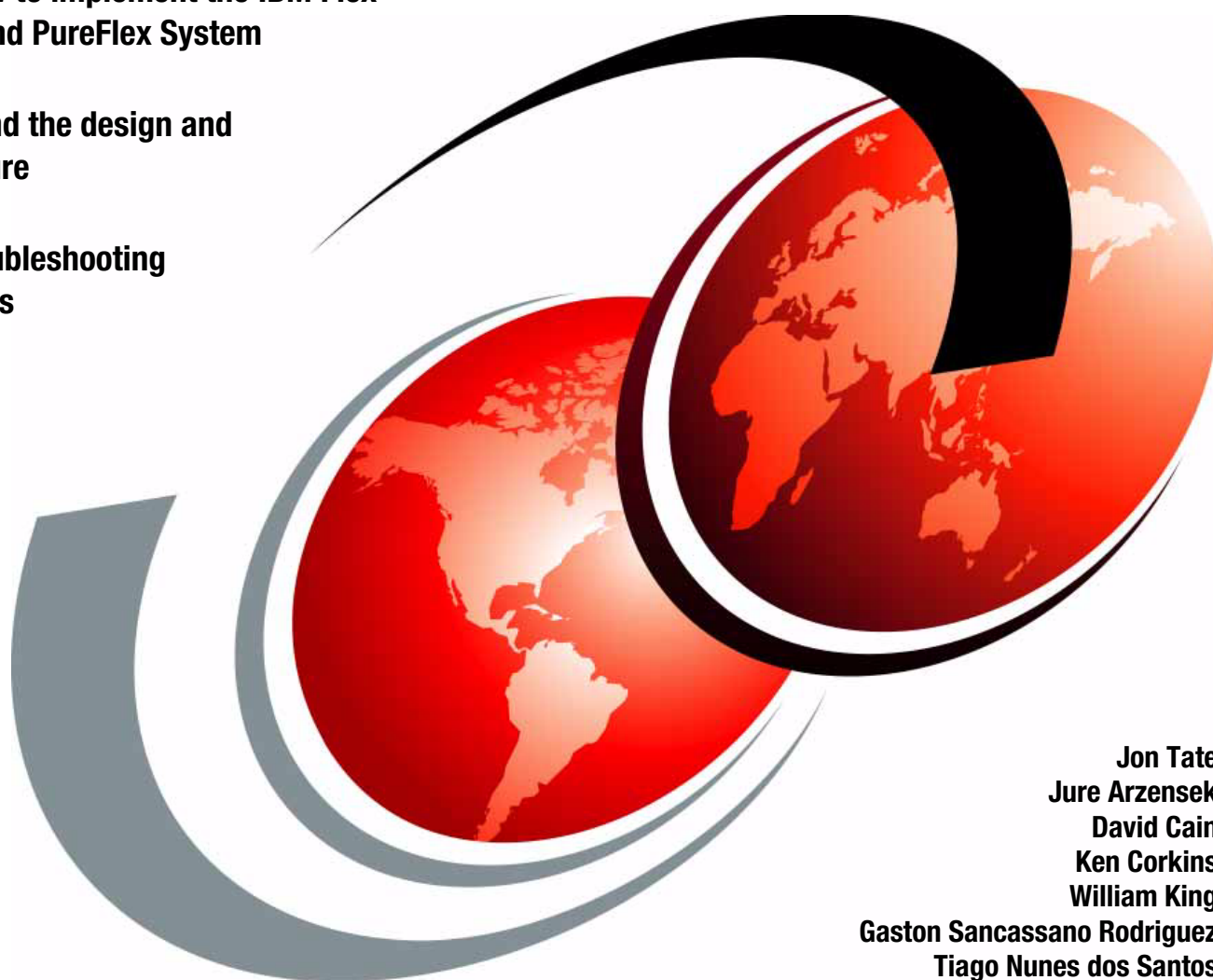


IBM Flex System and PureFlex System Network Implementation

Learn how to implement the IBM Flex
System and PureFlex System

Understand the design and
architecture

Learn troubleshooting
techniques



Jon Tate
Jure Arzensek
David Cain
Ken Corkins
William King

Gaston Sancassano Rodriguez
Tiago Nunes dos Santos

Redbooks



International Technical Support Organization

**IBM Flex System and PureFlex System Network
Implementation**

July 2013

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (July 2013)

This edition applies to the IBM PureFlex System software and hardware available in September 2012. This may, or may not, include pre-GA code.

© Copyright International Business Machines Corporation 2013. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
 Preface	 ix
Authors	ix
Now you can become a published author, too!	xi
Comments welcome	xi
Stay connected to IBM Redbooks	xii
 Chapter 1. Introduction to IBM PureFlex System Networking	 1
1.1 Introduction to IBM PureSystems	2
1.1.1 IBM Flex System Enterprise Chassis	2
1.2 IBM Flex System Networking product family	3
1.2.1 IBM Flex System Fabric EN4093 and EN4093R 10Gb Scalable Switch	4
1.2.2 IBM Flex System EN4091 10Gb Ethernet Pass-thru Module	5
1.2.3 IBM Flex System EN2092 1Gb Ethernet Scalable Switch	6
1.2.4 I/O modules and cables	8
1.3 IBM Flex System Ethernet adapters	9
1.3.1 IBM Flex System CN4054 10Gb Virtual Fabric Adapter	9
1.3.2 IBM Flex System EN4054 4-port 10Gb Ethernet Adapter	10
1.3.3 IBM Flex System EN2024 4-port 1Gb Ethernet Adapter	11
1.3.4 IBM Flex System EN4132 2-port 10Gb Ethernet Adapter	11
1.4 IBM Networking Operating System	12
1.4.1 Command-line interface	13
1.4.2 Browser-Based Interface	13
 Chapter 2. Data center design and architecture	 15
2.1 Open Data Center Interoperable Network	17
2.2 VMready	18
2.2.1 VMready benefits	20
2.2.2 VMready compatibility	20
2.2.3 VMready references	20
2.3 IBM Flex System data center high availability components	21
2.3.1 The Ethernet switch I/O module	21
2.3.2 Virtual local area networks	22
2.3.3 Scalability and performance	23
2.3.4 Link aggregation	25
2.3.5 Layer 2 failover	27
2.3.6 NIC teaming	28
2.4 Fibre Channel over Ethernet solution capabilities	29
2.4.1 FCoE references	30
2.5 Virtual Fabric vNIC solution capabilities	30
2.5.1 Virtual Fabric mode vNIC	31
2.5.2 Switch-independent mode vNIC	32
2.5.3 Virtual Fabric references	32
2.6 High availability use cases	33
2.6.1 Highly available topologies	34
2.7 Practical Use Case 1: Fully redundant with virtualized chassis technology (VSS/vPC/VLAG)	38

2.8 Other use cases	40
2.8.1 Practical Use Case 2: Fully redundant with traditional Spanning Tree	40
2.8.2 Practical Use Case 3: Fully redundant with Open Shortest Path First	40
2.9 Summary and conclusions	41
Chapter 3. Planning and hardware selection	43
3.1 Hardware selection and interoperability	44
3.1.1 Selecting the Ethernet switch module	44
3.1.2 Selecting the compute node NICs	45
3.1.3 Interoperability considerations	46
3.2 Virtual local area networks	46
3.3 Scalability and performance	47
3.4 High availability	49
3.4.1 Examples of topologies	50
3.4.2 Spanning Tree	53
3.4.3 Link aggregation	54
3.4.4 Network interface card teaming	55
3.4.5 Trunk failover	56
3.4.6 Virtual Router Redundancy Protocol	58
3.5 Virtual Fabric vNIC solution capabilities	59
3.5.1 Virtual Fabric mode vNIC	60
3.5.2 Switch-independent mode vNIC	61
3.6 Management	61
3.6.1 Management tools and their capabilities	62
Chapter 4. Initial configuration	65
4.1 Initial hardware installation	66
4.1.1 Installing a switch	67
4.2 Initial Software Configuration	69
4.2.1 Administration interfaces	69
4.2.2 First boot of the IBM Flex System Ethernet I/O modules	71
4.2.3 Connecting to the switch	74
4.2.4 Setup Tool	76
4.2.5 Setup command-line interface procedure	80
4.2.6 User management	82
Chapter 5. Compute node network configuration	85
5.1 Introduction and background	85
5.1.1 NIC teaming	85
5.2 Components used and setup	86
5.2.1 Testing methodology	87
5.2.2 Logical diagram	88
5.3 Microsoft Windows Server 2008	89
5.3.1 Implementation	89
5.4 Red Hat Enterprise Linux Server 6	100
5.4.1 Implementation	100
5.5 VMware ESXi 5	103
5.5.1 Implementation	103
Chapter 6. Implementation of IBM PureFlex Systems and IBM System Networking connectivity	111
6.1 Introduction	112
6.2 IBM System Networking components used	113
6.3 Physical setup	113

6.4	EN4093flex_1 configuration	113
6.5	G8264tor_1 configuration	118
6.6	Verification and show command output	120
6.7	Full configuration files	134
Chapter 7.	System management	157
7.1	Management network	158
7.2	Chassis Management Module	159
7.2.1	Overview	159
7.2.2	Chassis Management Module user interfaces	160
7.3	Security	162
7.4	Compute node management	163
7.4.1	Integrated management module II	164
7.4.2	Flexible service processor	165
7.5	I/O modules management	166
7.5.1	I/O module management in Chassis Management Module web interface	167
7.6	IBM Flex System Manager	171
7.6.1	Hardware overview	174
7.6.2	Software features	177
7.6.3	Supported agents, hardware, operating systems and tasks	180
7.7	IBM System Networking Element Manager Component	182
7.7.1	Health Summary pane	184
7.7.2	Viewing Health Status	184
7.7.3	Viewing reports	185
7.8	IBM System Networking Element Manager Solution	186
7.8.1	IBM System Networking Element Manager solution requirements	187
Chapter 8.	Troubleshooting and maintenance	189
8.1	Troubleshooting	190
8.1.1	Basic troubleshooting procedures	190
8.1.2	Connectivity troubleshooting	194
8.1.3	Port mirroring	195
8.1.4	Serial cable troubleshooting procedures	197
8.2	Configuration management	198
8.2.1	Configuration files	198
8.2.2	Configuration blocks	198
8.2.3	Managing configuration files	198
8.2.4	Resetting to factory defaults	200
8.2.5	Password recovery	206
8.3	Firmware management	206
8.3.1	Firmware images	206
8.3.2	Upgrading the firmware with ISCLI	208
8.3.3	Recovering from a failed firmware upgrade	212
8.4	Logging and reporting	214
8.4.1	System logs	214
8.4.2	Simple Network Management Protocol	217
8.4.3	Remote Monitoring	220
8.4.4	Using sFlow to monitor traffic	222
Appendix A.	An integration guide to the IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch	225
	Overview of the factory network configuration	226
	Purple network	228
	Green network	228

Gold network	228
EN4093/EN4093R VLAN configuration	228
Consolidating VLANs across Link Aggregation Groups	230
Consolidating VLANs across a single link	235
Nexus configuration	236
EN4093/EN4093R configuration	237
Cisco Nexus switch configuration	239
Changing the VLAN IDs in the Flex	244
Show EN4093/EN4093R running configuration	246
Adding a second IBM EN4093/EN4093R switch to the setup	251
Appendix B. Easy Connect	253
8.5 Introduction to IBM Easy Connect	254
8.6 Easy Connect Single Mode	254
8.6.1 Implementation	255
8.7 Storage Mode	256
8.7.1 Implementation	257
8.8 Easy Connect Multi-Chassis Mode	258
8.8.1 Implementation with CN/EN4093/R	259
8.8.2 Implementation with G8264	260
8.9 Client examples with diagrams	262
8.9.1 Telecommunications client	262
8.9.2 State government client	263
8.9.3 Medical center client	264
8.10 Easy Connect limitations	266
Related publications	267
IBM Redbooks	267
Online resources	267
Help from IBM	267

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	Micromuse®	RackSwitch™
Blade Network Technologies®	Netcool®	Redbooks®
BladeCenter®	POWER Hypervisor™	Redbooks (logo)  ®
BNT®	Power Systems™	Storwize®
developerWorks®	POWER7®	System Storage®
DS4000®	PowerVM®	System x®
Extreme Blue®	POWER®	Tivoli®
IBM Flex System™	PureApplication™	VMready®
IBM Flex System Manager™	PureData™	X-Architecture®
IBM PureData™	PureFlex™	
IBM®	PureSystems™	

The following terms are trademarks of other companies:

Intel Xeon, Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.

Preface

To meet today's complex and ever-changing business demands, you need a solid foundation of server, storage, networking, and software resources that are simple to deploy and can quickly and automatically adapt to changing conditions. You also need access to, and the ability to take advantage of, broad expertise and proven best practices in systems management, applications, hardware maintenance, and more.

IBM® PureFlex™ System, which is a part of the IBM PureSystems™ family of expert integrated systems, combines advanced IBM hardware and software along with patterns of expertise and integrates them into three optimized configurations that are simple to acquire and deploy so that you can achieve faster time to value.

If you want a preconfigured, preintegrated infrastructure with integrated management and cloud capabilities, factory tuned from IBM with x86 and Power Systems™ hybrid solution, IBM PureFlex System is the answer.

In this IBM Redbooks® publication, which is aimed at system and network administrators, we show the design and architecture, how to configure hosts and switches, maintain, and troubleshoot using the IBM Flex System™ Ethernet I/O modules (EN2092 1Gb Ethernet Scalable Switch and EN4093R 10Gb Scalable Switch).

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), San Jose Center.



Jon Tate is a Project Manager for IBM System Storage® SAN Solutions at the International Technical Support Organization, San Jose Center. Before joining the ITSO in 1999, he worked in the IBM Technical Support Center, providing Level 2 support for IBM storage products. Jon has 27 years of experience in storage software and management, services, and support, and is both an IBM Certified IT Specialist and an IBM SAN Certified Specialist. He is also the UK Chairman of the Storage Networking Industry Association.



Jure Arzensek is an Advisory IT Specialist for IBM Slovenia and works for the EMEA Level 2 team, supporting PureFlex and IBM BladeCenter® products. He has been with IBM since 1995 and has worked in various technical support and technical education roles. Jure holds a degree in Computer Science from the University of Ljubljana. His other areas of expertise include IBM System x® servers, SAN, System Storage DS3000, IBM DS4000®, and DS5000 products and network operating systems for the Intel platform. He has co-authored eleven other IBM Redbooks publications.



David Cain is a Network and Systems Engineer for the IBM Software Group in Research Triangle Park, North Carolina. He has nine years of experience in the data center, with expertise in Ethernet switching, storage, SAN, security, virtualization, IBM System x, and Linux server infrastructure. Dave holds a Bachelor of Science degree in Computer Science from North Carolina State University, and has co-authored two patents and invention disclosures in the networking field. He joined IBM full-time in 2006 after gaining valuable experience on various internships with IBM while he was a student, including an IBM Extreme Blue® internship in 2005.



Ken Corkins is a member of the IBM System x Advanced Technical Support Group in Dallas, Texas. He specializes in BladeCenter and networking. His responsibilities include assisting BladeCenter pre-sale clients in the Americas with proof of concepts and pilot testing. He organized the Nortel Champions program for North America in Dallas. He has 17 years of experience in the IT industry and holds the Cisco CCNP and CCDP certifications. Ken started with IBM in October 2000 as an IT Specialist (Networking) with IBM Global Services.



William King works for IBM Software Group, Tivoli® Division, IBM UK, as part of the Network Management team. His role is as a network architect developing scenarios on the test network used by the IBM Tivoli Network Manager and ITNCM development teams. As a former IBM Micromuse® employee, he has been working on the IBM Tivoli Netcool® suite of products for over 10 years. He is familiar with a wide range of different network equipment from optical and MPLS WAN topologies to data center Fibre Channel and iSCSI storage. He has worked with Cisco, Juniper, Huawei, Nortel, IBM System Networking, Brocade, Foundry, and Extreme equipment. He has a PhD in Immunology from Birmingham University.



Gaston Sancassano Rodriguez is a Network Specialist for IBM Uruguay. He has almost seven years of experience working in the design and implementation of Networking and Security projects. His main specialities include routing, switching, and wireless. He holds an Engineering degree in Telecommunications from Universidad ORT and several Cisco and Juniper certifications in Routing and Switching.



Tiago Nunes dos Santos is a Gold Redbooks author and the Infrastructure Strategy Leader for the IBM Linux Technology Center, IBM Brazil. He is a Staff Software Engineer and specialized System Administrator, and an expert on the Operating Systems/Application stack, network architecture, and IT User Support processes. Tiago has been working on both the enterprise and open source community for over seven years, accumulating expertise in innovation, IT architecture, and strategy leadership. His knowledge on IT infrastructure architecture helped him become an IBM Inventor, and he is also a member of the Brazilian IBM developerWorks® technical reviewing board.

Thanks to the following people for their contributions to this project:

Sangam Racherla
International Technical Support Organization, San Jose Center

Cathy Lemeshefsky
Pushkar Patil
Tim Shaughnessy
IBM San Jose

Scott Irwin
IBM Dallas

Scott Lorditch
IBM Denver

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- Send your comments in an email to:

redbooks@us.ibm.com

- Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>



Introduction to IBM PureFlex System Networking

Complexity has become a roadblock to data center efficiency. In order to have efficient, cost-effective data center operation, IT administrators must integrate and simplify network resources and data center management. The solution to simplifying management, reducing costs, easing complexity, and increasing time-to-value lies in addressing all the elements of the system: servers, storage, and especially networking.

IBM Flex System provides compute, storage, and networking resources in a single environment that is both efficient and easy to manage. IBM PureFlex System, IBM PureApplication™ System, and IBM PureData™ System each offer an integrated solution using components from the IBM Flex System portfolio. These components provide advanced networking with flexibility for various workloads.

The following topics are described in this chapter:

- ▶ An introduction to IBM PureSystems
 - IBM Flex System Enterprise Chassis
- ▶ IBM Flex System Networking product family
 - Ethernet I/O modules
 - Ethernet adapters
- ▶ IBM Network Operating System (Network OS)
 - Industry Standard command-line interface (ISCLI) and Browser-Based Interface (BBI)

1.1 Introduction to IBM PureSystems

IBM PureSystems combines the flexibility of general-purpose computer systems, the elasticity of cloud computing, and the simplicity of an appliance that is tuned to the workload at hand. Expert integrated systems are essentially the building blocks of capability in an enterprise data center. This category of systems represents the collective knowledge of thousands of deployments, established best practice guidelines, innovative thinking, IT leadership, and decades of experience and expertise.

The offerings in IBM PureSystems are designed to deliver value in the following ways:

- ▶ Built-in expertise helps you to address complex business and operational tasks automatically.
- ▶ Integration by design helps you to tune systems for optimal performance and efficiency.
- ▶ Simplified experience, from design to purchase to maintenance, creates efficiencies quickly.

The IBM PureSystems offerings are optimized for performance and virtualized for efficiency. These systems offer a no-compromise design with system-level upgradeability. IBM PureSystems are built for the next-generation data center, containing “built-in” flexibility and simplicity.

At IBM, expert integrated systems come in three types:

- ▶ IBM PureFlex System: An infrastructure system that provides an integrated computing environment—combining servers, storage, networking, virtualization, and management into a single offering.
- ▶ IBM PureApplication System: A platform system specifically designed and tuned for running applications. The system supports the use of patterns for easy deployment into its cloud environment.
- ▶ IBM PureData System: A platform that is designed and optimized for your data services.

1.1.1 IBM Flex System Enterprise Chassis

The IBM Flex System Enterprise Chassis offers compute, networking, and storage capabilities far exceeding those currently available. With the ability to handle up to 14 compute nodes, intermixing IBM Power Systems and Intel x86, the Enterprise Chassis provides flexibility and tremendous compute capacity in a 10U package. Additionally, the rear of the chassis accommodates four high-speed networking switches. With interconnecting compute nodes, networking, and storage using a high performance and scalable mid-plane, Enterprise Chassis can support 40 Gb speeds.

The IBM Flex System Enterprise Chassis (machine type 8721) is a 10U next-generation server platform with integrated chassis management. It is a compact, high-density, high-performance, rack-mounted, scalable server platform system. It supports up to 14 one-bay compute nodes that can share common resources, such as power, cooling, management, and I/O resources within a single Enterprise Chassis. In addition, it can also support up to seven 2-bay compute nodes or three 4-bay compute nodes when the shelves are removed from the chassis. You can mix and match 1-bay, 2-bay, and 4-bay compute nodes to meet your specific hardware needs.

The ability to support the workload demands of tomorrow's workloads is built in with a new I/O architecture, which provides choice and flexibility in fabric and speed. With the ability to use Ethernet, InfiniBand, Fibre Channel, Fibre Channel over Ethernet (FCoE), and

Internet Small Computer System Interface (iSCSI), the Enterprise Chassis is uniquely positioned to meet the growing and future I/O needs of businesses both large and small.

Figure 1-1 shows a frontal view of the IBM Flex System Enterprise Chassis.

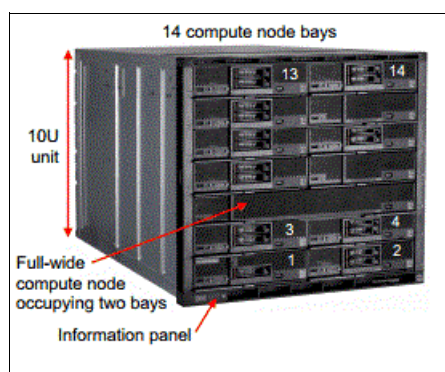


Figure 1-1 IBM Flex System Enterprise Chassis front view

Figure 1-2 shows the rear view of the IBM Flex System Enterprise Chassis.

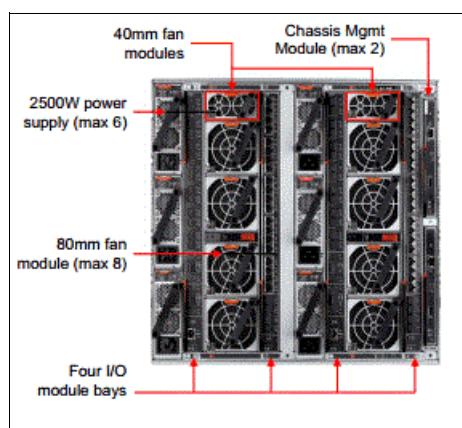


Figure 1-2 IBM Flex System Enterprise Chassis rear view

1.2 IBM Flex System Networking product family

The IBM Flex System Enterprise Chassis features a number of Ethernet I/O module solutions providing a combination of 1 Gb and 10 Gb ports to the node bays and 1 Gb, 10 Gb, and 40 Gb for uplink connectivity to the outside upstream infrastructure. The IBM Flex System Enterprise Chassis ensures that a suitable selection is available to meet the needs of the server nodes.

There are three Ethernet I/O modules available for deployment with the Enterprise Chassis:

- ▶ 1.2.1, “IBM Flex System Fabric EN4093 and EN4093R 10Gb Scalable Switch” on page 4
- ▶ 1.2.2, “IBM Flex System EN4091 10Gb Ethernet Pass-thru Module” on page 5
- ▶ 1.2.3, “IBM Flex System EN2092 1Gb Ethernet Scalable Switch” on page 6

1.2.1 IBM Flex System Fabric EN4093 and EN4093R 10Gb Scalable Switch

The IBM Flex System EN4093 and IBM Flex System EN4093R 10Gb Scalable Switches are 10Gb 64-port upgradeable midrange to high-end switch modules. They offer Layer 2/3 switching designed to install within the I/O module bays of the Enterprise Chassis.

The latest 4093R switch adds additional capabilities to the EN4093, that of stacking: Virtual network interface card (NIC) (Stacking), Unified fabric port (Stacking), Edge virtual bridging (Stacking), and CEE/FCoE (Stacking) and so is ideal for clients looking to implement a converged infrastructure with NAS, iSCSI, or FCoE. For an FCoE implementation, the EN4093R acts as a transit switch forwarding FCoE traffic upstream to another device such as the Brocade VDX or Cisco Nexus 5548/5596 where the Fibre Channel traffic diverges.

Each switch contains the following ports:

- ▶ Up to 42 internal 10-Gb ports
- ▶ Up to 14 external 10-Gb uplink ports (enhanced small form-factor pluggable (SFP+) connectors)
- ▶ Up to 2 external 40-Gb uplink ports (quad small form-factor pluggable (QSFP+) connectors), which can also be broken out into (8) 10-Gb SFP+ connections instead.

The EN4093/EN4093R 10-Gb Scalable Switch is shown in Figure 1-3.



Figure 1-3 The IBM Flex System EN4093/EN4093R 10-Gb Scalable Switch

This switch is considered particularly suited for clients who:

- ▶ Will be building a 10-Gb infrastructure
- ▶ Are implementing a virtualized environment
- ▶ Are preparing for the adaptation of 40-Gb Ethernet in the data center
- ▶ Want to reduce TCO, improve performance, while maintaining high levels of availability and security
- ▶ Want to avoid oversubscription (traffic from multiple internal ports attempting to pass through a lower quantity of external ports, leading to congestion and performance impact)

Table 1-1 on page 5 provides the IBM Flex System Fabric EN4093 10Gb Scalable Switch part numbers and port upgrades.

Table 1-1 IBM Flex System Fabric EN4093 10Gb Scalable Switch part numbers and port upgrades

Part number	Feature code ^a	Product description	Total ports enabled		
			Internal	10 Gb uplink	40 Gb uplink
49Y4270	A0TB / 3593	IBM Flex System Fabric EN4093 10Gb Scalable Switch ► 10x external 10-Gb uplinks ► 14x internal 10-Gb ports	14	10	0
95Y3309	A3J6/ESW7	IBM Flex System Fabric EN4093R 10Gb Scalable Switch ► 10x external 10-Gb uplinks ► 14x internal 10-Gb ports	14	10	0
49Y4798	A1EL / 3596	IBM Flex System Fabric EN4093 10Gb Scalable Switch (Upgrade 1) ► Adds 2x external 40-Gb uplinks ► Adds 14x internal 10-Gb ports	28	10	2
88Y6037	A1EM / 3597	IBM Flex System Fabric EN4093 10Gb Scalable Switch (Upgrade 2) (requires Upgrade 1): ► Adds 4x external 10-Gb uplinks ► Add 14x internal 10-Gb ports	42	14	2

a. The first feature code listed is for configurations ordered through System x sales channels (HVEC) using x-config. The second feature code is for configurations ordered through the IBM Power Systems channel (AAS) using e-config.

For more general information, see *IBM Flex System Fabric EN4093 and EN4093R 10Gb Scalable Switches*, TIPS0864.

For more technical information, see *IBM PureFlex System and IBM Flex System Products and Technology*, SG24-7984.

1.2.2 IBM Flex System EN4091 10Gb Ethernet Pass-thru Module

The EN4091 10Gb Ethernet Pass-thru Module offers a one-for-one connection between a single node bay and an I/O module uplink. It has no management interface, and can support both 1 Gb and 10 Gb dual-port adapters installed in the compute nodes. If quad-port adapters are installed in the compute nodes, only the first two ports have access to the pass-through module's ports.

The necessary 1 GbE or 10 GbE module (SFP, SFP+, or DAC) must also be installed in the external ports of the pass-through. This configuration supports the speed (1 Gb or 10 Gb) and medium (fiber optic or copper) for adapter ports on the compute nodes.

The IBM Flex System EN4091 10Gb Ethernet Pass-thru is shown in Figure 1-4.



Figure 1-4 The IBM Flex System EN4091 10Gb Ethernet Pass-thru

The part number for the EN4091 10Gb Ethernet Pass-thru Module is listed in Table 1-2. There are no upgrades available for this I/O module at the time of this writing.

Table 1-2 IBM Flex System EN4091 10Gb Ethernet Pass-thru part number and feature codes

Part number	Feature code ^a	Product name
88Y6043	A1QV/3700	IBM Flex System EN4091 10Gb Ethernet Pass-thru

a. The first feature code listed is for configurations ordered through System x sales channels (HVEC) using x-config. The second feature code is for configurations ordered through the IBM Power Systems channel (AAS) using e-config.

For more general information, see *IBM Flex System EN4091 10Gb Ethernet Pass-thru Module*, TIPS0865.

For more technical information, see *IBM PureFlex System and IBM Flex System Products and Technology*, SG24-7984.

1.2.3 IBM Flex System EN2092 1Gb Ethernet Scalable Switch

The EN2092 1Gb Ethernet Switch provides support for L2/L3 switching and routing. The switch has the following features:

- ▶ Up to 28 internal 1-Gb ports
- ▶ Up to 20 external 1-Gb ports (RJ45 connectors)
- ▶ Up to four external 10-Gb uplink ports (SFP+ connectors)

Figure 1-5 shows a view of the faceplate for the EN2092 1Gb Ethernet Switch.



Figure 1-5 The EN2092 1Gb Ethernet Switch

This switch is considered ideal for the following clients:

- ▶ Those still using 1 Gb as their networking infrastructure
- ▶ Are deploying virtualization and require multiple 1-Gb ports
- ▶ Want investment protection for 10-Gb uplinks
- ▶ Looking to reduce total cost of ownership (TCO) and improve performance, while maintaining high levels of availability and security
- ▶ Looking to avoid oversubscription (multiple internal ports attempting to pass through a lower quantity of external ports, leading to congestion or performance impact)

Ports that are enabled and available depend on the features activated on the I/O module. Table 1-3 describes the port configurations for the EN2092 1Gb Ethernet Switch.

Table 1-3 IBM Flex System EN2092 1Gb Ethernet Scalable Switch part numbers and port upgrades

Part number	Feature code ^a	Product description
49Y4294	A0TF/3598	IBM Flex System EN2092 1Gb Ethernet Scalable Switch <ul style="list-style-type: none"> ▶ 14 internal 1-Gb ports ▶ 10 external 1-Gb ports
90Y3562	A1QW/3594	IBM Flex System EN2092 1Gb Ethernet Scalable Switch (Upgrade 1) <ul style="list-style-type: none"> ▶ Adds 14 internal 1-Gb ports ▶ Adds 10 external 1-Gb ports
49Y4298	A1EN/3599	IBM Flex System EN2092 1Gb Ethernet Scalable Switch (10Gb Uplinks) <ul style="list-style-type: none"> ▶ Adds four external 10-Gb uplinks

a. The first feature code listed is for configurations ordered through System x sales channels (HVEC) using x-config. The second feature code is for configurations ordered through the IBM Power Systems channel (AAS) using e-config.

For more general information, see *IBM Flex System EN2092 1Gb Ethernet Scalable Switch*, TIPS0861.

For more technical information, see *IBM PureFlex System and IBM Flex System Products and Technology*, SG24-7984.

1.2.4 I/O modules and cables

The Ethernet I/O module support for interface modules and cables is shown in Table 1-4.

Table 1-4 Modules and cables supported in Ethernet I/O modules

Part number	Feature code (x-config / e-config)	Description	EN20921 GbE Switch	EN4093/ EN4093R 10 GbE Switch	EN4091 10 GbE Pass-thru
44W4408	4942/3382	10 GBase-SR SFP+ (MMFiber)	Yes	Yes	Yes
46C3447	5053/EB28	IBM SFP+ SR Transceiver (MMFiber)	Yes	Yes	Yes
90Y9412	A1PM/ECB9	IBM SFP+ LR Transceiver (SMFiber)	Yes	Yes	Yes
81Y1622	3269/EB2A	IBM SFP SX Transceiver	Yes	Yes	Yes
81Y1618	3268/EB29	IBM SFP RJ45 Transceiver (does not support 10/100 Mbps)	Yes	Yes	Yes
90Y9424	A1PN/ECB8	IBM SFP LX Transceiver	Yes	Yes	Yes
49Y7884	A1DR/EB27	IBM QSFP+ SR Transceiver 40 Gbase-SR	No	Yes	No
90Y9427	A1PH/ECB4	1m IBM Passive DAC SFP+ Cable	Yes	Yes	No
90Y9430	A1PJ/ECB5	3m IBM Passive DAC SFP+ Cable	Yes	Yes	No
90Y9433	A1PK/ECB6	5m IBM Passive DAC SFP+ Cable	Yes	Yes	No
49Y7886	A1DL/EB24	1m IBM Passive QSFP+ DAC Break Out	No	Yes	No
49Y7887	A1DM/EB25	3m IBM Passive QSFP+ DAC Break Out	No	Yes	No
49Y7888	A1DN/EB26	5m IBM Passive QSFP+ DAC Break Out	No	Yes	No
90Y3519	A1MM/EB2J	10m IBM MTP Fiber Optical Cable (requires transceiver 49Y7884)	No	Yes	No
90Y3521	A1MN/EC2K	30m IBM MTP Fiber Optical Cable (requires transceiver 49Y7884)	No	Yes	No
49Y7890	A1DP/EB2B	1m IBM Passive QSFP+ to QSFP+ DAC	No	Yes	No
49Y7891	A1DQ/EB2H	3m IBM Passive QSFP+ to QSFP+ DAC	No	Yes	No
95Y0323	A25A/None	1m IBM Active DAC SFP+ Cable	No	No	Yes
95Y0326	A25B/None	3m IBM Active DAC SFP+ Cable	No	No	Yes
95Y0329	A25C/None	5m IBM Active DAC SFP+ Cable	No	No	Yes
81Y8295	A18M/EN01	1m 10 GE Twinax Act Copper SFP+ DAC (active)	No	No	Yes
81Y8296	A18N/EN02	3m 10 GE Twinax Act Copper SFP+ DAC (active)	No	No	Yes
81Y8297	A18P/EN03	5m 10 GE Twinax Act Copper SFP+ DAC (active)	No	No	Yes

All Ethernet I/O modules are restricted to using the SFP/SFP+ modules listed in Table 1-4. However, OEM Direct Attached Cables can be used if they meet the multi-source agreement (MSA).

