received CCP packet matches the number of loaded CoreXL Firewall instances on this VSX Cluster Member or this Virtual System. | There is a mismatch between the number of CoreXL Firewall instances in the received CCP packet and the number of loaded CoreXL Firewall instances on this VSX Cluster Member or this Virtual System (see sk106912). |
| `Hibernating` | This pnote appears in VSX VSLS mode cluster with 3 and more Cluster Members. This pnote shows if this Virtual System is in "Backup" (hibernated) state. Also see sk114557. | This Virtual System is in "Backup" (hibernated) state on this Cluster Member. | |
| `admin_down` | Monitors the Critical Device "`admin_down`". | | User ran the `clusterXL_admin down` command on this Cluster Member. See *"The clusterXL_admin Script" on page 325*. |
| `host_monitor` | Monitors the Critical Device "`host_monitor`". User executed the `$FWDIR/bin/clusterXL_monitor_ips` script. See *"The clusterXL_monitor_ips Script" on page 329*. | All monitored IP addresses on this Cluster Member replied to pings. | At least one of the monitored IP addresses on this Cluster Member did not reply to at least one ping. |

Table: Built-in Critical Devices (continued)

| Critical Device | Description | Meaning of the "OK" state | Meaning of the "problem" state |
|---|---|---|---|
| A name of a user space process (except `fwd`, `routed`, `cvpnd`, `ted`) | Administrator executed the `$FWDIR/bin/clusterXL_monitor_process` script. See *"The clusterXL_monitor_process Script" on page 333*. | All monitored user space processes on this Cluster Member are running. | At least one of the monitored user space on this Cluster Member processes is not running. |

Table: Built-in Critical Devices (continued)

| Critical Device | Description | Meaning of the "OK" state | Meaning of the "problem" state |
|---|---|---|---|
| `Local Probing` | Monitors the probing mechanism on the cluster interfaces (see the term **Probing** in the *"Glossary" on page 354*). | CCP packets are received on all cluster interfaces. | At least one of the cluster interfaces on this Cluster Member does not (or did not) **receive** CCP packets for 5 seconds.<br>The probing started for the network connected to the affected interface. |

Table: Built-in Critical Devices (continued)

| Critical Device | Description | Meaning of the "OK" state | Meaning of the "problem" state |
|---|---|---|---|
| | | | |

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster members pnotes {all \| problem}` |
| Expert mode | `cphaprob [-l] [-ia] [-e] list` |

Where:

| Command | Description |
|---|---|
| `show cluster members pnotes all` | Shows the list of all Critical Devices |
| `show cluster members pnotes problem` | Shows the list of all the "`Built-in Devices`" and the "`Registered Devices`" that report their state as "problem" |
| `cphaprob -l` | Shows the list of all Critical Devices |
| `cphaprob -i list` | When there are no issues on the Cluster Member, shows: `There are no pnotes in problem state`<br>When a Critical Device reports a problem, prints only the Critical Device that reports its state as "`problem`". |
| `cphaprob -ia list` | When there are no issues on the Cluster Member, shows: `There are no pnotes in problem state`<br>When a Critical Device reports a problem, prints the Critical Device "`Problem Notification`" and the Critical Device that reports its state as "`problem`" |
| `cphaprob -e list` | When there are no issues on the Cluster Member, shows: `There are no pnotes in problem state`<br>When a Critical Device reports a problem, prints only the Critical Device that reports its state as "`problem`" |

### Related topics

- *"Reporting the State of a Critical Device" on page 219*
- *"Registering a Critical Device" on page 214*
- *"Registering Critical Devices Listed in a File" on page 221*
- *"Unregistering a Critical Device" on page 218*
- *"Unregistering All Critical Devices" on page 223*

## Examples

### Example 1 - Critical Device '`fwd`'

Critical Device `fwd` reports its state as `problem` because the `fwd` process is down.

```
[Expert@Member1:0]# cphaprob -l list

Built-in Devices:

Device Name: Interface Active Check
Current state: OK

Device Name: Recovery Delay
Current state: OK

Device Name: CoreXL Configuration
Current state: OK

Registered Devices:

Device Name: Fullsync
Registration number: 0
Timeout: none
Current state: OK
Time since last report: 1753.7 sec

Device Name: Policy
Registration number: 1
Timeout: none
Current state: OK
Time since last report: 1753.7 sec

Device Name: routed
Registration number: 2
Timeout: none
Current state: OK
Time since last report: 940.3 sec

Device Name: fwd
Registration number: 3
Timeout: 30 sec
Current state: problem
Time since last report: 1782.9 sec
Process Status: DOWN

Device Name: cphad
Registration number: 4
Timeout: 30 sec
Current state: OK
Time since last report: 1778.3 sec
Process Status: UP

Device Name: VSX
Registration number: 5
Timeout: none
Current state: OK
Time since last report: 1773.3 sec

Device Name: Init
Registration number: 6
Timeout: none
Current state: OK
Time since last report: 1773.3 sec

[Expert@Member1:0]#
```

### Example 2 - Critical Device '`CoreXL Configuration`'

Critical Device `CoreXL Configuration` reports its state as `problem` because the numbers of CoreXL Firewall instances do not match between the Cluster Members.

```
[Expert@Member1:0]# cphaprob -l list

Built-in Devices:

Device Name: Interface Active Check
Current state: OK

Device Name: Recovery Delay
Current state: OK

Device Name: CoreXL Configuration
Current state: problem (non-blocking)

Registered Devices:

Device Name: Fullsync
Registration number: 0
Timeout: none
Current state: OK
Time since last report: 1753.7 sec

Device Name: Policy
Registration number: 1
Timeout: none
Current state: OK
Time since last report: 1753.7 sec

Device Name: routed
Registration number: 2
Timeout: none
Current state: OK
Time since last report: 940.3 sec

Device Name: fwd
Registration number: 3
Timeout: 30 sec
Current state: OK
Time since last report: 1782.9 sec
Process Status: UP

Device Name: cphad
Registration number: 4
Timeout: 30 sec
Current state: OK
Time since last report: 1778.3 sec
Process Status: UP

Device Name: VSX
Registration number: 5
Timeout: none
Current state: OK
Time since last report: 1773.3 sec

Device Name: Init
Registration number: 6
Timeout: none
Current state: OK
Time since last report: 1773.3 sec

[Expert@Member1:0]#
```

# Viewing Cluster Interfaces

### Description

This command shows the state of the Cluster Member interfaces and the virtual cluster interfaces.

ClusterXL treats the interfaces as Critical Devices. ClusterXL makes sure that interfaces can send and receive CCP packets.

ClusterXL also sets the required minimal number of functional interfaces to the largest number of functional interfaces ClusterXL detected since the last reboot. If the number of functional interfaces is less than the required number, ClusterXL declares the Cluster Member as failed and starts a failover. The same applies to the synchronization interfaces, where only good synchronization interfaces are counted.

When an interface is DOWN, it means that the interface cannot receive or send CCP packets, or both. An interface may also be able to receive, but not send CCP packets. The time you see in the command's output is the number of seconds that elapsed since the interface was last able to receive or send a CCP packet.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | 1. `set virtual-system <VSID>`<br>2. `show cluster members interfaces {all \| secured \| virtual \| vlans}` |
| Expert mode | `cphaprob [-vs all] [-a] [-m] if` |

Where:

| Command | Description |
|---|---|
| `show cluster members interfaces all` | Shows full list of all cluster interfaces:<br>■ including the number of required interfaces<br>■ including Network Objective<br>■ including VLAN monitoring mode, or list of monitored VLAN interfaces |
| `show cluster members interfaces secured` | Shows only cluster interfaces (Cluster and Sync) and their states:<br>■ without Network Objective<br>■ without VLAN monitoring mode<br>■ without monitored VLAN interfaces |
| `show cluster members interfaces virtual` | Shows full list of cluster virtual interfaces and their states:<br>■ including the number of required interfaces<br>■ including Network Objective<br>■ without VLAN monitoring mode<br>■ without monitored VLAN interfaces |
| `show cluster members interfaces vlans` | Shows only monitored VLAN interfaces |
| `cphaprob if` | Shows only cluster interfaces (Cluster and Sync) and their states:<br>■ without Network Objective<br>■ without VLAN monitoring mode<br>■ without monitored VLAN interfaces |
| `cphaprob -a if` | Shows full list of cluster interfaces and their states:<br>■ including the number of required interfaces<br>■ including Network Objective<br>■ without VLAN monitoring mode<br>■ without monitored VLAN interfaces |

| Command | Description |
|---|---|
| `cphaprob -a -m if` | Shows full list of all cluster interfaces and their states:<br>■ including the number of required interfaces<br>■ including Network Objective<br>■ including VLAN monitoring mode, or list of monitored VLAN interfaces |

### Output

The output of these commands must be identical to the configuration in the cluster object's **Network Management** page in SmartConsole.

### Example

```
[Expert@Member1:0]# cphaprob -a -m if

CCP mode: Manual (Unicast)
Required interfaces: 4
Required secured interfaces: 1


Interface Name:      Status:

eth0                 UP
eth1 (S)             UP
eth2 (LM)            UP
bond1 (LS)           UP

S - sync, LM - link monitor, HA/LS - bond type

Virtual cluster interfaces: 3

eth0            192.168.3.247
eth2            44.55.66.247
bond1           77.88.99.247


No VLANs are monitored on the member

[Expert@Member1:0]#
```

Description of the "`cphaprob -a -m if`" command output fields:

Table: Description of the output fields

| Field, or Text | Description |
|---|---|
| **CCP mode** | Shows the CCP mode.<br>The default mode is `Unicast`.<br>**Important** - In R81.10, the CCP always runs in the unicast mode. |

Table: Description of the output fields (continued)

| Field, or Text | Description |
|---|---|
| **Required interfaces** | Shows the total number of monitored cluster interfaces, including the Sync interface.<br>This number is based on the configuration of the cluster object > **Network Management** page. |
| **Required secured interfaces** | Shows the total number of the required Sync interfaces.<br>This number is based on the configuration of the cluster object > **Network Management** page. |
| **Non-Monitored** | This means that Cluster Member does not monitor the state of this interface.<br>In SmartConsole, in the cluster object > **Network Management** page, administrator configured the Network Type **Private** for this interface. |
| **UP** | This means that Cluster Member monitors the state of this interface.<br>The current cluster state of this interface is UP, which means this interface can send and receive CCP packets.<br>In SmartConsole, in the cluster object > **Network Management** page, administrator configured one of these Network Types for this interface: **Cluster**, **Sync**, or **Cluster + Sync**. |
| **DOWN** | This means that Cluster Members monitors the state of this interface.<br>The current cluster state of this interface is DOWN, which means this interface cannot send CCP packets, receive CCP packets, or both.<br>In SmartConsole, in the cluster object > **Network Management** page, administrator configured one of these Network Types for this interface: **Cluster**, **Sync**, or **Cluster + Sync**. |
| **(S)** | This interface is a Sync interface.<br>In SmartConsole, in the cluster object > **Network Management** page, administrator configured one of these Network Types for this interface: **Sync**, or **Cluster + Sync**. |

Table: Description of the output fields (continued)

| Field, or Text | Description |
|---|---|
| **(LM)** | This interface is configured in the `$FWDIR/conf/cpha_link_monitoring.conf` file.<br>Cluster Member monitors only the link on this interface (does **not** monitor the received or sent CCP packets).<br>See *"Configuring Link Monitoring on the Cluster Interfaces" on page 119*. |
| **(HA)** | This interface is a Bond interface in High Availability mode. |
| **(LS)** | This interface is a Bond interface in Load Sharing mode. |
| **Virtual cluster interfaces** | Shows the total number of the configured virtual cluster interfaces.<br>This number is based on the configuration of the cluster object > **Network Management** page. |
| **No VLANs are monitored on the member** | Shows the VLAN monitoring mode - there are no VLAN interfaces configured on the cluster interfaces. |
| **Monitoring mode is Monitor all VLANs: All VLANs are monitored** | Shows the VLAN monitoring mode - there are some VLAN interfaces configured on the cluster interfaces, and Cluster Member monitors all VLAN IDs. |
| **Monitoring mode is Monitor specific VLAN: Only specified VLANs are monitored** | Shows the VLAN monitoring mode - there are some VLAN interfaces configured on the cluster interfaces, and Cluster Member monitors only specific VLAN IDs. |

# Viewing Bond Interfaces

### Description

This command shows the configuration of bond interfaces and their subordinate interfaces.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | 1. `show cluster bond {all \| name <bond_name>}`<br>2. `show bonding groups` |
| Expert mode | `cphaprob show_bond [<bond_name>]`<br>`cphaprob show_bond_groups` |

Where:

| Command | Description |
|---|---|
| `show cluster bond all`<br>`show bonding groups`<br>`cphaprob show_bond` | Shows configuration of all configured bond interfaces |
| `show cluster bond name <bond_name>`<br>`cphaprob show_bond <bond_name>` | Shows configuration of the specified bond interface |
| `cphaprob show_bond_groups` | Shows the configured Groups of Bonds and their settings. |

## Examples

### Example 1 - 'cphaprob show_bond'

```
[Expert@Member2:0]# cphaprob show_bond

                                    |Slaves     |Slaves  |Slaves
Bond name  |Mode                |State |configured |link up |required
-----------+--------------------+------+-----------+--------+--------
bond1      | High Availability | UP   | 2         | 2      | 1

Legend:
-------
UP!               - Bond interface state is UP, yet attention is required
Slaves configured - number of slave interfaces configured on the bond
Slaves link up    - number of operational slaves
Slaves required   - minimal number of operational slaves required for bond to be UP

[Expert@Member2:0]#

Member2> show bonding groups
Bonding Interface: 1
    Bond Configuration
        xmit-hash-policy Not configured
        down-delay 200
        primary Not configured
        lacp-rate Not configured
        mode active-backup
        up-delay 200
        mii-interval 100
        Bond Interfaces
            eth3
            eth4
Member2>
```

Description of the output fields for the "`cphaprob show_bond`" and "`show cluster bond all`" commands:

Table: Description of the output fields

| Field | Description |
|---|---|
| **Bond name** | Name of the Gaia bonding group. |
| **Mode** | Bonding mode of this Gaia bonding group.<br>One of these:<br>■ `High Availability`<br>■ `Load Sharing` |
| **State** | State of the Gaia bonding group:<br>■ `UP` - Bond interface is fully operational<br>■ `UP!` - Bond interface state is UP, yet attention is required<br>■ `DOWN` - Bond interface failed |
| **Slaves configured** | Total number of physical subordinate interfaces configured in this Gaia bonding group. |

Table: Description of the output fields (continued)

| Field | Description |
|---|---|
| **Slaves link up** | Number of operational physical subordinate interfaces in this Gaia bonding group. |
| **Slaves required** | Minimal number of operational physical subordinate interfaces required for the state of this Gaia bonding group to be UP. |

**Example 2 - 'cphaprob show_bond <bond_name>'**

```
[Expert@Member2:0]# cphaprob show_bond bond1

Bond name:      bond1
Bond mode:      High Availability
Bond status:    UP

Configured slave interfaces: 2
In use slave interfaces:     2
Required slave interfaces:   1

Slave name      | Status          | Link
----------------+-----------------+-------
eth4            | Active          | Yes
eth3            | Backup          | Yes

[Expert@Member2:0]#
```

Description of the output fields for the "`cphaprob show_bond <bond_name>`" and "`show cluster bond name <bond_name>`" commands:

Table: Description of the output fields

| Field | Description |
|---|---|
| **Bond name** | Name of the Gaia bonding group. |
| **Bond mode** | Bonding mode of this Gaia bonding group. One of these:<br>■ `High Availability`<br>■ `Load Sharing` |
| **Bond status** | Status of the Gaia bonding group. One of these:<br>■ `UP` - Bond interface is fully operational<br>■ `UP!` - Bond interface state is UP, yet attention is required<br>■ `DOWN` - Bond interface failed |
| **Configured slave interfaces** | Total number of physical subordinate interfaces configured in this Gaia bonding group. |
| **In use slave interfaces** | Number of operational physical subordinate interfaces in this Gaia bonding group. |

Table: Description of the output fields (continued)

| Field | Description |
|---|---|
| **Required slave interfaces** | Minimal number of operational physical subordinate interfaces required for the state of this Gaia bonding group to be UP. |
| **Slave name** | Names of physical subordinate interfaces configured in this Gaia bonding group. |
| **Status** | Status of physical subordinate interfaces in this Gaia bonding group. One of these:<br>■ `Active` - In High Availability or Load Sharing bonding mode. This subordinate interface is currently handling traffic.<br>■ `Backup` - In High Availability bonding mode only. This subordinate interface is ready and can support internal bond failover.<br>■ `Not Available` - In High Availability or Load Sharing bonding mode. The physical link on this subordinate interface is lost, or this Cluster Member is in status *Down*. The bond cannot failover internally in this state. |
| **Link** | State of the physical link on the physical subordinate interfaces in this Gaia bonding group. One of these:<br>■ `Yes` - Link is present<br>■ `No` - Link is lost |

**Example 3 - 'cphaprob show_bond_groups'**

```
[Expert@Member2:0]# cphaprob show_bond_groups

                                | Required     | Bonds    | Bonds
Group of bonds name | State     | active bonds | in group | status
--------------------+-----------+--------------+----------+--------+
GoB0                | UP        | 1            |          |
                    |           |              | bond1    |   UP
                    |           |              | bond2    |   UP


Legend:
---------
Bonds in group        - a list of the bonds in the bond group
Required active bonds - number of required active bonds
[Expert@Member2:0]#
```

Description of the output fields for the "`cphaprob show_bond_groups`" command:

Table: Description of the output fields

| Field | Description |
|---|---|
| **Group of bonds name** | Name of the Group of Bonds. |
| **State** | State of the Group of Bonds. One of these:<br>■ `UP` - Group of Bonds is fully operational<br>■ `DOWN` - Group of Bonds failed |
| **Required active bonds** | Number of required active bonds in this Group of Bonds. |
| **Bonds in group** | Names of the Gaia bond interfaces configured in this Group of Bonds. |
| **Bonds status** | State of the Gaia bond interface. One of these:<br>■ `UP` - Bond interface is fully operational<br>■ `DOWN` - Bond interface failed |

# Viewing Cluster Failover Statistics

### Description

This command shows the cluster failover statistics on the Cluster Member:

- Number of failovers that happened
- Failover reason
- The time of the last failover event

### Syntax to show the statistics

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster failover` |
| Expert mode | `cphaprob [-l <number>] show_failover` |

### Syntax to reset the statistics

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster failover reset {count \| history}` |
| Expert mode | `cphaprob -reset {-c \| -h} show_failover` |

### Parameters

| Parameter | Description |
|---|---|
| `-l <number>` | Specifies how many of last failover events to show (between 1 and 50) |
| `count`<br>`-c` | Resets the counter of failover events |
| `history`<br>`-h` | Resets the history of failover events |

## Example

```
[Expert@Member1:0]# cphaprob show_failover

Last cluster failover event:
   Transition to new ACTIVE:   Member 2 -> Member 1
   Reason:                     ADMIN_DOWN PNOTE
   Event time:                 Sun Sep  8 18:21:44 2019

Cluster failover count:
   Failover counter:           1
   Time of counter reset:      Sun Sep  8 16:08:34 2019 (reboot)


Cluster failover history (last 20 failovers since reboot/reset on Sun Sep  8 16:08:34 2019):

 No. Time:                     Transition:           CPU:  Reason:
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - -
 1   Sun Sep  8 18:21:44 2019  Member 2 -> Member 1  01    ADMIN_DOWN PNOTE

[Expert@Member1:0]#
```

# Viewing Software Versions on Cluster Members

### Description

This command shows information about the software version (including private hotfixes) on the local Cluster Member and its matches / mismatches with other Cluster Members.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster release` |
| Expert mode | `cphaprob release` |

### Example

```
[Expert@Member1:0]# cphaprob release

Release:                R80.40 T136

Kernel build:           994000117
FW1 build:              994000116
FW1 private fixes:      None

ID         SW release

1 (local)  R80.40 T136
2          R80.40 T136

[Expert@Member1:0]#
```

# Viewing Delta Synchronization

Heavily loaded clusters and clusters with geographically separated members pose special challenges.