1.3 IBM Flex System Ethernet adapters

The IBM Flex System portfolio contains a number of Ethernet I/O adapters. The cards are a combination of 1-Gb and 10-Gb ports and advanced function support that includes converged networks and virtual network interface cards (NICs).

The following Ethernet I/O adapters are covered:

- ▶ “IBM Flex System CN4054 10Gb Virtual Fabric Adapter”
- ▶ “IBM Flex System EN4054 4-port 10Gb Ethernet Adapter”
- ▶ “IBM Flex System EN2024 4-port 1Gb Ethernet Adapter”
- ▶ “IBM Flex System EN4132 2-port 10Gb Ethernet Adapter”

1.3.1 IBM Flex System CN4054 10Gb Virtual Fabric Adapter

The IBM Flex System CN4054 10Gb Virtual Fabric Adapter is a 4-port 10-Gb converged network adapter (CNA) for Intel processor-based compute nodes that can scale up to 16 virtual ports and support Ethernet, iSCSI, and FCoE. The adapter supports up to eight virtual NIC (vNIC) devices, where each physical 10 GbE port can be divided into four virtual ports with flexible bandwidth allocation. The CN4054 Virtual Fabric Adapter Upgrade adds FCoE and iSCSI hardware initiator functions.

The CN4054 adapter is shown in Figure 1-6.

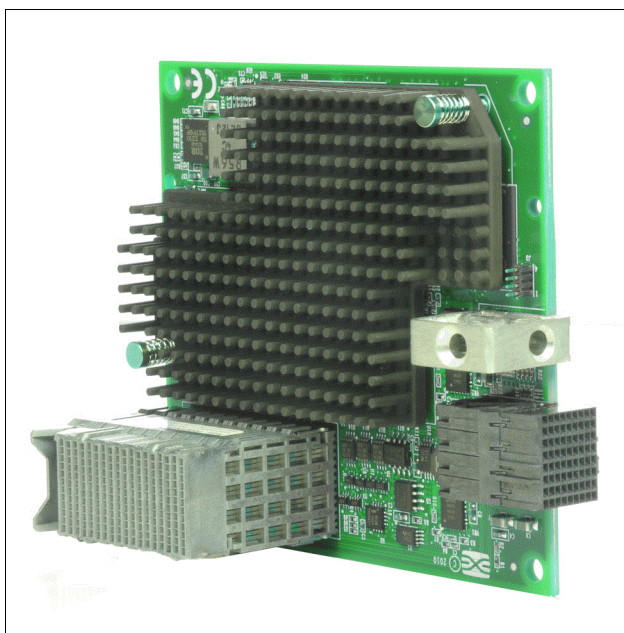


Figure 1-6 IBM Flex System CN4054 10Gb Virtual Fabric Adapter

Table 1-5 lists the ordering part numbers and feature codes.

Table 1-5 Ordering part numbers and feature codes

System x part number	System x feature code	Power feature code	Description
90Y3554	A1R1	None	IBM Flex System CN4054 10Gb Virtual Fabric Adapter

System x part number	System x feature code	Power feature code	Description
90Y3558	A1R0	None	IBM Flex System CN4054 Virtual Fabric Adapter Upgrade

For more general information, see *IBM Flex System CN4054 10Gb Virtual Fabric Adapter and EN4054 4-port 10Gb Ethernet Adapter*, TIPS0868.

For more technical information, see *IBM PureFlex System and IBM Flex System Products and Technology*, SG24-7984.

1.3.2 IBM Flex System EN4054 4-port 10Gb Ethernet Adapter

The IBM Flex System EN4054 4-port 10Gb Ethernet Adapter from Emulex enables the installation of four 10-Gb ports of high-speed Ethernet into an IBM Power Systems compute node. These ports interface to chassis switches or pass-through modules, enabling connections within and external to the IBM Flex System Enterprise Chassis.

The firmware for this 4-port adapter is provided by Emulex; the IBM AIX® driver and AIX tool support are provided by IBM.

Figure 1-7 shows the IBM Flex System EN4054 4-port 10Gb Ethernet Adapter.

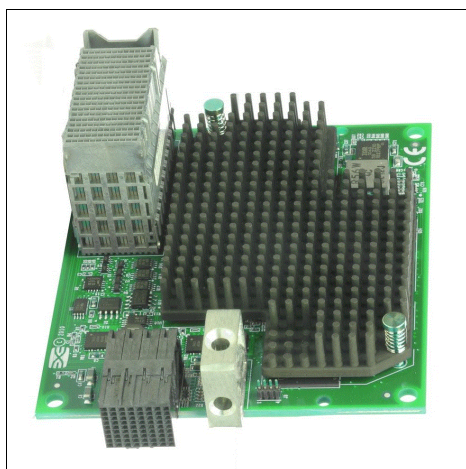


Figure 1-7 IBM Flex System EN4054 4-port 10Gb Ethernet Adapter

Table 1-6 lists the ordering part number and feature code.

Table 1-6 Ordering part number and feature code

Part number	System x feature code	Power feature code	Description
None	None	1762	EN4054 4-port 10Gb Ethernet Adapter

For more general information, see *IBM Flex System CN4054 10Gb Virtual Fabric Adapter and EN4054 4-port 10Gb Ethernet Adapter*, TIPS0868.

For more technical information, see *IBM PureFlex System and IBM Flex System Products and Technology*, SG24-7984.

1.3.3 IBM Flex System EN2024 4-port 1Gb Ethernet Adapter

The IBM Flex System EN2024 4-port 1Gb Ethernet Adapter is a quad-port Gigabit Ethernet network adapter. When it is combined with the IBM Flex System EN2092 1Gb Ethernet Switch, clients can use an end-to-end 1-Gb solution on the IBM Flex System Enterprise Chassis. The EN2024 adapter is based on the Broadcom 5718 controller and offers a Peripheral Component Interconnect Express (PCIe) 2.0 x1 host interface with MSI/MSI-X. It also supports I/O virtualization features, such as VMware NetQueue and Microsoft VMQ technologies.

The EN2024 adapter is shown in Figure 1-8.

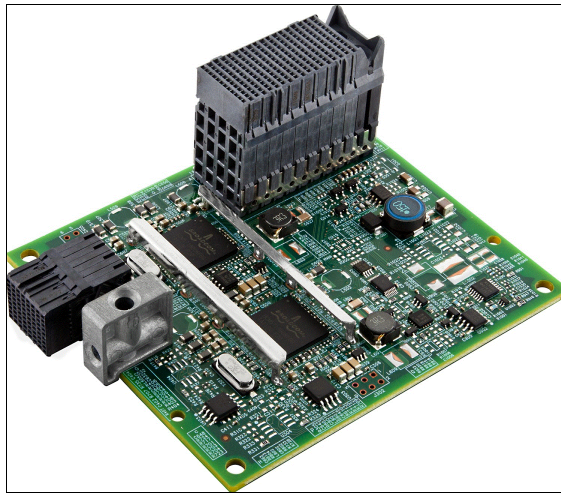


Figure 1-8 IBM Flex System EN2024 4-port 1Gb Ethernet Adapter

Table 1-7 lists the ordering part number and feature code.

Table 1-7 Ordering part number and feature code

Part number	System x feature code	Power feature code	Description
49Y7900	A1BR	1763	EN2024 4-port 1Gb Ethernet Adapter

For more general information, see *IBM Flex System EN2024 4-port 1Gb Ethernet Adapter*, TIPS0845.

For more technical information, see *IBM PureFlex System and IBM Flex System Products and Technology*, SG24-7984.

1.3.4 IBM Flex System EN4132 2-port 10Gb Ethernet Adapter

The IBM Flex System EN4132 2-port 10Gb Ethernet Adapter provides the highest-performing and most flexible interconnect solution for servers used in enterprise data centers, high-performance computing, and embedded environments.

The IBM Flex System EN4132 2-port 10Gb Ethernet Adapter is shown in Figure 1-9 on page 12.

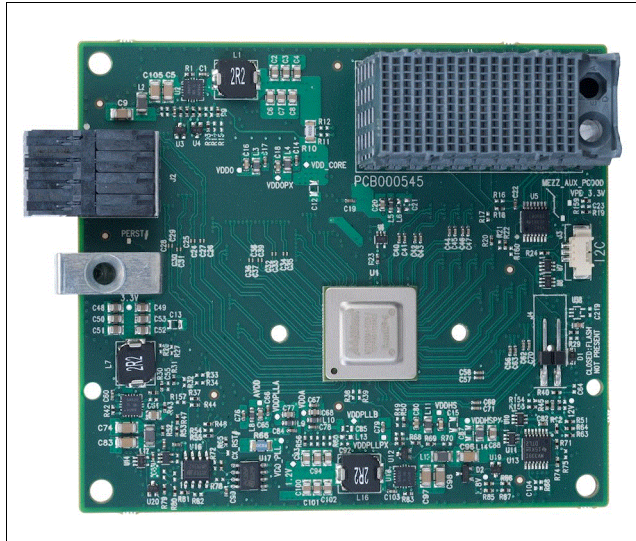


Figure 1-9 The EN4132 2-port 10Gb Ethernet Adapter for IBM Flex System

Table 1-8 lists the ordering part number and feature code.

Table 1-8 Ordering part number and feature code

Part number	System x feature code	Power feature code	Description
90Y3466	A1QY	None	EN4132 2-port 10Gb Ethernet Adapter

For more general information, see *IBM Flex System EN4132 2-port 10Gb Ethernet Adapter*, TIPS0873.

For more technical information, see *IBM PureFlex System and IBM Flex System Products and Technology*, SG24-7984.

1.4 IBM Networking Operating System

The IBM Networking Operating System (IBM Networking OS, or Network OS for short) is a data center-class network operating system progressively developed over the past 10 years to deliver highly reliable, high-performance Ethernet switching and interoperability with existing network infrastructures. IBM Networking OS delivers advanced capabilities for IBM RackSwitch™ and embedded PureSystems and BladeCenter switches through its robust feature set, stable implementation of industry standards, and IBM innovations such as VMready® and Virtual Fabric. IBM Networking OS supports the latest advances in cloud networking, flat networks, converged data and storage networking, virtualization awareness, and software-defined networking.

The most sensible and cost-effective approach to building a best-of-breed data center network is to base it upon open industry standards that are supported by multiple vendors. To that end, IBM has commissioned multiple independent studies by the Tolly Group to evaluate the IBM System Networking portfolio for interoperability between vendor products.

Table 1-9 on page 13 lists some of the most recent Tolly reports commissioned by IBM, and previously Blade Network Technologies® (BNT®), acquired by IBM in 2010.

Table 1-9 Tolly Group reports

IBM products evaluated	Publish date	Report title	Link to report
Flex System Fabric EN4093 Flex System EN2092 RackSwitch G8052 RackSwitch G8264	October 2012	Functionality and Interoperability Certification with Cisco Nexus 5548UP, 5596UP, C7009, and Catalyst 6509-E	http://tolly.com/DocDetail.aspx?DocNumber=212134
RackSwitch G8264	March 2011	IBM RackSwitch G8264: Competitive Performance Evaluation versus Cisco Systems, Inc. Nexus 5548P, Arista Networks 7148SX, and Juniper Networks EX4500	http://tolly.com/DocDetail.aspx?DocNumber=211108
RackSwitch G8124 Virtual Fabric 10G Switch Module	September 2010	IBM RackSwitch G8000, RackSwitch G8124 and Virtual Fabric 10G Switch Module: Functionality Certification and Cooperative Interoperability Evaluation with Cisco Nexus 5010	http://tolly.com/DocDetail.aspx?DocNumber=210140

IBM Networking OS provides network administrators a graphical user interface (GUI) and ISCLI delivering easy management.

1.4.1 Command-line interface

Network administrators can access and administer the switch through the command-line interface (CLI), which provides a simple, direct method for switch administration. Two distinct configuration modes exist for the CLI on IBM Networking OS:

- ▶ IBMNOS CLI Mode: This mode presents you with an organized hierarchy of menus, each with logically related submenus and commands.
- ▶ ISCLI Mode: This mode presents you with a more direct, command-driven means of administering the switch, emulating other industry standard interfaces for configuring networking equipment.

1.4.2 Browser-Based Interface

The network administrator can also access switch configuration and monitoring functions through the Browser-Based Interface (BBI), a web-based switch-management interface. The BBI has the following features:

- ▶ Many of the same configuration and monitoring functions as the CLI.
- ▶ An intuitive and easy-to-use interface structure.
- ▶ Nothing to install; the BBI is part of the IBM Network OS switch software.
- ▶ Automatically upgraded with each new software release.



Data center design and architecture

In this chapter, we describe specific topics that can lead the network and data center thought leaders to make optimal decisions regarding a healthy, secure, and highly available data center architecture.

We also describe the Open Data Center Interoperable Network (ODIN) initiative and IBM VMready that are new concepts that support and extend current industry capabilities, especially if you consider the virtualized driven-development direction that most companies are heading in.

Fibre Channel over Ethernet (FCoE) and Virtual Fabric are also described.

We also present the way that the IBM Flex System platform is designed to work with the data center networking infrastructure, and also its architecture. This introduces some new features and highlights relevant aspects of the hardware, related to industry concepts.

Use cases are presented to show how cutting-edge technology introduced by the IBM Flex System platform can interoperate with industry-standard procedures. These scenarios are designed to deliver high availability (HA), scalability, and performance with reasonable quality of service (QoS). In all the cases, we consider the usage of IPv6 as standard.

These are the subjects that are described in this chapter:

- ▶ Open Data Center Interoperable Network (ODIN)
- ▶ VMready
- ▶ IBM Flex System data center high availability components
 - The Ethernet switch I/O module
 - Virtual local area networks (VLANs)
 - Scalability and performance
 - Link aggregation
 - Layer 2 failover
 - Network interface card (NIC) teaming
- ▶ FCoE solution capabilities
- ▶ Virtual Fabric vNIC solution capabilities

- ▶ High availability use cases
 - Looped and blocking design
 - Non-looped, single upstream device design
 - Non-looped, multiple upstream devices design
- ▶ Practical Use Case 1: Fully redundant with Virtualized Chassis Technology (VSS/vPC/VLAG)
- ▶ Practical Use Case 2: Fully redundant with traditional Spanning Tree
- ▶ Practical Use Case 3: Fully redundant with Open Shortest Path First (OSPF)
- ▶ Summary and conclusions

2.1 Open Data Center Interoperable Network

Over the past several years, progressive data centers have undergone fundamental and profound architectural changes. Nowhere is this more apparent than in the data center network infrastructure.

An Open Data Center Interoperable Network (ODIN) is a flat, converged, virtualized data center network that is based on open industry standards, as shown in Figure 2-1.

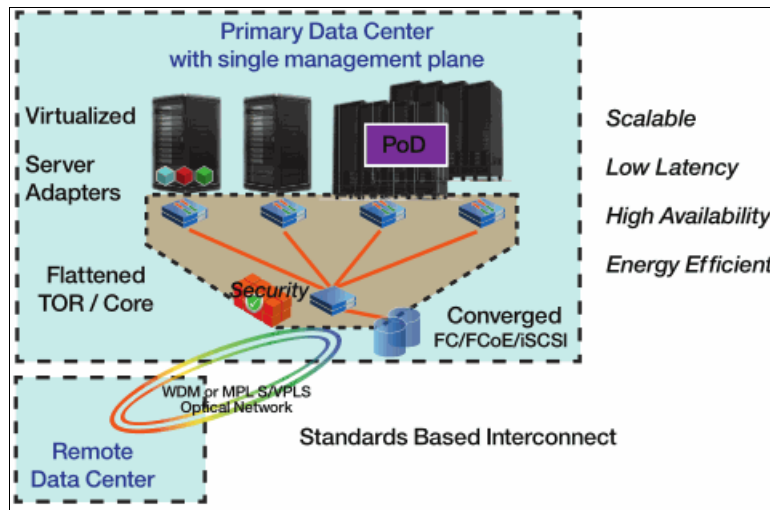


Figure 2-1 An overview of the ODIN ideal network

Instead of under-utilized devices, multi-tier networks, and complex management environments, the modern data center is characterized by highly utilized servers running multiple virtual machines (VMs); flattened, lower latency networks; and automated integrated management tools. New software-defined network approaches (including overlay networks and OpenFlow standards) greatly simplify the implementation of features such as dynamic workload provisioning, load balancing, and redundant paths for high availability and network reconfiguration. Furthermore, backbones with large bandwidth connecting remote virtualized data center resources extend the data center itself, making it capable of providing business continuity and backup and recovery of mission-critical data.

The practical, cost-effective evolution of data center networks should be based on open industry standards. This can be a challenging and often confusing proposition, because there are so many different emerging network architectures, both standard and proprietary. The ODIN materials created by IBM currently address issues such as virtualization and VM migration across flat Layer 2 networks, lossless Ethernet, Software-Defined Networking (SDN) and OpenFlow, and extended distance WAN connectivity (including low latency).

The benefits of ODIN include:

- ▶ Customer choice and designs that can be used in the future for data center networks, including vendor-neutral request for quotes (RFQs)
- ▶ Lower total cost of ownership (TCO) by enabling a multi-vendor network
- ▶ Avoiding confusion in the marketplace between proprietary and vendor-neutral solutions
- ▶ Providing best practices, relative maturity, and interpretation of networking standards

By taking advantage of ODIN, you can design a cost-effective and manageable data center that fully utilizes the potential of virtualization. It also gives you the flexibility to migrate to

federated data centers, in which computing, storage, and network resources can be treated as dynamically provisioned resource pools that can be rapidly partitioned into any wanted configuration.

Over time, emerging standards will help you improve dynamic resource provisioning, offer management flexibility, and deploy new features more rapidly. Similar technologies in software-defined networking will provide for an easily reconfigurable logical network that is aware of the requirements for workload mobility.

For more information about ODIN, see the following publications:

- ▶ *The IBM Networking Solutions - Open Data Center Interoperable Network (ODIN) website:*
<http://www-03.ibm.com/systems/networking/solutions/odin.html>
- ▶ *ODIN Volume 1: Transforming the Data Center Network:*
http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=STGE_QC_QC_USEN&htmlfid=QCW03019USEN&attachment=QCW03019USEN.PDF
- ▶ *ODIN Volume 2: ECMP Layer 3 Networks:*
http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=STGE_QC_QC_USEN&htmlfid=QCW03020USEN&attachment=QCW03020USEN.PDF
- ▶ *ODIN Volume 3: Software-Defined Networking and OpenFlow:*
http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=STGE_QC_QC_USEN&htmlfid=QCW03021USEN&attachment=QCW03021USEN.PDF
- ▶ *ODIN Volume 4: Lossless Ethernet:*
http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=STGE_QC_QC_USEN&htmlfid=QCW03022USEN&attachment=QCW03022USEN.PDF
- ▶ *ODIN Volume 5: WAN and Ultra Low Latency Applications:*
http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=STGE_QC_QC_USEN&htmlfid=QCW03023USEN&attachment=QCW03023USEN.PDF

2.2 VMready

VMready is a patented, unique solution that extends the concept of virtualization into the network. Network policies can be configured for virtual ports (v-ports), rather than just for physical ports. Each virtual machine can be assigned unique networking parameters such as security access control lists (ACLs), QoS, and VLANs.

VMready automatically synchronizes with VMware vCenter to create port groups. These port groups all have the same network configuration on all the required ESX vSwitches. This automatic configuration simplifies administrative tasks and reduces the chance of error due to misconfigurations. VMready also tracks the mobility of virtual machines across the data center and automatically reconfigures the network in real time as the virtual machines move. Figure 2-2 on page 19 presents the concept of usage of VMready.

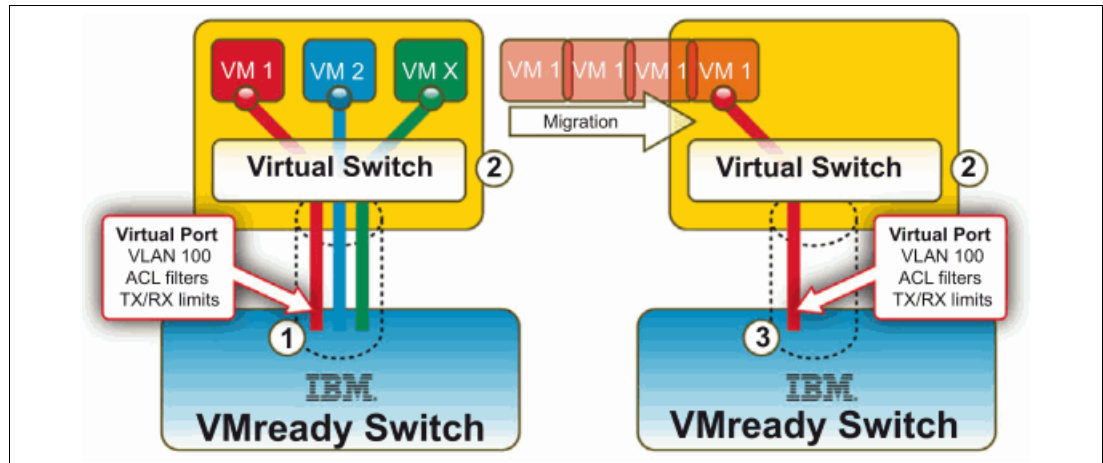


Figure 2-2 Example of VMready effects during the virtual machine live migration process

For server and network administrators, VMready greatly simplifies management by delivering consistent network policies that are enforced regardless of a virtual machine's physical location. VMready balances resources to the application workload using VMware VMotion, enabling truly dynamic data centers.

In addition, when you combine VMready with VMware vSphere, your company can deploy truly dynamic data centers where virtual machines can be automatically migrated to balance physical resources to application workloads.

These are some highlights of VMready:

- ▶ Supports open standards virtualization based on IEEE 802.1Qbg Edge Virtual Bridging
- ▶ Automatically discover and identify virtual machines in the data center
- ▶ Create groups of similar virtual machines with dedicated switch resources and isolated from other virtual machines
- ▶ Track all virtual machines during migrations
- ▶ Reconfigure network settings automatically
- ▶ Manage virtual machines in the network across the entire data center for any end device or edge switch that is compliant with the 802.1Qbg standard
- ▶ Enforce network policies on the virtual machine's resident inside the same hypervisor
- ▶ No proprietary tagging or changes to hypervisor software
- ▶ Works with all major hypervisors
- ▶ Scales across multiple physical switches and to any edge device supporting the 802.1Qbg standard
- ▶ Integration with VMware vCenter
- ▶ Track virtual machines during migrations with automatic reconfiguration of network settings
- ▶ VMready uses open standards and requires no proprietary tagging or changes to Hypervisor software
- ▶ VMready works with all major hypervisors including: VMware, Hyper-V, Xen, KVM, Oracle VM, and IBM PowerVM®

2.2.1 VMready benefits

VMready maximizes the advantages of virtualization while helping to eliminate the exposure to error that exists in traditional networking environments. It resolves virtual machine management issues and provides the simplicity, flexibility, and power needed for dynamic data centers. Administrators can configure the network parameters of virtual machines and track them as they migrate with an open-standards based solution. It can also help prevent security breaches and service outages that can be caused by improper network configuration.

VMready requires no additional server software or changes to hypervisors or virtual machines. This reduces complexity as it helps create energy-efficient, cost-effective data centers that allow enterprise applications to perform with the highest availability and performance. Because it supports VMware Enterprise and Advanced editions (VMware vSphere 4.x and ESX 3.x) without extra license fees, it helps reduce costs.

2.2.2 VMready compatibility

VMready works with the following products:

- ▶ VMware vSphere 4.x
- ▶ VMware ESX 3.x
- ▶ VMware vCenter
- ▶ IBM RackSwitch G8264
- ▶ IBM RackSwitch G8264T
- ▶ IBM RackSwitch G8124E
- ▶ IBM RackSwitch G8052
- ▶ IBM RackSwitch G8000
- ▶ IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch
- ▶ IBM Flex System EN2092 1 Gb Ethernet Scalable Switch
- ▶ IBM BladeCenter 1/10Gb Uplink Ethernet Switch Module

2.2.3 VMready references

For more information about VMready, see the following publications:

- ▶ *Implementing a VM-Aware Network Using VMready, SG24-7985:*
<http://www.redbooks.ibm.com/abstracts/sg247985.html>
- ▶ *The IBM Systems Networking VMready website:*
http://www-03.ibm.com/systems/networking/software/vmready/?csr=agus_sysnetwork-20111212&cm=k&cr=google&ct=USBRB301&S_TACT=USBRB301&ck=ibm_vm_ready&cmp=USBRB&mkwid=sRy5jKNXD_10896144046_432gdi13039
- ▶ *IBM VMready - Virtual machine-aware networking technical white paper:*
http://www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=SA&subtype=WH&htmlfid=QCW03005USEN&attachment=QCW03005USEN.PDF&appname=STGE_QC_QC_USEN_WH
- ▶ *IBM VMready specifications:*
<http://www-03.ibm.com/systems/networking/software/vmready/sysreq.html>

2.3 IBM Flex System data center high availability components

After physical redundancy requirements are met, it is necessary to consider logical elements to use this physical redundancy. The following logical features aid in high availability:

- ▶ NIC teaming/bonding on the server or compute node
- ▶ Layer 2 (L2) failover (also known as *trunk failover*) on the I/O modules
- ▶ Rapid Spanning Tree Protocol for looped environments
- ▶ Virtual link aggregation on upstream devices connected to the I/O modules
- ▶ Virtual Router Redundancy Protocol for redundant upstream default gateway
- ▶ Routing Protocols (such as Routing Information Protocol (RIP) or OSPF) on the I/O modules, if L2 adjacency is not a requirement

The IBM Flex System family presents a series of features to provide easy cooperation with industry spread network equipment.

In this section, we present the features from the Flex System and PureFlex point of view.

2.3.1 The Ethernet switch I/O module

Selecting the Ethernet I/O module that is best for an environment is a process specific to each client. The Enterprise Chassis offers a range of Ethernet connectivity options, but deciding which one to use can be made easier by following the selection process below. The following factors should be considered during the selection process:

- ▶ If your bandwidth requirements are such that any servers within the Enterprise Chassis require a 10 Gb connection, the EN4093/EN4093R 10Gb Scalable Switch is the best choice.
- ▶ If your upstream bandwidth requirements are such that 1 Gb (or even an aggregation of 1 Gb) links are insufficient, the EN4093/EN4093R 10Gb Scalable Switch module is the best choice.
- ▶ If you are installing the Enterprise Chassis into an environment where the upstream network is using 10 Gb (or there is a plan to upgrade eventually to 10 Gb), the EN4093/EN4093R 10Gb Scalable Switch module is the best choice.
- ▶ If you require the maximum bandwidth available for the nodes, the EN4093/EN4093R 10Gb Scalable Switch module is the best choice.
- ▶ If you need to take advantage of the advanced virtualization features, such as vNIC, that are offered in the Enterprise Chassis, the EN4093/EN4093R 10Gb Scalable Switch is the best choice.
- ▶ If there is no immediate need for 10 Gb, there are no plans to upgrade to 10 Gb in the foreseeable future, and you have no need for any of the advanced features offered in the 10 Gb solution, the EN2092 1Gb Ethernet Switch is the optimal solution.
- ▶ If you need a solution that is transparent to the network, has only one link for each compute node for each I/O module, and requires direct connections from the compute node to the external Top-of-Rack (ToR) switch, the EN4091 10Gb Ethernet Pass-thru is an option.

There are more criteria involved because each environment has its own unique attributes. However, the criteria reviewed in this section are a good starting point in the decision-making process.

The Ethernet I/O module selection criteria are summarized in Table 2-1.

Table 2-1 Switch module and adapter selection criteria

Suitable switch module and adapter	Switches		Adapters		
	EN2092 1Gb Ethernet Switch	EN4093/ EN4093 R 10Gb Scalable Switch	CN4054 10Gb Virtual Fabric Adapter	EN2024 4-port 1Gb Ethernet Adapter	EN4132 2-port 10Gb Ethernet Adapter
Requirement					
Gigabit Ethernet to nodes	Yes	Yes	Yes	Yes	No
10 Gb Ethernet to nodes	No	Yes	Yes	Yes	Yes
10 Gb Ethernet uplinks	Yes	Yes	Not applicable		
40 Gb Ethernet uplinks	No	Yes			
Basic Layer 2 switching (VLAN, port aggregation)	Yes	Yes			
Advanced Layer 2 switching: IEEE features (STP, QoS)	Yes	Yes			
Layer 3 IPv4 switching (forwarding, routing, ACL filtering)	Yes	Yes			
Layer 3 IPv6 switching (forwarding, routing, ACL filtering)	Yes	Yes			
10 Gb Ethernet CEE	No	Yes	Yes	No	No
FCoE	No	Yes ^a	Yes	No	No
Switch stacking	No	Yes ^a	Not applicable		
vNIC support	No	Yes	Yes	No	No
VMready	Yes	Yes	Not applicable		

a. Support for FCoE and trunk failover switch stacking is planned in future firmware releases.

2.3.2 Virtual local area networks

Virtual local area networks (VLANs) are commonly used in a Layer 2 network to split groups of networked systems into manageable broadcast domains, create logical segmentation of workgroups, and enforce security policies among logical segments. Primary VLAN considerations include the number and types of supported VLANs and VLAN tagging protocols.

All Ethernet I/O switch modules in the Enterprise Chassis support the following VLAN-related features:

- ▶ VLANs available in the range of 1 - 4094
 - Some VLANs might be reserved when certain features are enabled
- ▶ 1024 total VLANs active simultaneously of the 1 - 4094 range
- ▶ IEEE 802.1Q for VLAN tagging on links (also called *trunking* by some vendors)
 - Support for tagged or untagged native VLAN
- ▶ Port-based VLANs

- ▶ Protocol-based VLANs
- ▶ Spanning Tree Per VLAN (Per VLAN Rapid Spanning Tree)
- ▶ 802.1x Guest VLANs
- ▶ VLAN Maps for ACLs
- ▶ VLAN-based port mirroring

Specific to 802.1Q VLAN tagging, this feature is critical to maintain VLAN separation when packets in multiple VLANs must traverse a common link between devices. Without a tagging protocol, such as 802.1Q, maintaining VLAN separation between devices can be accomplished through a separate link for each VLAN, a less than optimal solution.

Important: In rare cases, there are some older nonstandards-based tagging protocols used by vendors. These protocols are not compatible with 802.1Q.

The need for 802.1Q VLAN tagging is not relegated only to networking devices. It is also supported and frequently used on end nodes, and is implemented differently by various operating systems (OSs). For example, in Windows based systems, a vendor driver is needed to subdivide the physical interface into logical NICs, with each logical NIC set for a specific VLAN. Typically, this setup is part of the teaming software from the NIC vendor.

For Linux, tagging is done by creating subinterfaces of a physical or logical NIC, such as `eth0.10` for VLAN 10.

For VMware ESX, tagging can be done within the vSwitch through port group tag settings (known as *Virtual Switch Tagging*). Tagging also can be done in the OS within the guest VM itself (called *Virtual Guest Tagging*).

From an OS perspective, having several logical interfaces can be useful when an application requires more than two separate interfaces and you do not want to dedicate an entire physical interface. It might also help to implement strict security policies for separating network traffic that uses VLANs and having access to server resources from different VLANs, without adding additional physical network adapters.

Review the documentation of the application to ensure that the application deployed on the system supports the use of logical interfaces often associated with VLAN tagging.

2.3.3 Scalability and performance

Each Enterprise Chassis has four I/O bays and, depending on the Ethernet switch module installed in the I/O bay, the license installed on the Ethernet switch, and the adapters installed on the node. Each bay can support many connections, both toward the nodes and up toward the external network.

The I/O switch modules available for the Enterprise Chassis are a scalable class of switch. This means that additional banks of ports (also known as *partitions* in this context) can be enabled as needed, thus scaling the switch to meet a particular requirement.

The architecture allows up to four switch partitions in each I/O module, for a total of up to 16 partitions within each chassis. The number of ports in these partitions available for use by a node depends on the following factors:

- ▶ The I/O module installed
- ▶ The partitions activated on the I/O module
- ▶ The I/O adapters installed in the nodes

The Ethernet I/O switch modules include an enabled base partition, and require upgrades to enable the extra partitions. Not all Ethernet I/O modules support the same number of partitions. A cross-reference of the number of partitions supported on each of the available I/O modules is shown in Table 2-2. The pass-through is a fixed function device, and as such has no real concept of port expansion.

Table 2-2 Module names and the number of switch partitions

Module name	Number of partitions supported
EN2092 1Gb Ethernet Switch	2
EN4093/EN4093R 10Gb Scalable Switch	3
EN4091 10Gb Ethernet Pass-thru	1

As shipped, all I/O modules have support for partition 1 (base), which includes 14 internal ports, one to each of the compute node bays up front, and some number of uplinks. Upgrades to the scalable switches to enable other partitions and uplinks are added as part of the Feature on Demand capability (FoD). Because of these upgrades, it is possible to increase ports without hardware changes. As each FoD is enabled, the ports controlled by the upgrade are activated. If the compute node has a suitable I/O adapter, the ports are available to the node.

The act of enabling a bank of ports by applying the FoD merely enables more ports for the switch to use. There is no logical or physical separation of these ports from a networking perspective, only from a licensing perspective. Adding these FoDs increases the size of the switch. The term *partition* in this context is only about increasing the number of ports available for use.

As an example of how this licensing works, the EN4093/EN4093R 10Gb Scalable Switch, by default, includes 14 internal available ports, together with 10 uplink enhanced small form-factor pluggable (SFP+) ports. Additional partitions can be enabled with an FoD upgrade, thus providing a second or third set of 14 internal ports and some number of uplinks, as shown in Figure 2-3 on page 25.

Internal-facing ports for partition 3 are currently not usable by any of the available I/O adapters in the nodes. The only reason to enable the second upgrade at this time is to obtain the extra four SFP+ uplinks included as part of the process to enable this partition.

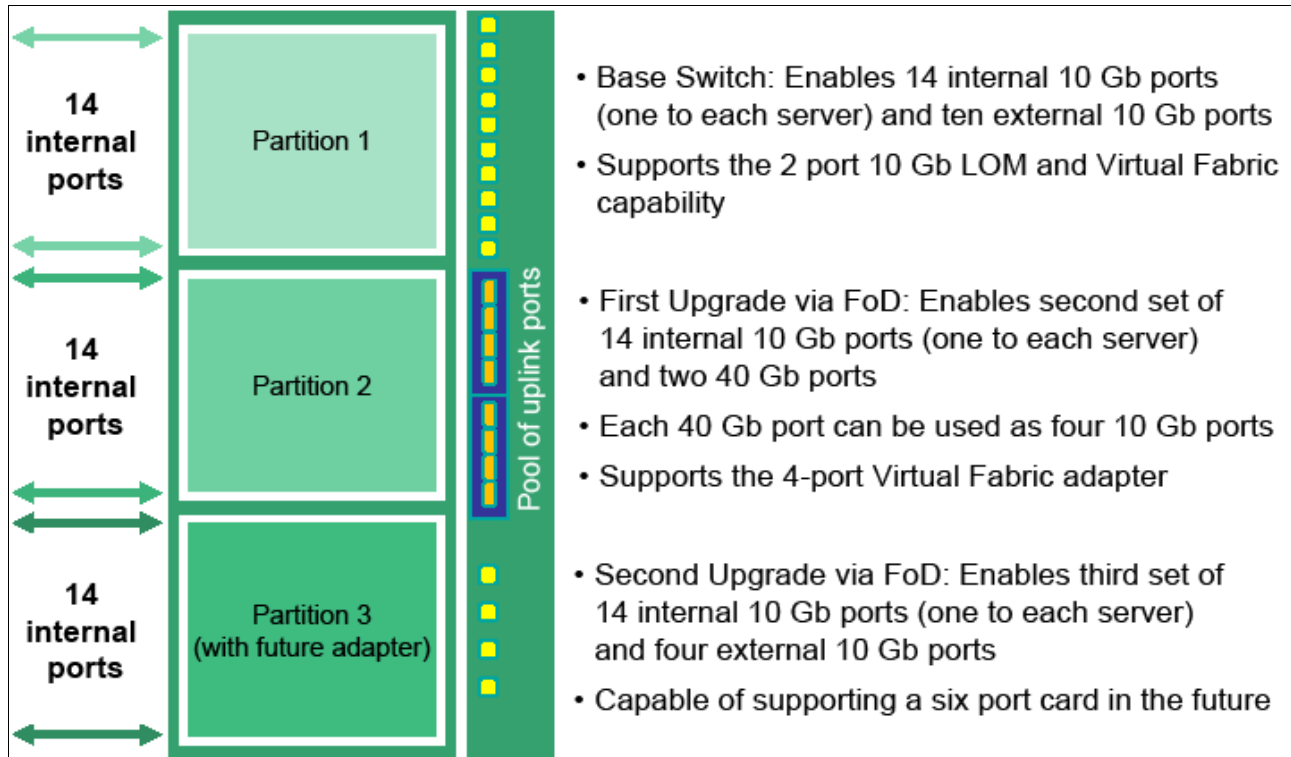


Figure 2-3 Partition layout for EN4093/EN4093R 10Gb Scalable Switch

The ability to add ports and bandwidth as needed is a critical element of a scalable platform.

2.3.4 Link aggregation

Sometimes referred to as *trunking*, *port channel*, or *Etherchannel*, link aggregation involves taking multiple physical links and binding them into a single common link for use between two devices. The primary purposes of aggregation are to improve high availability and increase bandwidth.

Bundling the links

Although there are several different kinds of aggregation, the two most common and that are supported by the Enterprise Chassis I/O modules are static and Link Aggregation Control Protocol (LACP).

Important: In rare cases, there are still some older non-standards-based aggregation protocols, such as Port Aggregation Protocol (PAgP), in use by some vendors. These protocols are not compatible with either static or LACP aggregations.

Static aggregation does not use any protocol to create the aggregation. Instead, static aggregation simply combines the ports based on the aggregation configuration applied on the ports and assumes that the other side of the connection does the same.

Important: In some cases, static aggregation is referred to as *static LACP*. This term is a contradictory term because it is difficult in this context to be both static and have a control protocol.

LACP is an IEEE standard that was defined in 802.3ad. The standard was later included in the mainline 802.3 standard but then was pulled out into the current standard 802.1AX-2008. LACP is a dynamic way of determining whether both sides of the link might be aggregating.

The decision to use static LACP is usually a question of what you use in your network. If there is no preference, we provide several advantages and disadvantages of each option to aid in the decision-making process.

Static aggregation is the quickest and easiest way to build an aggregated link. This method also is the most stable in high-bandwidth utilization environments, particularly if pause frames are being exchanged.

Using static aggregation can be advantageous in mixed vendor environments because it can help prevent possible interoperability issues. Because settings in the LACP standard do not have a suggested default, vendors are allowed to use different defaults, which can lead to unexpected interoperation. For example, the LACP Data Unit (LACPDU) timers can be set to be exchanged every 1 second or every 30 seconds. If one side is set to 1 second and one side is set to 30 seconds, the LACP aggregation can be unstable.

Important: Most vendors default to using the 30-second exchange of LACPDUs, including IBM switches. If you encounter a vendor that defaults to 1-second timers, we advise that the other vendor changes to operate with 30-second timers, rather than setting both to 1 second. This setting tends to produce a more robust aggregation over the 1-second timers.

One of the downsides to static aggregation is that it lacks a mechanism to detect whether the other side is correctly configured for aggregation. So, if one side is static and the other side is not configured, configured incorrectly, or is not connected to the correct ports, it is possible to cause a network outage by bringing up the links.

Based on the information presented in this section, if you are sure that your links are connected to the correct ports and that both sides are configured correctly for static aggregation, then static aggregation is a solid choice.

LACP has the inherent safety that a protocol brings to this process. At linkup, LACPDUs are exchanged and both sides must agree, they are using LACP before it attempts to bundle the links. Therefore, in the case of misconfiguration or incorrect connections, LACP helps protect the network from an unplanned outage. The disadvantages of using LACP are that it takes a small amount of time to negotiate the aggregation and form an aggregating link (usually under a second), and it can become unstable and unexpectedly fail in environments with heavy and continued pause frame activity.

Another factor to consider about aggregation is whether it is better to aggregate multiple low-speed links into a high-speed aggregation, or use a single high-speed link with a similar speed to all of the links in the aggregation.

If your primary goal is high availability, aggregations can offer a no-single-point-of-failure situation that a single high-speed link cannot offer.

If maximum performance and lowest possible latency are the primary goals, often a single high-speed link makes more sense. Another factor is cost. Often, one high-speed link can cost more to implement than a link that consists of an aggregation of multiple slower links.

Virtual link aggregations

Aside from the standard point-to-point aggregations covered in this section, there is a technology that provides multi-chassis aggregation, sometimes called *distributed aggregation* or *virtual link aggregation*.

Under the latest IEEE specifications, an aggregation is still defined as a bundle between only two devices. By this definition, you cannot create an aggregation on one device and have the links of that aggregation connect to more than a single device on the other side of the aggregation. Using only two devices limits the ability to offer certain robust designs.

Although the standards bodies are working on a solution that provides split aggregations across devices, most vendors devised their own version of multichassis aggregation. For example, Cisco has virtual Port Channel (vPC) on Nexus products, and Virtual Switch System (VSS) on the 6500 line. IBM offers virtual Link Aggregation Groups (vLAGs) on many of their ToR solutions, and on the EN4093/EN4093R 10Gb Scalable Switch. The primary goals of vLAG are to overcome the limits imposed by standard aggregation, and provide a distributed aggregation across a pair of switches instead of a single switch.

The decisions whether to aggregate and which method of aggregation is most suitable to a specific environment are not always straightforward. But if the decision is made to aggregate, the I/O modules for the Enterprise Chassis offer the necessary wanted features to integrate into the aggregated infrastructure.

2.3.5 Layer 2 failover

Layer 2 failover, also known as *trunk failover* or *link state tracking*, is an important feature for ensuring high availability in chassis-based computing. This feature is used with NIC teaming to ensure that the compute nodes can detect an uplink failure from the I/O modules.

With traditional NIC teaming and bonding, the decision process used by the teaming software to use a NIC is based on whether the link to the NIC is up or down. In a chassis-based environment, the link between the NIC and the internal I/O module rarely goes down unexpectedly. Instead, a more common occurrence might be the uplinks from the I/O module go down; for example, an upstream switch failed or cables were disconnected. In this situation, although the I/O module no longer has a path to send packets because of the upstream fault, the actual link to the internal server NIC is still up. The server might continue to send traffic to this unusable I/O module, leading to a black hole condition.

To prevent this black hole condition and to ensure continued connection to the upstream network, Layer 2 failover can be configured on the I/O modules. Depending on the configuration, Layer 2 failover monitors a set of uplinks. If these uplinks go down, Layer 2 failover takes down the configured server-facing links. This action alerts the server that this path is not available, and NIC teaming can take over and redirect traffic to the other NIC.

This feature is shown in detail in Figure 2-4 on page 28.

Layer 2 failover offers these features:

- ▶ Besides triggering on link up/down, Layer 2 failover also operates on the spanning-tree blocking state. From a data packet perspective, a blocked link is no better than a down link.
- ▶ Layer 2 failover can be configured to fail over if the number of links in a monitored aggregation falls below a certain number.
- ▶ Layer 2 failover can be configured to trigger on VLAN failure.

- ▶ When a monitored uplink comes back up, Layer 2 failover automatically brings back up the downstream links if it is not blocked and other attributes, such as the minimum number of links, are met for the trigger.
- ▶ For Layer 2 failover to work properly, it is assumed that there is an L2 path between the uplinks, external to the chassis. This path is most commonly found at the switches just above the chassis level in the design (but they can be higher) if there is an external L2 path between the Enterprise Chassis I/O modules.

Important: Other solutions to detect an indirect path failure were created, such as the VMware beacon probing. Although these solutions offer advantages, Layer 2 failover is the simplest and most non-intrusive way to provide this functionality. We encourage you to use Layer 2 failover whenever NIC teaming is configured on the compute nodes.

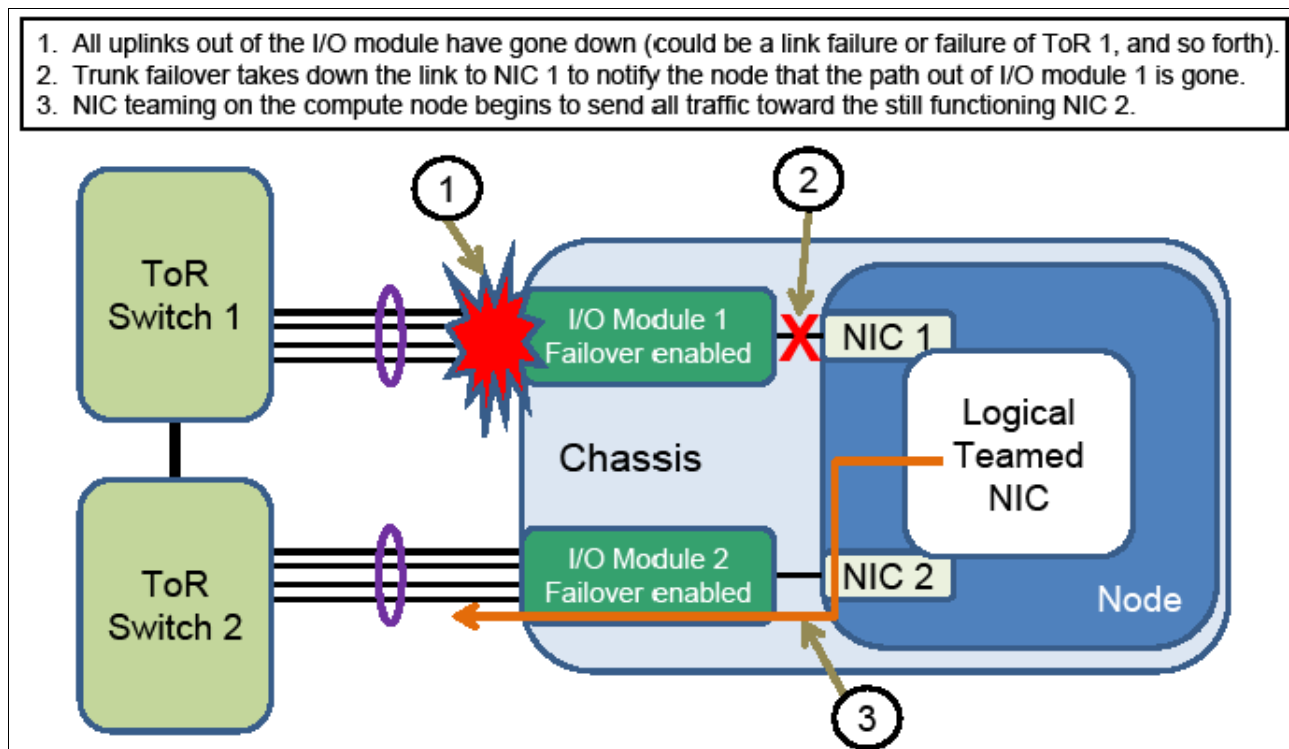


Figure 2-4 Layer 2 failover in action

Using Layer 2 failover with NIC teaming is a critical element for nodes requiring a highly available path from the Enterprise Chassis.

2.3.6 NIC teaming

NIC teaming, also known as *bonding*, is a solution used on servers to logically bond two or more NICs to form one or more logical NICs for purposes of high availability, increased performance, or both.

There are many forms of NIC teaming, and the type available for a server is often tied to the OS installed on the server.

For Microsoft Windows, the teaming software traditionally was provided by the NIC vendor and is installed as an add-on to the operating system. This software often also includes the

elements necessary to enable VLAN tagging on the logical NICs created by the teaming software. These logical NICs are seen by the OS as physical NICs and are treated as such when configuring them. Depending on the NIC vendor, the teaming software might offer several different types of failover, including simple active/standby, static aggregation, dynamic aggregation (LACP), and vendor-specific load balancing schemes.

For Linux-based systems, the bonding module is used to implement NIC teaming. There are a number of bonding modes available, most commonly mode 1 (active/standby) and mode 4 (LACP aggregation). Like Windows teaming, Linux bonding also offers logical interfaces to the OS that can be used as wanted. Unlike Windows teaming, VLAN tagging is controlled by different software, and can create sub interfaces for VLANs from both physical and logical entities, for example, `eth0.10` for VLAN 10 on physical `eth0`, or `bond0:20` for VLAN 20 on a logical NIC bond pair 0.

Like Linux, VMware ESX also has built-in teaming in the form of assigning multiple NICs to a common vSwitch (a logical switch that runs within an ESX host, which is shared by the VMs that require network access). VMware has several teaming modes, with the route-based default on the originating virtual port ID. This default mode provides a per VM load balance of physical NICs assigned to the vSwitch.

The teaming method that is best for a specific environment is unique to each situation. However, these common elements might help in the decision-making process:

- ▶ Do not select a mode that requires some form of aggregation (static or LACP) on the switch side unless the NICs in the team go to the same physical switch or logical switch created by a technology, such as vLAG or stacking.
- ▶ If using a mode that uses some form of aggregation, you must also perform proper configuration on the upstream switches to complete the aggregation on that side.
- ▶ The most stable solution is often active/standby, but this solution has the disadvantage of losing any bandwidth on a NIC that is in standby mode.
- ▶ Most teaming software offers proprietary forms of load balancing. The selection of these modes must be thoroughly tested for suitability to the task for an environment.
- ▶ Most teaming software incorporates the concept of *auto fallback*, which means that if a NIC went down and then came back up, it automatically fails back to the original NIC. Although this function helps ensure good load balancing, each time that a NIC fails, some small packet loss might occur, which can lead to unexpected instabilities. A flapping link occurs when a severe disruption to the network connection of the servers causes the link to flap back and forth. One way to mitigate this circumstance is to disable the auto fallback feature. After a NIC fails, the traffic falls back only in the event the original link is restored and something happened to the current link that requires a switchover.

It is your responsibility to understand your goals and the tools available to achieve those goals. NIC teaming is one tool for users that need high availability connections for their compute nodes.

2.4 Fibre Channel over Ethernet solution capabilities

One common way to reduce management points in an environment is by converging technologies that were implemented on separate infrastructures. Like collapsing office phone systems from a separate cabling plant and components into a common IP infrastructure, Fibre Channel networks also are experiencing this type of convergence. Like phones that move to Ethernet, Fibre Channel also is moving to Ethernet.

Fibre Channel over Ethernet (FCoE) removes the need for separate host bus adapters (HBAs) on the servers and separate Fibre Channel cables flowing out the back of the server or chassis. Instead, a Converged Network Adapter (CNA) is installed in the server. The CNA presents what appears to be both a NIC and an HBA to the operating system, but the output from the server is 10 Gb Ethernet.

When used with EN4091 10Gb Ethernet Pass-thru connected to the external FCoE-capable ToR switch, the CN4054 10Gb Virtual Fabric Adapter (VFA), or embedded VFA on x240 compute node with optional Virtual Fabric Upgrade offers this service. As of this writing, this solution allows the EN4091 10Gb Ethernet Pass-thru to connect a node that runs a CNA to an upstream switch that acts as an FCoE Forwarder (FCF). In this configuration, the Fibre Channel packet is extracted from the Ethernet packet and sent into the Fibre Channel SAN.

FCoE support on the EN4093/EN4093R 10Gb Scalable Switch is planned for future firmware releases.

2.4.1 FCoE references

- ▶ *An Introduction to Fibre Channel over Ethernet, and Fibre Channel over Convergence Enhanced Ethernet*, REDP-4493
- ▶ *Storage and Network Convergence Using FCoE and iSCSI*, SG24-7986

2.5 Virtual Fabric vNIC solution capabilities

Virtual Network Interface Controller (vNIC) is a way to divide a physical NIC into smaller logical NICs (or partition them) so that the OS has more ways to logically connect to the infrastructure. The vNIC feature is supported only on 10 Gb ports on the EN4093/EN4093R 10Gb Scalable Switch facing the compute nodes within the chassis, and requires a node adapter (CN4054 10Gb Virtual Fabric Adapter or embedded VFA) that also supports this functionality.

As of this writing, there are two primary forms of vNIC available: Virtual Fabric mode (or Switch dependent mode) and Switch independent mode. The Virtual Fabric mode also is subdivided into two submodes: dedicated uplink vNIC mode and shared uplink vNIC mode.

All vNIC modes share these common elements:

- ▶ They are supported only on 10 Gb connections.
- ▶ Each vNIC mode allows a NIC to be divided into up to 4 vNICs per physical NIC (can be less than 4, but not more).
- ▶ They all require an adapter that has support for one or more of the vNIC modes.
- ▶ When creating vNICs, the default bandwidth is 2.5 Gb for each vNIC, but can be configured to be anywhere from 100 Mb up to the full bandwidth of the NIC.
- ▶ The bandwidth of all configured vNICs on a physical NIC cannot exceed 10 Gb.

A summary of some of the differences and similarities of these modes is shown in Table 2-3. These differences and similarities are covered next.

Table 2-3 Attributes of vNIC modes

Capability	IBM Virtual Fabric mode		Switch independent mode
	Dedicated uplink	Shared uplink	
Requires support in the I/O module	Yes	Yes	No
Requires support in the NIC	Yes	Yes	Yes
Supports adapter transmit rate control	Yes	Yes	Yes
Support I/O module transmit rate control	Yes	Yes	No
Supports changing rate when needed	Yes	Yes	No
Requires a dedicated uplink per vNIC group	Yes	No	No
Support for node OS-based tagging	Yes	No	Yes
Support for failover per vNIC group	Yes	Yes	No
Support for more than one uplink per vNIC group	No	Yes	Yes

2.5.1 Virtual Fabric mode vNIC

Virtual Fabric mode or switch-dependent mode depends on the switch in the I/O switch module to participate in the vNIC process. Specifically, the I/O module that supports this mode in the Enterprise Chassis is the IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch. It requires an adapter on the node that also supports the vNIC switch-dependent mode feature.

In switch-dependent vNIC mode, configuration is performed on the switch itself and the configuration information is communicated between the switch and the adapter so that both sides agree on and enforce bandwidth controls. The mode can be changed to different speeds at any time without reloading either the OS or the I/O module.

Currently, there is only one type of switch-dependent vNIC mode: the dedicated uplink mode. That mode incorporates the concept of a vNIC group on the switch that is used to associate vNICs and physical ports into virtual switches within the chassis.

The dedicated uplink mode of switch-dependent vNIC modes has the following attributes:

- ▶ It conceptually is a vNIC group that must be created on the I/O module.
- ▶ Similar vNICs are bundled together into common vNIC groups.
- ▶ Each vNIC group is treated as a virtual switch within the I/O module. Packets in one vNIC group can get only to a different vNIC group by going to an external switch/router.
- ▶ For the purposes of Spanning Tree and packet flow, each vNIC group is treated as a unique switch by upstream connecting switches/routers.
- ▶ Both modes support the addition of physical NICs (pNIC) to vNIC groups, for internal communication to other pNICs and vNICs in that vNIC group, and then share any uplink associated with that vNIC group. pNICs are the NICs from nodes not using vNIC.

Dedicated uplink mode

Dedicated uplink mode is the default mode when vNIC is enabled on the I/O module. In dedicated uplink mode, each vNIC group must have its own dedicated physical or logical (aggregation) uplink. In this mode, no more than one physical or logical uplink to a vNIC group can be assigned and it is assumed that high availability is achieved by some combination of aggregation on the uplink or NIC teaming on the server.

In dedicated uplink mode, vNIC groups are VLAN agnostic to the nodes and the rest of the network, which means that you do not need to create VLANs for each VLAN used by the nodes. The vNIC group takes each packet (tagged or untagged) and moves it through the switch.

This mode is accomplished by the use of a form of Q-in-Q tagging. Each vNIC group is assigned some VLAN that is unique to each vNIC group. Any packet (tagged or untagged) that comes in on a downstream or upstream port in that vNIC group has a tag placed on it equal to the vNIC group VLAN. As that packet leaves the vNIC into the node or out an uplink, that tag is removed and the original tag (or no tag, depending on the original packet) is revealed.

2.5.2 Switch-independent mode vNIC

Switch-independent mode vNIC is completed only on the node itself, and the I/O module is unaware of this virtualization. The I/O module acts as a normal switch in all ways. This mode is enabled at the node directly, and has similar rules as dedicated vNIC mode regarding how you can divide the vNIC. But any bandwidth settings are limited to how the node sends traffic, not how the I/O module sends traffic back to the node. Also, the bandwidth settings cannot be changed in real time because they require a reload to change.

The mode that is best-suited for a user depends on the user's requirements. Virtual Fabric dedicated uplink mode offers the most control, and switch-independent mode offers the most flexibility with uplink connectivity.

2.5.3 Virtual Fabric references

For more information about VMready, see the following publications:

- ▶ Virtual Fabric Academy: <http://www.virtualfabric.net>
- ▶ IBM Networking Software - Virtual Fabric:
<http://www.ibm.com/systems/networking/software/virtualfabric.html>
- ▶ *IBM BladeCenter Virtual Fabric Solutions*, SG24-7966
- ▶ *IBM Flex System CN4054 10Gb Virtual Fabric Adapter and EN4054 4-port 10Gb Ethernet Adapter*, TIPS0868
- ▶ *IBM BladeCenter Virtual Fabric 10Gb Switch Module*, TIPS0708

2.6 High availability use cases

Customers often require continuous access to their network-based resources and applications. Providing high availability (HA) for client network resources can be a complex task that involves fitting multiple pieces together on a hardware and software level. Our focus is to provide high availability access to the network infrastructure.

Network infrastructure availability can be achieved by using various techniques and technologies. Most are widely used standards and can be deployed with everything from rack-mount servers to full iDataPlex racks, but some are specific to the IBM Flex System Enterprise Chassis. We review the most common technologies that can be implemented in an Enterprise Chassis environment to provide high availability to the network infrastructure.

A typical LAN infrastructure consists of server NICs, client NICs, and network devices, such as Ethernet switches and cables that connect them. Specific to the Enterprise Chassis, the potential failure areas for node network access include port failures (both on switches and the node adapters), the midplane, and the I/O modules.

The first step in achieving high availability is to provide physical redundancy of components connected to the infrastructure as a whole. Providing this redundancy typically means that the following measures are taken:

- ▶ Deploy node NICs in pairs
- ▶ Deploy top of rack switches or embedded switch modules in pairs
- ▶ Connect the pair of node NICs to separate I/O modules in the Enterprise Chassis
- ▶ Provide connections from each I/O module to a redundant upstream infrastructure

Figure 2-5 on page 34 shows an example of a node with a dual port adapter in adapter slot 1 and a quad port adapter in adapter slot 2. The associated lanes that the adapters take to the respective I/O modules in the rear also are shown. To ensure redundancy, when selecting NICs for a team, use NICs that connect to different physical I/O modules.

For example, if you were to select the first two NICs shown coming off the top of the quad port adapter, you realize twice the bandwidth and compute node redundancy. However, the I/O module in I/O bay 3 can become a single point of failure, making this configuration a poor design for HA.

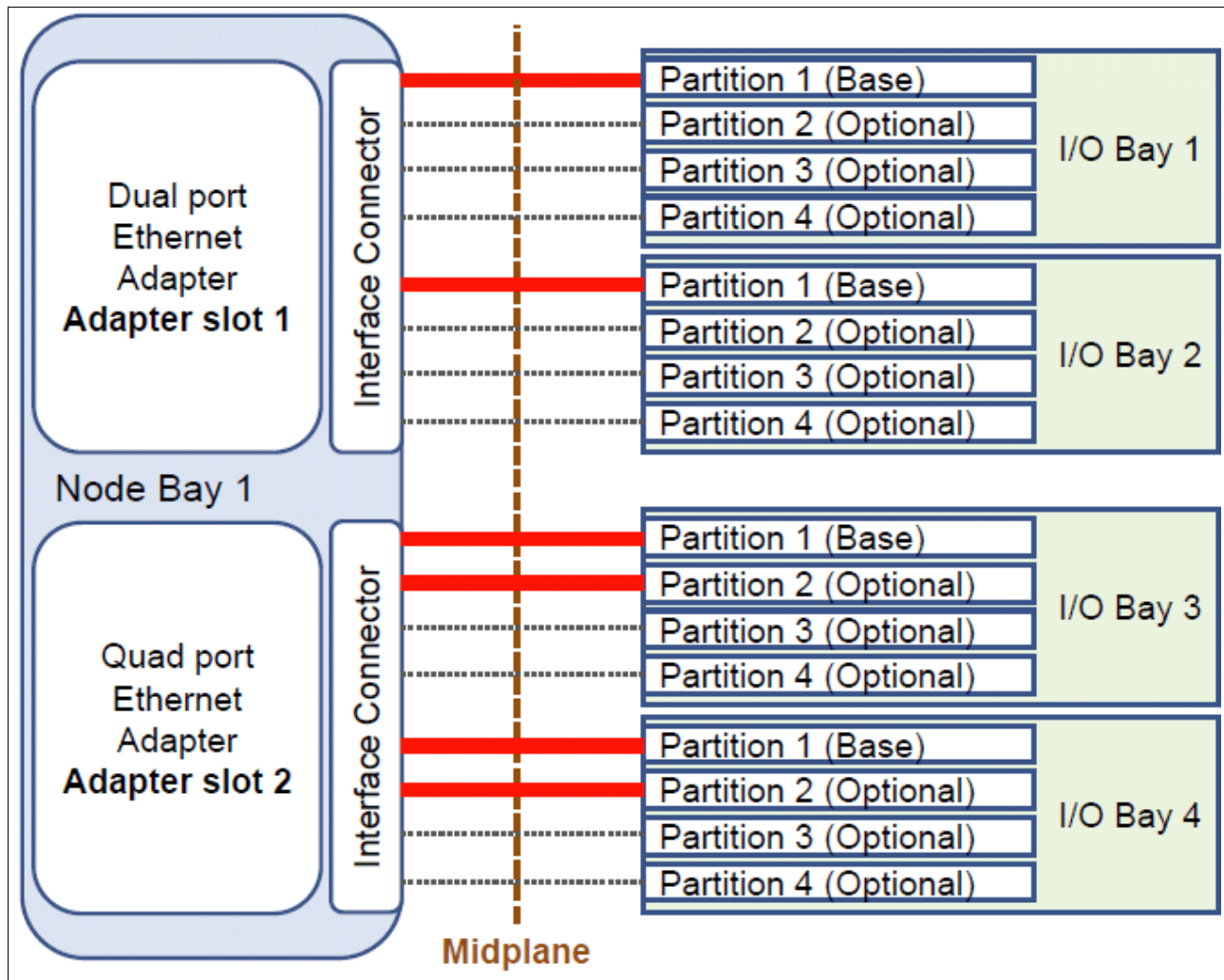


Figure 2-5 Active lanes shown in red based on adapter installed and partition enabled

2.6.1 Highly available topologies

The use cases described in the following sections were selected primarily based on input from IBM System Networking Consulting Engineers as to what has been observed most often in the field during client engagements.

Note: Although these implementation scenarios have been tested and verified to be compatible with upstream Juniper and Cisco networks in a lab environment, these are not the only design options available to the network architect. Customers should consult with their IBM Account Representative to engage IBM Worldwide System Networking Consulting Engineers for more in-depth design discussion.

Looped and blocking design

One of the most traditional designs for chassis HA server-based deployments is the looped and blocking design, as indicated in Figure 2-6 on page 35.

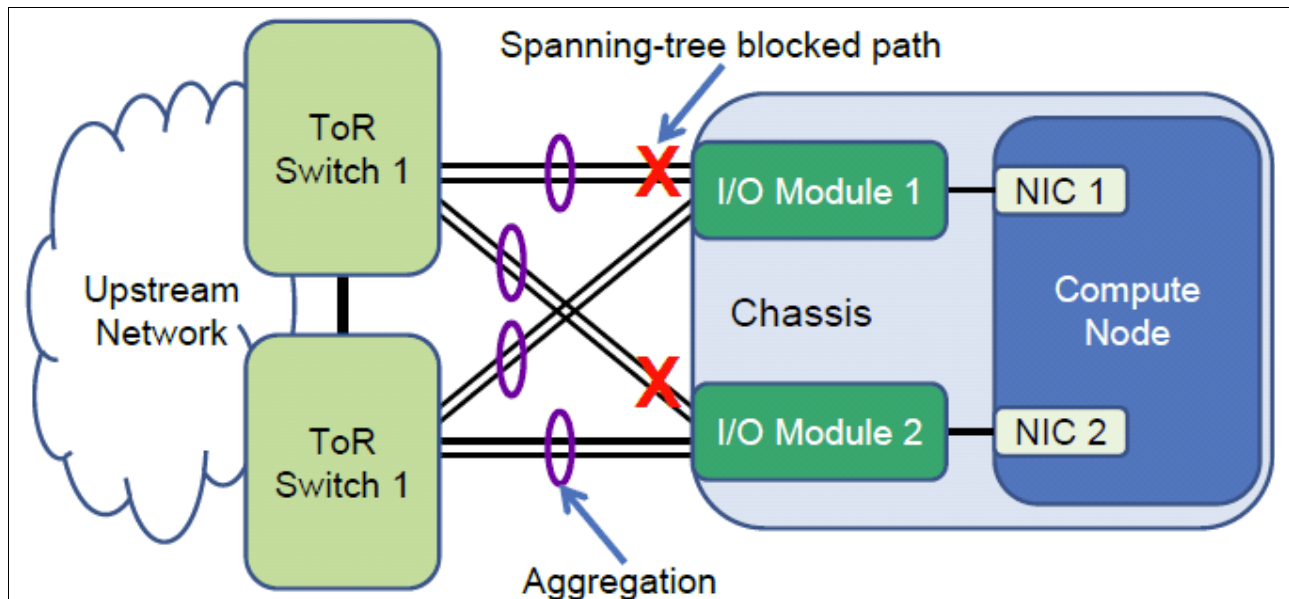


Figure 2-6 Looped and blocking design, no host NIC teaming

The looped and blocking design shows each I/O module in the Enterprise Chassis with two direct aggregations to a pair of upstream Top-of-Rack (ToR) switches. The specific number and speed of the external ports used for link aggregation in this and other designs shown in this section depend on the redundancy and bandwidth requirements of the client. This topology is a bit complicated and is suggested for environments in which hosts need network redundancy, but they are not themselves performing any NIC teaming. Although this choice offers complete network-level redundancy out of the chassis, the potential exists to lose half of the available links and bandwidth due to the Spanning Tree Protocol (STP) blocking them.