High connection rates, and large distances between the members can lead to delays that affect the operation of the cluster.

Monitor the operation of the State Synchronization mechanism in highly loaded and distributed clusters.

**Perform these troubleshooting steps:**

1. Examine the Delta Sync statistics counters:

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster statistics sync` |
| Expert mode | `cphaprob syncstat` |

2. Change the values of the applicable synchronization global configuration parameters.
3. Reset the Delta Sync statistics counters:

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster statistics sync reset` |
| Expert mode | `cphaprob -reset syncstat` |

4. Examine the Delta Sync statistics to see if the problem is solved.
5. Solve any identified problem.

### Example output of the "`show cluster statistics sync`" and "`cphaprob syncstat`" commands from a Cluster Member:

```
Delta Sync Statistics

Sync status: OK

Drops:
Lost updates.............................  0
Lost bulk update events..................  0
Oversized updates not sent...............  0

Sync at risk:
Sent reject notifications................  0
Received reject notifications............  0

Sent messages:
Total generated sync messages............  26079
Sent retransmission requests.............  0
Sent retransmission updates..............  0
Peak fragments per update................  1

Received messages:
Total received updates...................  3710
Received retransmission requests.........  0

Sync Interface:
Name.....................................  eth1
Link speed...............................  1000Mb/s
Rate.....................................  46000 [Bps]
Peak rate................................  46000 [Bps]
Link usage...............................   0%
Total....................................  376827[KB]

Queue sizes (num of updates):
Sending queue size.......................  512
Receiving queue size.....................  256
Fragments queue size.....................  50

Timers:
Delta Sync interval (ms).................  100

Reset on Sun Sep  8 16:09:15 2019 (triggered by fullsync).
```

Each section of the output is described below.

#### The "Sync status:" section

This section shows the status of the Delta Sync mechanism. One of these:

- `Sync status: OK`
- `Sync status: Off - Full-sync failure`
- `Sync status: Off - Policy installation failure`
- `Sync status: Off - Cluster module not started`
- `Sync status: Off - SIC failure`
- `Sync status: Off - Full-sync checksum error`
- `Sync status: Off - Full-sync received queue is full`

- Sync status: Off - Release version mismatch
- Sync status: Off - Connection to remote member timed-out
- Sync status: Off - Connection terminated by remote member
- Sync status: Off - Could not start a connection to remote member
- Sync status: Off - cpstart
- Sync status: Off - cpstop
- Sync status: Off - Manually disabled sync
- Sync status: Off - Was not able to start for more than X second
- Sync status: Off - Boot
- Sync status: Off - Connectivity Upgrade (CU)
- Sync status: Off - cphastop
- Sync status: Off - Policy unloaded
- Sync status: Off - Hibernation
- Sync status: Off - OSU deactivated
- Sync status: Off - Sync interface down
- Sync status: Fullsync in progress
- Sync status: Problem (Able to send sync packets, unable to receive sync packets)
- Sync status: Problem (Able to send sync packets, saving incoming sync packets)
- Sync status: Problem (Able to send sync packets, able to receive sync packets)
- Sync status: Problem (Unable to send sync packets, unable to receive sync packets)
- Sync status: Problem (Unable to send sync packets, saving incoming sync packets)
- Sync status: Problem (Unable to send sync packets, able to receive sync packets)

### The "Drops:" section

This section shows statistics for drops on the Delta Sync network.

Table: Description of the output fields

| Field | Description |
|---|---|
| **Lost updates** | Shows how many Delta Sync updates this Cluster Member considers as lost (based on sequence numbers in CCP packets).<br>If this counter shows a value greater than 0, this Cluster Member lost Delta Sync updates.<br>**Possible mitigation:**<br>Increase the size of the Sending Queue and the size of the Receiving Queue:<br>■ Increase the size of the Sending Queue, if the counter **Received reject notification** is increasing.<br>■ Increase the size of the Receiving Queue, if the counter **Received reject notification** is not increasing. |
| **Lost bulk update events** | Shows how many times this Cluster Member missed Delta Sync updates. (bulk update = twice the size of the local receiving queue)<br>This counter increases when this Cluster Member receives a Delta Sync update with a sequence number much greater than expected. This probably indicates some networking issues that cause massive packet drops.<br>This counter increases when the amount of missed Delta Sync updates is more than twice the local Receiving Queue Size.<br>**Possible mitigation:**<br>■ If the counter's value is steady, this might indicate a one-time synchronization problem that can be resolved by running manual Full Sync. See sk37029.<br>■ If the counter's value keeps increasing, probable there are some networking issues. Increase the sizes of both the Receiving Queue and Sending Queue. |
| **Oversized updates not sent** | Shows how many oversized Delta Sync updates were discarded before sending them.<br>This counter increases when Delta Sync update is larger than the local Fragments Queue Size.<br>**Possible mitigation:**<br>■ If the counter's value is steady, increase the size of the Sending Queue.<br>■ If the counter's value keeps increasing, contact *Check Point Support*. |

### The "Sync at risk:" section

This section shows statistics that the Sending Queue is at full capacity and rejects Delta Sync retransmission requests.

Table: Description of the output fields

| Field | Description |
|---|---|
| **Sent reject notifications** | Shows how many times this Cluster Member rejected Delta Sync retransmission requests from its peer Cluster Members, because this Cluster Member does not hold the requested Delta Sync update anymore. |
| **Received reject notification** | Shows how many reject notifications this Cluster Member received from its peer Cluster Members. |

### The "Sent updates:" section

This section shows statistics for Delta Sync updates sent by this Cluster Member to its peer Cluster Members.

Table: Description of the output fields

| Field | Description |
|---|---|
| **Total generated sync messages** | Shows how many Delta Sync updates were generated.<br>This counts the Delta Sync updates, Retransmission Requests, Retransmission Acknowledgments, and so on. |
| **Sent retransmission requests** | Shows how many times this Cluster Member asked its peer Cluster Members to retransmit specific Delta Sync update(s).<br>Retransmission requests are sent when certain Delta Sync updates (with a specified sequence number) are missing, while the sending Cluster Member already received Delta Sync updates with advanced sequences.<br>**Note** - Compare the number of **Sent retransmission requests** to the **Total generated sync messages** of the other Cluster Members.<br>A large counter's value can imply connectivity problems. If the counter's value is unreasonably high (more than 30% of the **Total generated sync messages** of other Cluster Members), contact *Check Point Support* equipped with the entire output and a detailed description of the network topology and configuration. |
| **Sent retransmission updates** | Shows how many times this Cluster Member retransmitted specific Delta Sync update(s) at the requests from its peer Cluster Members. |

Table: Description of the output fields (continued)

| Field | Description |
|---|---|
| **Peak fragments per update** | Shows the peak amount of fragments in the Fragments Queue on this Cluster Member (usually, should be 1). |

### The "Received updates:" section

This section shows statistics for Delta Sync updates that were received by this Cluster Member from its peer Cluster Members.

Table: Description of the output fields

| Field | Description |
|---|---|
| **Total received updates** | Shows the total number of Delta Sync updates this Cluster Member received from its peer Cluster Members.<br>This counts only Delta Sync updates (not Retransmission Requests, Retransmission Acknowledgments, and others). |
| **Received retransmission requests** | Shows how many retransmission requests this Cluster Member received from its peer Cluster Members.<br>A large counter's value can imply connectivity problems. If the counter's value is unreasonably high (more than 30% of the **Total generated sync messages** on this Cluster Member), contact *Check Point Support* equipped with the entire output and a detailed description of the network topology and configuration. |

### The "Queue sizes (num of updates):" section

This section shows the sizes of the Delta Sync queues.

Table: Description of the output fields

| Field | Description |
|---|---|
| **Sending queue size** | Shows the size of the cyclic queue, which buffers all the Delta Sync updates that were already sent until it receives an acknowledgment from the peer Cluster Members.<br>This queue is needed for retransmitting the requested Delta Sync updates.<br>Each Cluster Member has one Sending Queue.<br>Default: 512 Delta Sync updates, which is also the minimal value. |

Table: Description of the output fields (continued)

| Field | Description |
|---|---|
| **Receiving queue size** | Shows the size of the cyclic queue, which buffers the received Delta Sync updates in two cases:<br>■ When Delta Sync updates are missing, this queue is used to hold the remaining received Delta Sync updates until the lost Delta Sync updates are retransmitted (Cluster Members must keep the order, in which they save the Delta Sync updates in the kernel tables).<br>■ This queue is used to re-assemble a fragmented Delta Sync update.<br>Each Cluster Member has one Receiving Queue.<br>Default: 256 Delta Sync updates, which is also the minimal value. |
| **Fragments queue size** | Shows the size of the queue, which is used to prepare a Delta Sync update before moving it to the Sending Queue.<br>Notes:<br>■ This queue must be smaller than the Sending Queue.<br>■ This queue must be significantly smaller than the Receiving Queue.<br>Default: 50 Delta Sync updates, which is also the minimal value. |

### The "Timers:" section

This section shows the Delta Sync timers.

| Field | Description |
|---|---|
| **Delta Sync interval (ms)** | Shows the interval at which this Cluster Member sends the Delta Sync updates from its Sending Queue.<br>The base time unit is 100ms (or 1 *tick*).<br>Default: 100 ms, which is also the minimum value.<br>See *Increasing the Sync Timer*. |

### The "Reset on XXX (triggered XXX)" section

Shows the date and the time of last statistics reset.

In parentheses, it shows how the last statistics was triggered - "`manually`", or "`by fullsync`".

# Viewing IGMP Status

## Description

This command shows the IGMP membership status.

## Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster members igmp` |
| Expert mode | `cphaprob igmp` |

## Example

```
[Expert@Member1:0]# cphaprob igmp

IGMP Membership: Enabled
Supported Version: 2
Report Interval [sec]: 60

IGMP queries are replied only by Operating System

Interface        Host Group      Multicast Address  Last ver.  Last Query[sec]
-------------------------------------------------------------------------------
eth0             224.168.3.247   01:00:5e:28:03:f7     N/A        N/A
eth1             224.22.33.250   01:00:5e:16:21:fa     N/A        N/A
eth2             224.55.66.247   01:00:5e:37:42:f7     N/A        N/A

[Expert@Member1:0]#
```

# Viewing Cluster Delta Sync Statistics for Connections Table

### Description

This command shows Delta Sync statistics about the operations performed in the Connections Kernel Table (id 8158).

The output shows operations such as creating a new connection (`SET`), updating a connection (`REFRESH`), deleting a connection (`DELETE`), and so on.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster statistics transport [reset]` |
| Expert mode | `cphaprob [-reset] ldstat` |

The "`reset`" flag resets the kernel statistics, which were collected since the last reboot or reset.

### Example

```
[Expert@Member1:0]# cphaprob ldstat

Operand              Calls     Bytes     Average   Ratio %
-------------------------------------------------------------
ERROR                0         0         0         0
SET                  354       51404     145       33
RENAME               0         0         0         0
REFRESH              1359      70668     52        46
DELETE               290       10440     36        6
SLINK                193       12352     64        8
UNLINK               0         0         0         0
MODIFYFIELDS         91        7280      80        4
RECORD DATA CONN     0         0         0         0
COMPLETE DATA CONN   0         0         0         0

Total bytes sent: 161292 (0 MB) in 1797 packets. Average 89

[Expert@Member1:0]#
```

# Viewing Cluster IP Addresses

### Description

This command shows the IP addresses and interfaces of the Cluster Members.

### Syntax to see all interfaces

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster members ips` |
| Expert mode | `cphaprob tablestat` |

### Syntax to see only the monitored interfaces

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster members monitored`<br>**Note** - This command is available in the R81.10 Jumbo Hotfix Accumulator, Take 75 and higher. |
| Expert mode | `cphaprob -m tablestat` |

## Example

**Note** - To see name of interfaces that correspond to numbers in the "**Interface**" column, run the `fw ctl iflist` command.

```
[Expert@Member1:0]# cphaprob tablestat

----   Unique IP's Table  ----

Member          Interface       IP-Address
-----------------------------------------

(Local)
0               1               192.168.3.245
0               2               11.22.33.245
0               3               44.55.66.245

1               1               192.168.3.246
1               2               11.22.33.246
1               3               44.55.66.246

-----------------------------------------


[Expert@Member1:0]#
[Expert@Member1:0]# fw ctl iflist
1 : eth0
2 : eth1
3 : eth2
[Expert@Member1:0]#
```

# Viewing the Cluster Member ID Mode in Local Logs

### Description

This command shows how the local ClusterXL logs show the Cluster Member - by its Member ID (default), or its Member Name.

See *"Configuring the Cluster Member ID Mode in Local Logs" on page 213*.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster members idmode` |
| Expert mode | `cphaprob names` |

### Example

```
[Expert@Member1:0]# cphaprob names

Current member print mode in local logs is set to: ID

[Expert@Member1:0]#
```

# Viewing Interfaces Monitored by RouteD

### Description

This command shows the interfaces, which the RouteD daemon monitors on the Cluster Member when you configure OSPF.

The idea is that if you configure OSPF, Cluster Member monitors these interfaces and does not bring up the Cluster Member unless RouteD daemon says it is OK to bring up the Cluster Member. This is used mainly in ClusterXL High Availability Primary Up configuration to avoid premature failbacks.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `show ospf interfaces [detailed]` |
| Expert mode | `cphaprob routedifcs` |

### Example 1

```
[Expert@Member1:0]# cphaprob routedifcs

No interfaces are registered.

[Expert@Member1:0]#
```

### Example 2

```
[Expert@Member1:0]# cphaprob routedifcs

Monitored interfaces registered by routed:

eth0
[Expert@Member1:0]#
```

# Viewing Roles of RouteD Daemon on Cluster Members

### Description

This command shows on which Cluster Member the RouteD daemon runs as a Master.

**Notes:**

- In ClusterXL High Availability, the RouteD daemon must run as a *Master* only on the Active Cluster Member.
- In ClusterXL Load Sharing, the RouteD daemon must run as a *Master* only on one of the Active Cluster Members and as a Non-Master on all other Cluster Members.
- In VRRP Cluster, the RouteD daemon must run as a *Master* only on the VRRP Master Cluster Member.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster role` |
| Expert mode | `cphaprob roles` |

### Example

```
[Expert@Member1:0]# cphaprob roles

ID         Role

1 (local)  Master
2          Non-Master

[Expert@Member1:0]#
```

# Viewing Cluster Correction Statistics

### Background

The Cluster Correction Layer (CCL) is a mechanism that deals with asymmetric connections - when a client-to-server connection passes through one Cluster Member, and the server-to-client connection arrives at another Cluster Member.

The CCL provides stickiness for connections by "correcting" the packets to the correct Cluster Member (that processed the client-to-server connection):

- In most cases, the CCL makes the correction from the CoreXL SND instances.
- In some cases (like Dynamic Routing, or VPN), the CCL makes the correction from the CoreXLFirewall instances or from SecureXL.

**Note** - For more information about CoreXL, see the *R81.10 Performance Tuning Administration Guide*.

### Description

This command shows the statistics counters for the Cluster Correction Layer (CCL) on each Cluster Member.

The counters are reset in these cases:

- Dring each boot.
- When Check Point services are stopped and started.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | N / A |
| Expert mode | `cphaprob [{-d \| -f \| -s}] corr` |

Where:

| Command | Description |
|---|---|
| `cphaprob corr` | Shows Cluster Correction Layer (CCL) statistics for all traffic. |
| `cphaprob -d corr` | Shows Cluster Correction Layer (CCL) statistics for CoreXL SND instances only. |
| `cphaprob -f corr` | Shows Cluster Correction Layer (CCL) statistics for CoreXL Firewall instances only. |
| `cphaprob -s corr` | Shows Cluster Correction Layer (CCL) statistics for SecureXL only. |

**Notes:**

- In the output of "`cphaprob -d corr`", this row does not appear:
  `Local asymmetric conns:`
- In the output, these rows appear only on Scalable Platforms:
  - `ICMP ERROR forwarded packets:`
  - `ICMP ERROR forwarded bytes:`
  - `VS Stateless forwarded packets:`
  - `VS Stateless forwarded bytes:`
- In some cases, ClusterXL needs to send some data along with the corrected packet (for example, in VPN).
  For such packets, the output counters show "`with metadata`".