Important: Due to possible issues with looped designs in general, a strong recommendation of good L2 design is to pursue loop-free topologies if you can still offer hosts the high availability access necessary to function.

Non-looped, single upstream device design

An alternative take on the looped and blocking design shown in Figure 2-7 is the non-looped, single upstream device HA design.

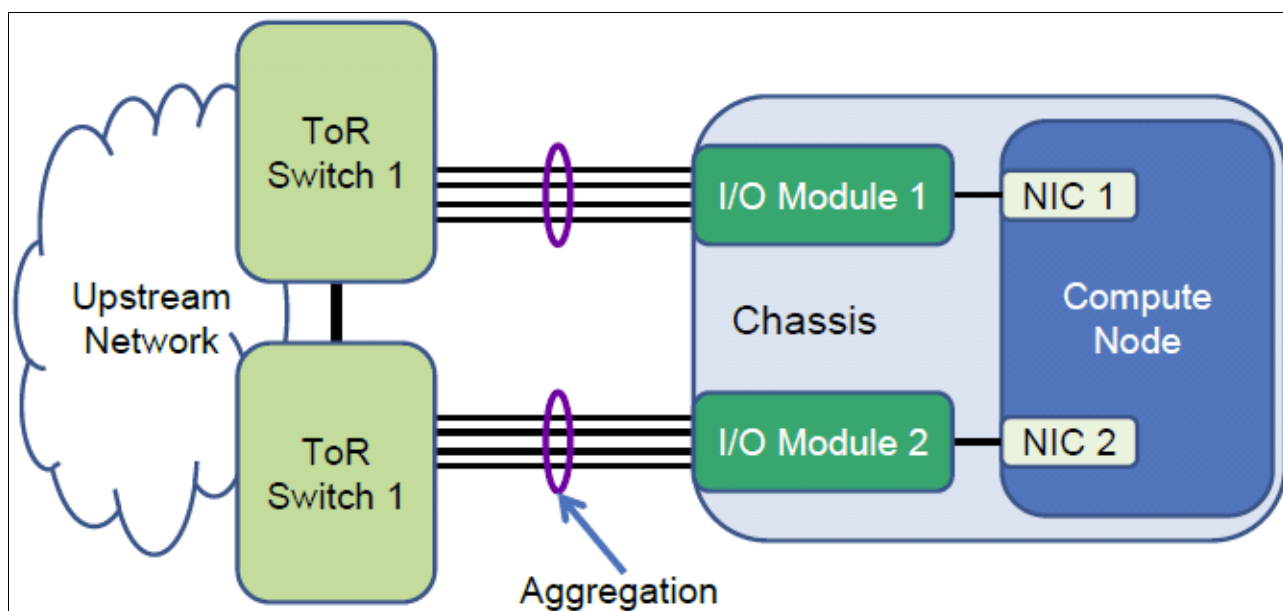


Figure 2-7 Non-looped, single upstream device design, with host NIC teaming

Figure 2-7 shows each I/O module in the Enterprise Chassis directly connected to a single ToR switch through aggregated links. This topology is highly suggested when servers or compute nodes use some form of NIC teaming. To ensure that the nodes correctly detect uplink failures from the I/O modules, Layer 2 Failover must be enabled and configured on the I/O modules. Should the uplinks go down with Layer 2 Failover enabled, the internal ports to the compute nodes are automatically shut down by the I/O module. NIC teaming and bonding also is used to fail the traffic over to the other NIC in the team, ensuring near seamless recovery for the nodes.

The combination of this architecture, NIC teaming on the host, and Layer 2 Failover on the I/O modules, provides for a highly available environment with no loops and thus no wasted bandwidth to spanning-tree blocked links.

Non-looped, multiple upstream devices design

With the recent advent of virtualized chassis and virtual port-channeling technology from networking vendors (including IBM) a third general topology becomes available, which is illustrated in Figure 2-8 on page 37.

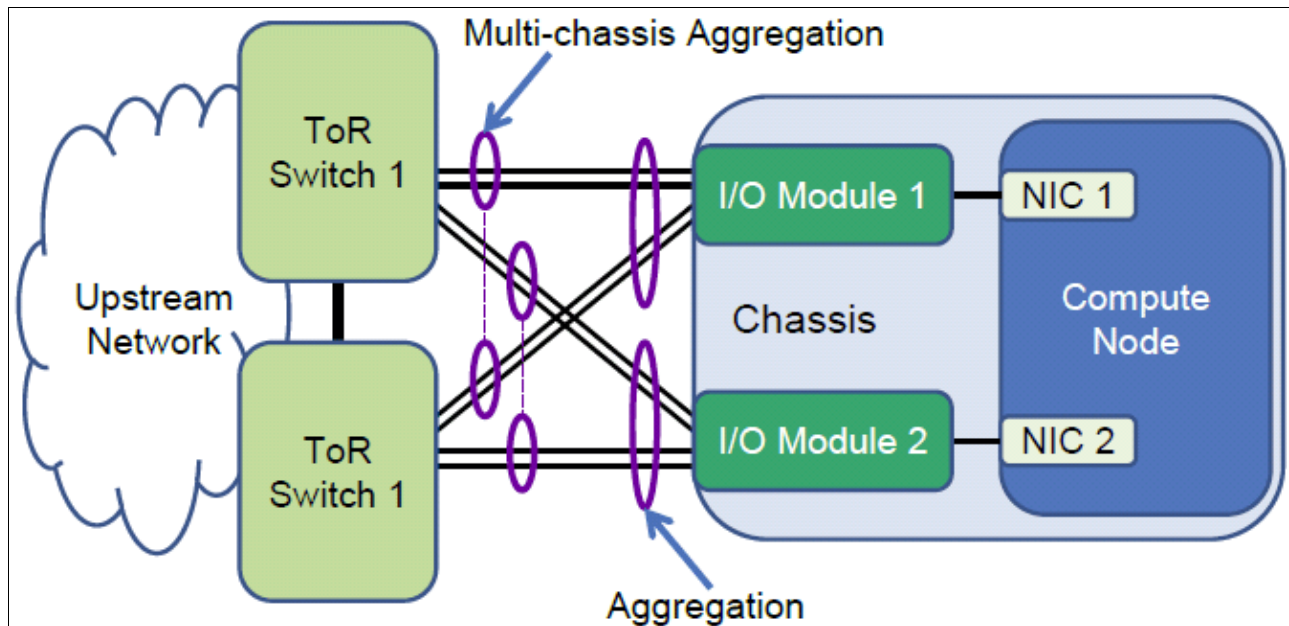


Figure 2-8 Non-looped, multiple upstream devices design, with hosts that can run either teamed or non-teamed NIC cards

The non-looped, multiple upstream devices design brings the best of both the “looped and blocking design” and the “non-looped, single upstream device design” in a robust, stable implementation, suitable for use with hosts that have either teamed or non-teamed NICs.

Offering the maximum bandwidth and high availability of the three topologies covered, this design requires the ToR switches to appear as a single logical switch to each I/O module in the Enterprise Chassis. This technology is vendor-specific at the time of this writing. However, the products of most major vendors support this functionality, including IBM System Networking products. The I/O modules do not need any special aggregation feature to take advantage of this functionality. Instead, normal static or LACP aggregation support is needed because the modules are a simple point-to-point aggregation to a single upstream device.

The designs reviewed in this section all assume that the L2/L3 boundary for the network is at or above the ToR switches in the diagrams. Ultimately, each environment needs to be analyzed to understand all the requirements and to ensure that the best design is selected and deployed.

2.7 Practical Use Case 1: Fully redundant with virtualized chassis technology (VSS/vPC/VLAG)

This implementation scenario incorporates switch virtualization features that allow a downstream switch the ability to be connected to two upstream, virtualized switches through the means of aggregated links, or port-channels. Inter-switch links (ISLs) between the same or similar products on the aggregation or access layer provide a loop-free design that is both redundant and fully available in terms of bandwidth to the eventual downstream nodes. The switches are peers of one another and synchronize their logical view of the access layer port structure and internally prevent implicit loops. This design is suggested for clients that want to use a best-practice implementation on a Cisco network using next generation networking features such as the Cisco Virtual Switching System (VSS) or Virtual Port Channel (vPC) technology.

Important: This is the industry nominated best practice.

Following are some advantages of this approach:

- ▶ Active/active uplinks help to avoid the wasted bandwidth associated with links blocked by Spanning Tree.
- ▶ Maximum redundancy and fault tolerance.
- ▶ Extremely fast convergence times.

Following are some disadvantages of this approach:

- ▶ Requires upstream equipment that supports virtualization features, and a network architect that is familiar with the implementation details.
- ▶ More cabling and connections are necessary.
- ▶ Careful implementation and planning to ensure correct operation.

Figure 2-9 shows the network topology for the fully redundant scenario with virtualized chassis technology.

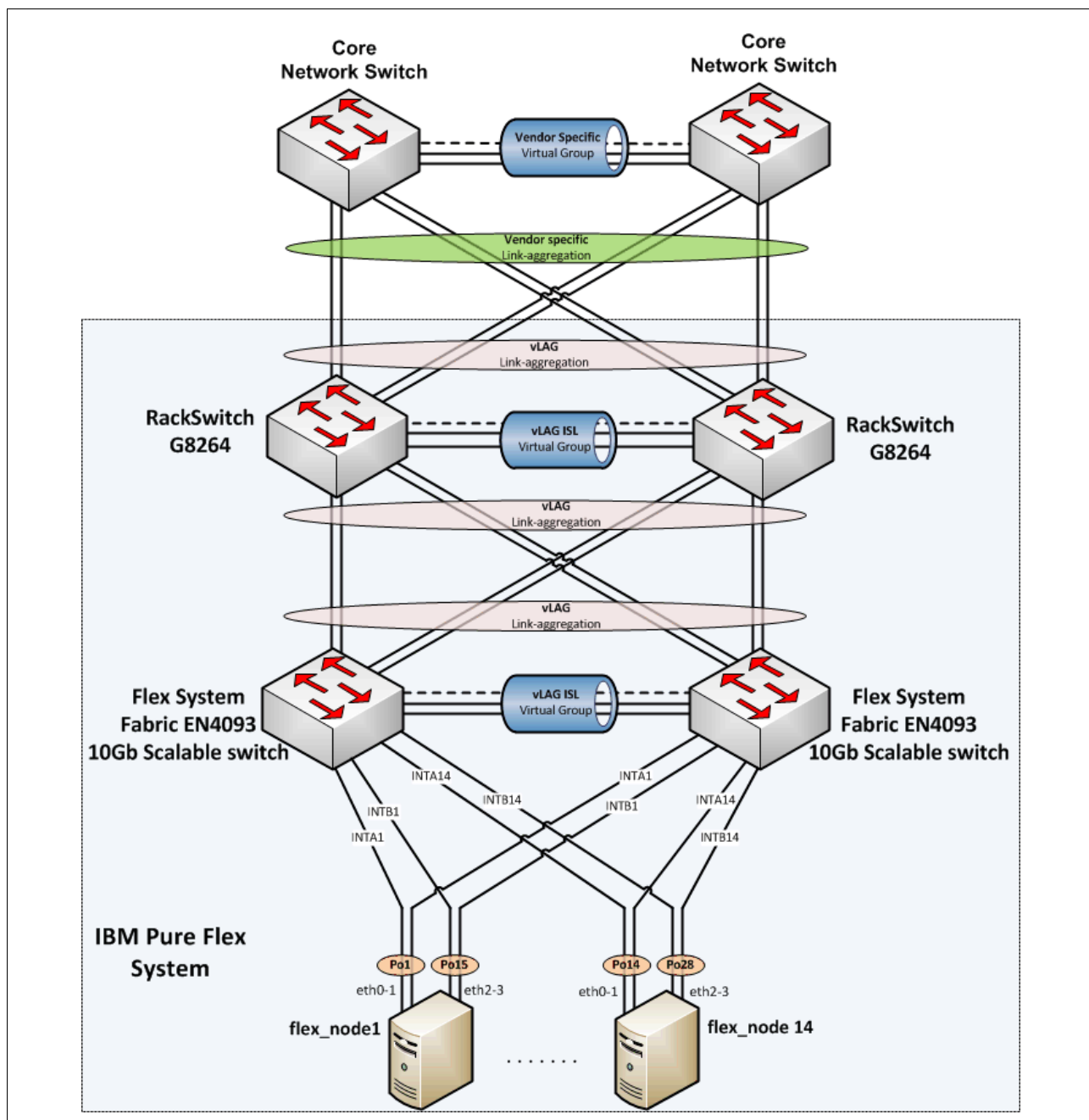


Figure 2-9 A fully redundant scenario with virtualized chassis technology

2.8 Other use cases

The following use cases are presented because they are still present in installations in the IT world.

2.8.1 Practical Use Case 2: Fully redundant with traditional Spanning Tree

This implementation scenario uses a more traditional, classic network design with the Spanning Tree Protocol serving as a protection against bridge or L2 loops. Clients with upstream Cisco equipment that might not have the ability to aggregate themselves from a virtualized standpoint (see Cisco Catalyst 6500 Virtual Switching System, or Cisco Virtual PortChannel on the Nexus platform), or if they are more comfortable with STP, can choose this implementation scenario.

Following are some advantages of this approach:

- ▶ Almost plug and play if Per VLAN Rapid Spanning Tree Protocol plus (PVRST+) is used on both Cisco (default selection in NX-OS) equipment and IBM equipment (default selection as of recent software versions of Network OS)
- ▶ Does not require extra steps or implementation experience in switch virtualization features and functionality in order to begin implementation.
- ▶ Can be done with almost any data center-class upstream Cisco switch.

Following are some disadvantages of this approach:

- ▶ Links are blocked by Spanning Tree to prevent bridging loops, wasting valuable bandwidth.
- ▶ Can take slightly longer convergence times in the event of a link failure.

Troubleshooting problems with Spanning Tree might be more difficult for less experienced network architects.

2.8.2 Practical Use Case 3: Fully redundant with Open Shortest Path First

This implementation scenario exclusively uses the Layer-3 routing protocol Open Shortest Path First (OSPF) to provide network connectivity to the Flex Enterprise Chassis. Although this design is different from the presented Layer-2 implementation scenarios, the end goal of providing a fully redundant infrastructure to the compute nodes still applies. Clients with upstream Cisco equipment that prefer to limit the exposure of Layer 2 down to the chassis can choose to implement OSPF all the way down to the embedded I/O modules instead, but understandability with some caveats.

Following are some advantages of this approach:

- ▶ Limited Layer-2 exposure to network infrastructure equipment, limiting the ability of a misconfiguration resulting in a broadcast storm, Address Resolution Protocol (ARP) flooding, or other negative consequence of Layer 2.
- ▶ OSPF builds adjacency matrixes and adjusts automatically to down equipment or links.

Following are some disadvantages of this approach:

- ▶ Less flexibility in exposing compute nodes to VLANs that might exist on other switches, either physically or geographically separated.

- Applications that specifically require Layer-2 adjacency for functionality, such as virtual machine-based mobility between hypervisors, will not function between differing chassis without Layer-2 adjacency.

2.9 Summary and conclusions

The IBM Flex System platform provides a unique set of features that enable the integration of leading-edge technologies and transformation approaches into the data centers. These IBM Flex System features ensure that the availability, performance, scalability, security, and manageability goals of the data center networking design are met as efficiently as possible.

The key data center technology implementation trends include the virtualization of servers, storage, and networks. Trends also include the steps toward infrastructure convergence that are based on mature 10 Gb Ethernet technology. In addition, the data center network is being flattened, and the logical overlay network becomes important in overall network design. These approaches and directions are fully supported by IBM Flex System offerings.

IBM Flex System data center networking capabilities provide solutions to many issues that arise in data centers where new technologies and approaches are being adopted:

- Network administrator responsibilities can no longer be limited by the NIC level. Administrators must consider the platforms of the server network-specific features and requirements, such as vSwitches. IBM offers Distributed Switch 5000V that provides standard functional capabilities and management interfaces to ensure smooth integration into a data center network management framework.
- After 10 Gb Ethernet networks reach their maturity and price attractiveness, they can provide sufficient bandwidth for virtual machines in virtualized server environments and become a foundation of unified converged infrastructure. IBM Flex System offers 10 Gb Ethernet Scalable Switches and Pass-thru Modules that can be used to build a unified converged fabric.
- Although 10 Gb Ethernet is becoming a prevalent server network connectivity technology, there is a need to go beyond 10 Gb to avoid oversubscription in switch-to-switch connectivity, thus freeing room for emerging technologies, such as 40 Gb Ethernet. IBM Flex System offers the industry's first 40 Gb Ethernet-capable switch, EN4093/EN4093R, to ensure that the sufficient bandwidth is available for inter-switch links.
- Network infrastructure must be VM-aware to ensure the end-to-end QoS and security policy enforcement. IBM Flex System network switches offer VMready capability that provides VM visibility to the network and ensures that the network policies are implemented and enforced end-to-end.
- Pay as you grow scalability becomes an essential approach as increasing network bandwidth demands must be satisfied in a cost-efficient way with no disruption in network services. IBM Flex System offers scalable switches that enable ports when required by purchasing and activates simple software FoD upgrades without the need to buy and install more hardware.
- Infrastructure management integration becomes more important because the interrelations between appliances and functions are difficult to control and manage. Without integrated tools that simplify the data center operations, managing the infrastructure box-by-box becomes cumbersome. IBM Flex System offers centralized systems management with the integrated management appliance, IBM Flex System Manager™, that integrates network management functions into a common data center management framework from a single pane of glass.



Planning and hardware selection

In this chapter, we cover networking planning considerations for the IBM Flex System platform in a data center to meet availability, performance, scalability, and systems management goals.

We describe the following topics from a planning perspective:

- ▶ Hardware selection and interoperability
- ▶ Virtual local area networks (VLANs)
- ▶ Scalability and performance
- ▶ High availability
- ▶ Virtual Fabric virtual network interface card (vNIC) capabilities
- ▶ Management

3.1 Hardware selection and interoperability

In this section, we list considerations that you must think about when selecting the right Ethernet switch modules and compute node NICs for your environment. The interoperability guide is an important tool when choosing hardware components, and it must be considered and fully understood.

3.1.1 Selecting the Ethernet switch module

The Enterprise Chassis offers a range of Ethernet connectivity options. Selecting the Ethernet switch module that is best for an environment is a process specific to each client. The following factors must be considered during the selection process:

- ▶ If your bandwidth requirements are such that any servers within the Enterprise Chassis require a 10 Gb connection, the EN4093/EN4093R 10Gb Scalable Switch is the best choice.
- ▶ If your upstream bandwidth requirements are such that 1 Gb (or even an aggregation of 1 Gb) links are insufficient, the EN4093/EN4093R 10Gb Scalable Switch module is the best choice.
- ▶ If you are installing the Enterprise Chassis into an environment where the upstream network is using 10 Gb (or there is a plan to upgrade to 10 Gb), the EN4093/EN4093R 10Gb Scalable Switch module is the best choice.
- ▶ If you require the maximum bandwidth available for the nodes, the EN4093/EN4093R 10Gb Scalable Switch module is the best choice.
- ▶ If you need to take advantage of the advanced virtualization features, such as vNIC, that are offered in the Enterprise Chassis, the EN4093/EN4093R 10Gb Scalable Switch is the best choice.
- ▶ If there is no immediate need for 10 Gb, there are no plans to upgrade to 10 Gb in the foreseeable future, and you have no need for any of the advanced features offered in the 10 Gb solution, the EN2092 1Gb Ethernet Switch is the optimal solution.
- ▶ If you need a solution that is transparent to the network, has only one link for each compute node for each I/O module, and requires direct connections from the compute node to the external Top-of-Rack (ToR) switch, the EN4091 10Gb Ethernet Pass-thru is an option.

There are more criteria involved because each environment has its own unique attributes. However, the criteria reviewed in this section are a good starting point in the decision-making process.

The Ethernet I/O module selection criteria are summarized in Table 3-1.

Table 3-1 Switch module selection criteria

	Switches	
	EN2092 1Gb Ethernet Switch	EN4093/ EN4093 R 10Gb Scalable Switch
Requirement		
Gigabit Ethernet to nodes	Yes	Yes
10 Gb Ethernet to nodes	No	Yes

	Switches	
	EN2092 1Gb Ethernet Switch	EN4093/ EN4093 R 10Gb Scalable Switch
Requirement		
10 Gb Ethernet uplinks	Yes	Yes
40 Gb Ethernet uplinks	No	Yes
Basic Layer 2 switching (VLAN, port aggregation)	Yes	Yes
Advanced Layer 2 switching: IEEE features (STP, QoS)	Yes	Yes
Layer 3 IPv4 switching (forwarding, routing, ACL filtering)	Yes	Yes
Layer 3 IPv6 switching (forwarding, routing, ACL filtering)	Yes	Yes
10 Gb Ethernet CEE	No	Yes
FCoE	No	Yes ^a
Switch stacking	No	Yes ^a
vNIC support	No	Yes
VMready	Yes	Yes

a. Support for FCoE and switch stacking is planned in future firmware releases.

3.1.2 Selecting the compute node NICs

Table 3-2 gives details about compute node network adapter selection criteria.

EN2024 (4-port 1Gb Ethernet Adapter) is an obvious choice when 10 Gbps speed to compute nodes is not required.

In cases when 10 Gbps speed to compute nodes is required, the need for advanced functions (Converged Enhanced Ethernet (CEE), Fibre Channel over Ethernet (FCoE), and vNIC support) will determine the most suitable 10-Gb adapter. If you require advanced features such as CEE, FCoE, and vNIC support, select the CN4054 10Gb Virtual Fabric Adapter.

Table 3-2 Compute node network adapter selection criteria

	Adapters		
	CN4054 10Gb Virtual Fabric Adapter	EN2024 4-port 1Gb Ethernet Adapter	EN4132 2-port 10Gb Ethernet Adapter
Requirement			
Gigabit Ethernet to nodes	Yes	Yes	No
10 Gb Ethernet to nodes	Yes	Yes	Yes
10 Gb Ethernet CEE	Yes	No	No
FCoE	Yes	No	No
vNIC support	Yes	No	No

3.1.3 Interoperability considerations

When selecting Ethernet switch modules and compute node Ethernet adapters, you need to ensure that you choose compatible and interoperable components. The most important resource for determining interoperability requirements is the *IBM Flex System Interoperability Guide*, REDP-FSIG, available for download at the following website:

<http://www.redbooks.ibm.com/abstracts/redpfsig.html>

Table 3-3 shows interoperability information for compute nodes and Ethernet adapters. As you can see, EN2024 (4-port 1Gb Ethernet Adapter) can be used universally on all compute node types. However, other Ethernet adapters are specific to different compute node types:

- ▶ EN4132 and CN4054 are supported on compute nodes x220, x240, and x440, but not on Power Systems compute nodes p24L, p260, and p460.
- ▶ EN4054 is supported on Power Systems compute nodes p24L, p260, and p460, but not on compute nodes x220, x240, and x440.

Table 3-3 Compute node and Ethernet adapter interoperability

Part number	Power feature code	Ethernet adapter	x220	x240	x440	p24L	p260	p460
49Y7900	1763	EN2024 4-port 1Gb Ethernet Adapter	Yes	Yes	Yes	Yes	Yes	Yes
90Y3466	None	EN4132 2-port 10Gb Ethernet Adapter	Yes	Yes	Yes	No	No	No
None	1762	EN4054 4-port Ethernet Adapter	No	No	No	Yes	Yes	Yes
90Y3554	None	CN4054 10Gb Virtual Fabric Adapter	Yes	Yes	Yes	No	No	No

In Table 3-4, we offer a different kind of interoperability information: Ethernet switch to adapter interoperability. All combinations except one are supported: EN2092 1Gb Ethernet Switch does not support EN4132 2-port 10Gb Ethernet Adapter.

Table 3-4 Ethernet switch to adapter interoperability

Part number	Power feature code	Ethernet adapter	EN4093 10Gb Scalable Switch	EN2092 1Gb Ethernet Switch	EN4091 10Gb Ethernet Pass-thru
49Y7900	1763	EN2024 4-port 1Gb Ethernet Adapter	Yes	Yes	Yes
90Y3466	None	EN4132 2-port 10Gb Ethernet Adapter	Yes	No	Yes
None	1762	EN4054 4-port Ethernet Adapter	Yes	Yes	Yes
90Y3554	None	CN4054 10Gb Virtual Fabric Adapter	Yes	Yes	Yes

3.2 Virtual local area networks

Virtual local area networks (VLANs) are commonly used in a Layer 2 network to split groups of networked systems into manageable broadcast domains, create logical segmentation of workgroups, and enforce security policies among logical segments. Primary VLAN considerations include the number and types of supported VLANs and VLAN tagging protocols.

All Ethernet I/O switch modules in the Enterprise Chassis support the following VLAN-related features:

- ▶ VLANs available in the range of 1 - 4094
 - Some VLANs might be reserved when certain features are enabled
 - VLAN 4095 is a reserved management VLAN
- ▶ Institute of Electrical and Electronics Engineers (IEEE) 802.1Q for VLAN tagging on links (also called *trunking* by some vendors)
 - Support for tagged or untagged native VLAN
- ▶ Port-based VLANs
- ▶ Protocol-based VLANs
- ▶ 802.1x Guest VLANs
- ▶ VLAN Maps for ACLs
- ▶ VLAN-based port mirroring

Specific to 802.1Q VLAN tagging, this feature is critical to maintain VLAN separation when packets in multiple VLANs must traverse a common link between devices. Without a tagging protocol, such as 802.1Q, maintaining VLAN separation between devices can be accomplished through a separate link for each VLAN, a less than optimal solution.

Important: In rare cases, there are some older nonstandards-based tagging protocols used by vendors. These protocols are not compatible with 802.1Q.

The need for 802.1Q VLAN tagging is not relegated only to networking devices. It is also supported and frequently used on end nodes, and is implemented differently in various operating systems. For example, in Windows based systems, a vendor driver is needed to subdivide the physical interface into logical NICs, with each logical network interface card (NIC) set for a specific VLAN. Typically, this setup is part of the teaming software from the NIC vendor.

For Linux, tagging is done by creating subinterfaces of a physical or logical NIC, such as eth0.10 for VLAN 10.

For VMware ESX, tagging can be done within the vSwitch through port group tag settings (known as *Virtual Switch Tagging*). Tagging also can be done in the OS within the guest VM itself (called *Virtual Guest Tagging*).

From an OS perspective, having several logical interfaces can be useful when an application requires more than two separate interfaces and you do not want to dedicate an entire physical interface. It might also help to implement strict security policies for separating network traffic that uses VLANs and having access to server resources from different VLANs, without adding more physical network adapters.

Review the documentation of the application to ensure that the application deployed on the system supports the use of logical interfaces often associated with VLAN tagging.

3.3 Scalability and performance

The IBM Flex Systems Enterprise Chassis has four I/O bays. Each bay can support many connections, both toward the nodes and toward the external network. The number of supported internal and external connections depends on the Ethernet switch module installed

in the I/O bay, the licenses installed on the Ethernet switch, and the adapters installed on the node.

The Ethernet switch modules available for the Enterprise Chassis are scalable. This means that additional banks of ports (also known as *partitions* in this context) can be enabled as needed, thus scaling the switch to meet a particular requirement.

The chassis architecture allows up to four switch partitions in each I/O module, for a total of up to 16 partitions within each chassis. The number of ports in these partitions available for use by a node depends on the following factors:

- ▶ The Ethernet I/O module installed
- ▶ The partitions activated on the I/O module
- ▶ The I/O adapters installed in the nodes

The Ethernet I/O switch modules include a base partition (always enabled), and require upgrades to enable the extra partitions. Not all Ethernet switch modules support the same number of partitions. A cross-reference of the number of partitions supported on each of the available switch modules is shown in Table 3-5. The pass-thru is a fixed function device and as such, has no real concept of port expansion.

Table 3-5 Module names and the number of switch partitions

Module name	Number of partitions supported
EN2092 1Gb Ethernet Switch	2
EN4093/EN4093R 10Gb Scalable Switch	3
EN4091 10Gb Ethernet Pass-thru	1

As shipped, all Ethernet switch modules have support for base partition, which includes 14 internal ports, one to each of the compute node bays upfront. Upgrades to the scalable switches to enable other partitions (and additional external ports) are added as part of the Feature on Demand (FoD) capability. Because of these upgrades, it is possible to increase ports without hardware changes. As each FoD is enabled, the ports controlled by the upgrade are activated. If the compute nodes have suitable I/O adapters, the ports are available to the nodes.

The act of enabling a bank of ports by applying the FoD merely enables more ports for the switch to use. There is no logical or physical separation of these ports from a networking perspective, only from a licensing perspective. Adding these FoDs increases the size of the switch. The term *partition* in this context is only about increasing the number of ports available for use.

As an example of how this licensing works, the EN4093/EN4093R 10Gb Scalable Switch, by default, includes 14 internal available ports, together with ten uplink 10Gb enhanced small form-factor pluggable (SFP+) ports. More partitions can be enabled with an FoD upgrade, thus providing a second or third set of 14 internal ports and additional uplinks, as shown in Figure 3-1 on page 49.

Internal-facing ports for partition 3 are currently not usable by any of the available I/O adapters in the nodes. The only reason to enable the second upgrade at this time is to obtain the extra four 10Gb SFP+ uplinks included as part of the process to enable this partition.

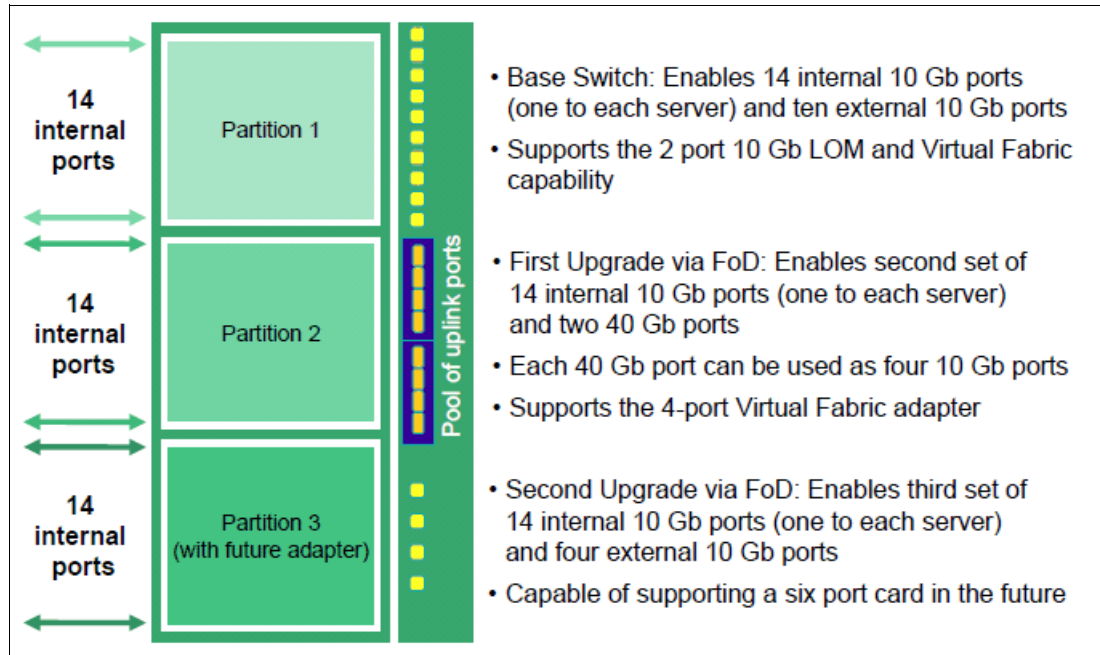


Figure 3-1 Partition layout for EN4093/EN4093R 10Gb Scalable Switch

The ability to add ports and bandwidth as needed is a critical element of a scalable platform.

3.4 High availability

Most clients require continuous access to their network-based resources and applications. Providing high availability (HA) for client network resources can be a complex task that involves fitting multiple pieces together on a hardware and software level. One key to system high availability is to provide high availability access to the network infrastructure.

Network infrastructure availability can be achieved by using certain techniques and technologies. Most techniques and technologies are widely used standards, but some are specific to the Enterprise Chassis. We review the most common technologies that can be implemented in an Enterprise Chassis environment to provide high availability to the network infrastructure.

A typical LAN infrastructure consists of server NICs, client NICs, network devices such as Ethernet switches, and cables that connect them. Specific to the Enterprise Chassis, the potential failure areas for compute node network access include port failures (both on switches and the compute node adapters), the midplane, and the Ethernet switch modules.

The first step in achieving high availability is to provide physical redundancy of components connected to the infrastructure. Physical redundancy typically means that the following measures are taken:

- ▶ Deploy compute node NICs in pairs
- ▶ Deploy switch modules in pairs
- ▶ Connect the pair of node NICs to separate Ethernet switch modules in the Enterprise Chassis
- ▶ Provide connections from each Ethernet switch module to a redundant upstream infrastructure

Shown in Figure 3-2 is an example of a node with a dual port adapter in adapter slot 1 and a quad port adapter in adapter slot 2. The associated lanes that the adapters take to the respective I/O modules in the rear are highlighted. To ensure redundancy, when selecting NICs for a team, use NICs that connect to different physical I/O modules. For example, if you select the first two NICs shown coming off the top of the quad port adapter for the team, you realize twice the bandwidth and compute node redundancy. However, the Ethernet switch module in I/O bay 3 can become a single point of failure, making this configuration a poor design for HA. It makes more sense to use a NIC that connects to the Ethernet switch module in I/O bay 3 and a NIC that connects to the switch module in I/O bay 4 for your team.

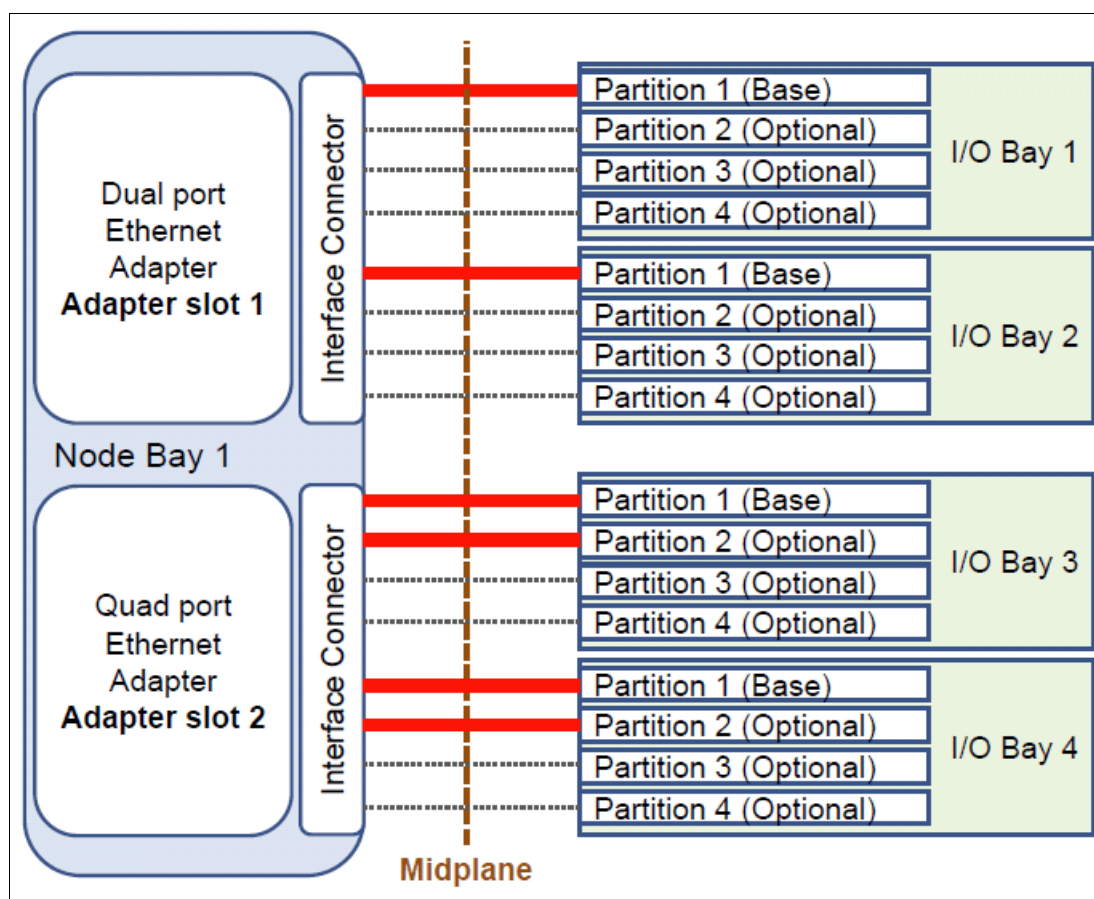


Figure 3-2 Active lanes shown in red based on adapter installed and partition enabled

After physical redundancy requirements are met, it is necessary to consider logical elements to use this physical redundancy. The following logical features aid in high availability:

- ▶ NIC teaming and bonding on the compute node
- ▶ Layer 2 (L2) failover (also known as *trunk failover*) on the I/O modules
- ▶ Rapid Spanning Tree Protocol for looped environments
- ▶ Virtual link aggregation on upstream devices connected to the I/O modules
- ▶ Virtual Router Redundancy Protocol for redundant upstream default gateway
- ▶ Routing Protocols (such as Routing Information Protocol (RIP) or Open Shortest Path First (OSPF)) on the I/O modules, if L2 adjacency is not a concern

3.4.1 Examples of topologies

The Enterprise Chassis can be connected to the upstream infrastructure in a number of possible combinations. Some examples of potential L2 designs are included here.

Important: There are many design options available to the network architect, and this section shows a small subset based on some useful L2 technologies. With the large feature set and high port densities, the Ethernet switch modules of the Enterprise Chassis can be used to implement many other designs.

One of the traditional designs for chassis server-based deployments is the looped and blocking design, as shown in Figure 3-3.

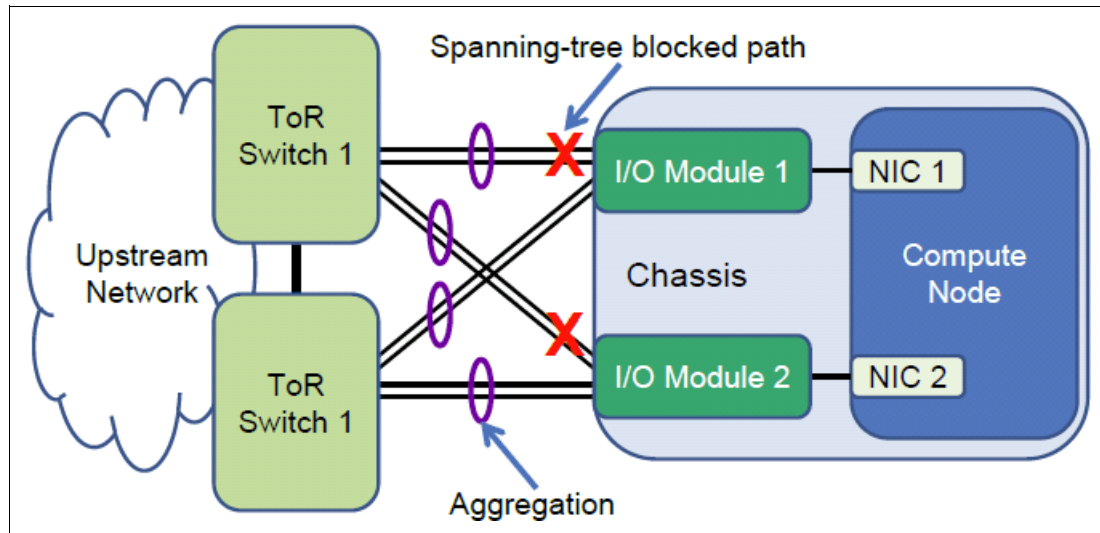


Figure 3-3 Topology 1: Typical looped and blocking topology

Topology 1 in Figure 3-3 features each Ethernet switch module in the Enterprise Chassis with two direct aggregations to a pair of ToR switches. The specific number and speed of the external ports used for link aggregation in this and other designs shown in this section depend on the redundancy and bandwidth requirements of the client. This topology is suggested for environments in which compute nodes need network redundancy, but they are not performing any NIC teaming on the node side. Although offering complete network-attach redundancy out of the chassis, the potential exists to lose half of the available bandwidth to Spanning Tree blocking.

Topology 2 in Figure 3-4 on page 52 features each switch module in the Enterprise Chassis directly connected to a single ToR switch through aggregated links. This topology is suggested when compute nodes use some form of NIC teaming. To ensure that the nodes correctly detect uplink failures from the Ethernet switch modules, trunk failover (as described in 3.4.5, “Trunk failover” on page 56) must be enabled and configured on the switch modules. With failover, if the uplinks go down, internal ports to the nodes shut down. This works with NIC teaming and bonding to fail the traffic over to the other NIC in the team. The combination of this architecture, NIC teaming on the node, and trunk failover on the I/O modules, provides for a highly available environment with no loops and thus no wasted bandwidth to spanning-tree blocked links.

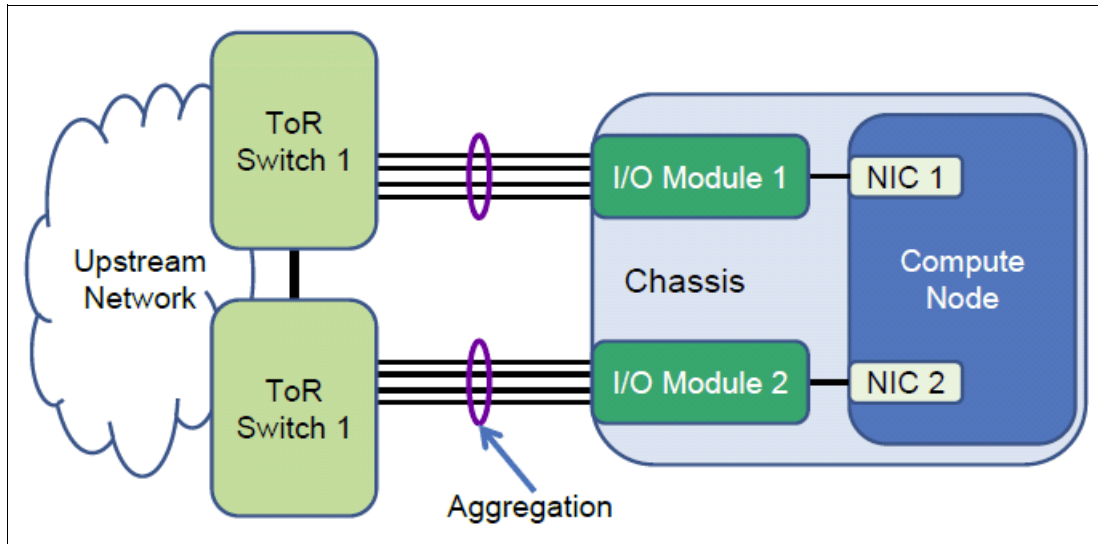


Figure 3-4 Topology 2: Non-looped HA design

Topology 3, as shown in Figure 3-5, brings the best of both topology 1 and 2 together in a robust design, suitable for use with nodes that run either teamed or non-teamed NICs.

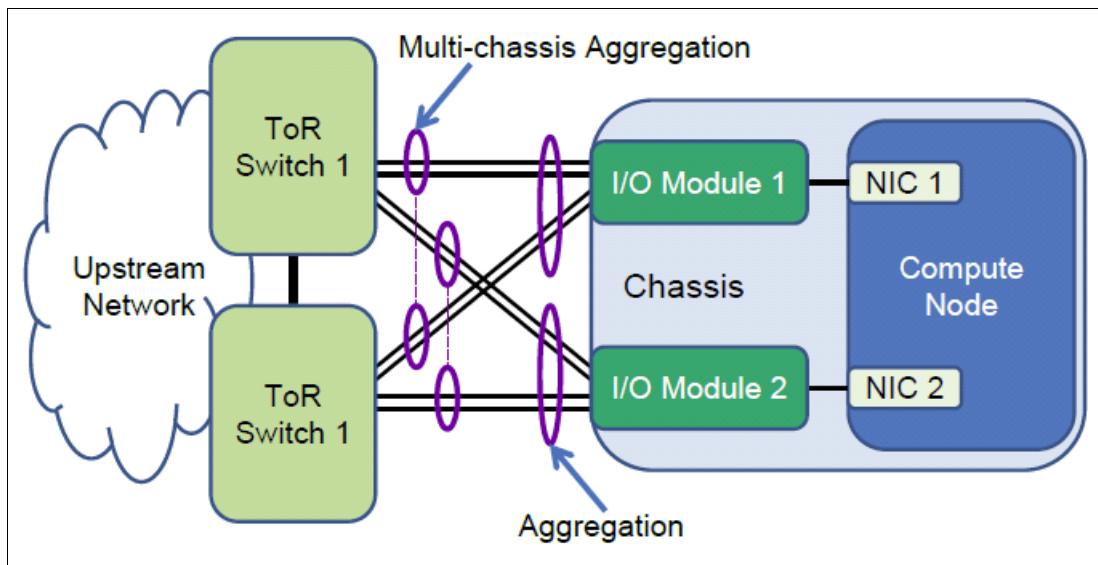


Figure 3-5 Topology 3: Non-looped design using multi-chassis aggregation

Offering the maximum bandwidth and high availability, this design requires that the ToR switches provide a form of multi-chassis aggregation that allows an aggregation to be split between two physical switches. The design requires the ToR switches to appear as a single logical switch to each Ethernet switch module in the Enterprise Chassis. At the time of this writing, this functionality is vendor-specific; however, the products of most major vendors, including IBM ToR products, support this type of function. The Ethernet switch modules do not need any special aggregation feature to take advantage of this functionality. Instead, normal static or LACP aggregation support is needed because the modules are a simple point-to-point aggregation to a single upstream device.

To further enhance the design shown in Figure 3-5 on page 52, enable the uplink failover feature (see 3.4.5, “Trunk failover” on page 56) on the Enterprise Chassis Ethernet switch modules, thus ensuring the most robust design possible.

The designs reviewed in this section all assume that the L2/L3 boundary for the network is at or above the ToR switches in the diagrams. We touched only on a few of the many possible ways to interconnect the Enterprise Chassis to the network infrastructure. Ultimately, each environment must be analyzed to understand all the requirements to ensure that the best design is selected and deployed.

3.4.2 Spanning Tree

Spanning Tree is defined in the IEEE specification 802.1D. The primary goal of Spanning Tree is to ensure a loop-free design in an L2 network. Loops are not allowed in an L2 network because there is no mechanism in an L2 frame to aid in the detection and prevention of looping packets, such as a time to live field or a hop count (all part of the L3 header portion of some headers, but not seen by L2 switching devices). Packets might loop indefinitely and consume bandwidth that could be used for other purposes. Ultimately, an L2-looped network eventually fails as broadcast and multicast packets rapidly multiply through the loop.

The entire process used by Spanning Tree to control loops is beyond the scope of this document. In its simplest terms, Spanning Tree controls loops by exchanging Bridge Protocol Data Units (BPDUs) and building a tree that blocks redundant paths until they might be needed, for example, if the path selected for forwarding went down.

The Spanning Tree specification evolved considerably. Other standards, such as 802.1w (rapid Spanning Tree) and 802.1s (multi-instance Spanning Tree) are included in the current Spanning Tree specification, 802.1D-2004. As some features were added, other features, such as the original non-rapid Spanning Tree, are no longer part of the specification.

Both the EN2092 1Gb Ethernet Switch and the EN4093/EN4093R 10Gb Scalable Switch support the 802.1D-2004 specification. They also support a Cisco proprietary version of Spanning Tree called *Per VLAN Rapid Spanning Tree* (PVRST). The following list shows the four Spanning Tree modes currently supported on the IBM Ethernet switch modules:

- ▶ Rapid Spanning Tree Protocol (RSTP), also known as *mono-instance Spanning Tree Protocol*
- ▶ Multi-instance Spanning Tree Protocol (MSTP)
- ▶ PVRST
- ▶ Disabled (turns off Spanning Tree on the switch)

The default Spanning Tree for the Enterprise Chassis Ethernet switch modules is PVRST. This Spanning Tree allows seamless integration into the largest and most commonly deployed infrastructures in use today. This mode also allows for better potential load balancing of redundant links (because blocking and forwarding is determined per VLAN rather than per physical port) over RSTP, and without some of the configuration complexities involved with implementing an MSTP environment.

With PVRST, as VLANs are created or deleted, an instance of Spanning Tree is created or deleted for each VLAN automatically.

Other supported forms of Spanning Tree can be enabled and configured if required, thus allowing the Enterprise Chassis to be readily deployed into the most varied environments.

3.4.3 Link aggregation

Link aggregation involves bundling multiple physical links into a single common link for use between two devices. The primary purposes of aggregation are to improve high availability and increase bandwidth.

Bundling the links

There are several different kinds of aggregation, but the two most common that are supported by the Enterprise Chassis Ethernet switch modules are static aggregation and Link Aggregation Control Protocol (LACP).

Important: In rare cases, there are still some older nonstandards-based aggregation protocols, such as Port Aggregation Protocol (PAgP), in use by some vendors. These protocols are not compatible with either static or LACP aggregations.

Static aggregation does not use any protocol to create the aggregated link. Instead, it simply combines the ports based on the (static) aggregation configuration and assumes that the other side of the connection does the same.

Important: In some cases, static aggregation is referred to as *static LACP*. This term is contradictory because it is difficult in this context to be both static and have a control protocol.

LACP is an IEEE standard that was defined in 802.3ad. The standard was later included in the mainline 802.3 standard but then was pulled out into the current standard 802.1AX-2008. LACP is a dynamic way of determining whether both sides of the link might be aggregating.

The decision to use static aggregation is usually a question of what a client uses in the network. If there is no preference, we provide several advantages and disadvantages of each option to aid in the decision-making process.

Static aggregation is the quickest and easiest way to build an aggregated link. This method is also the most stable in high-bandwidth utilization environments, particularly if pause frames are being exchanged.

Using static aggregation can be advantageous in mixed vendor environments because it can help prevent possible interoperability issues. Because settings in the LACP standard do not have a recommended default, vendors are allowed to use different defaults, which can lead to unexpected interoperability problems. For example, the LACP Data Unit (LACPDU) timers can be set to be exchanged every 1 second or every 30 seconds. If one side is set to 1 second and one side is set to 30 seconds, the LACP aggregation can be unstable.

Important: Most vendors default to using the 30-seconds exchange of LACPDU's, including IBM switches. If you encounter a vendor that defaults to 1-second timers, we suggest that the other vendor switch is set to operate with 30-second timers, rather than setting both to 1 second. This setting tends to produce a more robust aggregation over the 1-second timers.

One of the downsides to static aggregation is that it lacks a mechanism to detect if the other side is correctly configured for aggregation. So, if one side is static and the other side is not configured, configured incorrectly, or is not connected to the correct ports, it is possible to cause a network outage by bringing up the links.

Based on the information presented in this section, If you are sure that your links are connected to the correct ports and that both sides are configured correctly for static aggregation, then static aggregation is a solid choice.

LACP has the inherent safety that a protocol brings to this process. At linkup, LACPDUs are exchanged and both sides must agree they are using LACP before it attempts to bundle the links. So, in the case of misconfiguration or incorrect connections, LACP helps protect the network from an unplanned outage. The disadvantages of using LACP are that it takes a small amount of time to negotiate the aggregation and form an aggregating link (usually under a second), and it can become unstable and unexpectedly fail in environments with heavy and continued pause frame activity.

Another factor to consider about aggregation is whether it is better to aggregate multiple low-speed links into a high-speed aggregation, or use a single high-speed link instead.

If your primary goal is high availability, aggregations can offer a no-single-point-of-failure situation that a single high-speed link cannot offer.

If maximum performance and lowest possible latency are the primary goals, often a single high-speed link makes more sense. Another factor is cost. Often, one high-speed link can cost more to implement than a link that consists of an aggregation of multiple slower links.

However, with 10-Gb uplinks, the above consideration is usually not applicable. We aggregate multiple 10-Gb links to achieve both higher speed and high availability.

Virtual link aggregations

Aside from the standard point-to-point aggregations covered in this section, there is a technology that provides multi-chassis aggregation, sometimes called *distributed aggregation* or *virtual link aggregation*.

Under the latest IEEE specifications, an aggregation is still defined as a bundle between only two devices. By this definition, you cannot create an aggregation on one device and have the links of that aggregation connect to more than a single device on the other side of the aggregation. Using only two devices limits the ability to offer certain robust designs.

Although the standards bodies are working on a solution that provides split aggregations across devices, most vendors devised their own version of multi-chassis aggregation. For example, Cisco has virtual Port Channel (vPC) on Nexus products, and Virtual Switch System (VSS) on the 6500 line. IBM offers virtual Link Aggregation Groups (vLAGs) on many of their ToR solutions, and on the EN4093/EN4093R 10Gb Scalable Switch. The primary goals of vLAG are to overcome the limits imposed by standard aggregation, and provide a distributed aggregation across a pair of switches instead of a single switch.

The decisions whether to aggregate and which method of aggregation is most suitable to a specific environment are not always straightforward. But if the decision is made to aggregate, the I/O modules for the Enterprise Chassis offer the necessary features to integrate into the aggregated infrastructure.

3.4.4 Network interface card teaming

Network interface card (NIC) teaming, also known as *bonding*, is a solution used on servers to logically bond two or more NICs to form one or more logical NICs for purposes of high availability, increased performance, or both.

There are many forms of NIC teaming, and the type available for a server is often tied to the OS installed on the server.

For Microsoft Windows, the teaming software traditionally was provided by the NIC vendor and is installed as an add-on to the operating system. This software often also includes the elements necessary to enable VLAN tagging on the logical NICs created by the teaming software. These logical NICs are seen by the OS as physical NICs and are treated as such when configuring them. Depending on the NIC vendor, the teaming software might offer several different types of failover, including simple active/standby, static aggregation, dynamic aggregation (LACP), and vendor-specific load balancing schemes.

For Linux-based systems, the bonding module is used to implement NIC teaming. There are a number of bonding modes available, most commonly mode 1 (active/standby) and mode 4 (LACP aggregation). Like Windows teaming, Linux bonding also offers logical interfaces to the OS that can be used as wanted. Unlike Windows teaming, VLAN tagging is controlled by different software, and can create sub-interfaces for VLANs from both physical and logical entities, for example, `eth0.10` for VLAN 10 on physical `eth0`, or `bond0.20` for VLAN 20 on a logical NIC bond pair 0.

Like Linux, VMware ESX also has built-in teaming in the form of assigning multiple NICs to a common vSwitch (a logical switch that runs within an ESX host, which is shared by the VMs that require network access). VMware has several teaming modes, with the route-based default on the originating virtual port ID. This default mode provides a per VM load balance of physical NICs assigned to the vSwitch.

The teaming method that is best for a specific environment is unique to each situation. However, these common elements might help in the decision-making process:

- ▶ Do not select a mode that requires some form of aggregation (static or LACP) on the switch side unless the NICs in the team go to the same physical switch or logical switch created by a technology, such as virtual link aggregation or stacking.
- ▶ If using a mode that uses some form of aggregation, you must also perform proper configuration on the upstream switches to complete the aggregation on that side.
- ▶ The most stable solution is often active/standby, but this solution has the disadvantage of losing any bandwidth on a NIC that is in standby mode.
- ▶ Most teaming software offers proprietary forms of load balancing. The selection of these modes must be thoroughly tested for suitability to the task for an environment.
- ▶ Most teaming software incorporates the concept of *auto fallback*, which means that if a NIC went down and then came back up, it automatically fails back to the original NIC. Although this function helps ensure good load balancing, each time that a NIC fails, some small packet loss might occur, which can lead to unexpected instabilities. A flapping link occurs when a severe disruption to the network connection of the servers causes the link to flap back and forth. One way to mitigate this circumstance is to disable the auto fallback feature. After a NIC fails, the traffic falls back only in the event that the original link is restored and something happened to the current link that requires a switchover.

It is your responsibility to understand your goals and the tools available to achieve those goals. NIC teaming is one tool for users that need high availability connections for their compute nodes.

3.4.5 Trunk failover

Trunk failover, also known as *failover* or *link state tracking*, is an important feature for ensuring high availability in chassis-based computing. This feature is used with NIC teaming to ensure the compute nodes can detect an uplink failure from the I/O modules.

With traditional NIC teaming and bonding, the decision process used by the teaming software to use a NIC is based on whether the link to the NIC is up or down. In a chassis-based environment, the link between the NIC and the internal I/O module rarely goes down unexpectedly. Instead, a more common occurrence might be the uplinks from the I/O module go down; for example, an upstream switch failed or cables were disconnected. In this situation, although the I/O module no longer has a path to send packets because of the upstream fault, the actual link to the internal server NIC is still up. The server might continue to send traffic to this unusable I/O module, leading to a black hole condition.

To prevent this black hole condition and to ensure continued connection to the upstream network, trunk failover can be configured on the I/O modules. Depending on the configuration, trunk failover monitors a set of uplinks. If these uplinks go down, trunk failover takes down the configured server-facing links. This action alerts the server that this path is not available, and NIC teaming can take over and redirect traffic to the other NIC.

This feature is shown in detail in Figure 3-6 on page 58.

Trunk failover offers these features:

- ▶ Besides triggering on link up/down, trunk failover also operates on the Spanning-Tree blocking state. From a data packet perspective, a blocked link is no better than a down link.
- ▶ Trunk failover can be configured to fail over if the number of links in a monitored aggregation falls below a certain number.
- ▶ Trunk failover can be configured to trigger on VLAN failure.
- ▶ When a monitored uplink comes back up, trunk failover automatically brings back up the downstream links if Spanning Tree is not blocking and other attributes, such as the minimum number of links are met for the trigger.
- ▶ For trunk failover to work properly, it is assumed that there is an L2 path between the uplinks, external to the chassis. This path is most commonly found at the switches just above the chassis level in the design (but they can be higher) if there is an external L2 path between the Enterprise Chassis I/O modules.

Important: Other solutions to detect an indirect path failure were created, such as the VMware beacon probing. Although these solutions offer advantages, trunk failover is the simplest and most nonintrusive way to provide this functionality. We encourage you to use trunk failover whenever NIC teaming is configured on the compute nodes.

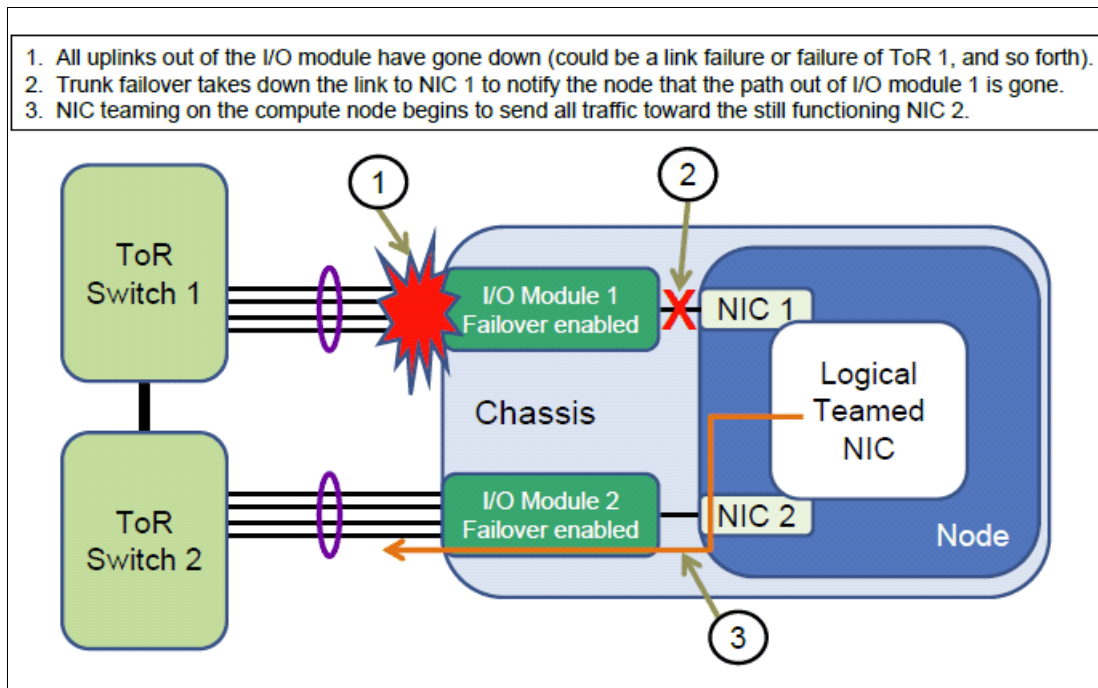


Figure 3-6 Trunk failover in action

Using trunk failover with NIC teaming is a critical element for nodes requiring a highly available path from the Enterprise Chassis.

3.4.6 Virtual Router Redundancy Protocol

Rather than having every server make its own routing decisions (not scalable), most servers implement a default gateway. In this configuration, if the server sends a packet to a device on a subnet that is not the same as its own, the server sends the packets to a *default gateway* and lets the default gateway determine where to send the packets.

If this default gateway is a stand-alone router and it goes down, the servers that point their default gateway setting at the router cannot route off their own subnet.

To prevent this type of single point of failure, most data center routers that offer a default gateway service implement a redundancy protocol so that one router can take over for the other when one router fails.

Although there are nonstandard solutions to this issue, for example, Hot Standby Router Protocol (HSRP), most routers now implement standards-based Virtual Router Redundancy Protocol (VRRP).

Important: Although they offer similar services, HSRP and VRRP are not compatible.

In its simplest form, two routers that run VRRP share a common IP address (called the *Virtual IP address*). One router traditionally acts as master and the other as a backup in the event the master goes down. Information is constantly exchanged between the routers to ensure that one can provide the services of the default gateway to the devices that point at its virtual IP address. Servers requiring a default gateway service point the default gateway service at the virtual IP address, and redundancy is provided by the pair of routers that run VRRP.

Both the EN2092 1Gb Ethernet Switch and the EN4093/EN4093R 10Gb Scalable Switch offer support for VRRP directly within the Enterprise Chassis. Most common data center designs place this function in the routing devices above the chassis (or even higher). The design depends on how important it is in the L2 adjacencies requirements to have a large, common L2 network between nodes. This function can be moved within the Enterprise Chassis as networking requirements dictate.

3.5 Virtual Fabric vNIC solution capabilities

Virtual network interface controller (vNIC) is a way to partition a physical NIC into multiple logical NICs so that the OS has more ways to logically connect to the infrastructure. The vNIC feature is supported only on 10-Gb ports on the EN4093/EN4093R 10Gb Scalable Switch facing the compute nodes within the chassis. It requires an adapter in compute node that also supports this functionality (CN4054 10Gb Virtual Fabric Adapter).

As of this writing, there are two primary forms of vNIC available: IBM Virtual Fabric mode (or Switch dependent mode) and Switch independent mode. The Virtual Fabric mode also is subdivided into two submodes: Dedicated uplink vNIC mode and shared uplink vNIC mode.

All vNIC modes share these common elements:

- ▶ They are supported only on 10-Gb connections.
- ▶ Each vNIC mode allows a 10-Gb physical NIC port to be divided into up to four vNICs. This means that you can configure up to eight vNICs on a NIC with two physical 10-Gb ports.
- ▶ They all require an adapter that has support for one or more of the vNIC modes.
- ▶ When creating vNICs, the default bandwidth is 2.5 Gbps for each vNIC, but can be configured to be anywhere from 100 Mbps up to the full bandwidth of the NIC.
- ▶ The cumulative bandwidth of all configured vNICs on a physical NIC port cannot exceed 10 Gbps.

A summary of some of the differences and similarities of these modes is shown in Table 3-6. These differences and similarities are covered next.

Table 3-6 Attributes of vNIC modes

Capability	IBM Virtual Fabric mode		Switch independent mode
	Dedicated uplink	Shared uplink	
Requires support in the I/O module	Yes	Yes	No
Requires support in the NIC	Yes	Yes	Yes
Supports adapter transmit rate control	Yes	Yes	Yes
Support I/O module transmit rate control	Yes	Yes	No
Supports changing rate when needed	Yes	Yes	No
Requires a dedicated uplink per vNIC group	Yes	No	No
Support for node OS-based tagging	Yes	No	Yes
Support for failover per vNIC group	Yes	Yes	No
Support for more than one uplink per vNIC group	No	Yes	Yes

3.5.1 Virtual Fabric mode vNIC

IBM Virtual Fabric mode or switch-dependent mode depends on the switch module to participate in the vNIC process. Specifically, the I/O module that supports this mode in the Enterprise Chassis is the IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch. It requires an adapter on the node that also supports the vNIC switch-dependent mode feature.