### Calculating how much traffic is corrected out of total traffic on each Cluster Member

Large percentage values indicate a large number of assymetic connections that arrive at a cluster.

1. Get the output for the Cluster Correction Statistics (CCL) counters.
2. Copy the values of these CCL counters:
   - `Sent packets`
   - `Received packets`
3. Run:

   ```
   cpview
   ```

4. At the top, click **Network** > **Interfaces** >**Traffic**.
   a. In the section "`RX Traffic`", refer to the column "`packets`" > refer to the row "`TOTAL`".
   b. In the section "`TX Traffic`", refer to the column "`packets`" > refer to the row "`TOTAL`".
5. Calculate the percentage of the corrected received packets:

   ```
   (CCL "Received packets" x 100%) / (CPView "RX Traffic"
   packets)
   ```

6. Calculate the percentage of the corrected transmitted packets:

   ```
   (CCL "Sent packets" x 100%) / (CPView "TX Traffic" packets)
   ```

### Example 1 - CCL statistics for all traffic

```
[Expert@Member1:0]# cphaprob corr

Cluster Correction Stats (All Traffic):
-------------------------------------------------------
Sent packets:                   0 (0 with metadata)
Sent bytes:                     0
Received packets:               0 (0 with metadata)
Received bytes:                 0
Send errors:                    0
Receive errors:                 0
Local asymmetric conns:         0
ICMP ERROR forwarded packets:   0
ICMP ERROR forwarded bytes:     0
VS Stateless forwarded packets: 0
VS Stateless forwarded bytes:   0
[Expert@Member1:0]#
```

### Example 2 - CCL statistics for CoreXL SND instances only

```
[Expert@Member1:0]# cphaprob -d corr

Cluster Correction Stats (Dispatcher Corrections only):
-------------------------------------------------------
Sent packets:                   0 (0 with metadata)
Sent bytes:                     0
Received packets:               0 (0 with metadata)
Received bytes:                 0
Send errors:                    0
Receive errors:                 0
[Expert@Member1:0]#
```

### Example 3 - CCL statistics for CoreXL Firewall instances only

```
[Expert@Member1:0]# cphaprob -f corr

Cluster Correction Stats (Firewall instances only):
---------------------------------------------------
Sent packets:                   0 (0 with metadata)
Sent bytes:                     0
Received packets:               0 (0 with metadata)
Received bytes:                 0
Send errors:                    0
Receive errors:                 0
Local asymmetric conns:         0
[Expert@Member1:0]#
```

### Example 4 - CCL statistics for SecureXL only

```
[Expert@Member1:0]# cphaprob -s corr

Getting stats for SXL device 0, may take a few seconds...

Cluster Correction Stats (SXL Devices only):
---------------------------------------------------
Sent packets:                   0 (0 with metadata)
Sent bytes:                     0
Received packets:               0 (0 with metadata)
Received bytes:                 0
Send errors:                    0
Receive errors:                 0
Local asymmetric conns:         0
[Expert@Member1:0]#
```

# Viewing the Cluster Control Protocol (CCP) Settings

### Description

- You can view the Cluster Control Protocol (CCP) mode on the Cluster Members.
- You can view the Cluster Control Protocol (CCP) Encryption on the Cluster Members - enabled or disabled (and the encryption key).

See *"Configuring the Cluster Control Protocol (CCP) Settings" on page 224*

### Syntax for viewing the Cluster Control Protocol (CCP) mode

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster members interfaces virtual` |
| Expert mode | `cphaprob -a if` |

**Important** - In R81.10, the CCP always runs in the unicast mode.

### Syntax for viewing the Cluster Control Protocol (CCP) Encryption

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster members ccpenc` |
| Expert mode | `cphaprob ccp_encrypt`<br>`cphaprob ccp_encrypt_key` |

# Viewing the State of the Multi-Version Cluster Mechanism

### Description

This command shows the state of the Multi-Version Cluster (MVC) Mechanism - enabled (`ON`) or disabled (`OFF`).

See *"Configuring the Multi-Version Cluster Mechanism" on page 232*.

### Syntax

| Shell | Command |
|---|---|
| Gaia Clish | `show cluster members mvc` |
| Expert mode | `cphaprob mvc` |

### Example

```
Member1> show cluster members mvc

ON

Member1>
```

# Viewing Full Connectivity Upgrade Statistics

## Description

This command shows the Full Connectivity Upgrade statistics when you upgrade between minor versions.

## Syntax

| Shell | Command |
|---|---|
| Gaia Clish | N / A |
| Expert mode | `cphaprob fcustat` |

## Example

```
[Expert@Member1:0]# cphaprob fcustat

During FCU.....................   no
Connection module map..........   none

Table id map (remote->local).....   none

Table handlers ..................
        8151 --> 0x0x7f97c421d860 (sip_state)
        8158 --> 0x0x7f97c43d8e30 (connections)
LD handlers.....................
        ok - 0
        failed - 0

Global handlers .................   none

[Expert@Member1:0]#
```

# Monitoring Cluster Status in SmartConsole

## Background

To see the applicable logs in SmartConsole:

- From the left navigation panel, click **Logs & Monitor** > **Logs**.

To get logs about changes in the state of Cluster Members:

1. From the left navigation panel, click **Gateways & Servers**.
2. Open the cluster object.
3. From the left tree, click **ClusterXL and VRRP**.
4. Open the cluster object.
5. In the **Tracking** field, select **Log**.
6. Click **OK**.
7. Install the Access Control Policy on the cluster object.

## ClusterXL Log Messages

This section uses these conventions:

1. Square brackets are used to indicate place holders, which are substituted by relevant data when an actual log message is issued (for example, "`[NUMBER]`" is replaced by a numeric value).
2. Angle brackets are used to indicate alternatives, one of which is used in actual log messages.

   The different alternatives are separated with a vertical line (for example, "`<up|down>`" indicates that either "`up`" or "`down`" is used).
3. These placeholders are frequently used:
   - **ID**: A unique Cluster Member identifier, starting from "1". This corresponds to the order, in which Cluster Members are sorted in the cluster object on the **Cluster Members** page.
   - **IP**: Any unique IP address that belongs to the Cluster Member.
   - **MODE**: The cluster mode (for example, **New HA**, **LS Multicast**, and so on).
   - **STATE**: The state of the member (for example, **active**, **down**, **standby**).

- **DEVICE**: The name of a Critical Device (for example, **Interface Active Check**, **fwd**).

## General Logs

| Log | Description |
|---|---|
| `Starting <ClusterXL\|State Synchronization>.` | Indicates that ClusterXL (or State Synchronization, for 3rd party clusters) was successfully started on the reporting Cluster Member.<br>This message is usually issued after a member boots, or after an explicit call to `cphastart`. |
| `Stopping <ClusterXL\|State Synchronization>.` | Informs that ClusterXL (or State Synchronization) was deactivated on this Cluster Member.<br>The Cluster Member is no longer a part of the cluster (even if configured to be so), until ClusterXL is restarted. |
| `Unconfigured cluster Computers changed their MAC Addresses. Please reboot the cluster so that the changes take affect.` | This message is usually issued when a Cluster Member is shut down, or after an explicit call to the `cphastop`. |

## State Logs

| Log | Description |
|---|---|
| `Mode inconsistency detected: member [ID] ([IP]) will change its mode to [MODE]. Please re-install the security policy on the cluster.` | This message should rarely happen.<br>It indicates that another Cluster Member has reported a different cluster mode than is known to the local Cluster Member.<br>This is usually the result of a failure to install the Access Control Policy on all Cluster Members.<br>To correct this problem, install the Access Control Policy again.<br>**Note** - The cluster continues to operate after a mode inconsistency is detected by altering the mode of the reporting Cluster Member to match the other Cluster Members. However, we highly recommend to re-install the policy as soon as possible. |

| Log | Description |
|---|---|
| `State change of member [ID] ([IP]) from [STATE] to [STATE] was cancelled, since all other members are down. Member remains [STATE].` | When a Cluster Member needs to change its state (for example, when an Active Cluster Member encounters a problem and needs to change its state to "Down"), it first queries the other Cluster Members for their state.<br>If all other Cluster Members are down, this Cluster Member cannot change its state to a non-active one (otherwise the cluster cannot function).<br>Thus, the reporting Cluster Member continues to function, despite its problem (and usually reports its state as "`Active(!)`"). |
| `member [ID] ([IP]) <is active\|is down\|is stand-by\|is initializing> ([REASON]).` | This message is issued whenever a Cluster Member changes its state.<br>The log text specifies the new state of the Cluster Member. |

**Critical Device Logs**

| Log | Description |
|---|---|
| `[DEVICE] on member [ID] ([IP]) status OK ([REASON])` | The Critical Device is working normally. |
| `[DEVICE] on member [ID] ([IP]) detected a problem ([REASON]).` | Either an error was detected by the Critical Device, or the Critical Device has not reported its state for a number of seconds (as set by the "`timeout`" option of the Critical Device) |
| `[DEVICE] on member [ID] ([IP]) is initializing ([REASON]).` | Indicates that the Critical Device has registered itself with the Critical Device mechanism, but has not yet determined its state. |
| `[DEVICE] on member [ID] ([IP]) is in an unknown state ([STATE ID]) ([REASON]).` | This message should not normally appear.<br>Contact *Check Point Support*. |

# Interface Logs

| Log | Description |
|---|---|
| `interface [INTERFACE NAME] of member [ID] ([IP]) is up.` | Indicates that this interface is working normally - it is able to receive and transmit Cluster Control Protocol (CCP) packets on the expected subnet. |
| `interface [INTERFACE NAME] of member [ID] ([IP]) is down (receive <up\|down>, transmit <up\|down>).` | This message is issued whenever an interface encounters a problem, either in receiving or transmitting Cluster Control Protocol (CCP) packets.<br>Note that in this case the interface may still be working properly, as far as the OS is concerned, but is unable to communicate with other Cluster Members. |
| `interface [INTERFACE NAME] of member [ID] ([IP]) was added.` | Indicates that a new interface was registered with the Cluster Member (meaning that Cluster Control Protocol (CCP) packets arriving on this interface).<br>Usually, this message is the result of activating an interface (such as issuing the "`ifconfig up`" command).<br>The interface is now included in the ClusterXL reports (in the output of the applicable CLI commands.<br>Note that the interface may still be reported as "`Disconnected`", in case it was configured as such for ClusterXL. |
| `interface [INTERFACE NAME] of member [ID] ([IP}) was removed.` | Indicates that an interface was detached from the Cluster Member, and is therefore no longer monitored by ClusterXL. |

## Reason Strings

| Log | Description |
|---|---|
| `member [ID] ([IP]) reports more interfaces up.` | This text can be included in a Critical Device log message describing the reasons for a problem report: another Cluster Member has more interfaces reported to be working, than the local Cluster Member does. Usually, this means that the local Cluster Member has a faulty interface, and that its peer Cluster Member can do a better job as a Cluster Member. The local Cluster Member changes it state to "`Down`", leaving the peer Cluster Member specified in the message to handle traffic. |
| `member [ID] ([IP]) has more interfaces - check your disconnected interfaces configuration in the <discntd.if file\|registry>` | This message is issued when Cluster Members in the same cluster have a different number of interfaces.<br>A Cluster Member with fewer interfaces than the maximal number in the cluster (the reporting Cluster Member) may not be working properly, as it is missing an interface required to operate against a cluster IP address, or a synchronization network.<br>If some of the interfaces on the other Cluster Member are redundant, and should not be monitored by ClusterXL, they should be explicitly designated as "Non-Monitored". See *"Defining Non-Monitored Interfaces" on page 162*. |
| `[NUMBER] interfaces required, only [NUMBER] up.` | ClusterXL has detected a problem with one or more of the monitored interfaces.<br>This does not necessarily mean that the Cluster Member changes its state to "`Down`", as the other Cluster Members may have less operational interfaces. In such a condition, the Cluster Member with the largest number of operational interfaces will remain up, while the others go down. |

# Working with SNMP Traps

You can configure and see SNMP traps for ClusterXL High Availability.

**To configure an SNMP trap:**

1. Connect to the command line on the Management Server.
2. Log in to the Expert mode.
3. On a Multi-Domain Server, go to the context of the applicable Domain Management Server:

   ```
   mdsenv <IP Address or Name of Domain Management Server>
   ```

4. Run:

   ```
   threshold_config
   ```

   For more information, see the *R81.10 CLI Reference Guide* > Chapter *Security Management Server Commands* > Section *threshold_config*.

5. From the **Threshold Engine Configuration Options** menu, select **(9) Configure Thresholds**.
6. From the **Threshold Categories** menu, select **(2) High Availability**.
7. Select the applicable traps.
8. Select and configure these actions for the specified trap:
   - **Enable/Disable Threshold**
   - **Set Severity**
   - **Set Repetitions**
   - **Configure Alert Destinations**
9. From the **Threshold Engine Configuration Options** menu, select **(7) Configure alert destinations**.
10. Configure your alert destinations.
11. From the **Threshold Engine Configuration Options** menu, select **(3) Save policy**.

    You can optionally save the policy to a file.
12. In SmartConsole, install the Access Control Policy on this cluster object.

**Note** - You can download the most recent Check Point MIB files from sk90470.

# How to Initiate Cluster Failover

For more information on initiating manual cluster failovers, see sk55081.

**Method 1 (recommended)**

| Change in the State of the Cluster Member | Command in Gaia Clish | Command in the Expert Mode | Notes |
|---|---|---|---|
| Change the state of the Cluster Member to DOWN | `set cluster member admin down` | `clusterXL_admin down` | Does not disable Delta Sync. |
| Change the state of the Cluster Member to UP | `set cluster member admin up` | `clusterXL_admin up` | Does not initiate Full Sync. |

See:

- *"Initiating Manual Cluster Failover" on page 225*
- *"The clusterXL_admin Script" on page 325*

**Method 2 (not recommended)**

| Change in the State of the Cluster Member | Command in the Expert mode | Notes |
|---|---|---|
| Change the state of the Cluster Member to DOWN | 1. `cphaconf set_pnote -d <Name of Critical Device> -t 0 -s ok register`<br>2. `cphaconf set_pnote -d <Name of Critical Device> -s problem report` | Does not disable Delta Sync. |
| Change the state of the Cluster Member to UP | 1. `cphaconf set_pnote -d <Name of Critical Device> -s ok report`<br>2. `cphaconf set_pnote -d <Name of Critical Device> unregister` | Does not initiate Full Sync. |

See:

- *"Registering a Critical Device" on page 214*
- *"Reporting the State of a Critical Device" on page 219*
- *"Unregistering a Critical Device" on page 218*

**Notes:**

- In Load Sharing mode, the cluster distributes the traffic load between the remaining Active members.
- In High Availability mode, the cluster fails over to a Standby Cluster Member with the highest priority.

# Troubleshooting Issues with the Critical Device "routed"

## Background

The Critical Device called **routed** monitors the state of the Dynamic Routing on the Cluster Member (see *"Viewing Critical Devices" on page 246*).

This Critical Device makes sure that traffic is not assigned to a Cluster Member before it is ready to handle the traffic.

The Gaia OS RouteD daemon handles all routing (static and dynamic) operations.

There can be an issue with Dynamic Routing with one or more of these symptoms:

- Cluster IP address connectivity problems
- Unexpected cluster failovers
- The state of the Critical Device **routed** is **problem**.

  Example:

```
Device Name: routed
Registration number: 2
Timeout: none
Current state: problem
Time since last report: 10 sec
```

These are some of the common causes of this issue:

- Cluster misconfiguration
- Traffic on TCP port 2010 between Cluster Members is blocked
- The RouteD daemon did not get all of its routes
- The RouteD daemon did not start correctly

## Standard Behavior of the Critical Device "routed"

Typically, the Critical Device **routed**reports its current state as **problem** when:

- A Cluster Member fails over
- A Cluster Member reboots
- There is an inconsistency in the Dynamic Routing configuration on Cluster Members

the Critical Device **routed**reports its current state as **OK** when

- A ClusterXL member tells the RouteD daemon that it is a Master
- The RouteD daemon gets the entire routing state from the Master

# Basic Troubleshooting Steps

1. Examine the cluster interfaces to make sure they are configured correctly.

   See *"Viewing Cluster Interfaces" on page 258*.

2. In ClusterXL High Availability mode, make sure that the RouteD daemon is running on the Active Cluster Member.
3. Make sure traffic on TCP port 2010 between the Cluster Members is not blocked.
4. Generate RouteD cluster messages.

   Run this command in the Expert mode on the cluster member:

   ```
   dbset
   routed:instance:default:traceoptions:traceoptions:Cluster
   ```

   Examine the `/var/log/routed/log` file.

5. In ClusterXL High Availability mode with the OSPF configuration, make sure the OSPF interface is up on the Standby Cluster Member.
6. In the OSPF configuration, look for a "`router-id`" mismatch.

For advanced troubleshooting procedures and more information, see sk92787.

For troubleshooting OSPF and the RouteD daemon, see sk84520.

# ClusterXL Error Messages

Each important cluster event that affects the Cluster Members has a unique code that appears in the `/var/log/messages` file and `dmesg`.

Each cluster event message starts with the prefix **CLUS-XXXXXX-Y**.

Example: `CLUS-214802-1`

| Part of the message | Description |
|---|---|
| **CLUS-** | Constant string. |
| **XXXXXX** | A six-digit error code.<br>These error codes specify the events:<br>■ Events related to Critical Devices<br>■ Events related to the states of Cluster Members<br>■ Events related to cluster synchronization<br>■ Events related to policy installation<br>The *first digit* shows which Cluster Member generated the event:<br>■ 1 - local member<br>■ 2 - remote member<br>The *second digit* represents the cluster type event<br>The meaning of the *other digits* depends on the cluster event type. |
| **Y** | Shows the ID or the NAME of the local Cluster Member that generated this log message.<br>See *"Configuring the Cluster Member ID Mode in Local Logs" on page 213*. |

For more information, see sk125152.