In switch-dependent vNIC mode, configuration is performed on the switch itself and the configuration information is communicated between the switch and the adapter so that both sides agree on and enforce bandwidth controls. The mode can be changed to different speeds at any time without reloading either the OS or the I/O module.

There are two types of switch-dependent vNIC mode: Dedicated uplink mode and shared uplink mode. Both modes incorporate the concept of a vNIC group on the switch that is used to associate vNICs and physical ports into virtual switches within the chassis. How these vNIC groups are used is the primary difference between dedicated uplink mode and shared uplink mode.

Switch-dependent vNIC modes share these common attributes:

- ▶ They conceptually are a vNIC group that must be created on the I/O module.
- ▶ Similar vNICs are bundled together into common vNIC groups.
- ▶ Each vNIC group is treated as a virtual switch within the I/O module. Packets in one vNIC group can get only to a different vNIC group by going to an external switch or router.
- ▶ For the purposes of Spanning Tree and packet flow, each vNIC group is treated as a unique switch by upstream connecting switches and routers.
- ▶ Both modes support the addition of physical NICs (pNICs, the NICs from nodes not using vNIC) to vNIC groups for internal communication to other pNICs and vNICs in that vNIC group, and share any uplink associated with that vNIC group.

Dedicated uplink mode

In dedicated uplink mode, each vNIC group must have its own dedicated physical or logical (aggregation) uplink. In this mode, no more than one physical or logical uplink to a vNIC group can be assigned and it is assumed that high availability is achieved by some combination of aggregation on the uplink or NIC teaming on the server.

In dedicated uplink mode, vNIC groups are VLAN agnostic to the nodes and the rest of the network, which means that you do not need to create VLANs for each VLAN used by the nodes. The vNIC group takes each packet (tagged or untagged) and moves it through the switch.

This mode is accomplished by the use of a form of Q-in-Q tagging. Each vNIC group is assigned a VLAN that is unique to each vNIC group. Any packet (tagged or untagged) that comes in on a downstream or upstream port in that vNIC group has a tag placed on it equal to the vNIC group VLAN. As that packet leaves the vNIC into the node or out an uplink, that tag is removed and the original tag (or no tag, depending on the original packet) is revealed.

Shared uplink mode

Shared uplink mode allows the switch module to share uplinks among vNIC groups, thus reducing the number of uplinks required.

It also changes the way that the vNIC groups process packets for tagging. In shared uplink mode, it is expected that the servers will no longer use tags. Instead, the vNIC group VLAN acts as the tag that is placed on the packet.

Important: At the time of writing, EN4093/EN4093R switch modules support only dedicated uplink mode. Shared uplink mode is planned to be supported in a future firmware release.

3.5.2 Switch-independent mode vNIC

Switch-independent mode vNIC is enabled and configured only on the node itself, and the switch module is unaware of this virtualization. Because of this, switch-independent mode also works with EN4091 Ethernet Pass-thru Module and a TOR switch.

This mode is enabled on the node directly, and has similar rules as dedicated vNIC mode regarding how you can divide the vNIC. But any bandwidth settings are applicable only to how the node sends traffic, not how the switch module sends traffic back to the node. Also, the bandwidth settings cannot be changed in real time because they require a compute node restart to change.

The mode that is best suited for a user depends on the user's requirements. IBM Virtual Fabric mode offers the most control, and switch-independent mode supports more connectivity options (in addition to the EN4093 Ethernet switch, it also works with EN4091 Ethernet Pass-thru Module and a TOR switch).

3.6 Management

The Enterprise Chassis is managed as an integrated solution. It also offers the ability to manage each element as an individual product.

From an I/O module perspective, the Ethernet switch modules can be managed through the IBM Flex System Manager (FSM), an integrated management appliance for all IBM Flex System solution components.

Network Control, a component of FSM, provides advanced network management functions for IBM Flex System Enterprise Chassis network devices. The following functions are included in network control:

- ▶ Discovery
- ▶ Inventory
- ▶ Network topology
- ▶ Health and status monitoring
- ▶ Configuring network devices

Network Control is a preinstalled plug-in that builds on base management software capabilities. This build is done by integrating the launch of vendor-based device management tools, topology views of network connectivity, and subnet-based views of servers and network devices.

Network Control offers the following network management capabilities:

- ▶ Discover network devices in your environment.
- ▶ Review network device inventory in tables or a network topology view.

- ▶ Monitor the health and status of network devices.
- ▶ Manage devices by groups: Ethernet switches, Fibre Channel over Ethernet, or subnet.
- ▶ View network device configuration settings and apply templates to configure devices, including Converged Enhanced Ethernet quality of service (QoS), VLANs, and Link Layer Discovery Protocol (LLDP).
- ▶ View systems according to VLAN and subnet.
- ▶ Run network diagnostic tools, such as ping and traceroute.
- ▶ Create logical network profiles to quickly establish VLAN connectivity.
- ▶ Simplify VM connections management by configuring multiple characteristics of a network when virtual machines are part of a network system pool.
- ▶ With management software VMControl, maintain network state (VLANs and ACLs) as a virtual machine is migrated (kernel-based virtual machine (KVM)).
- ▶ Manage virtual switches, including virtual Ethernet bridges.
- ▶ Configure port profiles, a collection of network settings associated with a virtual system.
- ▶ Automatically configure devices in network systems pools.

Ethernet I/O modules can also be managed by the command-line interface (CLI), web interface, IBM System Network Element Manager (SNEM), or a third-party Simple Network Management Protocol-based (SNMP-based) management tool.

Both the EN4093/EN4093R 10Gb Scalable Switch and the EN2092 1Gb Ethernet Switch modules offer two CLI options (because it is a non-managed device, the EN4091 Ethernet Pass-thru Module has no user interface). The default CLI for the Ethernet switch modules is the IBM Networking OS CLI, which is a menu-driven interface. A user can also enable an optional CLI known as Industry Standard CLI (ISCLI) that closely resembles Cisco IOS CLI.

For more information about how to configure various features and the operation of the various user interfaces, IBM Flex Systems Information Center contains *Application and Command Reference* guides for Ethernet switch modules at the following web page:

<http://publib.boulder.ibm.com/infocenter/flexsys/information/index.jsp>

EN4093 documentation is available at the following web page:

http://publib.boulder.ibm.com/infocenter/flexsys/information/topic/com.ibm.acc.net.workdevices.doc/Io_module_compassplus.html?resultof=%22%65%6e%34%30%39%33%22%20

EN2092 documentation is available on this web page:

http://publib.boulder.ibm.com/infocenter/flexsys/information/topic/com.ibm.acc.net.workdevices.doc/Io_module_pollux.html

3.6.1 Management tools and their capabilities

The various user interfaces available for the I/O modules, whether the CLI or the web-based GUI, offer the ability to fully configure and manage all features available to the switches. Some elements of the modules can also be configured from the Chassis Management Module (CMM) user interface.

The best tool for you often depends on your experience with different interfaces and their knowledge of networking features. Most commonly, the CLI is used by those who work with networks as part of their day-to-day jobs. The CLI offers the quickest way to accomplish tasks, such as scripting an entire configuration. The downside to the CLI is that it tends to be more

cryptic to those that do not use it daily. For those users that do not need the power of the CLI, the web-based GUI allows the configuration and management of all switch features.

IBM System Networking Element Manager

Aside from the tools that run directly on the modules, IBM also offers System Networking Element Manager (SNEM), a tool that provides the following functions:

- ▶ Improved network visibility
- ▶ Increased availability and performance with correlation, event de-duplication, automated diagnostics, and root-cause analysis
- ▶ Simplified management of large groups of switches with automatic discovery of switches in the network
- ▶ Automated and integrated management, deployment, and monitoring
- ▶ SNMP-based configuration and management
- ▶ Support of network policies for virtualization
- ▶ Authentication and authorization
- ▶ Real-time root cause analysis and problem resolution
- ▶ Integration with IBM Systems Director and VMware Virtual Center and vSphere clients

For more information about SNEM, see the IBM System Networking Switch Center website:

<http://www.ibm.com/systems/networking/software/snem>

Any third-party management platforms that support SNMP can be used to configure and manage the modules as well.

IBM Fabric Manager

By using IBM Fabric Manager, you can quickly replace and recover compute nodes in your environment.

Fabric Manager assigns Ethernet Media Access Control (MAC), Fibre Channel worldwide name (WWN), and serial-attached SCSI (SAS) WWN addresses to compute node bays so that any compute nodes plugged into those bays take on the assigned addresses. These assignments enable the Ethernet and Fibre Channel infrastructure to be configured once and before any compute nodes are connected to the chassis.

With Fabric Manager, you can monitor the health of compute nodes and automatically, without user intervention, replace a failed compute node from a designated pool of spare compute nodes. After receiving a failure alert, Fabric Manager attempts to power off the failing compute node, read the Fabric Manager virtualized addresses and boot target parameters, apply these parameters to the next compute node in the standby pool, and power on the standby compute node.

You can also pre-assign MAC and WWN addresses and storage boot targets for up to 256 chassis or 3584 compute nodes. By using an enhanced GUI, you can create addresses for compute nodes and save the address profiles. You can then deploy the addresses to the bays in the same chassis or in up to 256 different chassis without any compute nodes installed in the chassis. Additionally, you can create profiles for chassis that are not installed in the environment by associating an IP address to the future chassis.

Fabric Manager is available as an FoD through the IBM Flex System Manager management software.



Initial configuration

In this chapter, we describe the steps to be performed for the hardware installation and initial configuration of the IBM Flex System Ethernet I/O modules (EN2092 1-Gb Ethernet Scalable Switch and EN4093/EN4093R 10-Gb Scalable Switch).

4.1 Initial hardware installation

This section provides instructions for installing a switch in the IBM Flex System chassis. See the documentation for your Flex System chassis for information about I/O bay locations and the components that can be installed in them that are specific to your Flex System chassis type.

Note: For detailed information and documentation for your IBM Flex System chassis, see the following website:

<http://www-03.ibm.com/systems/flex/chassis/>

You can install up to four I/O modules in the Flex System chassis, including Ethernet switches, Fibre Channel switches, InfiniBand, and pass-thru modules.

Figure 4-1 shows an example of a Flex System chassis (rear-view) with the I/O bays identified.

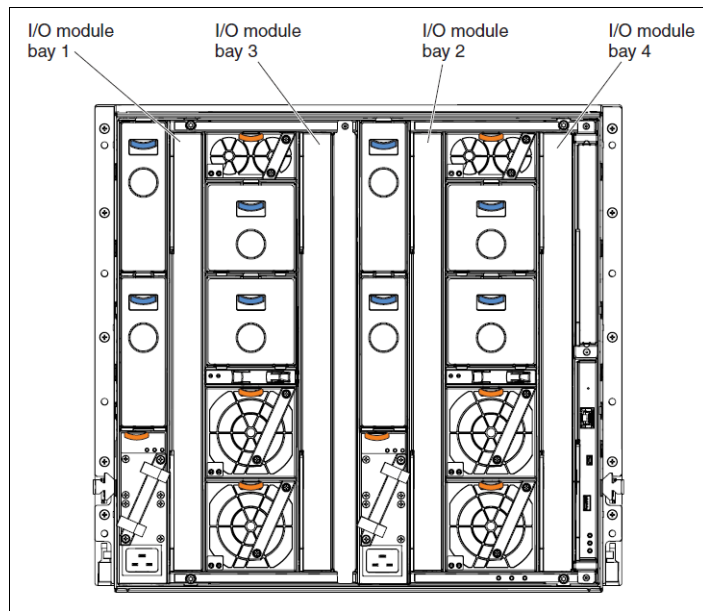


Figure 4-1 IBM Flex System chassis rear-view

An IBM Flex System network adapter must be installed in each compute node with which you want to communicate. To enable the switch to communicate with a compute node, at least one switch must be installed in the IBM Flex System chassis.

Installing a second switch enables a redundant path and a separate connection from the compute node to the external Ethernet network:

- ▶ The IBM Flex System chassis supports a maximum of four IBM Flex System EN4093/EN4093R 10Gb Scalable Switches Modules. The IBM Flex System chassis supports a maximum of 28 network adapters.
- ▶ The IBM Flex System chassis supports a maximum of four IBM Flex System EN2092 1Gb Ethernet Scalable Switch Modules. Depending on the type of IBM Flex System chassis that you are using, the IBM Flex System chassis supports a maximum of 10 or 14 network adapters.

Note: When an IBM Flex System EN4093/EN4093R 10Gb Scalable Switches Module is installed in a IBM Flex System chassis, the internal ports operate at 10 Gbps. The external ports can operate at 10 Gbps or 1 Gbps, depending on the SFP module type.

4.1.1 Installing a switch

Use the following instructions to install a switch in the IBM Flex System chassis. You can install a switch while the IBM Flex System chassis is powered on. For redundancy support, you must install I/O modules of the same type in I/O bays 1 and 2, and I/O modules of the same type in bays 3 and 4 of the chassis.

To install a switch, complete the following steps:

1. Verify that the switch is compatible with the chassis. For a list of supported optional devices for the IBM Flex System chassis see the following website:
<http://www-03.ibm.com/systems/info/x86servers/serverproven/compat/us/flex.html>
2. Select the I/O bay in which to install the switch, remove the filler module from the selected bay (store the filler module for future use).
3. Remove the switch from its static-protective package and ensure that the release handles on the switch are in the open position (perpendicular to the switch as shown in Figure 4-2). Then, slide the switch into the applicable I/O bay until it stops.

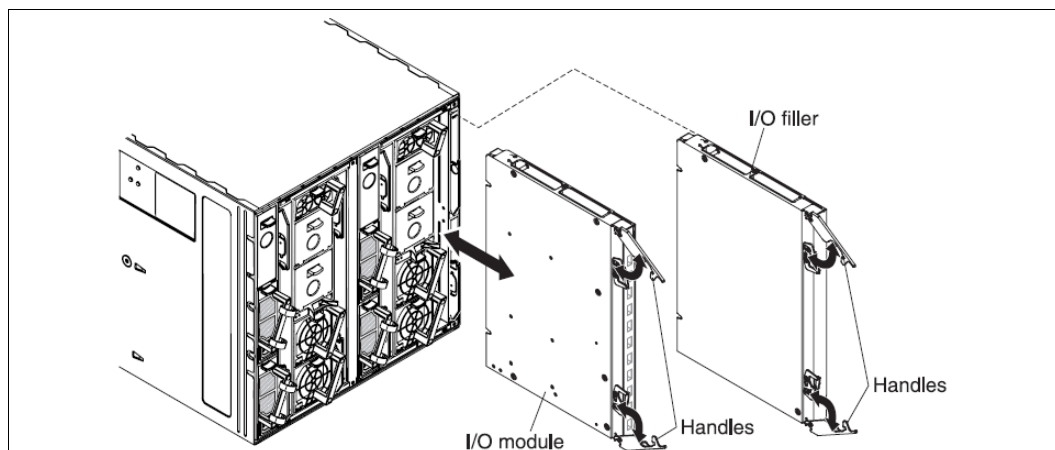


Figure 4-2 Switch installation

4. Push both release handles on the front of the switch to the closed position. After you insert and lock the switch, it is turned on, and a power-on self-test (POST) occurs to verify that the switch is operating correctly (takes approximately 100 seconds to complete the POST).
5. When POST has successfully completed, the Power LED remains on and the Error LED is off. See “LEDs on EN4093/EN4093R” on page 190.
6. Install the enhanced small form-factor pluggable (SFP+) or quad small form-factor pluggable (QSFP+) modules in the switch and attach any cables that are required by the switch.
7. Ensure that the external ports on the switch are enabled through one of the Chassis Management Module interfaces, such as the web-based interface or the command-line interface (CLI).

Note: For more information about installation guidelines, system reliability guidelines, and handling static-sensitive devices, see Chapter 2 of *IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch Users Guide*:

<http://publib.boulder.ibm.com/infocenter/flexsys/information/topic/com.ibm.acc.networkdevices.doc/8ly1206.pdf>

Installing SFP+ and QSFP+ modules

The EN4093/EN4093R switch supports the 10 Gb SFP+ module, the 1 Gb small form-factor pluggable (SFP) module, and a QSFP+ module. The SFP+, SFP, and QSFP+ modules are laser products that convert electrical signals to optical signals. It also supports multi-source agreement (MSA)-compliant copper direct-attach cables (DAC), up to 5 m (23 ft.) in length.

To install an SFP+/QSFP+ module, complete the following steps:

1. Remove the SFP+/QSFP+ module from its static-protective package and then remove the protective cap.
2. Insert the SFP+/QSFP+ module into the SFP+ or QSFP+ module port until it clicks into place. Use minimal pressure when you insert the module into the port. Do not use excessive force when you insert the module; you can damage the module or the SFP+/QSFP+ port.
3. Connect the fiber optic cable.

Note: To avoid damage to the cable or the SFP+/QSFP+ module, ensure that you do not connect the fiber optic cable before you install the SFP+ module.

For more information about installing and removing SFP+ and QSFP+, see the following website:

<http://publib.boulder.ibm.com/infocenter/flexsys/information/topic/com.ibm.acc.networkdevices.doc/88y7927.pdf>

4.2 Initial Software Configuration

In this section, we describe the steps to be performed for the initial configuration of the IBM Flex System Ethernet I/O modules (EN2092 1Gb Ethernet Scalable Switch and EN4093/EN4093R 10Gb Scalable Switch).

Every time that you receive a new switch and you want to install it in your network, there is a set of basic initial configuration tasks that should be done. These include setting the date and time, changing the administrator's password, and some other basic ones. And, to perform these tasks, you first connect to the switch. In the following sections, we describe how to perform this connection and these tasks.

4.2.1 Administration interfaces

IBM Flex System Ethernet I/O modules have many management interfaces where you can perform the initial setup tasks. In this section, we cover the common methods that provide IBM Networking OS, and also the ones that are specific to the Flex System Ethernet modules.

IBM Networking OS, the operating system that runs inside the switches, provides three main interfaces for administration purposes:

- ▶ A text-based CLI
- ▶ The Browser-Based Interface (BBI), which can be used with a standard web browser
- ▶ Simple Network Management Protocol (SNMP) support for access with network management software, such as IBM Systems Director and many others

For the IBM Flex System Ethernet I/O modules, one additional interface can be used: The Flex Chassis Management Module (CMM) interface, which is used also for general management of the chassis.

Console, Telnet, and Secure Shell

The IBM Networking OS CLI provides a simple and direct method for switch administration. Using a basic terminal, you have an organized hierarchy of menus, each with logically related submenus and commands. You can use these items to view detailed information and statistics about the switch, and to perform any necessary configuration and switch software maintenance.

You can access the CLI in any one of the following ways:

- ▶ Using a Telnet connection over the network
- ▶ Using a Secure Shell (SSH) connection over the network
- ▶ Using a serial connection via the serial port

The factory default settings allow initial switch administration through the built-in serial port and forms the CMM of the Flex Chassis.

Browser-Based interface

The web interface, also called the *Browser-Based Interface (BBI)*, for the switches is available on the IP address of the management port of the switch. To open the BBI, open your web browser and point to the IP address of the management port. By default, HTTPS access is enabled, so you need to specify it in the web browser navigation bar. Then, you can enable HTTP access, but it is not advised because it is an unsecured connection.

In our example, one of our switches has the IP address 172.25.101.238. This switch is the IBM Flex System Ethernet I/O module embedded switch in the Flex System chassis.

To access the BBI, enter the switch IP interface address in the web browser's URL field and specify to use HTTPS, in our case: `https://172.25.101.238`. As shown in Figure 4-3, you are prompted to log in. The default login credentials are user `admin` and password `admin`.

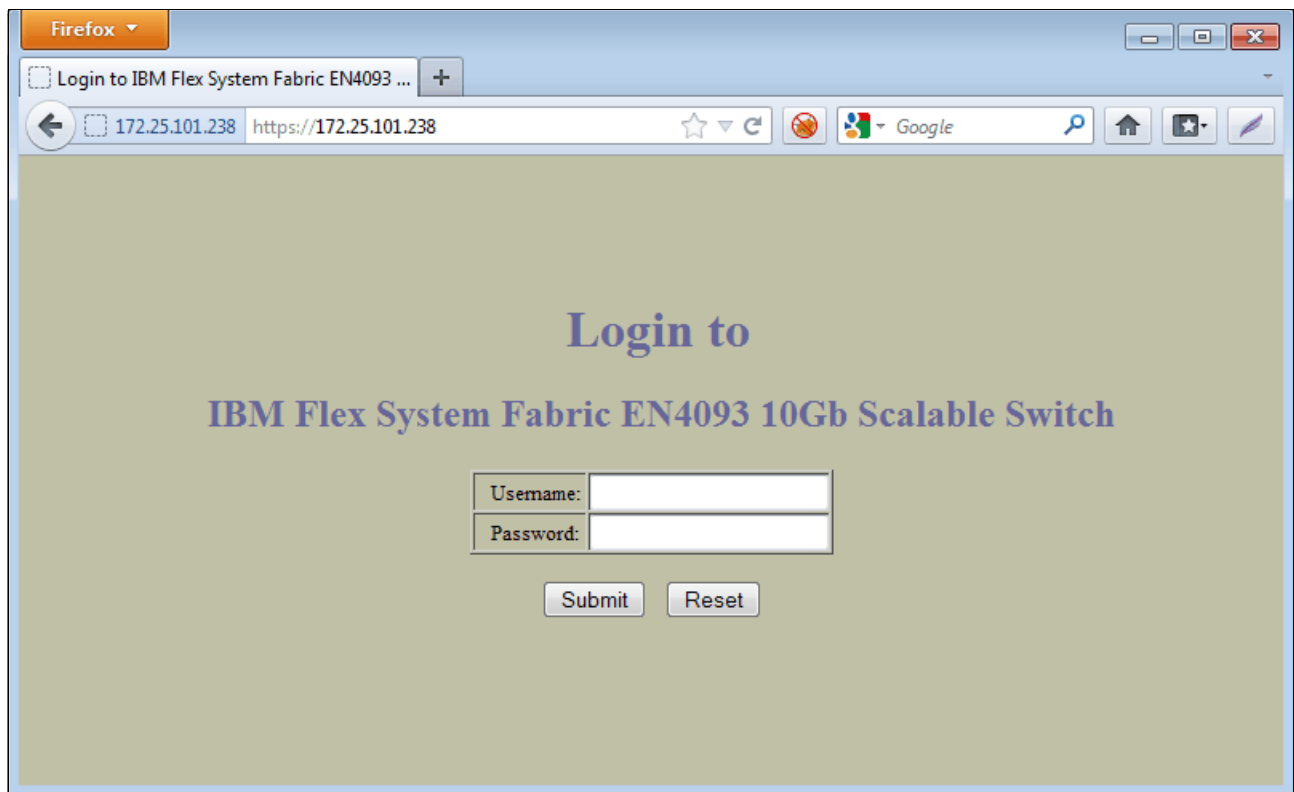


Figure 4-3 IBM Flex System Ethernet module Browser-Based-Interface

After successfully logging in, the Switch Dashboard is displayed, as shown in Figure 4-4 on page 71.

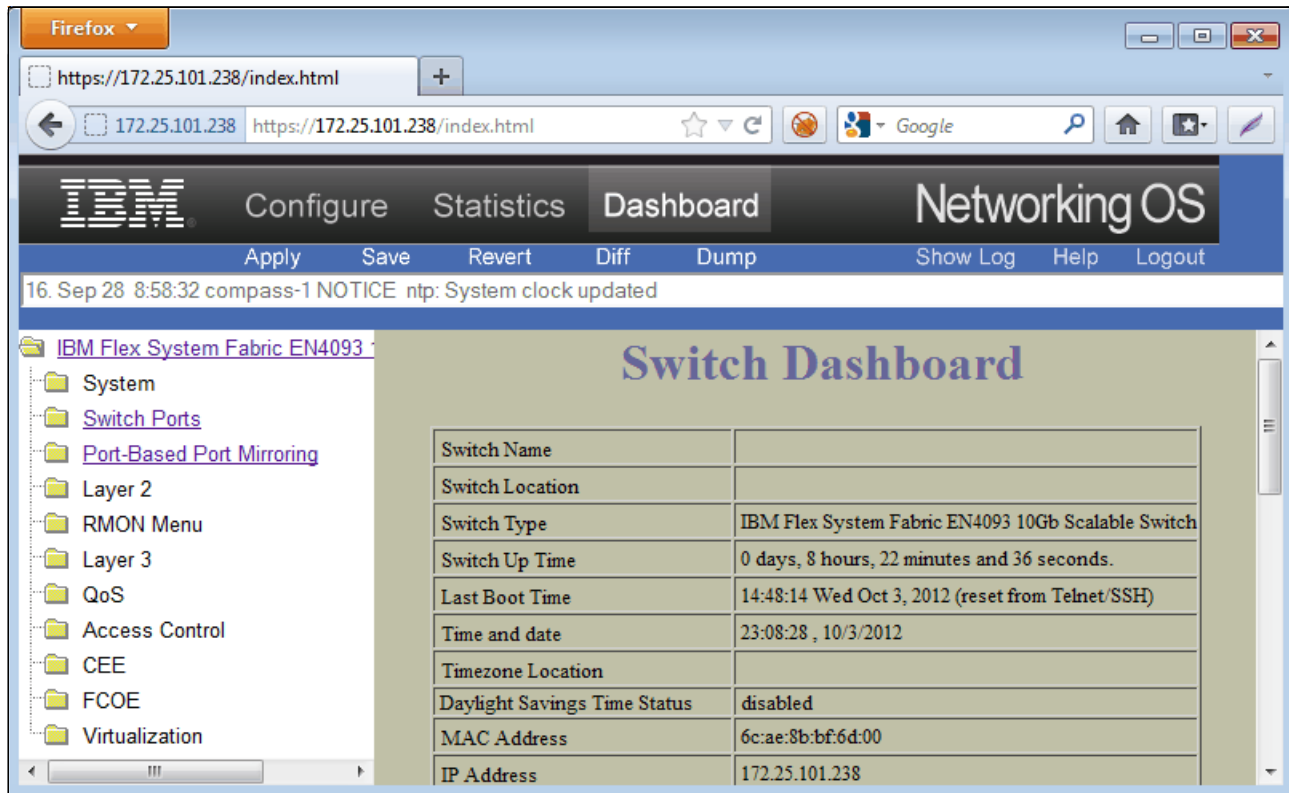


Figure 4-4 IBM Flex System Ethernet module Dashboard

For more information about the web interface, see the following documents:

- ▶ BBI Quick Guide for the IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch:
<http://publib.boulder.ibm.com/infocenter/flexsys/information/topic/com.ibm.acc.networkdevices.doc/88y7944.pdf>
- ▶ BBI Quick Guide for the EN2092 1Gb Ethernet Scalable Switch:
<http://publib.boulder.ibm.com/infocenter/flexsys/information/topic/com.ibm.acc.networkdevices.doc/88y7949.pdf>

During the first boot of your switch, you configure the basic options needed to continue working with the switch, such as the IP address for the management interface, gateways, and some other options.

4.2.2 First boot of the IBM Flex System Ethernet I/O modules

In the IBM Flex System Ethernet switches embedded in the Flex System chassis, the initial boot and setup are different than other IBM System Networking switches such as the Top-of-Rack (ToR) switches. The management interfaces are accessible from the Chassis Management Module (CMM) of the Flex chassis. First, log in to the CMM and then configure the I/O modules that correspond to the switches that you want to configure. During this section, we describe this process and use the web interface.

I/O Modules IP Configuration

Log on to the CMM by using your user name and password. When in the web interface, expand **Chassis Management** and click **I/O Modules**. Then, you see a list with all the

switches' modules that are connected to the chassis. In this list, you can identify the switch that you want to work on, as shown in Figure 4-5.

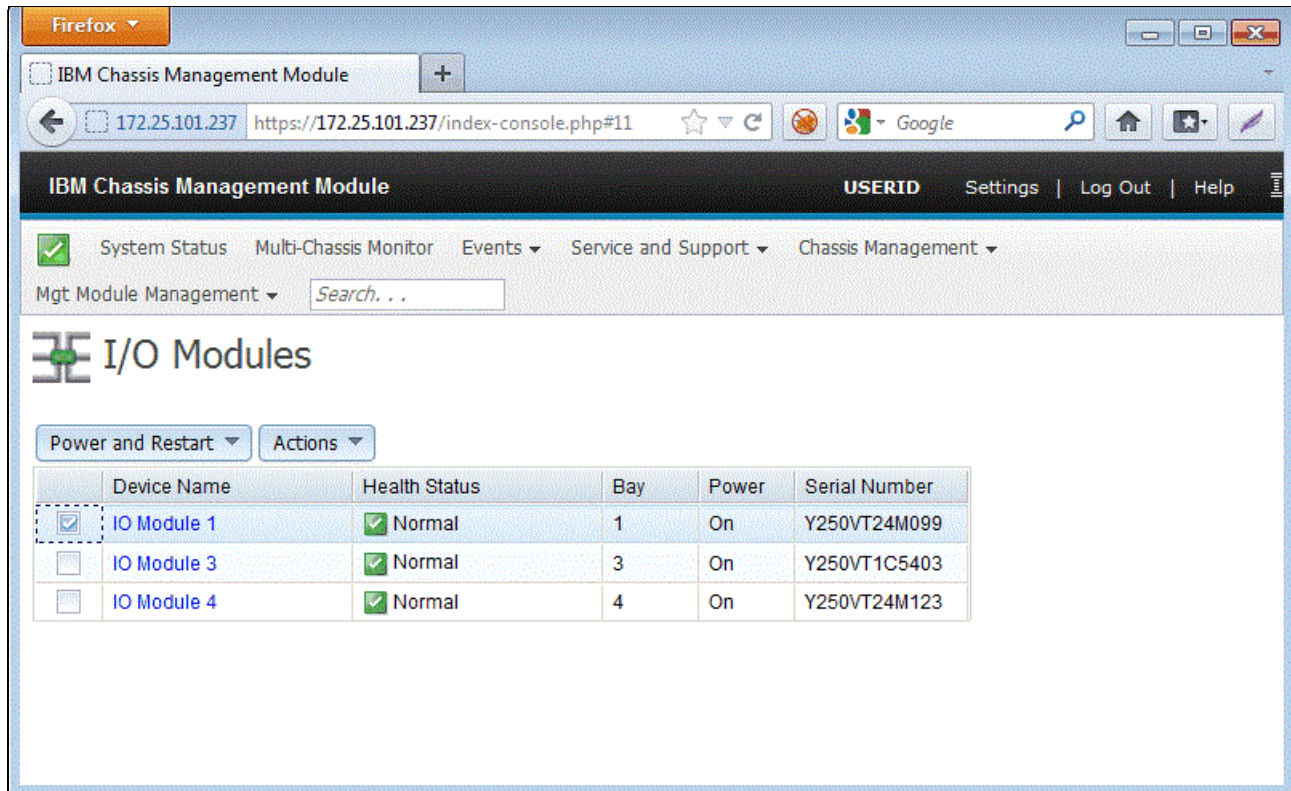


Figure 4-5 I/O module switches

Here, you can perform some tasks and access more detailed information about the switch, such as the following:

- ▶ POST results (to view the start messages)
- ▶ Power on/off
- ▶ Restore factory defaults
- ▶ Send **ping** requests (to check that the switch receives **ping** commands)
- ▶ Start a BBI session (to manage the switch)

In this book, we focus only on the options that are relevant to our topic. For more information about the IBM Flex System Chassis Management Module, see the following PDF:

http://publib.boulder.ibm.com/infocenter/flexsys/information/topic/com.ibm.acc.cmm.doc/dw1kt_cmm_ug_pdf.pdf

To configure the IP address, the subnet mask, and the gateway of the management port of the embedded switches, expand **Chassis Management** and click **Component IP Configuration**. You see a list with all of the switches connected into the different bays, and if you click one of these switches, the window shown in Figure 4-6 opens. Select the **IPv4** pane and then set the **Configuration Method** to **Use Static IP Address**. Here, you can configure the IP address, subnet mask, and gateway address of the management port.

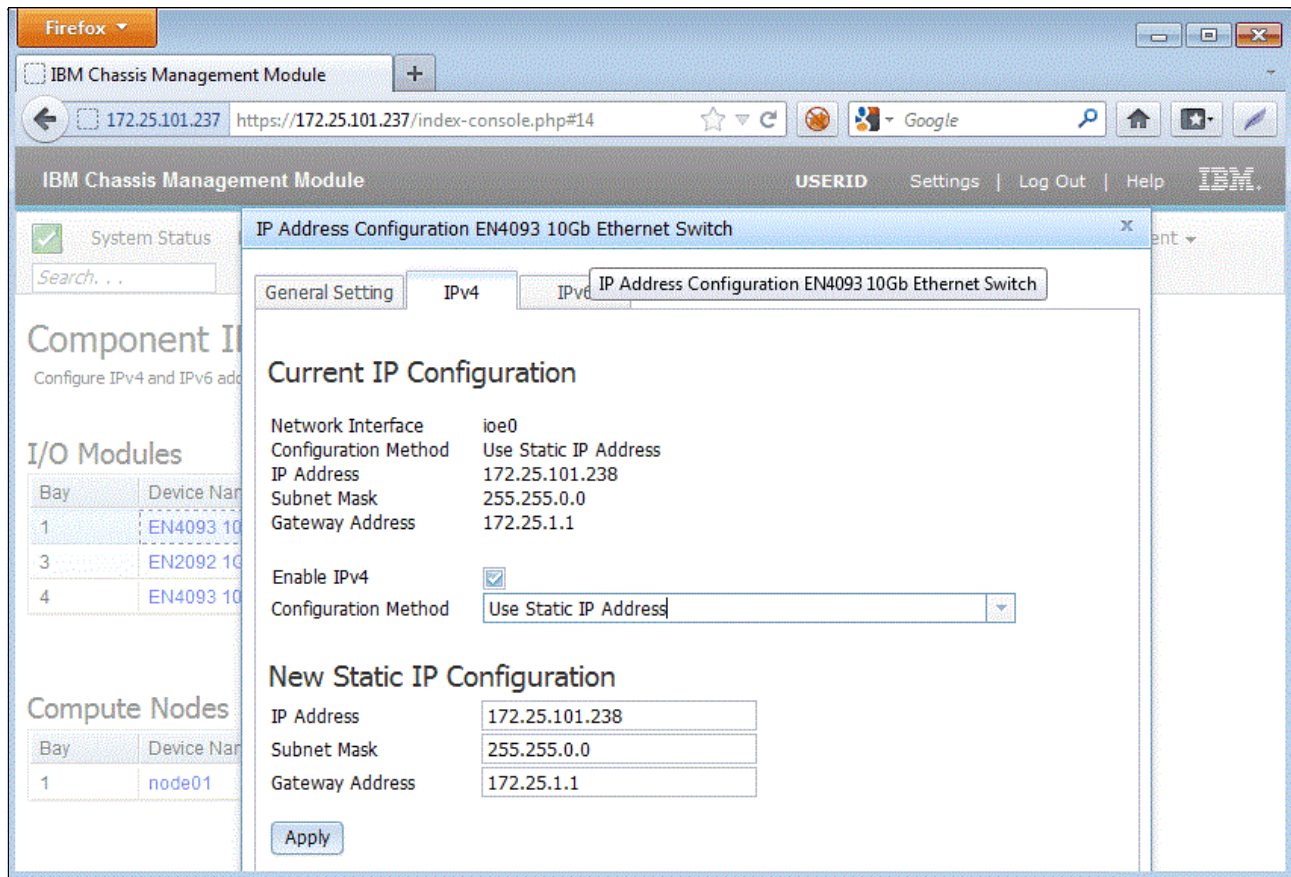


Figure 4-6 Switch management port IP configuration

After the IP address configuration is done, you should be able to ping the device and connect to it through SSH or HTTPS.

Note: By default, Telnet and HTTP access is disabled because they are unsecured connections. To use them, first you must enable them.

Note: By default, the CMM assigns an IPv4 address of 192.168.70.1xx to each I/O module installed, where xx is also based on the number of the bay into which each I/O module (EN4093/EN4093R, EN2092, EN4091) is installed. For example, the switch installed in Bay 1, by default, will have the 192.168.70.120 management IP address.

4.2.3 Connecting to the switch

After your switch is configured to have an IP address visible from your network, you can work with the switch remotely by using any SSH client. Start an SSH session and connect to the IP address previously configured.

If it is the first time that you connect to the switch using SSH, you must exchange encryption keys with the switch to establish the connection. Your SSH client handles this exchange automatically. In our case, the SSH client displayed a window to confirm the key exchange, as shown in Figure 4-7.

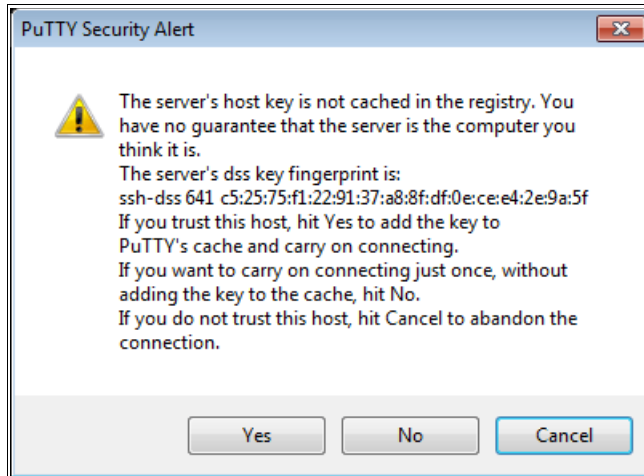


Figure 4-7 SSH key exchange message

When you first log on to the switch, and if there is no other user logged in to the switch, you are prompted to choose the CLI that you would like to use. IBM switches support two kinds of CLI:

- ▶ IBM Networking OS, which is the default
- ▶ Industry Standard CLI (ISCLI)

You will see a prompt similar to the one in Figure 4-8 on page 75. Choose the CLI that you want to use.

Note: If there are other users logged in, you are not prompted to choose the CLI to use. You must use the CLI that was chosen by the first logged in user.

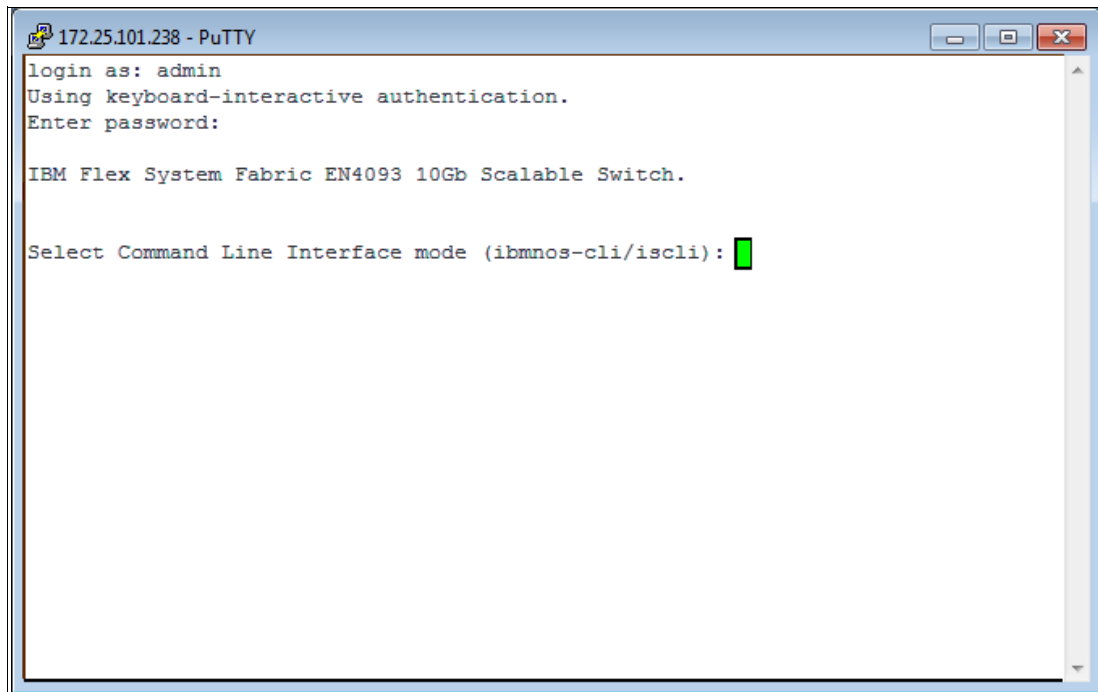


Figure 4-8 CLI selection

Note: For the illustrated examples in this chapter and in the following chapters, we use the ISCLI because it is the most used by network administrators.

When you are logged in to the system, you can continue with the initial configuration or setup tool.

The Industry Standard CLI has three command modes, each with access to different command sets:

- | | |
|---------------------------|--|
| User mode | This is the first mode that a user has access to after logging in to the switch. The user mode can be identified by the > prompt following the switch name. This mode allows the user to run only the basic commands, such as those that show the system's status. The system cannot be configured or restarted from this mode. |
| Privileged mode | This mode allows users to view the system configuration, restart the system, and enter configuration mode. It also allows all the commands that are available in user mode. Privileged mode can be identified by the # prompt following the router name. The user mode enable command tells the operating system that the user wants to enter privileged mode. If an enable password has been set, the user must enter the correct password to be granted access to privileged mode. Privileged mode allows the user to do anything on the switch. To exit privileged mode, the user runs the disable command. |
| Configuration mode | This mode allows users to modify the running system configuration. To enter configuration mode, enter the configure terminal command from privileged mode. Configuration mode has various submodes, starting with global configuration mode, which can be identified by the (config)# prompt following the switch name. As the configuration mode submodes change depending on what is being configured, the |

words inside the parentheses change. For example, when you enter interface configuration submode, the prompt changes to (config-if)# following the switch name. To exit configuration mode, the user can enter **end** or press Ctrl-Z.

4.2.4 Setup Tool

The IBM Networking OS includes a setup utility to make the initial configuration of your switch. The setup utility prompts you to enter all the necessary information for the basic configuration of the switch. Whenever you log on as the system administrator under the factory default configuration, you are prompted whether you want to run the setup utility. Also, the setup utility can be activated manually from the CLI any time after you log on by running the **setup** privileged mode command. (Example 4-1).

Example 4-1 Entering setup tool

```
EN4093-1#setup
```

Information needed for the setup utility

The setup utility requires the following information to perform the basic configuration:

- ▶ Basic system information:
 - Date and time
 - Whether to use Spanning Tree Protocol or not
- ▶ Optional configuration for each port:
 - Speed, duplex, flow control, and negotiation mode (as appropriate)
 - Whether to use virtual local area network (VLAN) tagging or not (as appropriate)
- ▶ Optional configuration for each VLAN:
 - Name of VLAN
 - Which ports are included in the VLAN
- ▶ Optional configuration of IP parameters:
 - IP address/mask and VLAN for each IP interface
 - IP addresses for default gateway
 - Whether IP forwarding is enabled or not

After the **setup** command is executed, the setup tool then prompts you with a series of questions. The first ones are the date and time, and whether to activate the Spanning Tree Protocol. In our case, we defined the date and time that corresponded to our time zone, and we defined Spanning Tree Group as 0N. See Example 4-2.

Important: If you do not enter a value when prompted, the setup utility uses the defaults that are defined by the IBM Networking OS, or leaves the requested value empty.

Example 4-2 Date, time, and Spanning Tree Protocol configuration

```
"Set Up" will walk you through the configuration of
System Date and Time, Spanning Tree, Port Speed/Mode,
VLANs, and IP interfaces. [type Ctrl-C to abort "Set Up"]
```

```
System Date:
Enter year [2012]:      2012
```

```
Enter month [10]:      10
Enter day [5]:         5
System clock set to 19:49:09 Fri Oct  5, 2012.
```

```
System Time:
Enter hour in 24-hour format [19]: 13
13
Enter minutes [49]:
Enter seconds [9]:
System clock set to 13:49:09 Fri Oct  5, 2012.
```

```
Spanning Tree:
Current Spanning Tree Group 1 setting: ON
Turn Spanning Tree Group 1 OFF? [y/n] n
```

Ports and VLANs configuration

Then, the setup tool prompts you to configure VLANs and VLAN tagging for the ports. If you want to change settings for VLANs, enter *y*, or enter *n* to skip port tagging and VLAN configuration. Example 4-3 shows the configuration for INTA1 (Flow Control, Auto Negotiation, and VLAN tagging).

Example 4-3 Port configuration

```
Port Config:
Will you configure VLANs and VLAN tagging for ports? [y/n] y
Enter port (INTA1-B14, EXT1-10, EXT15-22):      INTA1

Gig Link Configuration:
Port Flow Control:
Current Port INTA1 flow control setting:         both
Enter new value ["rx"/"tx"/"both"/"none"]:

Port Auto Negotiation:
Current Port INTA1 autonegotiation:              on
Enter new value ["on"/"off"]:

Port VLAN tagging config (tagged port can be a member of multiple VLANs):
Current VLAN tag support: disabled
Enter new VLAN tag support [d/e]: e
Port INTA1 changed to tagged.
Enter port (INTA1-B14, EXT1-10, EXT15-22):
```

After you finish the configuration of the port, the system prompts you to configure the next port. This step is repeated for all the ports in the switch. When you are done configuring the ports, press Enter without specifying any port, and the setup utility continues with the VLANs configuration.

If you want to change settings for individual VLANs, enter the number of the VLAN you want to configure (Example 4-4). To skip VLAN configuration, press Enter without entering a VLAN number.

Example 4-4 VLAN configuration

```
VLAN Config:
Enter VLAN number from 2 to 4094, NULL at end: 50
Current VLAN name:
Pending new VLAN name: VLAN 50
Enter new VLAN name:

Define Ports in VLAN:
Current VLAN 50: empty
Enter ports one per line, NULL at end:
> INTA1
Port INTA1 is an UNTAGGED port and its PVID is changed from 1 to 50
>
Current ports for VLAN 50:      empty
Pending new ports for VLAN 50:  INTA1

Spanning Tree Group membership:
Enter new Spanning Tree Group index [1-127]: 50
Enter VLAN number from 2 to 4094, NULL at end:
```

Entering a new VLAN name is optional. To use the pending new VLAN name, press Enter.

To assign ports to the VLAN, enter each port, by port number or port alias, and confirm the placement of the port into this VLAN. When you are finished adding ports to this VLAN, press Enter without specifying any port.

After the VLANs are configured, you must configure the Spanning Tree Group membership for the VLAN.

Repeat the steps in this section until all VLANs are configured. When all VLANs are configured, press Enter without specifying any VLAN to stop the VLAN configuration.

IP configuration

IP interfaces are used for defining the networks to which the switch belongs. Up to 128 IP interfaces can be configured on the EN4093/EN4093R 10Gb Scalable Switch and also on the EN2092. The IP address assigned to each IP interface provides the switch with an IP presence on your network. No two IP interfaces can be on the same IP network.

The interfaces can be used for connecting to the switch for remote configuration, and for routing between subnets and VLANs (if used).

Note: IP Interface 127 and 128 are reserved for switch management. If the IPv6 feature is enabled on the switch, IP Interface 125 and 126 are also reserved.

To configure individual IP interfaces, enter the number of the IP interface you want to configure. To skip IP interface configuration, press Enter without typing an interface number.

Example 4-5 shows the configuration of interface 22 to which was assigned the IP address 10.22.22.1 and mask 255.255.255.0.

Example 4-5 IPv4 interface configuration

IP Config:

IP interfaces:

Enter interface number: (1-125, 127) 22

Enter new IP address: 10.22.22.1

Pending new subnet mask: 255.0.0.0

Reminder: IP Interface 22 needs to be enabled.

Current subnet mask: 0.0.0.0

Pending new subnet mask: 255.0.0.0

Enter new subnet mask: 255.255.255.0

Current VLAN: 1

Enter new VLAN [1-4094]: 22

Reminder: VLAN 22 needs to be created and enabled.

Enable IP interface? [y/n] y

Enter interface number: (1-125, 127)

To complete the configuration of the IP interface, a VLAN must be assigned to the interface, and then enable it. In the example, VLAN 22 was assigned to interface 22. Then, the system prompts you to configure another interface

Repeat the steps in this section until all IP interfaces are configured. When all the interfaces are configured, press Enter without specifying any interface number.

Default Gateway

The switch can be configured with up to four IPv4 gateways. Gateways 1 - 3 are reserved for default gateways. Gateway 4 is reserved for switch management and can be configured only by the CMM.

Example 4-6 shows the configuration of the default gateway.

Example 4-6 Default Gateway configuration

IP default gateways:

Enter default gateway number: (1-2, 3, 4) 4

The MGT default gateway can be configured only by MM.

Enter default gateway number: (1-2, 3, 4) 1

Current IP address: 0.0.0.0

Enter new IP address: 10.22.22.254

Enable default gateway? [y/n] y

Enter default gateway number: (1-2, 3, 4)

When you have enabled it, the system prompts you to configure another default gateway.

Repeat the steps in this section until all default gateways have been configured. When all default gateways have been configured, press Enter without specifying any number.

IBM System Networking (SN) switches offer the ability to have multiple default gateways configured at the same time. Each gateway is health checked and if the first goes away, traffic is sent to a different default gateway.

IP Routing

When IP interfaces are configured for the various IP subnets that are attached to your switch, IP routing between them can be performed entirely within the switch. This eliminates the need to send inter-subnet communication to an external router device. Routing on more complex networks, where subnets might not have a direct presence on the EN4093/EN4093R, can be accomplished through configuring static routes or by letting the switch learn routes dynamically.

This part of the setup program prompts you to enable or disable the IP forwarding capability for IP Routing, as shown in Example 4-7.

Example 4-7 Enable and disable IP forwarding

```
Enable IP forwarding? [y/n] y
```

Enter y to enable IP forwarding or n to skip and keep the current setting.

Final steps

As shown in Example 4-8, in the final steps you are prompted whether to restart Setup Utility or continue.

Example 4-8 Applying changes and save configuration

```
Would you like to run from top again? [y/n] n
```

```
Apply the changes? [y/n] y
```

```
Copy changes to startup-config? [y/n]y
```

Then, you are prompted to decide whether to apply the changes or not. If you decide not to apply them, you are prompted to stop all changes or not. If you decide to apply them, the changes are applied and you are prompted to save changes in startup-config.

Note: It is suggested that you change default switch passwords after initial configuration and as regularly as required under your network security policies.

4.2.5 Setup command-line interface procedure

The configuration made using the Setup Utility can also be achieved by using the appropriate commands directly from the CLI.

First, you need to enter privileged mode by using the **enable** command. Then, run the **configuration terminal** command that places you in configuration mode, as shown in Example 4-9.

Example 4-9 Entering configuration mode

```
EN4093-1>enable
```

```
Enable privilege granted.
```

```
EN4093-1#configure terminal
```

```
Enter configuration commands, one per line. End with Ctrl/Z.
```

```
EN4093-1(config)#
```

To configure the system host name, date, and time, run the commands that are shown in Example 4-10, changing the date and time as appropriate.

Example 4-10 System host name, date, and time configuration

```
Router(config)#hostname EN4093-1
```

```
EN4093-1(config)#system date 2012 10 18  
System date set to 18:58:28 Thu Oct 18, 2012.
```

```
EN4093-1(config)#system time 11:58:33  
System clock set to 11:58:33 Thu Oct 18, 2012.
```

To configure ports auto negotiation and flow control, use the commands shown in Example 4-11 (you cannot configure auto negotiation for the external ports because they are fixed ports).

Example 4-11 Port layer 1 configuration

```
EN4093-1(config)#int port INTA1  
EN4093-1(config-if)#auto  
EN4093-1(config-if)#flowcontrol both  
EN4093-1(config-if)#exit
```

To define a VLAN, enable the VLAN, define the VLAN name, and assign ports to the VLAN, use the following commands. Example 4-12 shows the configuration of VLAN 50 and how to assign port INTA1 to it.

To verify the VLAN configuration, run **show vlan** command from privileged mode.

Example 4-12 VLAN configuration

```
EN4093-1(config)#vlan 50
```

```
VLAN number 50 with name "VLAN 50" created.  
Warning: VLAN 50 was assigned to STG 50.  
EN4093-1(config-vlan)#enable  
EN4093-1(config-vlan)#name Vlan50  
EN4093-1(config-vlan)#member INTA1  
EN4093-1(config-vlan)#exit
```

To configure individual IP interfaces, run the commands in Example 4-13, which shows the configuration of interface 22 to which was assigned the IP address of 10.22.22.1 and mask 255.255.255.0. Then, VLAN 50 is assigned to the interface, and it is enabled.

Example 4-13 IPv4 interfaces configuration

```
EN4093-1(config)#interface ip 22  
EN4093-1(config-ip-if)#ip address 10.22.22.1 255.255.255.0  
EN4093-1(config-ip-if)#vlan 50  
EN4093-1(config-ip-if)#enable  
EN4093-1(config-ip-if)#exit
```

Example 4-14 shows the configuration of the Default Gateway and how to enable IP Forwarding.No.

Example 4-14 Configuring Default Gateway and IP Forwarding

```
EN4093-1(config)#ip gateway 1 address 10.22.22.254
EN4093-1(config)#ip routing
EN4093-1(config)#exit
```

The final step is to save the configuration (Example 4-15).

Example 4-15 Configuration save

```
EN4093-1#copy running-config startup-config
```

4.2.6 User management

To enable better switch management and user accountability, three levels or classes of user access have been implemented on the EN4093/EN4093R. Levels of access to CLI, web management functions, and screens increase as needed to perform various switch management tasks. Conceptually, access classes are defined as follows:

User	Interaction with the switch is completely passive. Nothing can be changed on the EN4093/EN4093R. Users can display information that has no security or privacy implications, such as switch statistics and current operational state information.
Oper	Operators can make temporary changes on the EN4093/EN4093R. These changes are lost when the switch is rebooted/reset. Operators have access to the switch management features used for daily switch operations. Because any changes an operator makes are undone by a reset of the switch, operators cannot severely affect switch operation.
Admin	Administrators are the only ones that can make permanent changes to the switch configuration—changes that are persistent across a reboot/reset of the switch. Administrators can access switch functions to configure and troubleshoot problems on the EN4093/EN4093R. Because administrators can also make temporary (operator-level) changes as well, they must be aware of the interactions between temporary and permanent changes.

Access to switch functions is controlled by using unique user names and passwords. When you are connected to the switch via Telnet or SSH, you are prompted to enter a username and password. The default username and password for each access level are listed in Table 4-1.

Table 4-1 Default user accounts and passwords

User account	Password	Description and tasks performed
User	user	The User has no direct responsibility for switch management. This person can view all switch status information and statistics, but cannot make any configuration changes to the switch.
Oper	oper	The Operator can make temporary changes that are lost when the switch is rebooted/reset. Operators have access to the switch management features used for daily switch operations.

User account	Password	Description and tasks performed
Admin	admin	The superuser Administrator has complete access to all command modes, information, and configuration commands on the EN4093/EN4093R 10Gb Scalable Switch, including the ability to change both the user and administrator passwords.

Change the default passwords for your administrator *User*. To accomplish this task, perform the commands that are shown in Example 4-16.

Example 4-16 Changing administrator user password

```
EN4093-1(config)#access user administrator-password
Changing ADMINISTRATOR password; validation required:
Enter current local admin password:
Enter new admin password (max 128 characters):
Re-enter new admin password:
New admin password accepted.
EN4093-1(config)#
```

You can list the existing users in the switch by running **show access user**. As shown in Example 4-17, you will see the default users (*user*, *oper*, and *admin*) and the user *USERID* with administrator privilege that is used by the CMM for administration purposes.

Example 4-17 Showing the user of the system

```
EN4093-1#sh access
Current System Access settings:

IP Management currently allowed from *ALL* IP addresses

Usernames:
  user      - enabled      - offline
  oper      - disabled     - offline
  admin     - Always Enabled - online      1 session.
Current User ID table:
  1: name USERID , ena, cos admin , password valid, offline

Current strong password settings:
  strong password status: disabled

HTTP access currently disabled
HTTPS server access currently enabled on TCP port 443
SNMP access currently read-write
User configuration from BBI currently disabled
Telnet/SSH access configuration from BBI currently disabled
Telnet access currently disabled
TFTP occurs over port 69
EN4093-1#
```

IBM Networking OS allows an administrator to define user accounts that allow users to perform operation tasks through the switch by using CLI commands. After user accounts are configured and enabled, the switch requires user name and password authentication.

If you want to create a user, run the commands shown in Example 4-18. You define the access rights to the user that you want to create, and then provide a password for it (you are prompted for the admin password to validate the creation). Finally, you must enable the created user (by default, all users are disable).

To create a user, you must provide a User Index number. In our example, we create the user, *John*, with a User Index number of 2, which was the first User Index number available.

Example 4-18 Adding a user

```
EN4093-1(config)#access user 2 name John
EN4093-1(config)#access user 2 password
Changing John password; validation required:
Enter current admin password:
Enter new John password (max 128 characters):
Re-enter new John password:
New John password accepted.
EN4093-1(config)#access user 3 enable
```

The user is, by default, assigned to the user access level (also known as *class of service (COS)*). COS for all user accounts has global access to all resources except for User COS, which has access to view only resources that the user owns. To change the user's level, select one of the options shown in Example 4-19.

Example 4-19 User access level

```
EN4093-1(config)#access user 2 level administrator
```

To confirm that the creation of the user is done correctly, run **show access user uid #** (Example 4-20).

Example 4-20 User details

```
EN4093-1#show access user uid 2
Current User ID 2:
      name John      , ena, cos admin  , password valid, offline
EN4093-1#
```

You can also disable or delete one user. As shown in Example 4-21, the first command disables the user *John*, and the second command deletes the user *John*.

Example 4-21 Disabling and deleting users

```
EN4093-1(config)#no access user 2 enable

EN4093-1(config)#no access user 2
```

IBM Networking OS also supports other authentication systems, such as remote authentication dial-in user service (RADIUS), Terminal Access Controller Access-Control System (TACACS+), and Lightweight Directory Access Protocol (LDAP). With these security solutions, the user management is done external to the switch. Therefore, the only thing that must be done is to configure the switch to access the external security server.

Note: By default, the switch will disconnect your Telnet or SSH session after ten minutes of inactivity. This function is controlled by the following configuration mode command: **system idle <1-60>**.



Compute node network configuration

In this chapter, we show examples of configuring the network interface cards (NICs) of the compute nodes in the IBM Flex System Enterprise Chassis.

5.1 Introduction and background

The information in this IBM Redbooks publication is focused on best of breed, highly available solutions with the IBM PureSystems line, with redundancy a centerpiece of the discussion. IBM clients demand a resilient infrastructure with no single points of failure. This desire extends all the way down to the network interface cards (NICs) embedded in the compute nodes themselves, and are described in detail in the following chapter.

5.1.1 NIC teaming

NIC teaming, also known as *bonding*, is a solution used on servers to logically bond two or more NICs to form one or more logical NICs for purposes of high availability, increased performance, or both.

There are many forms of NIC teaming, and the type available for a server is often tied to the operating system that is installed.

For Microsoft Windows, the teaming software traditionally was provided by the NIC vendor and is installed as an add-on to the operating system. This software often also includes the elements necessary to enable VLAN tagging on the logical NICs created by the teaming software. These logical NICs are seen by the OS as physical NICs and are treated as such when configuring them. Depending on the NIC vendor, the teaming software might offer several different types of failover, including simple active/standby, static aggregation, dynamic aggregation (Link Aggregation Control Protocol (LACP)), and vendor-specific load balancing schemes.

For Linux-based systems, the bonding module is used to implement NIC teaming. There are a number of bonding modes available, most commonly mode 1 (active/standby) and mode 4 (LACP aggregation). Like Windows teaming, Linux bonding also offers logical interfaces to the OS that can be used as wanted. Unlike Windows teaming, VLAN tagging is controlled by different software. It can create sub interfaces for VLANs off both physical and logical entities, for example, `eth0.10` for VLAN 10 on physical `eth0`, or `bond0:20`, for VLAN 20 on a logical NIC bond pair 0.

Like Linux, VMware ESXi also has built-in teaming in the form of assigning multiple NICs to a common vSwitch (a logical switch that runs within an ESXi host, which is shared by the VMs that require network access). VMware has several teaming modes, with the route-based default on the originating virtual port ID. This default mode provides a per VM load balance of physical NICs assigned to the vSwitch.

The teaming method that is best for a specific environment is unique to each situation. However, these common elements might help in the decision-making process:

- ▶ Do not select a mode that requires some form of aggregation (static/LACP) on the switch side unless the NICs in the team go to the same physical switch or logical switch created by a specific technology, such as virtual link aggregation or stacking.
- ▶ If using a mode that uses some form of aggregation, you must also perform correct configuration on the upstream switches to complete the aggregation on that side.
- ▶ The most stable solution is often *active/standby*, but this solution has the disadvantage of losing any bandwidth on a NIC that is in standby mode.
- ▶ Most teaming software offers proprietary forms of load balancing. The selection of these modes must be thoroughly tested for suitability to the task for an environment.
- ▶ Most teaming software incorporates the concept of *auto failback*, which means that if a NIC went down and then came back up, it automatically fails back to the original NIC. Although this function helps ensure good load balancing, each time that a NIC fails, some small packet loss might occur, which can lead to unexpected instabilities. A flapping link occurs when a severe disruption to the network connection of the servers causes the link to flap back and forth. One way to mitigate this circumstance is to disable the auto failback feature. After a NIC fails, the traffic falls back only in the event that the original link is restored and something happened to the current link that requires a switchover.

It is your responsibility to understand your goals and the tools available to achieve those goals. NIC teaming is one tool for users that need high availability connections for their compute nodes.

5.2 Components used and setup

For the scenarios below, we used the following equipment to illustrate how to configure host-based networking:

- ▶ IBM Flex System Enterprise Chassis (quantity 1)
- ▶ IBM Flex System x240 Compute Node (quantity 1)
IBM Flex System CN4054 10 Gb Virtual Fabric Adapter (quantity 1)
- ▶ IBM Flex System Fabric EN4093R/EN4093R 10Gb Scalable Switches (quantity 2)

Switches: Throughout this book, we use EN4093/EN4093R to denote that either switch can be used.

Our compute node (called *flexnode1*), was installed in Slot 1 of an Enterprise Chassis. The CN4054 card was installed in Slot 1 of the x240. One EN4093/EN4093R switch was installed in Slot 1 of the Enterprise Chassis, and the other EN4093/EN4093R installed in Slot 2 for redundancy.

The following operating systems were used in this chapter for illustration of OS networking implementation:

- ▶ Windows Server 2008 R2 with Service Pack 1
- ▶ Red Hat Enterprise Linux 6.3 Server
- ▶ VMware ESXi 5.1

5.2.1 Testing methodology

For all of the operating systems selected for illustration in this chapter, we have decided to show how to configure active-backup NIC teaming and describe the steps in doing so on VLAN 4092 using 802.1Q tagging. The operating system must be “VLAN-aware” and tag the packets leaving the adapter with an ID of 4092 in order to be forwarded by the EN4093/EN4093R switches.

Although other teaming scenarios can be employed on the IBM Flex System Compute Nodes, active-backup NIC teaming is the easiest to set up and is generally supported across numerous operating systems. Although no traffic actively traverses the backup link in the team, redundancy prevents the failure of either link from taking the node off the network.

All but two of the internal ports on the CN4054 Virtual Fabric Adapter were disabled in the Unified Extensible Firmware Interface (UEFI) basic input/output system (BIOS) to present easy configuration guidance for the reader. If more ports are needed for capacity or redundancy purposes, these can be re-enabled and implemented using the guidance provided below.

Installation of the various operating systems on the compute nodes is outside of the scope of this chapter. It is assumed that the reader already has their chosen OS built and operational on the compute node before attempting the instructions below. This includes any prerequisite programs that might not be a part of the operating system, or drivers/modules that might be needed before the hardware itself can be recognized.

In all cases illustrated below, we found it most helpful to be connected to the OS via the integrated management module (IMM) built in to the compute node during these configurations, because it is a dedicated management path to the node. Interruptions to network service during any configuration step will not affect the reader’s administrative session.

5.2.2 Logical diagram

A network diagram of the testing setup is illustrated in Figure 5-1. The Enterprise Chassis is not shown in the diagram itself for brevity and to give the reader a sense of how the networking connections are displayed and function logically rather than physically.