# Command Line Reference

See the *R81.10 CLI Reference Guide*.

# ClusterXL Configuration Commands

### Description

These commands let you configure internal behavior of the Clustering Mechanism.

**Important:**

- We do not recommend that you run these commands. These commands must be run automatically only by the Security Gateway or the Check Point Support.
- In a Cluster, you must configure all the Cluster Members in the same way.

### Syntax

**Notes:**

- In Gaia Clish:
  Enter the **`set cluster<ESC><ESC>`** to see all the available commands.
- In Expert mode:
  Run the **`cphaconf`** command see all the available commands.
  You can run the `cphaconf` commands only from the Expert mode.
- Syntax legend:
  1. Curly brackets or braces **{ }**:
     Enclose a list of available commands or parameters, separated by the vertical bar **|**, from which user can enter only one.
  2. Angle brackets **< >**:
     Enclose a variable - a supported value user needs to specify explicitly.
  3. Square brackets or brackets **[ ]**:
     Enclose an optional command or parameter, which user can also enter.
- You can include these commands in scripts to run them automatically.
  The meaning of each command is explained in the next sections.

Table: ClusterXL Configuration Commands

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Configure how to show the Cluster Member in local ClusterXL logs - by its Member ID or its Member Name (see *"Configuring the Cluster Member ID Mode in Local Logs" on page 213*) | `set cluster member idmode {id \| name}` | `cphaconf mem_id_mode {id \| name}` |

Table: ClusterXL Configuration Commands (continued)

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Register a single Critical Device (Pnote) on the Cluster Member (see *"Registering a Critical Device" on page 214*) | `N / A` | `cphaconf set_pnote -d <Name of Device> -t <Timeout in Sec> -s {ok\|init\|problem} [-p] [-g] register` |
| Unregister a single Critical Device (Pnote) on the Cluster Member (see *"Unregistering a Critical Device" on page 218*) | `N / A` | `cphaconf set_pnote -d <Name of Device> [-p] [-g] unregister` |
| Report (change) a state in a single Critical Device (Pnote) on the Cluster Member (see *"Reporting the State of a Critical Device" on page 219*) | `N / A` | `cphaconf set_pnote -d <Name of Device> -s {ok\|init\|problem} [-g] report` |
| Register several Critical Devices (Pnotes) from a file on the Cluster Member (see *"Registering Critical Devices Listed in a File" on page 221*) | `N / A` | `cphaconf set_pnote -f <Name of File> [-g] register` |
| Unregister all Critical Devices (Pnotes) on the Cluster Member (see *"Unregistering All Critical Devices" on page 223*) | `N / A` | `cphaconf set_pnote -a [-g] unregister` |
| Configure the Cluster Control Protocol (CCP) Encryption on the Cluster Member (see *"Configuring the Cluster Control Protocol (CCP) Settings" on page 224*) | `set cluster member ccpenc {off \| on}` | `cphaconf ccp_ encrypt {off \| on}`<br>`cphaconf ccp_ encrypt_key <Key String>` |
| Configure the Cluster Forwarding Layer on the Cluster Member (controls the forwarding of traffic between Cluster Members)<br>**Note** - For Check Point use only. | `set cluster member forwarding {off \| on}` | `cphaconf forward {off \| on}` |
| Print the current cluster configuration as loaded in the kernel on the Cluster Member (for details, see sk93306) | `N / A` | `cphaconf debug_data` |

Table: ClusterXL Configuration Commands (continued)

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Start internal failover between subordinate interfaces of specified bond interface - only in Bond High Availability mode (for details, see sk93306) | `N / A` | `cphaconf failover_bond <bond_name>` |
| Configure what happens during a failover after a Bond already failed over internally (for details, see sk93306) | `N / A` | `cphaconf enable_bond_failover <bond_name>` |
| Initiate manual cluster failover (see *"Initiating Manual Cluster Failover" on page 225*) | `set cluster member admin {down \| up}` | `clusterXL_admin {down \| up}` |
| Configure the minimal number of required subordinate interfaces for Bond Load Sharing (see *"Configuring the Minimal Number of Required Subordinate Interfaces for Bond Load Sharing" on page 229*) | `N / A` | `cphaconf bond_ls {set <Bond Name> <Value> \| remove <Bond Name>}` |
| Configuring Link Monitoring on the Cluster Interfaces (see *"Configuring Link Monitoring on the Cluster Interfaces" on page 119*) | `N / A` | `N / A` |
| Configuring the Multi-Version Cluster Mechanism (see *"Configuring the Multi-Version Cluster Mechanism" on page 232*) | `N / A` | `cphaconf mvc {off \| on}` |
| Configuring duration of forced cluster failover in the High Availability mode (see *"Configuring Duration of Forced Failover in High Availability Mode" on page 233*) | `set cluster force_failover member_id <ID> {time <Minutes> on \| off}` | `cphaconf force_failover member_id <ID> {time <Minutes> on \| off}` |

### List of the Gaia Clish "`set cluster`" commands

```
set cluster member admin {down | up} [permanent]
set cluster member ccpenc {off | on}
set cluster member forwarding {off | on}
set cluster member idmode {id | name}
set cluster member mvc {off | on}
set cluster force_failover member_id <ID> {time <Minutes> on |
off}
```

### List of the Expert mode "`cphaconf`" commands

**Note** - Some commands are not applicable to 3rd-party clusters.

```
cphaconf [-D] <options> start
cphaconf stop
cphaconf [-t <Sync IF 1>...] [-d <Non-Monitored IF 1>...] add
cphaconf clear-secured
cphaconf clear-non-monitored
cphaconf debug_data
cphaconf delete_link_local [-vs <VSID>] <IF name>
cphaconf set_link_local [-vs <VSID>] <IF name> <Cluster IP>
cphaconf mem_id_mode {id | name}
cphaconf failover_bond <bond_name>
cphaconf [-s] {set | unset | get} var <Kernel Parameter Name>
[<Value>]
cphaconf bond_ls {set <Bond Name> <Value> | remove <Bond Name>}
cphaconf set_pnote -d <Device> -t <Timeout in sec> -s {ok | init |
problem} [-p] [-g] register
cphaconf set_pnote -f <File> [-g] register
cphaconf set_pnote -d <Device> [-p] [-g] unregister
cphaconf set_pnote -a [-g] unregister
cphaconf set_pnote -d <Device> -s {ok | init | problem} [-g]
report
cphaconf ccp_encrypt {off | on}
cphaconf ccp_encrypt_key <Key String>
cphaconf mvc {on | off}
cphaconf force_failover member_id <ID> {time <Minutes> on | off}
```

**Note** - When a ClusterXL object has only one interface with the role "Cluster" or "Cluster+Sync", and this one interface is a Bond interface, then the Expert mode command "`cphaconf failover_bond <Name of Bond Interface>`" triggers the internal Bond failover only on the ClusterXL member on which you execute this command. The peer ClusterXL members do not automatically perform the interface failover in this Bond. See sk183496.

# ClusterXL Monitoring Commands

### Description

Use the monitoring commands to make sure that the cluster and the Cluster Members work properly, and to define Critical Devices. A Critical Device (also known as a *Problem Notification*, or *pnote*) is a special software device on each Cluster Member, through which the critical aspects for cluster operation are monitored. When the critical monitored component on a Cluster Member fails to report its state on time, or when its state is reported as problematic, the state of that member is immediately changed to 'Down'.

### Syntax

**Notes:**

- In Gaia Clish:
  Enter the **`show cluster<ESC><ESC>`** to see all the available commands.
- In Expert mode:
  Run the **`cphaprob`** command see all the available commands.
  You can run the `cphaprob` commands from Gaia Clish as well.
- Syntax legend:
  1. Curly brackets or braces **{ }**:
     Enclose a list of available commands or parameters, separated by the vertical bar **|**, from which user can enter only one.
  2. Angle brackets **< >**:
     Enclose a variable - a supported value user needs to specify explicitly.
  3. Square brackets or brackets **[ ]**:
     Enclose an optional command or parameter, which user can also enter.
- You can include these commands in scripts to run them automatically.
  The meaning of each command is explained in the next sections.

Table: ClusterXL Monitoring Commands

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Show states of Cluster Members and their names (see *"Viewing Cluster State" on page 240*) | `show cluster state` | `cphaprob [-vs <VSID>] state` |
| Show Critical Devices (Pnotes) and their states on the Cluster Member (see *"Viewing Critical Devices" on page 246*) | `show cluster members pnotes {all \| problem}` | `cphaprob [-l] [-ia] [-e] list` |

Table: ClusterXL Monitoring Commands (continued)

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Show cluster interfaces on the cluster member (see *"Viewing Cluster Interfaces" on page 258*) | `show cluster members interfaces {all \| secured \| virtual \| vlans}` | `cphaprob [-vs all] [-a] [-m] if` |
| Show cluster bond configuration on the Cluster Member (see *"Viewing Bond Interfaces" on page 263*) | `show cluster bond {all \| name <bond_name>}` | `cphaprob show_bond [<bond_name>]` |
| Show groups of bonds on the Cluster Member (see *"Viewing Bond Interfaces" on page 263*) | `N / A` | `cphaprob show_bond_groups` |
| Show (and reset) cluster failover statistics on the Cluster Member (see *"Viewing Cluster Failover Statistics" on page 268*) | `show cluster failover [reset {count \| history}]` | `cphaprob [-reset {-c \| -h}] [-l <count>] show_failover` |
| Show information about the software version (including hotfixes) on the local Cluster Member and its matches/mismatches with other Cluster Members (see *"Viewing Software Versions on Cluster Members" on page 270*) | `show cluster release` | `cphaprob release` |
| Show Delta Sync statistics on the Cluster Member (see *"Viewing Delta Synchronization" on page 271*) | `show cluster statistics sync [reset]` | `cphaprob [-reset] syncstat` |
| Show Delta Sync statistics for the Connections table on the Cluster Member (see *"Viewing Cluster Delta Sync Statistics for Connections Table" on page 279*) | `show cluster statistics transport [reset]` | `cphaprob [-reset] ldstat` |
| Show the Cluster Control Protocol (CCP) mode on the Cluster Member (see *"Viewing Cluster Interfaces" on page 258*) | `show cluster members interfaces virtual` | `cphaprob [-vs all] -a if` |

Table: ClusterXL Monitoring Commands (continued)

| Description of Command | Command in Gaia Clish | Command in Expert Mode |
|---|---|---|
| Show the IGMP membership of the Cluster Member (see *"Viewing IGMP Status" on page 278*) | `show cluster members igmp` | `cphaprob igmp` |
| Show cluster unique IP's table on the Cluster Member (see *"Viewing Cluster IP Addresses" on page 280*) | `show cluster members ips`<br>`show cluster members monitored` | `cphaprob tablestat`<br>`cphaprob -m tablestat` |
| Show the Cluster Member ID Mode in local logs - by Member ID (default) or Member Name (see *"Viewing the Cluster Member ID Mode in Local Logs" on page 282*) | `show cluster members idmode` | `cphaprob names` |
| Show interfaces, which the RouteD monitors on the Cluster Member when you configure OSPF (see *"Viewing Interfaces Monitored by RouteD" on page 283*) | `show ospf interfaces [detailed]` | `cphaprob routedifcs` |
| Show roles of RouteD daemon on Cluster Members (see *"Viewing Roles of RouteD Daemon on Cluster Members" on page 284*) | `show cluster roles` | `cphaprob roles` |
| Show Cluster Correction Statistics (see *"Viewing Cluster Correction Statistics" on page 285*) | `N / A` | `cphaprob [{-d \| -f \| -s}] corr` |
| Show the Cluster Control Protocol (CCP) mode (see *"Viewing the Cluster Control Protocol (CCP) Settings" on page 289*) | `show cluster members interfaces virtual` | `cphaprob -a if` |
| Show the Cluster Control Protocol (CCP) Encryption settings (see *"Viewing the Cluster Control Protocol (CCP) Settings" on page 289*) | `show cluster members ccpenc` | `cphaprob ccp_encrypt` |
| Shows the state of the Multi-Version Cluster (see *"Viewing the State of the Multi-Version Cluster Mechanism" on page 290*) | `show cluster members mvc` | `N / A` |
| Show Full Connectivity Upgrade statistics (see *"Viewing Full Connectivity Upgrade Statistics" on page 291*) | `N / A` | `cphaprob fcustat` |

### List of the Gaia Clish "`show cluster`" commands

```
show cluster
      bond
            all
            name <Name of Bond>
      failover
      members
            ccpenc
            idmode
            igmp
            interfaces
                  all
                  secured
                  virtual
                  vlans
            ips
            monitored
            mvc
            pnotes
                  all
                  problem
      release
      roles
      state
      statistics
            sync [reset]
            transport [reset]
```

### List of the Expert mode "cphaprob" commands

**Note** - Some commands are not applicable to 3rd party clusters.

```
cphaprob [-vs <VSID>] state
cphaprob [-reset {-c | -h}] [-l <count>] show_failover
cphaprob names
cphaprob [-reset] [-a] syncstat
cphaprob [-reset] ldstat
cphaprob [-l] [-i[a]] [-e] list
cphaprob [-vs all] [-a] [-m] if
cphaprob show_bond [<bond_name>]
cphaprob show_bond_groups
cphaprob igmp
cphaprob fcustat
cphaprob [-m] tablestat
cphaprob routedifcs
cphaprob roles
cphaprob release
cphaprob ccp_encrypt
cphaprob [{-d | -f | -s}] corr
```

# cpconfig

### Description

This command starts the Check Point Configuration Tool.

This tool configures specific settings for the installed Check Point products.

**Important:**

- In a Cluster, you must configure all the Cluster Members in the same way.
- On Scalable Platforms (Maestro and Chassis), you must connect to the applicable Security Group.

### Syntax on a Security Gateway / Cluster Member in Gaia Clish or the Expert mode

```
cpconfig
```

### Syntax on a Scalable Platform Security Group in Gaia gClish or the Expert mode

```
cpconfig
```

### Menu Options

**Note** - The options shown depend on the configuration and installed products.

| Menu Option | Description |
|---|---|
| **Licenses and contracts** | Manages Check Point licenses and contracts on this Security Gateway or Cluster Member. |
| **SNMP Extension** | Obsolete. Do **not** use this option anymore.<br>To configure SNMP, see the *R81.10 Gaia Administration Guide* - Chapter *System Management* - Section *SNMP*. |
| **PKCS#11 Token** | Register a cryptographic token, for use by Gaia Operating System.<br>See details of the token, and test its functionality. |
| **Random Pool** | Configures the RSA keys, to be used by Gaia Operating System. |

| Menu Option | Description |
|---|---|
| **Secure Internal Communication** | Manages SIC on the Security Gateway or Cluster Member. This change requires a restart of Check Point services on the Security Gateway or Cluster Member.<br>For more information, see:<br>■ The *R81.10 Security Management Administration Guide*.<br>■ sk65764: How to reset SIC. |
| **Enable cluster membership for this gateway** | Enables the cluster membership on the Security Gateway. This change requires a reboot of the Security Gateway.<br>For more information, see the *R81.10 Installation and Upgrade Guide*.<br>**Note** - This section does **not** apply to Scalable Platforms (Maestro and Chassis). |
| **Disable cluster membership for this gateway** | Disables the cluster membership on the Security Gateway. This change requires a reboot of the Security Gateway.<br>For more information, see the *R81.10 Installation and Upgrade Guide*.<br>**Note** - This section does **not** apply to Scalable Platforms (Maestro and Chassis). |
| **Enable Check Point Per Virtual System State** | Enables Virtual System Load Sharing on the VSX Cluster Member.<br>For more information, see the *R81.10 VSX Administration Guide*.<br>**Note** - This section does **not** apply to Scalable Platforms (Maestro and Chassis). |
| **Disable Check Point Per Virtual System State** | Disables Virtual System Load Sharing on the VSX Cluster Member.<br>For more information, see the *R81.10 VSX Administration Guide*.<br>**Note** - This section does **not** apply to Scalable Platforms (Maestro and Chassis). |
| **Enable Check Point ClusterXL for Bridge Active/Standby** | Enables Check Point ClusterXL for Bridge mode.<br>This change requires a reboot of the Cluster Member.<br>For more information, see the *R81.10 Installation and Upgrade Guide*.<br>**Note** - This section does **not** apply to Scalable Platforms (Maestro and Chassis). |

| Menu Option | Description |
|---|---|
| **Disable Check Point ClusterXL for Bridge Active/Standby** | Disables Check Point ClusterXL for Bridge mode.<br>This change requires a reboot of the Cluster Member.<br>For more information, see the *R81.10 Installation and Upgrade Guide*.<br>**Note** - This section does **not** apply to Scalable Platforms (Maestro and Chassis). |
| **Check Point CoreXL** | Manages CoreXL and Firewall mode on the Security Gateway / Cluster Member / Scalable Platform Security Group.<br>After all changes in CoreXL configuration, you must reboot the Security Gateway / Cluster Member / Security Group.<br>For more information, see the *R81.10 Performance Tuning Administration Guide*. |
| **Automatic start of Check Point Products** | Shows and controls which of the installed Check Point products start automatically during boot. |
| **Exit** | Exits from the Check Point Configuration Tool. |

### Example 1 - Menu on a single Security Gateway

```
[Expert@MySingleGW:0]# cpconfig
This program will let you re-configure
your Check Point products configuration.


Configuration Options:
----------------------
(1) Licenses and contracts
(2) SNMP Extension
(3) PKCS#11 Token
(4) Random Pool
(5) Secure Internal Communication
(6) Enable cluster membership for this gateway
(7) Check Point CoreXL
(8) Automatic start of Check Point Products

(9) Exit

Enter your choice (1-9) :
```

## Example 2 - Menu on a Cluster Member

```
[Expert@MyClusterMember:0]# cpconfig
This program will let you re-configure
your Check Point products configuration.


Configuration Options:
----------------------
(1) Licenses and contracts
(2) SNMP Extension
(3) PKCS#11 Token
(4) Random Pool
(5) Secure Internal Communication
(6) Disable cluster membership for this gateway
(7) Enable Check Point Per Virtual System State
(8) Enable Check Point ClusterXL for Bridge Active/Standby
(9) Check Point CoreXL
(10) Automatic start of Check Point Products

(11) Exit

Enter your choice (1-11) :
```

# cphastart

### Description

Starts the cluster configuration on a Cluster Member after it was stopped with the *"cphastop" on page 318* command.

**Best Practice** - To start a Cluster Member, use the "`cpstart`" command.

**Note** - This command does **not** initiate a Full Synchronization on the Cluster Member.

### Syntax

```
cphastart
        [-h]
        [-d]
```

### Parameters

| Parameter | Description |
|---|---|
| `-h` | Shows the applicable built-in usage. |
| `-d` | Runs the command in debug mode.<br>Use only if you troubleshoot the command itself.<br>**Best Practice** - If you use this parameter, then redirect the output to a file, or use the `script` command to save the entire CLI session.<br>Refer to:<br>■ These lines in the output file:<br>`prepare_command_args: -D ... start`<br>`/opt/CPsuite-R81.10/fw1/bin/cphaconf clear-secured`<br>`/opt/CPsuite-R81.10/fw1/bin/cphaconf -D ...`<br>*`(truncated here for brevity)`*`... start`<br>■ The `$FWDIR/log/cphastart.elg` log file. |

# cphastop

### Description

Stops the cluster software on a Cluster Member.

**Best Practice** - To stop a Cluster Member, use the "`cpstop`" command.

**Notes:**

- This command stops the Cluster Member from passing traffic.
- This command stops the State Synchronization between this Cluster Member and its peer Cluster Members.
- After you run this command, you can still open connections directly to this Cluster Member.
- To start the cluster software, run the *"cphastart" on page 317* command.

### Syntax

```
cphastop
```

# cp_conf fullha

**Important** - This command does **not** apply to Scalable Platforms (Maestro and Chassis).

### Description

Manages the state of the Full High Availability Cluster:

- Enables the Full High Availability Cluster
- Disables the Full High Availability Cluster
- Deletes the Full High Availability peer
- Shows the Full High Availability state

**Important** - To configure a Full High Availability cluster, follow the *R81.10 Installation and Upgrade Guide*.

### Syntax

```
cp_conf fullha
      enable
      del_peer
      disable
      state
```

### Parameters

| Parameter | Description |
|---|---|
| `enable` | Enables the Full High Availability on this computer. |
| `del_peer` | Deletes the Full High Availability peer from the configuration. |
| `disable` | Disables the Full High Availability on this computer. |
| `state` | Shows the Full High Availability state on this computer. |

### Example

```
[Expert@Cluster_Member:0]# cp_conf fullha state
FullHA is currently enabled
[Expert@Cluster_Member:0]#
```

# cp_conf ha

**Important** - This command does **not** apply to Scalable Platforms (Maestro and Chassis).

### Description

Enables or disables cluster membership on this Security Gateway.

**Important** - This command is for Check Point use only. To configure cluster membership, you must use the "`cpconfig`" command.

### Syntax

```
cp_conf ha {enable | disable} [norestart]
```

### Parameters

| Parameter | Description |
|---|---|
| `enable` | Enables cluster membership on this Security Gateway.<br>This command is equivalent to the option **Enable cluster membership for this gateway** in the "`cpconfig`" menu. |
| `disable` | Disables cluster membership on this Security Gateway.<br>This command is equivalent to the option **Disable cluster membership for this gateway** in the "`cpconfig`" menu. |
| `norestart` | **Optional:** Specifies to apply the configuration change without the restart of Check Point services. The new configuration takes effect only after reboot. |

### Example 1 - Enable the cluster membership without restart of Check Point services

```
[Expert@MyGW:0]# cp_conf ha enable norestart

Cluster membership for this gateway was enabled successfully
Important: This change will take effect after reboot.

[Expert@MyGW:0]#
```

### Example 2 - Disable the cluster membership without restart of Check Point services

```
[Expert@MyGW:0]# cp_conf ha disable norestart
cpwd_admin:
Process CPHAMCSET process has been already terminated

Cluster membership for this gateway was disabled successfully
Important: This change will take effect after reboot.

[Expert@MyGW:0]#
```

# fw hastat

### Description

Shows information about Check Point computers in High Availability configuration and their states.

**Note** - This command is outdated. On cluster members, run the Gaia Clish command "`show cluster state`", or the Expert mode command "`cphaprob state`". See *"Viewing Cluster State" on page 240*.

### Syntax

```
fw hastat [<Target1>] [<Target2>] ... [<TargetN>]
```

### Parameters

| Parameter | Description |
|---|---|
| <*Target1*><br><*Target2*> ...<br><*TargetN*> | Specifies the Check Point computers to query.<br>If you run this command on the Management Server, you can enter the applicable IP address, or the resolvable HostName of the managed Security Gateway or Cluster Member.<br>If you do not specify the target, the command queries the local computer. |

### Example - Querying the local Cluster Member

```
[Expert@Member1:0]# fw hastat
HOST NUMBER HIGH AVAILABILITY STATE MACHINE STATUS
192.168.3.52 1 active OK
[Expert@Member1:0]#
```

# fwboot ha_conf

### Description

Configures the cluster mechanism during boot.

**Important** - This command is for Check Point use only.

**Notes:**

- You must run this command from the Expert mode.
- To install a cluster, see the *R81.10 Installation and Upgrade Guide*.

### Syntax

```
[Expert@HostName:0]# $FWDIR/boot/fwboot ha_conf
```

# ClusterXL Scripts

You can use special scripts to change the state of Cluster Members.

# The clusterXL_admin Script

### Description

You can use the **clusterXL_admin** script to initiate a manual fail-over from a Cluster Member.

Location of this script on your Cluster Members is:

```
$FWDIR/bin/clusterXL_admin
```

### Script Workflow

This shell script does one of these:

- Registers a Critical Device called "`admin_down`" and reports the state of that Critical Device as "`problem`".

  This gracefully changes the state of the Cluster Member to "`DOWN`".

- Reports the state of the registered Critical Device "`admin_down`" as "`ok`".

  This gracefully changes the state of the Cluster Member to "`UP`".

  Then, the script unregisters the Critical Device "`admin_down`".

For more information, see sk55081.

### Example

```
#! /bin/csh -f
#
# The script will cause the machine to get into down state, thus the member will not filter
packets.
# It will supply a simple way to initiate a failover by registering a new device in problem
state when
# a failover is required and will unregister the device when wanting to return to normal
operation.
# USAGE:
# clusterXL_admin <up|down>

set PERSISTENT = ""

# checking number of arguments
if ( $#argv > 2 || $#argv < 1 ) then
        echo "clusterXL_admin : Invalid Argument Count"
        echo "Usage: clusterXL_admin <up|down> [-p]"
        exit 1
else if ( "$1" != "up" && "$1" != "down" ) then
        echo "clusterXL_admin : Invalid Argument ($1)"
        echo "Usage: clusterXL_admin <up|down> [-p]"
        exit 1
else if ( $#argv == 2 ) then
        if ( "$2" != "-p" ) then
                echo "clusterXL_admin : Invalid Argument ($2)"
                echo "Usage: clusterXL_admin <up|down> [-p]"
                exit 1
        endif
        set PERSISTENT = "-p"
endif

#checking if cpha is started
$FWDIR/bin/cphaprob stat | grep "Cluster" > /dev/null
if ($status) then
        echo "HA is not started"
        exit 1
endif

# Inform the user that the command can run with persistent mode.
if ("$PERSISTENT" != "-p") then
        echo "This command does not survive reboot. To make the change permanent, please run
'set cluster member admin down/up permanent' in clish or add '-p' at the end of the command in
expert mode"
endif

if ( $1 == "up" ) then
        echo "Setting member to normal operation ..."
        $FWDIR/bin/cphaconf set_pnote -d admin_down $PERSISTENT unregister > & /dev/null
        if ( `uname` == 'IPSO' ) then
                sleep 5
        else
                sleep 1
        endif

        set stateArr = `$FWDIR/bin/cphaprob stat | grep "local"`

        $FWDIR/bin/cphaprob stat | egrep "Sync only|Bridge Mode" > /dev/null
        #If it's third party or bridge mode, use column 4 , otherwise 5
        if ($status) then
                set state = $stateArr[5]
        else
                set state = $stateArr[4]
        endif

        echo "Member current state is $state"
        if (($state != "Active" && $state != "Standby") && ($state != "ACTIVE" && $state !=
"STANDBY" && $state != "ACTIVE(!)")) then
                echo "Operation failed: member is still down, please run 'show cluster members
pnotes problem' in clish or 'cphaprob list' in expert mode for further details"
        endif
        exit 0
endif
```

```
if ( $1 == "down" ) then
        echo "Setting member to administratively down state ..."
        $FWDIR/bin/cphaconf set_pnote -d admin_down -t 0 -s problem $PERSISTENT register > &
/dev/null
        sleep 1

        set stateArr = `$FWDIR/bin/cphaprob stat | grep "local"`

        $FWDIR/bin/cphaprob stat | egrep "Sync only|Bridge Mode" > /dev/null
        #If it's third party or bridge mode, use column 4 , otherwise 5
        if ($status) then
                set state = $stateArr[5]
        else
                set state = $stateArr[4]
        endif

        echo "Member current state is $state"
        if ( $state == "Active attention" || $state == "ACTIVE(!)" ) then
                echo "All the members within the cluster have problem/s and the local member was
chosen to become active"
        else
                if ( $state != "Down" && $state != "DOWN" ) then
                        echo "Operation failed: member is still down, please run 'show cluster
members pnotes problem' in clish or 'cphaprob list' in expert mode for further details"
                endif
        endif
        exit 0
else
        echo "clusterXL_admin : Invalid Option ($1)"
        echo "Usage: clusterXL_admin <up|down> [-p]"
        exit 1
endif
```

# The clusterXL_monitor_ips Script

### Description

You can use the **clusterXL_monitor_ips** script to ping a list of predefined IP addresses and change the state of the Cluster Member to `DOWN` or `UP` based on the replies to these pings. For this script to work, you must write the IP addresses in the `$FWDIR/conf/cpha_hosts` file - each IP address on a separate line. This file does not support comments or spaces.

Location of this script on your Cluster Members is:

```
$FWDIR/bin/clusterXL_monitor_ips
```

### Script Workflow

1. Registers a Critical Device called "`host_monitor`" with the status "`ok`".
2. Starts to send pings to the list of predefined IP addresses in the `$FWDIR/conf/cpha_hosts` file.
3. While the script receives responses to its pings, it does not change the status of that Critical Device.
4. If the script does not receive a response to even one ping, it reports the state of that Critical Device as "`problem`".

   This gracefully changes the state of the Cluster Member to `DOWN`.

   If the script receives responses to its pings again, it changes the status of that Critical Device to "`ok`" again.

For more information, see sk35780.

**Important** - You must do these changes on all Cluster Members.

### Example

```
#!/bin/sh
#
# The script tries to ping the hosts written in the file $FWDIR/conf/cpha_hosts. The names (must
be resolveable) ot the IPs of the hosrs must be written in seperate lines.
# the file must not contain anything else.
# We ping the given hosts every number of seconds given as parameter to the script.
# USAGE:
# cpha_monitor_ips X silent
# where X is the number of seconds between loops over the IPs.
# if silent is set to 1, no messages will appear on the console
#
# We initially register a pnote named "host_monitor" in the problem notification mechanism
# when we detect that a host is not responding we report the pnote to be in "problem" state.
# when ping succeeds again - we report the pnote is OK.

silent=0

if [ -n "$2" ]; then
        if [ $2 -le 1 ]; then
                silent=$2
        fi
fi
hostfile=$FWDIR/conf/cpha_hosts
arch=`uname -s`
if [ $arch = "Linux" ]
then
        #system is linux
        ping="ping -c 1 -w 1"
else
        ping="ping"
fi
$FWDIR/bin/cphaconf set_pnote -d host_monitor -t 0 -s ok register
TRUE=1
while [ "$TRUE" ]
do
        result=1
        for hosts in `cat $hostfile`
        do
              if [ $silent = 0 ]
                then
                        echo "pinging $hosts using command $ping $hosts"
              fi
                if [ $arch = "Linux" ]
                then
                        $ping $hosts > /dev/null 2>&1
                else
                        $ping $hosts $1 > /dev/null 2>&1
                fi
                status=$?
                if [ $status = 0 ]
                then
                     if [ $silent = 0 ]
                        then
                            echo " $hosts is alive"
                     fi
                else
                     if [ $silent = 0 ]
                        then
                            echo " $hosts is not responding "
                     fi
                        result=0
                fi
        done
        if [ $silent = 0 ]
           then
                echo "done pinging"
        fi
        if [ $result = 0 ]
        then
              if [ $silent = 0 ]
              then
                  echo " Cluster member should be down!"
```

```
                fi
                  $FWDIR/bin/cphaconf set_pnote -d host_monitor -s problem report
        else
                if [ $silent = 0 ]
                then
                          echo " Cluster member seems fine!"
                fi
                  $FWDIR/bin/cphaconf set_pnote -d host_monitor -s ok report
        fi
        if [ "$silent" = 0 ]
        then
                  echo "sleeping"
        fi
        sleep $1
        echo "sleep $1"
done
```

# The clusterXL_monitor_process Script

### Description

You can use the **clusterXL_monitor_process** script to monitor if the specified user space processes run, and cause cluster fail-over if these processes do not run. For this script to work, you must write the correct case-sensitive names of the monitored processes in the `$FWDIR/conf/cpha_proc_list` file - each process name on a separate line. This file does not support comments or spaces.

Location of this script on your Cluster Members is:

```
$FWDIR/bin/clusterXL_monitor_process
```

### Script Workflow

1. Registers Critical Devices (with the status "`ok`") called as the names of the processes you specified in the `$FWDIR/conf/cpha_proc_list` file.
2. While the script detects that the specified process runs, it does not change the status of the corresponding Critical Device.
3. If the script detects that the specified process do not run anymore, it reports the state of the corresponding Critical Device as "`problem`".

   This gracefully changes the state of the Cluster Member to "`DOWN`".

   If the script detects that the specified process runs again, it changes the status of the corresponding Critical Device to "`ok`" again.

For more information, see sk92904.

**Important** - You must do these changes on all Cluster Members.

### Example

```
#!/bin/sh
#
# This script monitors the existance of processes in the system. The process names should be
written
# in the $FWDIR/conf/cpha_proc_list file one every line.
#
# USAGE :
# cpha_monitor_process X silent
# where X is the number of seconds between process probings.
# if silent is set to 1, no messages will appear on the console.
#
#
# We initially register a pnote for each of the monitored processes
# (process name must be up to 15 charachters) in the problem notification mechanism.
# when we detect that a process is missing we report the pnote to be in "problem" state.
# when the process is up again - we report the pnote is OK.

if [ "$2" -le 1 ]
then
        silent=$2
else
        silent=0
fi
if [ -f $FWDIR/conf/cpha_proc_list ]
then
        procfile=$FWDIR/conf/cpha_proc_list
else
        echo "No process file in $FWDIR/conf/cpha_proc_list "
        exit 0
fi

arch=`uname -s`

for process in `cat $procfile`
do
        $FWDIR/bin/cphaconf set_pnote -d $process -t 0 -s ok -p register > /dev/null 2>&1
done

while [ 1 ]
do

        result=1

        for process in `cat $procfile`
        do
                ps -ef  | grep $process | grep -v grep > /dev/null 2>&1

                status=$?

                if [ $status = 0 ]
                then
                         if [ $silent = 0 ]
                            then
                                  echo " $process is alive"
                        fi
#                       echo "3, $FWDIR/bin/cphaconf set_pnote -d $process -s ok report"
                        $FWDIR/bin/cphaconf set_pnote -d $process -s ok report
                else
                      if [ $silent = 0 ]
                         then
                                echo " $process is down"
                         fi

                     $FWDIR/bin/cphaconf set_pnote -d $process -s problem report
                      result=0
                fi

        done

        if [ $result = 0 ]

        then
```

```
                if [ $silent = 0 ]
                    then
                  echo " One of the monitored processes is down!"
                fi
         else
                if [ $silent = 0 ]
                    then
                        echo " All monitored processes are up "
              fi

        fi
        if [ "$silent" = 0 ]
        then
                echo "sleeping"
        fi

       sleep $1

done
```

# Cluster Management APIs

## Introduction

The purpose of Cluster APIs is to provide automation / orchestration of Check Point cluster in a way similar to simple-gateway APIs.

These Cluster APIs support common cluster operations - such as creating a new cluster object, modifying an existing cluster object (for example, adding or removing cluster members, manipulation of interfaces).

These Cluster APIs are called "simple" because they do not support all cluster object features.

For operations on cluster objects that are not provided by these APIs, use SmartConsole.

## List of APIs

| API Category | API | Description |
|---|---|---|
| **Asynchronous** | `add simple-cluster` | Creates a new simple cluster object from scratch |
| | `set simple-cluster` | Modifies an existing simple cluster object |
| **Synchronous** | `show simple-cluster` | Shows an existing simple cluster object specified by its Name or UID |
| | `show simple-clusters` | Shows all existing simple cluster objects |
| | `delete simple-cluster` | Deletes an existing simple cluster object |

# API Examples

### Example 1 - Adding a simple cluster object

#### API command:

Use this API to add a simple cluster object.

```
add simple-cluster
```

Once the API command finishes, and the session is published, a new cluster object appears in SmartConsole.

#### Prerequisite:

1. All Cluster Members must already be installed.
2. The applicable interfaces on each Cluster Member must already be configured.

#### Example description:

- A simple ClusterXL in High Availability mode called **cluster1**
- With two cluster members called **member1** and **member2**
- With three interfaces: **eth0** (external), **eth1** (sync), and **eth2** (internal)
- Only the **Firewall** Software Blade is enabled (the **IPsec VPN** blade is disabled)
- Cluster software version is R81.10

#### Example cluster object topology:

| Interface | Cluster | Member1 | Member2 |
|---|---|---|---|
| eth0 (External) | 172.23.5.254 / 255.255.255.0 | 172.23.5.1 / 255.255.255.0 | 172.23.5.2 / 255.255.255.0 |
| eth1 (Sync) | N / A | 1.1.1.1 / 255.255.255.0 | 1.1.1.2 / 255.255.255.0 |
| eth2 (Internal) | 192.168.1.254 / 255.255.255.0 | 192.168.1.1 / 255.255.255.0 | 192.168.1.2 / 255.255.255.0 |

**API example:**

**Important** - In the API command you must use the same One-Time Password you used on Cluster Members during the First Time Configuration Wizard.

```
{
  "name" : "cluster1",
  "color" : "yellow",
  "version" : "R81.10",
  "ip-address" : "172.23.5.254",
  "os-name" : "Gaia",
  "cluster-mode" : "cluster-xl-ha",
  "firewall" : true,
  "vpn" : false,
  "interfaces" : [
    {
      "name" : "eth0",
      "ip-address" : "172.23.5.254",
      "network-mask" : "255.255.255.0",
      "interface-type" : "cluster",
      "topology" : "EXTERNAL",
      "anti-spoofing" : "true"
    },
   {
      "name" : "eth1",
      "interface-type" : "sync",
      "topology" : "INTERNAL",
      "topology-settings": {
        "ip-address-behind-this-interface": "network defined by the interface ip and net
mask",
         "interface-leads-to-dmz": false
      }
    },
    {
      "name" : "eth2",
      "ip-address" : "192.168.1.254",
      "network-mask" : "255.255.255.0",
      "interface-type" : "cluster",
      "topology" : "INTERNAL",
      "anti-spoofing" : "true",
      "topology-settings": {
        "ip-address-behind-this-interface": "network defined by the interface ip and net
mask",
        "interface-leads-to-dmz": false
      }
    }
  ],
  "members" : [ {
    "name" : "member1",
    "one-time-password" : "abcd",
    "ip-address" : "172.23.5.1",
    "interfaces" : [
      {
      "name" : "eth0",
      "ip-address" : "172.23.5.1",
      "network-mask" : "255.255.255.0"
      },
      {
      "name" : "eth1",
      "ip-address" : "1.1.1.1",
      "network-mask" : "255.255.255.0"
      },
      {
      "name" : "eth2",
      "ip-address" : "192.168.1.1",
      "network-mask" : "255.255.255.0"
      }
    ]
    },
```

```
    {
    "name" : "member2",
    "one-time-password" : "abcd",
    "ip-address" : "172.23.5.2",
    "interfaces" : [
      {
      "name" : "eth0",
      "ip-address" : "172.23.5.2",
      "network-mask" : "255.255.255.0"
      },
      {
      "name" : "eth1",
      "ip-address" : "1.1.1.2",
      "network-mask" : "255.255.255.0"
      },
      {
      "name" : "eth2",
      "ip-address" : "192.168.1.2",
      "network-mask" : "255.255.255.0"
      }
    ]
   }
  ]
}
```

**Example 2 - Modifying an existing cluster object - adding a cluster member**

**API command:**

Use this API to add (scale up) Cluster Members.

```
set simple-cluster
```

**Example description:**

Adding a Cluster Member called **member3**.

**Example cluster object topology:**

| Interface | Cluster | Member1 | Member2 | Member3 |
|---|---|---|---|---|
| eth0 (External) | 172.23.5.254 / 255.255.255.0 | 172.23.5.1 / 255.255.255.0 | 172.23.5.2 / 255.255.255.0 | *172.23.5.3 / 255.255.255.0* |
| eth1 (Sync) | N / A | 1.1.1.1 / 255.255.255.0 | 1.1.1.2 / 255.255.255.0 | *1.1.1.3 / 255.255.255.0* |
| eth2 (Internal) | 192.168.1.254 / 255.255.255.0 | 192.168.1.1 / 255.255.255.0 | 192.168.1.2 / 255.255.255.0 | *192.168.1.3 / 255.255.255.0* |

**API example:**

```
{
  "name" : "cluster1",
  "members" : { "add" :
    {
    "name" : "member3",
    "ipv4-address" : "172.23.5.3",
        "one-time-password" : "aaaa",
        "interfaces" : [
         {
           "name" : "eth0",
           "ip-address" : "172.23.5.3",
           "network-mask" : "255.255.255.0"
         },
        {
           "name" : "eth1",
           "ip-address" : "1.1.1.3",
           "network-mask" : "255.255.255.0"
         },
         {
           "name" : "eth2",
           "ip-address" : "192.168.1.3",
           "network-mask" : "255.255.255.0"
         }
        ]
    }
  }
}
```

**Example 3 - Modifying an existing cluster object - removing a cluster member**

### API command:

Use this API to remove (scale down) Cluster Members.

```
set simple-cluster
```

### Example description:

Removing a Cluster Member called **member3**.

### Example cluster object topology:

| Interface | Cluster | Member1 | Member2 | Member3 |
|---|---|---|---|---|
| eth0 (External) | 172.23.5.254 / 255.255.255.0 | 172.23.5.1 / 255.255.255.0 | 172.23.5.2 / 255.255.255.0 | *172.23.5.3 / 255.255.255.0* |
| eth1 (Sync) | N / A | 1.1.1.1 / 255.255.255.0 | 1.1.1.2 / 255.255.255.0 | *1.1.1.3 / 255.255.255.0* |
| eth2 (Internal) | 192.168.1.254 / 255.255.255.0 | 192.168.1.1 / 255.255.255.0 | 192.168.1.2 / 255.255.255.0 | *192.168.1.3 / 255.255.255.0* |

#### API example:

```
{
  "name" : "cluster1",
  "members" :  { "remove" : "member3" }
}
```

### Example 4 - Modifying an existing cluster object - adding a cluster interface

#### API command:

Use this API to add a cluster interface.

```
set simple-cluster
```

#### Example description:

Adding a cluster interface called **eth3**.

#### Example cluster object topology:

| Interface | Cluster | Member1 | Member2 |
|---|---|---|---|
| eth0 (External) | 172.23.5.254 / 255.255.255.0 | 172.23.5.1 / 255.255.255.0 | 172.23.5.2 / 255.255.255.0 |
| eth1 (Sync) | N / A | 1.1.1.1 / 255.255.255.0 | 1.1.1.2 / 255.255.255.0 |
| eth2 (Internal) | 192.168.1.254 / 255.255.255.0 | 192.168.1.1 / 255.255.255.0 | 192.168.1.2 / 255.255.255.0 |
| *eth3 (Internal)* | *10.10.10.254 / 255.255.255.0* | *10.10.10.1 / 255.255.255.0* | *10.10.10.2 / 255.255.255.0* |

#### API example:

```
{
  "name" : "cluster1",
  "interfaces" : { "add" :
    {
        "name" : "eth3",
        "ip-address" : "10.10.10.254",
        "ipv4-mask-length" : "24",
        "interface-type" : "cluster",
        "topology" : "INTERNAL",
        "anti-spoofing" : "true"
    }
  },
  "members" : { "update" :
    [{
    "name" : "member1" ,
    "interfaces" :
      {"name" : "eth3",
      "ipv4-address" : "10.10.10.1",
      "ipv4-network-mask" : "255.255.255.0"}
    },
    {
    "name" : "member2" ,
    "interfaces" :
      {"name" : "eth3",
      "ipv4-address" : "10.10.10.2",
      "ipv4-network-mask" : "255.255.255.0"}
    }
    ]}
}
```

**Example 5 - Modifying an existing cluster object - removing a cluster interface**

### API command:

Use this API to remove a cluster interface.

```
set simple-cluster
```

### Example description:

Removing a cluster interface called **eth3**.

**Example cluster object topology:**

| Interface | Cluster | Member1 | Member2 |
|---|---|---|---|
| eth0 (External) | 172.23.5.254 / 255.255.255.0 | 172.23.5.1 / 255.255.255.0 | 172.23.5.2 / 255.255.255.0 |
| eth1 (Sync) | N / A | 1.1.1.1 / 255.255.255.0 | 1.1.1.2 / 255.255.255.0 |
| eth2 (Internal) | 192.168.1.254 / 255.255.255.0 | 192.168.1.1 / 255.255.255.0 | 192.168.1.2 / 255.255.255.0 |
| *eth3 (Internal)* | *10.10.10.254 / 255.255.255.0* | *10.10.10.1 / 255.255.255.0* | *10.10.10.2 / 255.255.255.0* |

**API example:**

```
{
  "name" : "cluster1",
  "interfaces" :  { "remove" : "eth3" }
}
```

**Example 6 - Modifying an existing cluster object - changing settings of a cluster interface**

**API command:**

Use this API to change settings of a cluster interface.

```
set simple-cluster
```

**Example description:**

Changing the IP address of the cluster interfaces called **eth2** from 192.168.x.254 / 255.255.255.0 to 172.30.1.x / 255.255.255.0

**Example cluster object topology:**

| Interface | Cluster | Member1 | Member2 |
|---|---|---|---|
| eth0 (External) | 172.23.5.254 / 255.255.255.0 | 172.23.5.1 / 255.255.255.0 | 172.23.5.2 / 255.255.255.0 |
| eth1 (Sync) | N / A | 1.1.1.1 / 255.255.255.0 | 1.1.1.2 / 255.255.255.0 |
| eth2 (Internal) | From: 192.168.1.254 / 255.255.255.0 To: *172.30.1.254 / 255.255.255.0* | From: 192.168.1.1 / 255.255.255.0 To: *172.30.1.1 / 255.255.255.0* | From: 192.168.1.2 / 255.255.255.0 To: *172.30.1.2 / 255.255.255.0* |

**API example:**

```
{
 "name" : "cluster1",
 "interfaces" : { "update" :
    {
      "name" : "eth2",
      "ip-address" : "172.30.1.254",
      "ipv4-mask-length" : "24",
      "interface-type" : "cluster",
      "topology" : "INTERNAL",
      "anti-spoofing" : "true"
    }
  },
  "members" : { "update" : [
  {
      "name" : "member1" ,
      "interfaces" :
        {"name" : "eth2",
        "ipv4-address" : "172.30.1.1",
        "ipv4-mask-length" : "24"}
    },
  {
      "name" : "member2" ,
      "interfaces" :
        {"name" : "eth2",
        "ipv4-address" : "172.30.1.2",
        "ipv4-mask-length" : "24"}
    }
  ]}
}
```

**Example 7 - Modifying an existing cluster object - reestablishing SIC**

### API command:

Use this API to reestablish SIC with Cluster Members.

```
set simple-cluster
```

### Prerequisite:

SIC must already be reset on the Cluster Members.

### API example:

**Important** - In the API command you must use the same One-Time Password you used on Cluster Members during the SIC reset.

```
{
  "name" : "cluster1",
  "members" : { "update" :
  [
       {
       "name" : "member1",
       "one-time-password" : "aaaa"
        },
       {
       "name" : "member2",
       "one-time-password" : "aaaa"
        }
     ]
  }
}
```

## Example 8 - Modifying an existing cluster object - enabling / disabling blades

### API command:

Use this API to enable and disable Software Blades on Cluster Members.

```
set simple-cluster
```

**Notes:**

- To enable a Software Blade, set its value to `true` in the API command.
- To disable a Software Blade, set its value to `false` in the API command.

### API example:

To *enable* all Software Blades supported by the Cluster API:

```
{
   "name" : "cluster1",
   "vpn" : true,
   "application-control" : true,
   "url-filtering" : true,
   "ips" : true,
   "content-awareness" : true,
   "anti-bot" : true,
   "anti-virus" : true,
   "threat-emulation" : true
}
```

To *disable* all Software Blades supported by the Cluster API:

```
{
   "name" : "cluster1",
   "vpn" : false,
   "application-control" : false,
   "url-filtering" : false,
   "ips" : false,
   "content-awareness" : false,
   "anti-bot" : false,
   "anti-virus" : false,
   "threat-emulation" : false
}
```

## Example 9 - Viewing an existing cluster object

### API command:

Use this API to view a specific existing cluster object.

```
show simple-cluster
```

**Note** - By default, the output shows up to 50 configured cluster interfaces.

### API example - request:

```
{
  "limit-interfaces" : "10",
  "name" : "cluster1"
}
```

### API example - response:

```
{
    "uid": "e0ce560b-8a0a-4468-baa9-5f8eb2658b96",
    "name": "cluster1",
    "type": "simple-cluster",
    "domain": {
        "uid": "41e821a0-3720-11e3-aa6e-0800200c9fde",
        "name": "SMC User",
        "domain-type": "domain"
    },
    "meta-info": {
        "lock": "unlocked",
        "validation-state": "ok",
        "last-modify-time": {
            "posix": 1567417185885,
            "iso-8601": "2019-09-02T12:39+0300"
        },
        "last-modifier": "aa",
        "creation-time": {
            "posix": 1567417140278,
            "iso-8601": "2019-09-02T12:39+0300"
        },
        "creator": "aa"
    },
    "tags": [],
    "read-only": false,
    "comments": "",
    "color": "yellow",
    "icon": "NetworkObjects/cluster",
    "groups": [],
    "ipv4-address": "172.23.5.254",
    "dynamic-ip": false,
    "version": "R81.10",
    "os-name": "Gaia",
    "hardware": "Open server",
    "firewall": true,
    "firewall-settings": {
        "auto-maximum-limit-for-concurrent-connections": true,
        "maximum-limit-for-concurrent-connections": 25000,
        "auto-calculate-connections-hash-table-size-and-memory-pool": true,
        "connections-hash-size": 131072,
        "memory-pool-size": 6,
        "maximum-memory-pool-size": 30
    },
    "vpn": false,
    "application-control": false,
    "url-filtering": false,
    "content-awareness": false,
    "ips": false,
    "anti-bot": false,
    "anti-virus": false,
    "threat-emulation": false,
    "save-logs-locally": false,
    "send-alerts-to-server": [
```

```
        "harry-main-take-96"
    ],
    "send-logs-to-server": [
        "harry-main-take-96"
    ],
    "send-logs-to-backup-server": [],
    "logs-settings": {
        "rotate-log-by-file-size": false,
        "rotate-log-file-size-threshold": 1000,
        "rotate-log-on-schedule": false,
        "alert-when-free-disk-space-below-metrics": "mbytes",
        "alert-when-free-disk-space-below": true,
        "alert-when-free-disk-space-below-threshold": 3000,
        "alert-when-free-disk-space-below-type": "popup alert",
        "delete-when-free-disk-space-below-metrics": "mbytes",
        "delete-when-free-disk-space-below": true,
        "delete-when-free-disk-space-below-threshold": 5000,
        "before-delete-keep-logs-from-the-last-days": false,
        "before-delete-keep-logs-from-the-last-days-threshold": 0,
        "before-delete-run-script": false,
        "before-delete-run-script-command": "",
        "stop-logging-when-free-disk-space-below-metrics": "mbytes",
        "stop-logging-when-free-disk-space-below": true,
        "stop-logging-when-free-disk-space-below-threshold": 100,
        "reject-connections-when-free-disk-space-below-threshold": false,
        "reserve-for-packet-capture-metrics": "mbytes",
        "reserve-for-packet-capture-threshold": 500,
        "delete-index-files-when-index-size-above-metrics": "mbytes",
        "delete-index-files-when-index-size-above": false,
        "delete-index-files-when-index-size-above-threshold": 100000,
        "delete-index-files-older-than-days": false,
        "delete-index-files-older-than-days-threshold": 14,
        "forward-logs-to-log-server": false,
        "perform-log-rotate-before-log-forwarding": false,
        "update-account-log-every": 3600,
        "detect-new-citrix-ica-application-names": false,
        "turn-on-qos-logging": true
    },
    "interfaces": {
        "total": 3,
        "from": 1,
        "to": 3,
        "objects": [
            {
                "name": "eth0",
                "ipv4-address": "172.23.5.254",
                "ipv4-network-mask": "255.255.255.0",
                "ipv4-mask-length": 24,
                "ipv6-address": "",
                "topology": "external",
                "anti-spoofing": true,
                "anti-spoofing-settings": {
                    "action": "prevent"
                },
                "security-zone": false,
                "comments": "",
                "color": "black",
```

```
             "comments": "",
                "color": "black",
                "icon": "NetworkObjects/network",
                "interface-type": "sync"
            },
            {
                "name": "eth2",
                "ipv4-address": "192.168.1.254",
                "ipv4-network-mask": "255.255.255.0",
                "ipv4-mask-length": 24,
                "ipv6-address": "",
                "topology": "internal",
                "topology-settings": {
                    "ip-address-behind-this-interface": "network defined by the interface ip
and net mask",
                    "interface-leads-to-dmz": false
                },
                "anti-spoofing": true,
                "anti-spoofing-settings": {
                    "action": "prevent"
                },
                "security-zone": false,
                "comments": "",
                "color": "black",
                "icon": "NetworkObjects/network",
                "interface-type": "cluster"
            }
        ]
    },
    "cluster-mode": "cluster-xl-ha",
    "cluster-members": [
        {
            "name": "member1",
            "sic-state": "initialized",
            "sic-message": "Initialized but trust not established",
            "ip-address": "172.23.5.1",
            "interfaces": [
                {
                    "name": "eth0",
                    "ipv4-address": "172.23.5.1",
                    "ipv4-network-mask": "255.255.255.0",
                    "ipv4-mask-length": 24,
                    "ipv6-address": "",
                    "ipv6-network-mask": "::",
                    "ipv6-mask-length": 0
                },
                {
                    "name": "eth1",
                    "ipv4-address": "1.1.1.1",
                    "ipv4-network-mask": "255.255.255.0",
                    "ipv4-mask-length": 24,
                    "ipv6-address": "",
                    "ipv6-network-mask": "::",
                    "ipv6-mask-length": 0
                },
                {
                    "name": "eth2",
                    "ipv4-address": "192.168.1.1",
                    "ipv4-network-mask": "255.255.255.0",
                    "ipv4-mask-length": 24,
                    "ipv6-address": "",
                    "ipv6-network-mask": "::",
                    "ipv6-mask-length": 0
```

```
```

**Example 10 - Viewing all existing cluster objects**

**API command:**

Use this API to view all existing cluster objects.

```
show simple-clusters
```

**Example 11 - Deleting an existing cluster object**

**API command:**

Use this API to delete a specific cluster object.

```
delete simple-cluster
```

**API example:**

```
{
  "name" : "cluster1"
}
```

# Known Limitations

- These Cluster APIs support only subset of cluster operations.
- These Cluster APIs support only basic configuration of Software Blades (similar to "`simple-gateway`" APIs - see the *Check Point Management API Reference* (at the top, select the correct version) ).
- These Cluster APIs support only ClusterXL High Availability, ClusterXL Load Sharing, and CloudGuard OPSEC clusters.
- These Cluster APIs do **not** support the configuration of a Cluster Virtual IP address on a different subnet than the IP addresses of the Cluster Members.

  For such configuration, use SmartConsole.
- These Cluster APIs do **not** support VRRP Clusters (either on Gaia OS or IPSO OS).
- These Cluster APIs support a limited subset of interface settings.

  To change interface settings such as Topology, Anti-Spoofing and Security Zone, you must replace the interface.

# Glossary

## #

**3rd-party Cluster**
Cluster of Check Point Security Gateways that work together in a redundant configuration. These Check Point Security Gateways are installed on X-Series XOS, or IPSO OS. VRRP Cluster on Gaia OS is also considered a 3rd-party cluster. The 3rd-party cluster handles the traffic, and Check Point Security Gateways perform only State Synchronization.

## A

**Active**
State of a Cluster Member that is fully operational: (1) In ClusterXL, this applies to the state of the Security Gateway component (2) In 3rd-party / OPSEC cluster, this applies to the state of the cluster State Synchronization mechanism.

**Active-Active**
A cluster mode (in versions R80.40 and higher), where cluster members are located in different geographical areas (different sites, different cloud availability zones). This mode supports the configuration of IP addresses from different subnets on all cluster interfaces, including the Sync interfaces. Each cluster member inspects all traffic routed to it and synchronizes the recorded connections to its peer cluster members. The traffic is not balanced between the cluster members.

**Active Up**
ClusterXL in High Availability mode that was configured as Maintain current active Cluster Member in the cluster object in SmartConsole: (1) If the current Active member fails for some reason, or is rebooted (for example, Member_A), then failover occurs between Cluster Members - another Standby member will be promoted to be Active (for example, Member_B). (2) When former Active member (Member_A) recovers from a failure, or boots, the former Standby member (Member_B) will remain to be in Active state (and Member_A will assume the Standby state).

**Active(!)**
In ClusterXL, state of the Active Cluster Member that suffers from a failure. A problem was detected, but the Cluster Member still forwards packets, because it is the only member in the cluster, or because there are no other Active members in the cluster. In any other situation, the state of the member is Down. Possible states: ACTIVE(!), ACTIVE(!F) - Cluster Member is in the freeze state, ACTIVE(!P) - This is the Pivot Cluster Member in Load Sharing Unicast mode, ACTIVE(!FP) - This is the Pivot Cluster Member in Load Sharing Unicast mode and it is in the freeze state.

**Anti-Bot**
Check Point Software Blade on a Security Gateway that blocks botnet behavior and communication to Command and Control (C&C) centers. Acronyms: AB, ABOT.

**Anti-Spam**
Check Point Software Blade on a Security Gateway that provides comprehensive protection for email inspection. Synonym: Anti-Spam & Email Security. Acronyms: AS, ASPAM.

**Anti-Virus**
Check Point Software Blade on a Security Gateway that uses real-time virus signatures and anomaly-based protections from ThreatCloud to detect and block malware at the Security Gateway before users are affected. Acronym: AV.

**Application Control**
Check Point Software Blade on a Security Gateway that allows granular control over specific web-enabled applications by using deep packet inspection. Acronym: APPI.

**ARP Forwarding**
Forwarding of ARP Request and ARP Reply packets between Cluster Members by encapsulating them in Cluster Control Protocol (CCP) packets. Introduced in R80.10 version.

**Audit Log**
Log that contains administrator actions on a Management Server (login and logout, creation or modification of an object, installation of a policy, and so on).

## B

**Backup**
(1) In VRRP Cluster on Gaia OS - State of a Cluster Member that is ready to be promoted to Master state (if Master member fails). (2) In VSX Cluster configured in Virtual System Load Sharing mode with three or more Cluster Members - State of a Virtual System on a third (and so on) VSX Cluster Member. (3) A Cluster Member or Virtual System in this state does not process any traffic passing through cluster.

**Blocking Mode**
Cluster operation mode, in which Cluster Member does not forward any traffic (for example, caused by a failure).

**Bridge Mode**
Security Gateway or Virtual System that works as a Layer 2 bridge device for easy deployment in an existing topology.

## C

**Cluster**
Two or more Security Gateways that work together in a redundant configuration - High Availability, or Load Sharing.

**Cluster Control Protocol**
Proprietary Check Point protocol that runs between Cluster Members on UDP port 8116, and has the following roles: (1) State Synchronization (Delta Sync), (2) Health checks (state of Cluster Members and of cluster interfaces): Health-status Reports, Cluster-member Probing, State-change Commands, Querying for cluster membership. Note: CCP is located between the Check Point Firewall kernel and the network interface (therefore, only TCPdump should be used for capturing this traffic). Acronym: CCP.

**Cluster Correction Layer**
Proprietary Check Point mechanism that deals with asymmetric connections in Check Point cluster. The CCL provides connections stickiness by "correcting" the packets to the correct Cluster Member: In most cases, the CCL makes the correction from the CoreXL SND; in some cases (like Dynamic Routing, or VPN), the CCL makes the correction from the Firewall or SecureXL. Acronym: CCL.

**Cluster Interface**
An interface on a Cluster Member, whose Network Type was set as Cluster in SmartConsole in cluster object. This interface is monitored by cluster, and failure on this interface will cause cluster failover.

**Cluster Member**
Security Gateway that is part of a cluster.

**Cluster Mode**
Configuration of Cluster Members to work in these redundant modes: (1) One Cluster Member processes all the traffic - High Availability or VRRP mode (2) All traffic is processed in parallel by all Cluster Members - Load Sharing.

**Cluster Topology**
Set of interfaces on all members of a cluster and their settings (Network Objective, IP address / Net Mask, Topology, Anti-Spoofing, and so on).

**ClusterXL**
Cluster of Check Point Security Gateways that work together in a redundant configuration. The ClusterXL both handles the traffic and performs State Synchronization. These Check Point Security Gateways are installed on Gaia OS: (1) ClusterXL supports up to 5 Cluster Members, (2) VRRP Cluster supports up to 2 Cluster Members, (3) VSX VSLS cluster supports up to 13 Cluster Members. Note: In ClusterXL Load Sharing mode, configuring more than 4 Cluster Members significantly decreases the cluster performance due to amount of Delta Sync traffic.

**Compliance**
Check Point Software Blade on a Management Server to view and apply the Security Best Practices to the managed Security Gateways. This Software Blade includes a library of Check Point-defined Security Best Practices to use as a baseline for good Security Gateway and Policy configuration.

**Content Awareness**
Check Point Software Blade on a Security Gateway that provides data visibility and enforcement. Acronym: CTNT.

**CoreXL**
Performance-enhancing technology for Security Gateways on multi-core processing platforms. Multiple Check Point Firewall instances are running in parallel on multiple CPU cores.

**CoreXL Firewall Instance**
On a Security Gateway with CoreXL enabled, the Firewall kernel is copied multiple times. Each replicated copy, or firewall instance, runs on one processing CPU core. These firewall instances handle traffic at the same time, and each firewall instance is a complete and independent firewall inspection kernel. Synonym: CoreXL FW Instance.

**CoreXL SND**
Secure Network Distributer. Part of CoreXL that is responsible for: Processing incoming traffic from the network interfaces; Securely accelerating authorized packets (if SecureXL is enabled); Distributing non-accelerated packets between Firewall kernel instances (SND maintains global dispatching table, which maps connections that were assigned to CoreXL Firewall instances). Traffic distribution between CoreXL Firewall instances is statically based on Source IP addresses, Destination IP addresses, and the IP 'Protocol' type. The CoreXL SND does not really "touch" packets. The decision to stick to a particular FWK daemon is done at the first packet of connection on a very high level, before anything else. Depending on the SecureXL settings, and in most of the cases, the SecureXL can be offloading decryption calculations. However, in some other cases, such as with Route-Based VPN, it is done by FWK daemon.

**CPHA**
General term in Check Point Cluster that stands for Check Point High Availability (historic fact: the first release of ClusterXL supported only High Availability) that is used only for internal references (for example, inside kernel debug) to designate ClusterXL infrastructure.

**CPUSE**
Check Point Upgrade Service Engine for Gaia Operating System. With CPUSE, you can automatically update Check Point products for the Gaia OS, and the Gaia OS itself.

**Critical Device**
A special software device on each Cluster Member, through which the critical aspects for cluster operation are monitored. When the critical monitored component on a Cluster Member fails to report its state on time, or when its state is reported as problematic, the state of that member is immediately changed to Down. The complete list of the configured critical devices (pnotes) is printed by the 'cphaprob -ia list' command or 'show cluster members pnotes all' command. Synonyms: Pnote, Problem Notification.

## D

**DAIP Gateway**
Dynamically Assigned IP (DAIP) Security Gateway is a Security Gateway, on which the IP address of the external interface is assigned dynamically by the ISP.

**Data Loss Prevention**
Check Point Software Blade on a Security Gateway that detects and prevents the unauthorized transmission of confidential information outside the organization. Acronym: DLP.

**Data Type**
Classification of data in a Check Point Security Policy for the Content Awareness Software Blade.

**Dead**
State reported by a Cluster Member when it goes out of the cluster (due to 'cphastop' command (which is a part of 'cpstop'), or reboot).

**Decision Function**
A special cluster algorithm applied by each Cluster Member on the incoming traffic in order to decide, which Cluster Member should process the received packet. Each Cluster Members maintains a table of hash values generated based on connections tuple (source and destination IP addresses/Ports, and Protocol number).

**Delta Sync**
Synchronization of kernel tables between all working Cluster Members - exchange of CCP packets that carry pieces of information about different connections and operations that should be performed on these connections in relevant kernel tables. This Delta Sync process is performed directly by Check Point kernel. While performing Full Sync, the Delta Sync updates are not processed and saved in kernel memory. After Full Sync is complete, the Delta Sync packets stored during the Full Sync phase are applied by order of arrival.

**Delta Sync Retransmission**
It is possible that Delta Sync packets will be lost or corrupted during the Delta Sync operations. In such cases, it is required to make sure the Delta Sync packet is re-sent. The Cluster Member requests the sending Cluster Member to retransmit the lost/corrupted Delta Sync packet. Each Delta Sync packet has a sequence number. The sending member has a queue of sent Delta Sync packets. Each Cluster Member has a queue of packets sent from each of the peer Cluster Members. If, for any reason, a Delta Sync packet was not received by a Cluster Member, it can ask for a retransmission of this packet from the sending member. The Delta Sync retransmission mechanism is somewhat similar to a TCP Window and TCP retransmission mechanism. When a member requests retransmission of Delta Sync packet, which no longer exists on the sending member, the member prints a console messages that the sync is not complete.

**Distributed Deployment**
Configuration in which the Check Point Security Gateway and the Security Management Server products are installed on different computers.

**Down**
State of a Cluster Member during a failure when one of the Critical Devices reports its state as "problem": In ClusterXL, applies to the state of the Security Gateway component; in 3rd-party / OPSEC cluster, applies to the state of the State Synchronization mechanism. A Cluster Member in this state does not process any traffic passing through cluster.

**Dying**
State of a Cluster Member as assumed by peer members, if it did not report its state for 0.7 second.

**Dynamic Object**
Special object type, whose IP address is not known in advance. The Security Gateway resolves the IP address of this object in real time.

## E

**Endpoint Policy Management**
Check Point Software Blade on a Management Server to manage an on-premises Harmony Endpoint Security environment.

**Expert Mode**
The name of the elevated command line shell that gives full system root permissions in the Check Point Gaia operating system.

## F

**Failback**
Recovery of a Cluster Member that suffered from a failure. The state of a recovered Cluster Member is changed from Down to either Active, or Standby (depending on Cluster Mode). Synonym: Fallback.

**Failed Member**
A Cluster Member that cannot send or accept traffic because of a hardware or software problem.

**Failover**
Transferring of a control over traffic (packet filtering) from a Cluster Member that suffered a failure to another Cluster Member (based on internal cluster algorithms). Synonym: Fail-over.

**Failure**
A hardware or software problem that causes a Security Gateway to be unable to serve as a Cluster Member (for example, one of cluster interface has failed, or one of the monitored daemon has crashed). Cluster Member that suffered from a failure is declared as failed, and its state is changed to Down (a physical interface is considered Down only if all configured VLANs on that physical interface are Down).

**Flapping**
Consequent changes in the state of either cluster interfaces (cluster interface flapping), or Cluster Members (Cluster Member flapping). Such consequent changes in the state are seen in the 'Logs & Monitor' > 'Logs' (if in SmartConsole > cluster object, the cluster administrator set the 'Track changes in the status of cluster members' to 'Log').

**Flush and ACK**

Cluster Member forces the Delta Sync packet about the incoming packet and waiting for acknowledgments from all other Active members and only then allows the incoming packet to pass through. In some scenarios, it is required that some information, written into the kernel tables, will be Sync-ed promptly, or else a race condition can occur. The race condition may occur if a packet that caused a certain change in kernel tables left Member_A toward its destination and then the return packet tries to go through Member_B. In general, this kind of situation is called asymmetric routing. What may happen in this scenario is that the return packet arrives at Member_B before the changes induced by this packet were Sync-ed to this Member_B. Example of such a case is when a SYN packet goes through Member_A, causing multiple changes in the kernel tables and then leaves to a server. The SYN-ACK packet from a server arrives at Member_B, but the connection itself was not Sync-ed yet. In this condition, the Member_B will drop the packet as an Out-of-State packet (First packet isn't SYN). In order to prevent such conditions, it is possible to use the "Flush and ACK" (F&A) mechanism. This mechanism can send the Delta Sync packets with all the changes accumulated so far in the Sync buffer to the other Cluster Members, hold the original packet that induced these changes and wait for acknowledgment from all other (Active) Cluster Members that they received the information in the Delta Sync packet. When all acknowledgments arrived, the mechanism will release the held original packet. This ensures that by the time the return packet arrived from a server at the cluster, all the Cluster Members are aware of the connection. F&A is being operated at the end of the Inbound chain and at the end of the Outbound chain (it is more common at the Outbound). Synonyms: FnA, F&A.

**Forwarding**

Process of transferring of an incoming traffic from one Cluster Member to another Cluster Member for processing. There are two types of forwarding the incoming traffic between Cluster Members - Packet forwarding and Chain forwarding. For more information, see "Forwarding Layer in Cluster" and "ARP Forwarding".

**Forwarding Layer**

The Forwarding Layer is a ClusterXL mechanism that allows a Cluster Member to pass packets to peer Cluster Members, after they have been locally inspected by the firewall. This feature allows connections to be opened from a Cluster Member to an external host. Packets originated by Cluster Members are hidden behind the Cluster Virtual IP address. Thus, a reply from an external host is sent to the cluster, and not directly to the source Cluster Member. This can pose problems in the following situations: (1) The cluster is working in High Availability mode, and the connection is opened from the Standby Cluster Member. All packets from the external host are handled by the Active Cluster Member, instead. (2) The cluster is working in a Load Sharing mode, and the decision function has selected another Cluster Member to handle this connection. This can happen since packets directed at a Cluster IP address are distributed between Cluster Members as with any other connection. If a Cluster Member decides, upon the completion of the firewall inspection process, that a packet is intended for another Cluster Member, it can use the Forwarding Layer to hand the packet over to that Cluster Member. In High Availability mode, packets are forwarded over a Synchronization network directly to peer Cluster Members. It is important to use secured networks only, as encrypted packets are decrypted during the inspection process, and are forwarded as clear-text (unencrypted) data. In Load Sharing mode, packets are forwarded over a regular traffic network. Packets that are sent on the Forwarding Layer use a special source MAC address to inform the receiving Cluster Member that they have already been inspected by another Cluster Member. Thus, the receiving Cluster Member can safely hand over these packets to the local Operating System, without further inspection.

**Full High Availability**

A special Cluster Mode supported only on Check Point appliances running Gaia OS (R75.40 and higher) or SecurePlatform OS (R77.30 and lower), where each Cluster Member also runs as a Security Management Server. This provides redundancy both between Security Gateways (only High Availability is supported) and between Security Management Servers (only High Availability is supported). Synonyms: Full HA Cluster Mode, Full HA, FullHA.

**Full Sync**
Process of full synchronization of applicable kernel tables by a Cluster Member from the working Cluster Member(s) when it tries to join the existing cluster. This process is meant to fetch a "snapshot" of the applicable kernel tables of already Active Cluster Member(s). Full Sync is performed during the initialization of Check Point software (during boot process, the first time the Cluster Member runs policy installation, during 'cpstart', during 'cphastart'). Until the Full Sync process completes successfully, this Cluster Member remains in the Down state, because until it is fully synchronized with other Cluster Members, it cannot function as a Cluster Member. Meanwhile, the Delta Sync packets continue to arrive, and the Cluster Member that tries to join the existing cluster, stores them in the kernel memory until the Full Sync completes. The whole Full Sync process is performed by fwd daemons on TCP port 256 over the Sync network (if it fails over the Sync network, it tries the other cluster interfaces). The information is sent by fwd daemons in chunks, while making sure they confirm getting the information before sending the next chunk. Also see "Delta Sync".

## G

**Gaia**
Check Point security operating system that combines the strengths of both SecurePlatform and IPSO operating systems.

**Gaia Clish**
The name of the default command line shell in Check Point Gaia operating system. This is a restricted shell (role-based administration controls the number of commands available in the shell).

**Gaia Portal**
Web interface for the Check Point Gaia operating system.

**Geo Cluster**
A High Availability cluster mode (in versions R81.20 and higher), where cluster members are located in different cloud availability zones. This mode supports the configuration of IP addresses from different subnets on all cluster interfaces, including the Sync interfaces. The Active cluster member inspects all traffic routed to the cluster and synchronizes the recorded connections to its peer cluster members. The traffic is not balanced between the cluster members. See "High Availability".

## H

**HA not started**
Output of the 'cphaprob <flag>' command or 'show cluster <option>' command on the Cluster Member. This output means that Check Point clustering software is not started on this Security Gateway (for example, this machine is not a part of a cluster, or 'cphastop' command was run, or some failure occurred that prevented the ClusterXL product from starting correctly).

**High Availability**
A redundant cluster mode, where only one Cluster Member (Active member) processes all the traffic, while other Cluster Members (Standby members) are ready to be promoted to Active state if the current Active member fails. In the High Availability mode, the Cluster Virtual IP address (that represents the cluster on that network) is associated: (1) With physical MAC Address of Active member (2) With virtual MAC Address. Synonym: Active/Standby. Acronym: HA.

**Hotfix**
Software package installed on top of the current software version to fix a wrong or undesired behavior, and to add a new behavior.

**HTTPS Inspection**
Feature on a Security Gateway that inspects traffic encrypted by the Secure Sockets Layer (SSL) protocol for malware or suspicious patterns. Synonym: SSL Inspection. Acronyms: HTTPSI, HTTPSi.

**HTU**
Stands for "HA Time Unit". All internal time in ClusterXL is measured in HTUs (the times in cluster debug also appear in HTUs). Formula in the Check Point software: 1 HTU = 10 x fwha_timer_base_res = 10 x 10 milliseconds = 100 ms.

**Hybrid**
Starting in R80.20, on Security Gateways with 40 or more CPU cores, Software Blades run in the user space (as 'fwk' processes). The Hybrid Mode refers to the state when you upgrade Cluster Members from R80.10 (or below) to R80.20 (or above). The Hybrid Mode is the state, in which the upgraded Cluster Members already run their Software Blades in the user space (as fwk processes), while other Cluster Members still run their Software Blades in the kernel space (represented by the fw_worker processes). In the Hybrid Mode, Cluster Members are able to synchronize the required information.

## I

**ICA**
Internal Certificate Authority. A component on Check Point Management Server that issues certificates for authentication.

**Identity Awareness**
Check Point Software Blade on a Security Gateway that enforces network access and audits data based on network location, the identity of the user, and the identity of the computer. Acronym: IDA.

**Identity Logging**
Check Point Software Blade on a Management Server to view Identity Logs from the managed Security Gateways with enabled Identity Awareness Software Blade.

**Init**
State of a Cluster Member in the phase after the boot and until the Full Sync completes. A Cluster Member in this state does not process any traffic passing through cluster.

**Internal Network**
Computers and resources protected by the Firewall and accessed by authenticated users.

**IP Tracking**
Collecting and saving of Source IP addresses and Source MAC addresses from incoming IP packets during the probing. IP tracking is a useful for Cluster Members to determine whether the network connectivity of the Cluster Member is acceptable.

**IP Tracking Policy**
Internal setting that controls, which IP addresses should be tracked during IP tracking: (1) Only IP addresses from the subnet of cluster VIP, or from subnet of physical cluster interface (this is the default) (2) All IP addresses, also outside the cluster subnet.

**IPS**
Check Point Software Blade on a Security Gateway that inspects and analyzes packets and data for numerous types of risks (Intrusion Prevention System).

**IPsec VPN**
Check Point Software Blade on a Security Gateway that provides a Site to Site VPN and Remote Access VPN access.

## J

**Jumbo Hotfix Accumulator**
Collection of hotfixes combined into a single package. Acronyms: JHA, JHF, JHFA.

## K

**Kerberos**
An authentication server for Microsoft Windows Active Directory Federation Services (ADFS).

## L

**Load Sharing**
A redundant cluster mode, where all Cluster Members process all incoming traffic in parallel. For more information, see "Load Sharing Multicast Mode" and "Load Sharing Unicast Mode". Synonyms: Active/Active, Load Balancing mode. Acronym: LS.

**Load Sharing Multicast**
Load Sharing Cluster Mode, where all Cluster Members process all traffic in parallel. Each Cluster Member is assigned the equal load of [ 100% / number_of_members ]. The Cluster Virtual IP address (that represents the cluster on that network) is associated with Multicast MAC Address 01:00:5E:X:Y:Z (which is generated based on last 3 bytes of cluster Virtual IP address on that network). A ClusterXL decision algorithm (Decision Function) on all Cluster Members decides, which Cluster Member should process the given packet.

**Load Sharing Unicast**
Load Sharing Cluster Mode, where one Cluster Member (called Pivot) accepts all traffic. Then, the Pivot member decides to process this traffic, or to forward it to other non-Pivot Cluster Members. The traffic load is assigned to Cluster Members based on the hard-coded formula per the value of "Pivot_overhead" attribute in the cluster object. The Cluster Virtual IP address (that represents the cluster on that network) is associated with: (1) Physical MAC Address of Pivot member (2) Virtual MAC Address.

**Log Server**
Dedicated Check Point server that runs Check Point software to store and process logs.

**Logging & Status**
Check Point Software Blade on a Management Server to view Security Logs from the managed Security Gateways.

## M

**Management Interface**
(1) Interface on a Gaia Security Gateway or Cluster member, through which Management Server connects to the Security Gateway or Cluster member. (2) Interface on Gaia computer, through which users connect to Gaia Portal or CLI.

**Management Server**
Check Point Single-Domain Security Management Server or a Multi-Domain Security Management Server.

**Manual NAT Rules**
Manual configuration of NAT rules by the administrator of the Check Point Management Server.

**Master**
State of a Cluster Member that processes all traffic in cluster configured in VRRP mode.

**Mobile Access**
Check Point Software Blade on a Security Gateway that provides a Remote Access VPN access for managed and unmanaged clients. Acronym: MAB.

**Multi-Domain Log Server**
Dedicated Check Point server that runs Check Point software to store and process logs in a Multi-Domain Security Management environment. The Multi-Domain Log Server consists of Domain Log Servers that store and process logs from Security Gateways that are managed by the corresponding Domain Management Servers. Acronym: MDLS.

**Multi-Domain Server**
Dedicated Check Point server that runs Check Point software to host virtual Security Management Servers called Domain Management Servers. Synonym: Multi-Domain Security Management Server. Acronym: MDS.

**Multi-Version Cluster**
The Multi-Version Cluster mechanism lets you synchronize connections between cluster members that run different versions. This lets you upgrade to a newer version without a loss in connectivity and lets you test the new version on some of the cluster members before you decide to upgrade the rest of the cluster members. Acronym: MVC.

## N

**Network Object**
Logical object that represents different parts of corporate topology - computers, IP addresses, traffic protocols, and so on. Administrators use these objects in Security Policies.

**Network Objective**
Defines how the cluster will configure and monitor an interface - Cluster, Sync, Cluster+Sync, Monitored Private, Non-Monitored Private. Configured in SmartConsole > cluster object > 'Topology' pane > 'Network Objective'.

**Network Policy Management**
Check Point Software Blade on a Management Server to manage an on-premises environment with an Access Control and Threat Prevention policies.

**Non-Blocking Mode**
Cluster operation mode, in which Cluster Member keeps forwarding all traffic.

**Non-Monitored Interface**
An interface on a Cluster Member, whose Network Type was set as Private in SmartConsole, in cluster object.

**Non-Pivot**
A Cluster Member in the Unicast Load Sharing cluster that receives all packets from the Pivot Cluster Member.

**Non-Sticky Connection**
A connection is called non-sticky, if the reply packet returns via a different Cluster Member, than the original packet (for example, if network administrator has configured asymmetric routing). In Load Sharing mode, all Cluster Members are Active, and in Static NAT and encrypted connections, the Source and Destination IP addresses change. Therefore, Static NAT and encrypted connections through a Load Sharing cluster may be non-sticky.

## O

**Open Server**
Physical computer manufactured and distributed by a company, other than Check Point.

## P

**Packet Selection**
Distinguishing between different kinds of packets coming from the network, and selecting, which member should handle a specific packet (Decision Function mechanism): CCP packet from another member of this cluster; CCP packet from another cluster or from a Cluster; Member with another version (usually older version of CCP); Packet is destined directly to this member; Packet is destined to another member of this cluster; Packet is intended to pass through this Cluster Member; ARP packets.

**Pingable Host**
Some host (that is, some IP address) that Cluster Members can ping during probing mechanism. Pinging hosts in an interface's subnet is one of the health checks that ClusterXL mechanism performs. This pingable host will allow the Cluster Members to determine with more precision what has failed (which interface on which member). On Sync network, usually, there are no hosts. In such case, if switch supports this, an IP address should be assigned on the switch (for example, in the relevant VLAN). The IP address of such pingable host should be assigned per this formula: IP_of_pingable_host = IP_of_physical_interface_on_member + ~10. Assigning the IP address to pingable host that is higher than the IP addresses of physical interfaces on the Cluster Members will give some time to Cluster Members to perform the default health checks. Example: IP address of physical interface on a given subnet on Member_A is 10.20.30.41; IP address of physical interface on a given subnet on Member_B is 10.20.30.42; IP address of pingable host should be at least 10.20.30.5

**Pivot**
A Cluster Member in the Unicast Load Sharing cluster that receives all packets. Cluster Virtual IP addresses are associated with Physical MAC Addresses of this Cluster Member. This Pivot Cluster Member distributes the traffic between other Non-Pivot Cluster Members.

**Preconfigured Mode**
Cluster Mode, where cluster membership is enabled on all Cluster Members to be. However, no policy had been yet installed on any of the Cluster Members - none of them is actually configured to be primary, secondary, and so on. The cluster cannot function, if one Cluster Member fails. In this scenario, the "preconfigured mode" takes place. The preconfigured mode also comes into effect when no policy is yet installed, right after the Cluster Members came up after boot, or when running the 'cphaconf init' command.

**Primary Up**
ClusterXL in High Availability mode that was configured as Switch to higher priority Cluster Member in the cluster object in SmartConsole: (1) Each Cluster Member is given a priority (SmartConsole > cluster object > 'Cluster Members' pane). Cluster Member with the highest priority appears at the top of the table, and Cluster Member with the lowest priority appears at the bottom of the table. (2) The Cluster Member with the highest priority will assume the Active state. (3) If the current Active Cluster Member with the highest priority (for example, Member_A), fails for some reason, or is rebooted, then failover occurs between Cluster Members. The Cluster Member with the next highest priority will be promoted to be Active (for example, Member_B). (4) When the Cluster Member with the highest priority (Member_A) recovers from a failure, or boots, then additional failover occurs between Cluster Members. The Cluster Member with the highest priority (Member_A) will be promoted to Active state (and Member_B will return to Standby state).

**Private Interface**
An interface on a Cluster Member, whose Network Type was set as 'Private' in SmartConsole in cluster object. This interface is not monitored by cluster, and failure on this interface will not cause any changes in Cluster Member's state.

**Probing**
If a Cluster Member fails to receive status for another member (does not receive CCP packets from that member) on a given segment, Cluster Member will probe that segment in an attempt to illicit a response. The purpose of such probes is to detect the nature of possible interface failures, and to determine which module has the problem. The outcome of this probe will determine what action is taken next (change the state of an interface, or of a Cluster Member).

**Provisioning**
Check Point Software Blade on a Management Server that manages large-scale deployments of Check Point Security Gateways using configuration profiles. Synonyms: SmartProvisioning, SmartLSM, Large-Scale Management, LSM.

## Q

**QoS**
Check Point Software Blade on a Security Gateway that provides policy-based traffic bandwidth management to prioritize business-critical traffic and guarantee bandwidth and control latency.

## R

**Ready**
State of a Cluster Member during after initialization and before promotion to the next required state - Active / Standby / VRRP Master / VRRP Backup (depending on Cluster Mode). A Cluster Member in this state does not process any traffic passing through cluster. A member can be stuck in this state due to several reasons.

**Rule**
Set of traffic parameters and other conditions in a Rule Base (Security Policy) that cause specified actions to be taken for a communication session.

**Rule Base**
All rules configured in a given Security Policy. Synonym: Rulebase.

## S

**Same VMAC**
Same Virtual MAC Address (see "VMAC"). When this feature is enabled in a ClusterXL (in the High Availability or Load Sharing Unicast mode), the Cluster Members use Virtual MAC (VMAC) addresses on the cluster interfaces instead of the real MAC addresses. Cluster interfaces that belong to the same subnet get the same VMAC address instead of their real MAC address. This feature helps avoid issues during the cluster operation, when switches block ports connected to the Cluster Members.

**SecureXL**
Check Point product on a Security Gateway that accelerates IPv4 and IPv6 traffic that passes through a Security Gateway.

**Security Gateway**
Dedicated Check Point server that runs Check Point software to inspect traffic and enforce Security Policies for connected network resources.

**Security Management Server**
Dedicated Check Point server that runs Check Point software to manage the objects and policies in a Check Point environment within a single management Domain. Synonym: Single-Domain Security Management Server.

**Security Policy**
Collection of rules that control network traffic and enforce organization guidelines for data protection and access to resources with packet inspection.

**Selection**
The packet selection mechanism is one of the central and most important components in the ClusterXL product and State Synchronization infrastructure for 3rd-party clustering solutions. Its main purpose is to decide (to select) correctly what has to be done to the incoming and outgoing traffic on the Cluster Member. (1) In ClusterXL, the packet is selected by Cluster Member(s) depending on the cluster mode: In HA modes - by Active member; In LS Unicast mode - by Pivot member; In LS Multicast mode - by all members. Then the Cluster Member applies the Decision Function (and the Cluster Correction Layer). (2) In 3rd-party / OPSEC cluster, the 3rd-party software selects the packet, and Check Point software just inspects it (and performs State Synchronization).

**SIC**
Secure Internal Communication. The Check Point proprietary mechanism with which Check Point computers that run Check Point software authenticate each other over SSL, for secure communication. This authentication is based on the certificates issued by the ICA on a Check Point Management Server.

**Silent Standby**
In the ClusterXL High Availability mode, this feature configures the Standby cluster member to communicate only through the Active cluster member. This feature is useful when it is necessary to connect from Standby cluster members to a host / server on the network.

**SmartConsole**
Check Point GUI application used to manage a Check Point environment - configure Security Policies, configure devices, monitor products and events, install updates, and so on.

**SmartDashboard**
Legacy Check Point GUI client used to create and manage the security settings in versions R77.30 and lower. In versions R80.X and higher is still used to configure specific legacy settings.

**SmartProvisioning**
Check Point Software Blade on a Management Server (the actual name is "Provisioning") that manages large-scale deployments of Check Point Security Gateways using configuration profiles. Synonyms: Large-Scale Management, SmartLSM, LSM.

**SmartUpdate**
Legacy Check Point GUI client used to manage licenses and contracts in a Check Point environment.

**Software Blade**
Specific security solution (module): (1) On a Security Gateway, each Software Blade inspects specific characteristics of the traffic (2) On a Management Server, each Software Blade enables different management capabilities.

**Standalone**
Configuration in which the Security Gateway and the Security Management Server products are installed and configured on the same server.

**Standby**
State of a Cluster Member that is ready to be promoted to Active state (if the current Active Cluster Member fails). Applies only to ClusterXL High Availability Mode.

**State Synchronization**
Technology that synchronizes the relevant information about the current connections (stored in various kernel tables on Check Point Security Gateways) among all Cluster Members over Synchronization Network. Due to State Synchronization, the current connections are not cut off during cluster failover.

**Sticky Connection**
A connection is called sticky, if all packets are handled by a single Cluster Member (in High Availability mode, all packets reach the Active Cluster Member, so all connections are sticky).

**Subscribers**
User Space processes that are made aware of the current state of the ClusterXL state machine and other clustering configuration parameters. List of such subscribers can be obtained by running the 'cphaconf debug_data' command.

**Sync Interface**
An interface on a Cluster Member, whose Network Type was set as Sync or Cluster+Sync in SmartConsole in cluster object. This interface is monitored by cluster, and failure on this interface will cause cluster failover. This interface is used for State Synchronization between Cluster Members. The use of more than one Sync Interfaces for redundancy is not supported because the CPU load will increase significantly due to duplicate tasks performed by all configured Synchronization Networks. Synonyms: Secured Interface, Trusted Interface.

**Synchronization Network**
A set of interfaces on Cluster Members that were configured as interfaces, over which State Synchronization information will be passed (as Delta Sync packets ). The use of more than one Synchronization Network for redundancy is not supported because the CPU load will increase significantly due to duplicate tasks performed by all configured Synchronization Networks. Synonyms: Sync Network, Secured Network, Trusted Network.

## T

**Threat Emulation**
Check Point Software Blade on a Security Gateway that monitors the behavior of files in a sandbox to determine whether or not they are malicious. Acronym: TE.

**Threat Extraction**
Check Point Software Blade on a Security Gateway that removes malicious content from files. Acronym: TEX.

## U

**Updatable Object**
Network object that represents an external service, such as Microsoft 365, AWS, Geo locations, and more.

**URL Filtering**
Check Point Software Blade on a Security Gateway that allows granular control over which web sites can be accessed by a given group of users, computers or networks. Acronym: URLF.

**User Directory**
Check Point Software Blade on a Management Server that integrates LDAP and other external user management servers with Check Point products and security solutions.

## V

**VMAC**
Virtual MAC Address. When this feature is enabled in a ClusterXL (in the High Availability or Load Sharing Unicast mode), the current Active or Pivot Cluster Member sends Gratuitous ARP Requests (G-ARP) for its Cluster Virtual IP (VIP) addresses and Virtual MAC (VMAC) addresses in G-ARP updates. Cluster Members create a VMAC address for each Cluster VIP address. This feature helps avoid issues during a cluster failover, when switches do not integrate G-ARP updates into their ARP cache table.

**VSX**
Virtual System Extension. Check Point virtual networking solution, hosted on a computer or cluster with virtual abstractions of Check Point Security Gateways and other network devices. These Virtual Devices provide the same functionality as their physical counterparts.

**VSX Gateway**
Physical server that hosts VSX virtual networks, including all Virtual Devices that provide the functionality of physical network devices. It holds at least one Virtual System, which is called VS0.

## Z

**Zero Phishing**
Check Point Software Blade on a Security Gateway (R81.20 and higher) that provides real-time phishing prevention based on URLs. Acronym: ZPH.