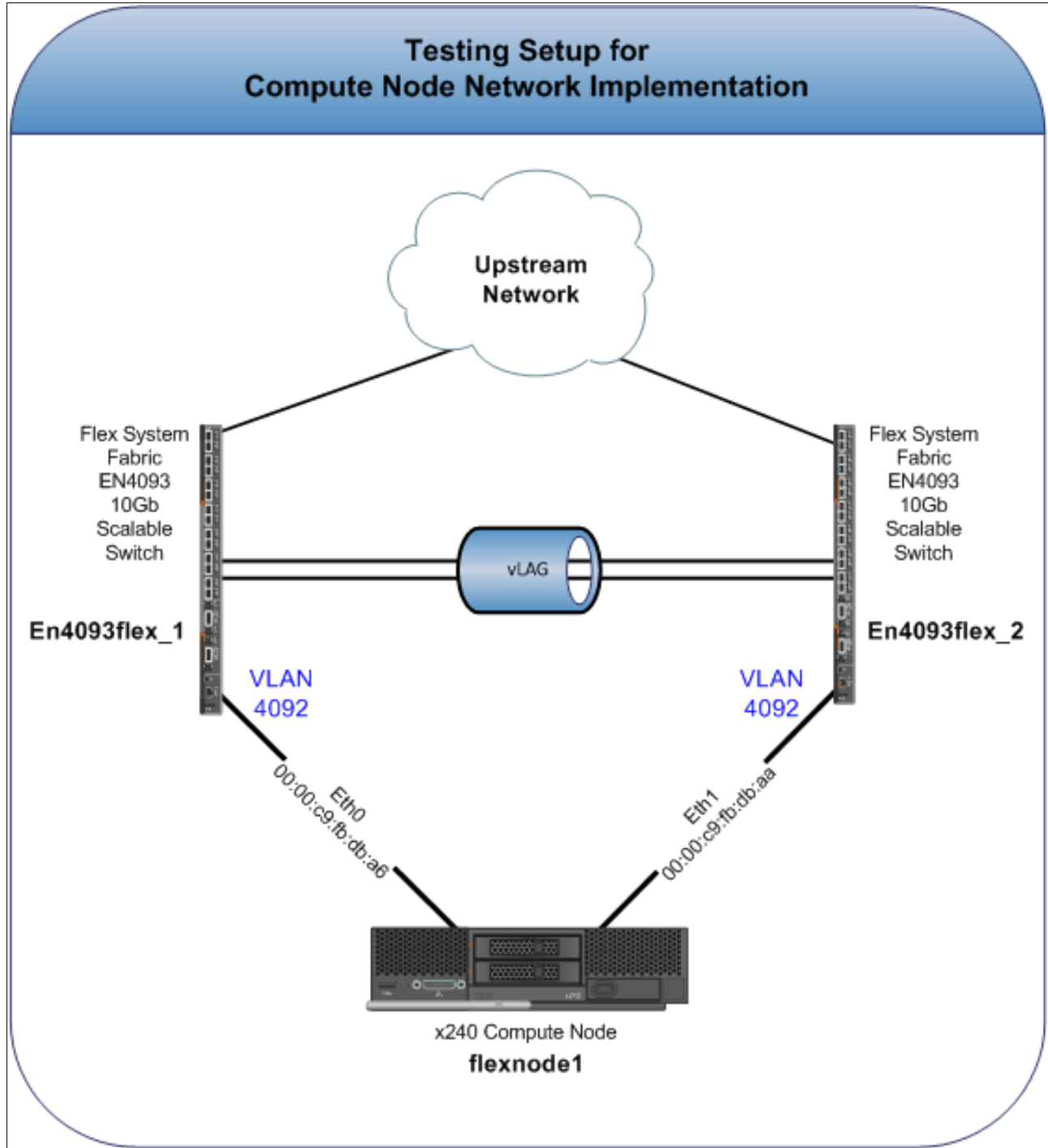


Figure 5-1 Logical diagram for network implementation setup

5.3 Microsoft Windows Server 2008

In our setup, Windows Server 2008 NIC Teaming is done through the vendor-provided “OneCommand NIC Teaming and VLAN Manager” utility by Emulex. This important tool can be downloaded from the Emulex website, or the IBM FixCentral website:

<http://www.ibm.com/support/fixcentral/>

Ensure that the latest NIC drivers are installed and that the NIC cards are recognized by Windows before installing the **OneCommand** utility. Our experience was that Windows did not recognize the Emulex adapters at all until the NIC drivers were installed from the IBM FixCentral repository. After they are recognized, installation of the **OneCommand** utility can be done by the server administrator.

5.3.1 Implementation

The server administrator can install the **OneCommand** utility by taking the following steps:

1. We begin by showing that our two Emulex NIC ports are active and operational from the perspective of Windows. Interfaces have been renamed to correspond with which upstream EN4093/EN4093R switch that we are connected to as a means of easy identification (Figure 5-2). Although renaming these adapters is optional, it does aid in problem identification if port settings on the EN4093/EN4093R need to be investigated.

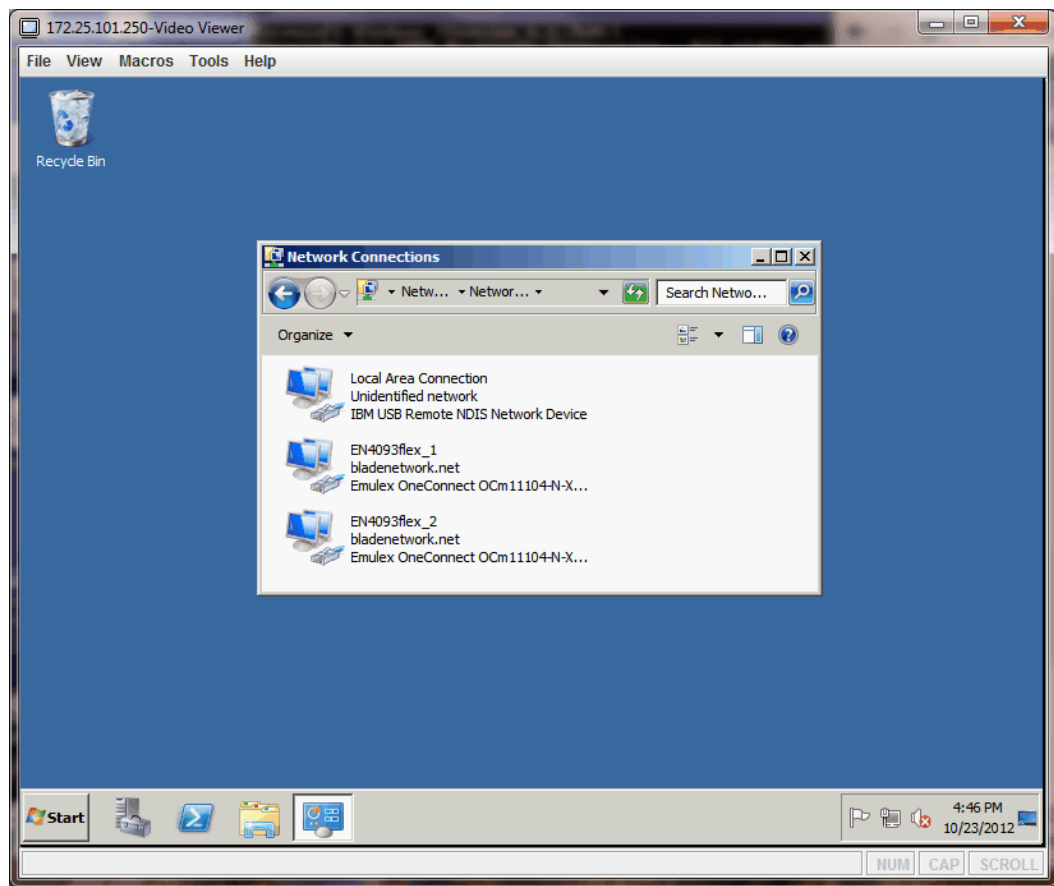


Figure 5-2 Emulex interfaces in Network Connections window

2. Start the “OneCommand NIC Teaming and VLAN Manager”, as shown in Figure 5-3.

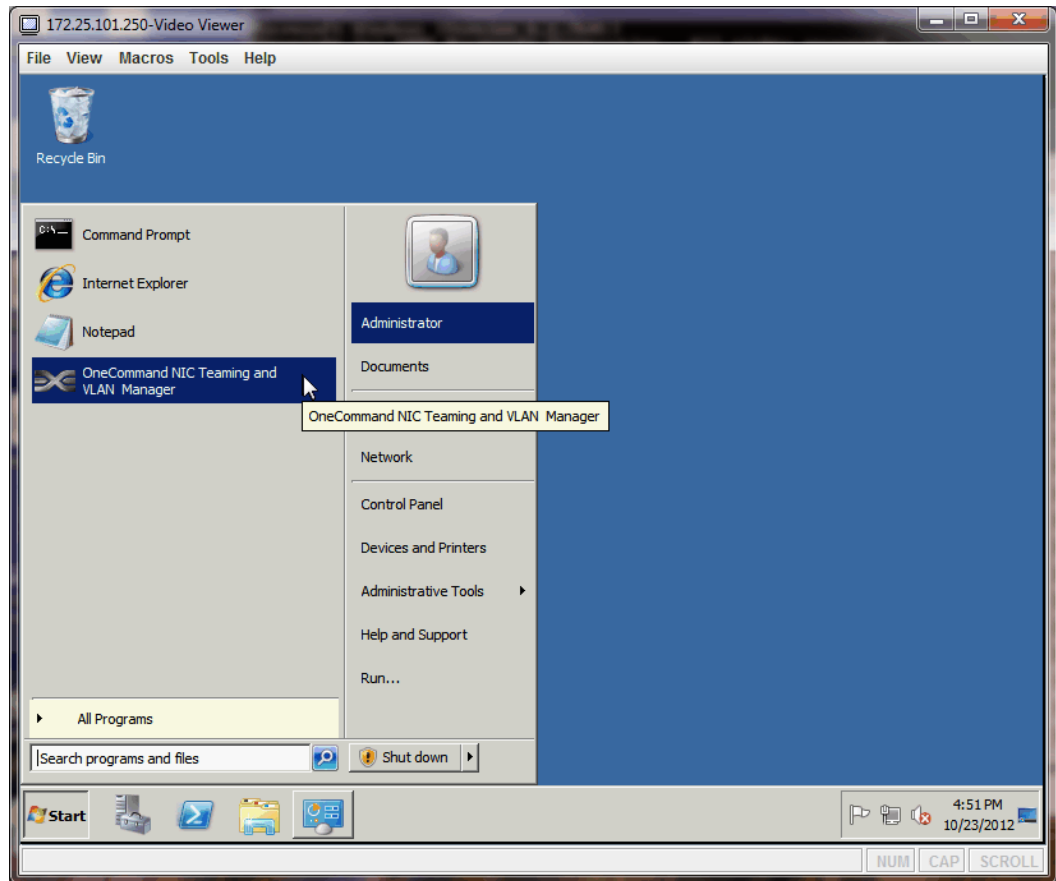


Figure 5-3 Launching the OneCommand utility by Emulex

3. If you see the following message in Figure 5-4 after starting the **OneCommand** utility, follow the provided instructions by disabling the OneCommand background process so that configuration changes can be made.

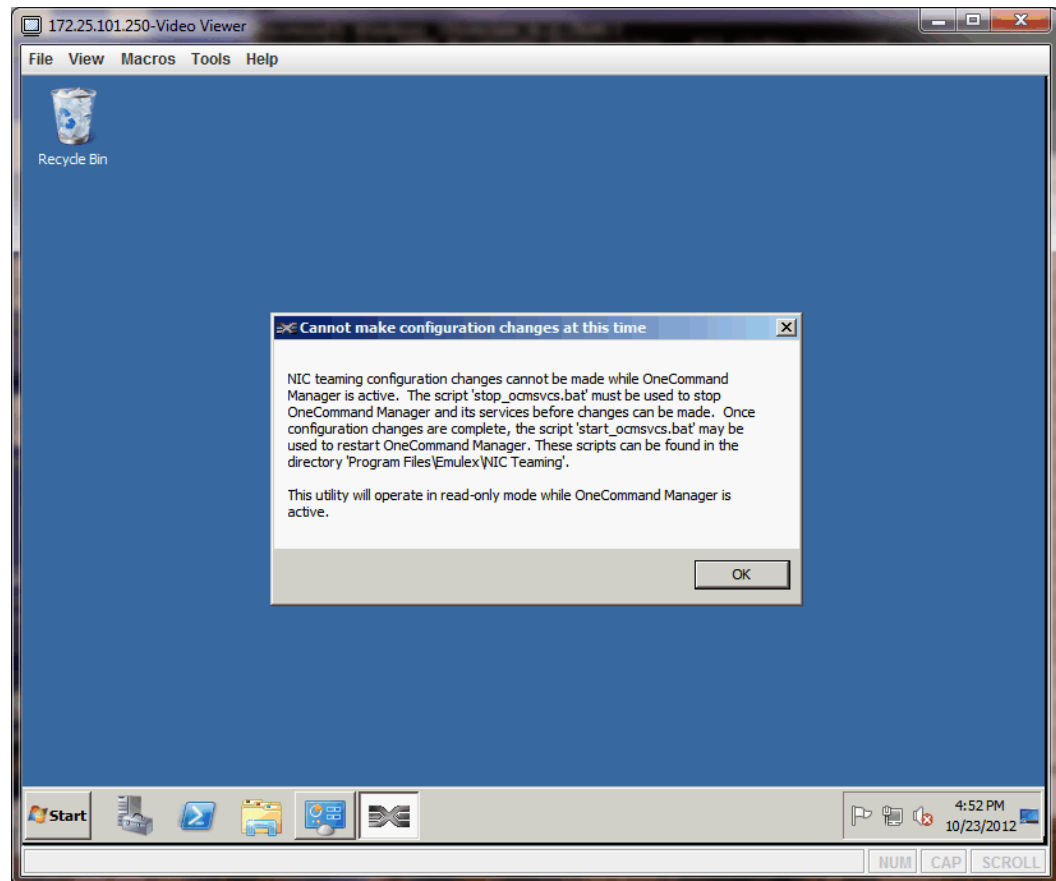


Figure 5-4 Warning message to notify the user to disable the OneCommand background process before configuration can continue

4. After starting the **OneCommand** utility, click **Create Team**, as shown in Figure 5-5.

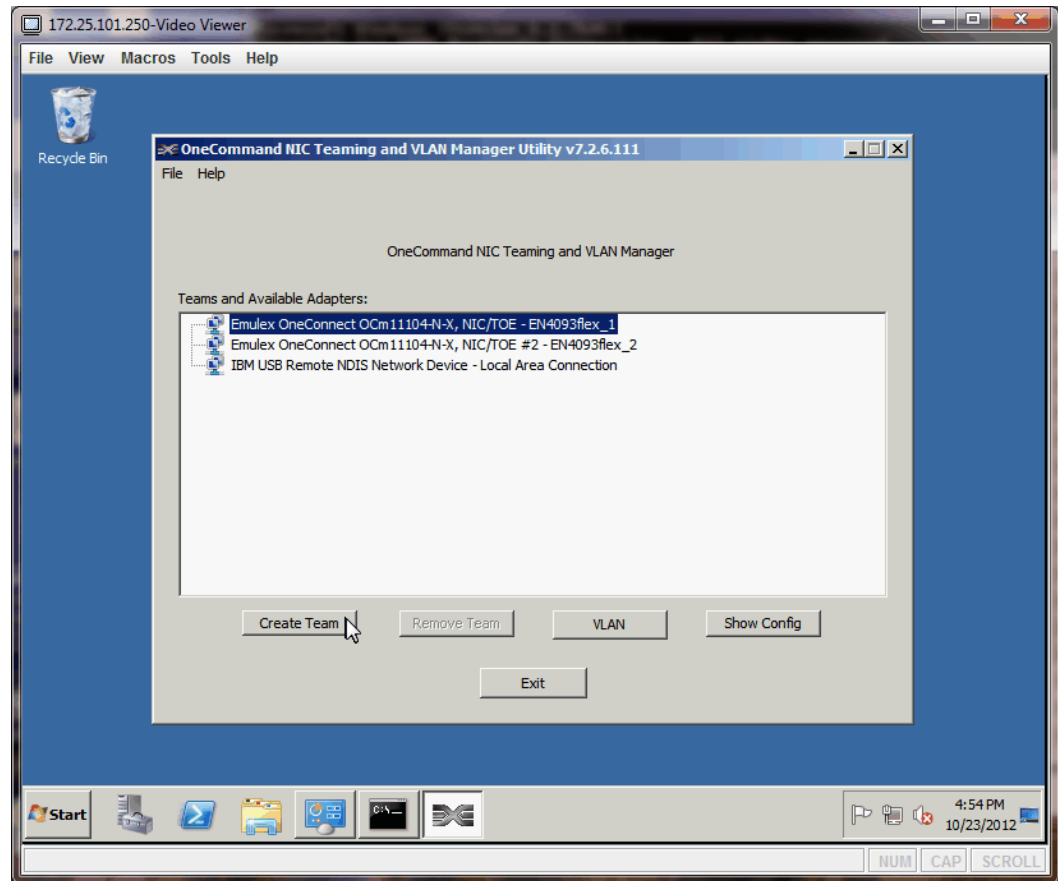


Figure 5-5 Creating a NIC Team in the OneCommand utility

5. Give the Team Name a unique name and description and ensure that **FailOver** is selected in the Team Type pull-down menu. Select both Emulex adapters and click **Add** to move them to the Team Member Adapters section of the configuration window, as illustrated in Figure 5-6. Click **OK** to return to the previous window.

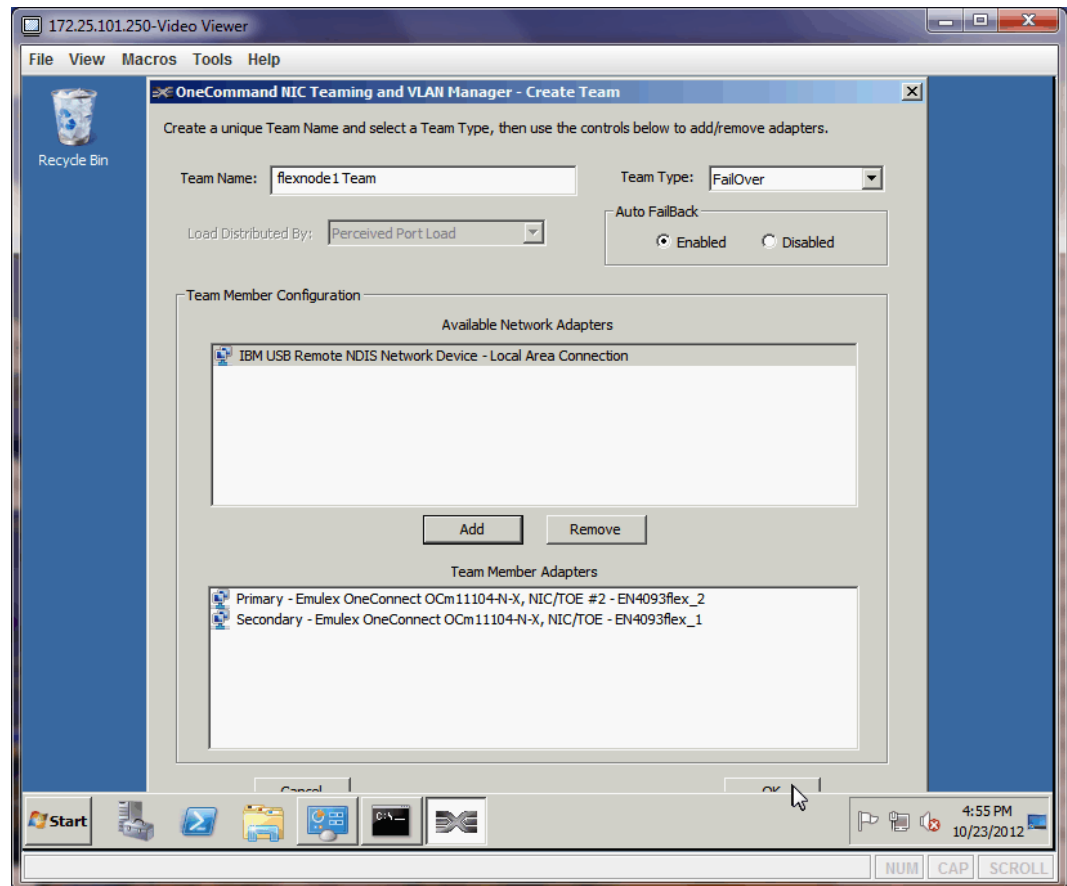


Figure 5-6 Building NIC Team using both Emulex adapters in our compute node

6. Notice that the NIC bond you created in the previous step is now listed on the main window with the Emulex adapters bundled together. We must ensure that VLAN 4092 is being tagged because our I/O modules are expecting the traffic to be tagged with 802.1Q frames before leaving the adapter. Click **VLAN**, as shown in Figure 5-7.

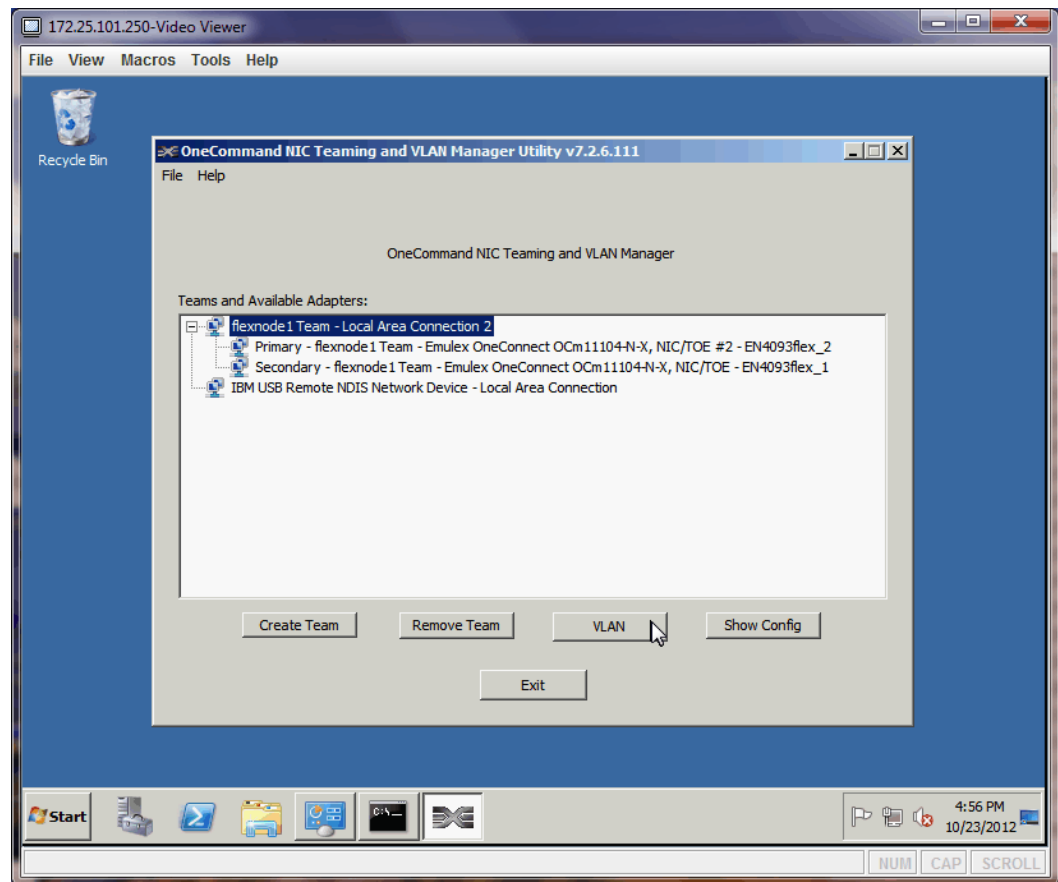


Figure 5-7 Inspecting the new NIC Team and configuring VLAN information

7. In the VLAN ID field, enter “4092”, as shown in Figure 5-8, and click **Add**.

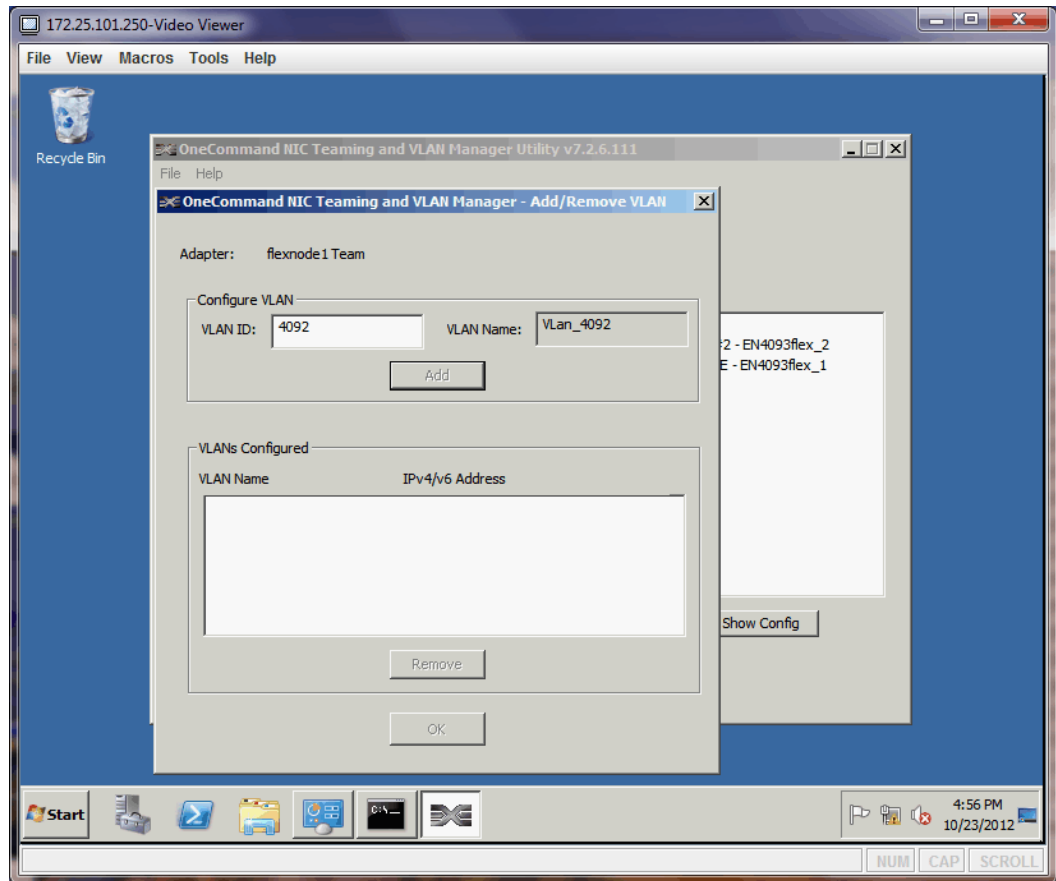


Figure 5-8 Adding VLAN 4092 to the NIC Team

8. After a few seconds, the VLAN name and observed IP address is displayed in the configuration window. The IP address might need to be changed or statically assigned. In our network, a Dynamic Host Configuration Protocol (DHCP) server provided the address that is shown in Figure 5-9, which we modify in the next step.

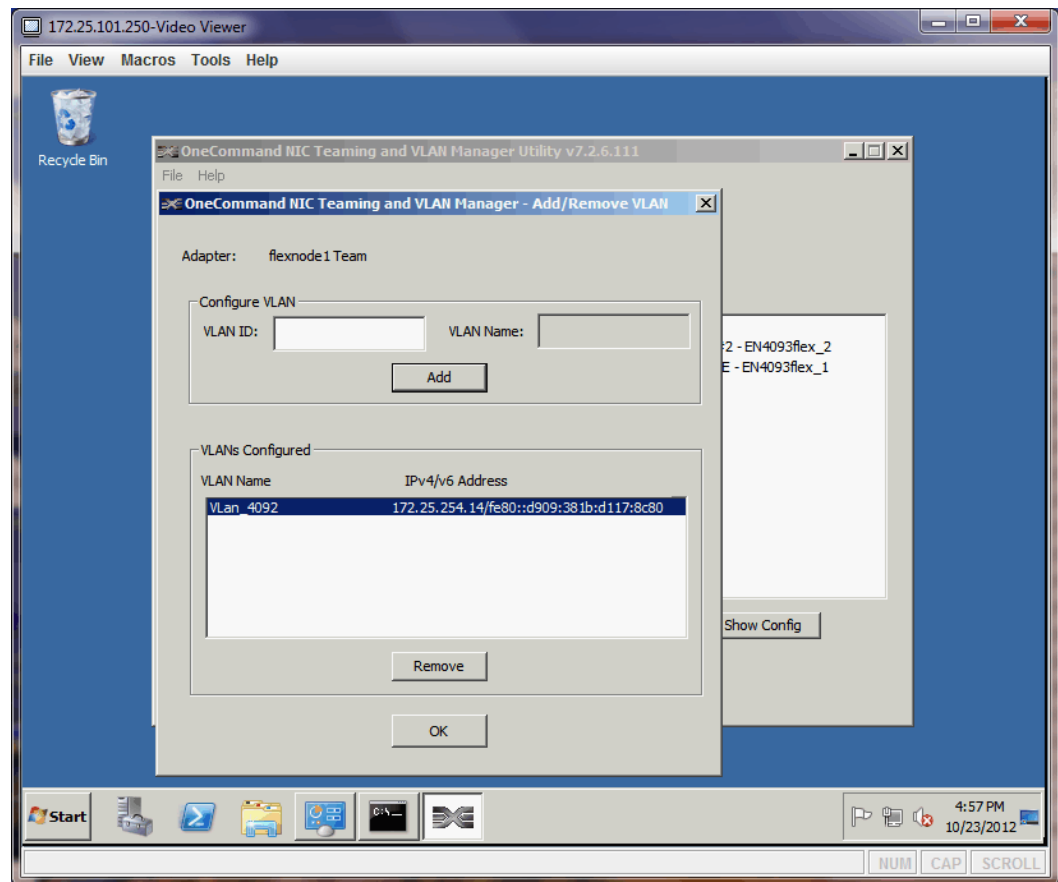


Figure 5-9 VLAN 4092 added to the NIC Team

- To statically assign an address to the new NIC Team formed in the previous steps, we must go back to the Network Connections panel in Windows and modify the newly created LAN connection, as shown in Figure 5-10.

Important: Ensure that you select the new NIC Team device to apply an IPv4 or IPv6 management address to rather than the individual physical interfaces. Network redundancy will not be implemented properly if addresses are assigned to the Emulex physical interfaces themselves.

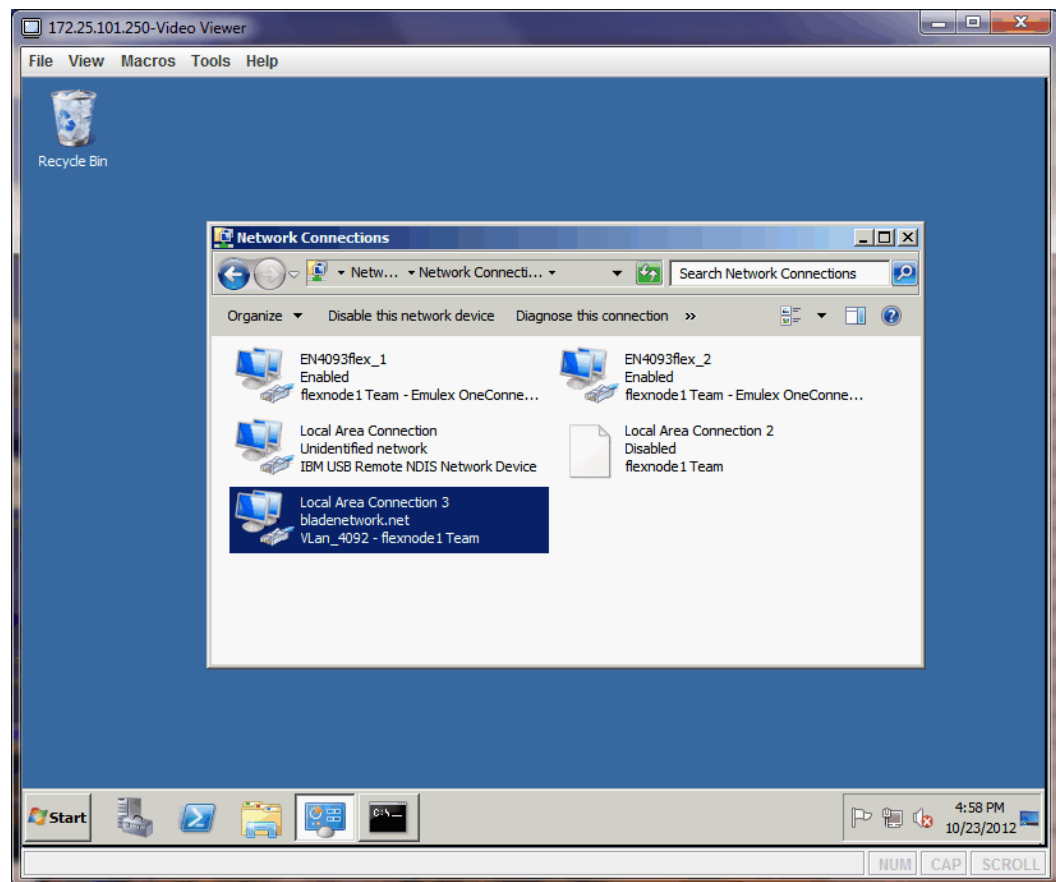


Figure 5-10 Selecting the new NIC Team device in Network Connections to apply a management IP address to

10. Navigate to the TCP/IP properties for the NIC Team device for IPv4 and input the correct IP address, subnet mask, and gateway shown in Figure 5-11. We have chosen IPv4 for this example, but IPv6 addresses can be assigned in a similar manner by choosing the IPv6 stack instead.

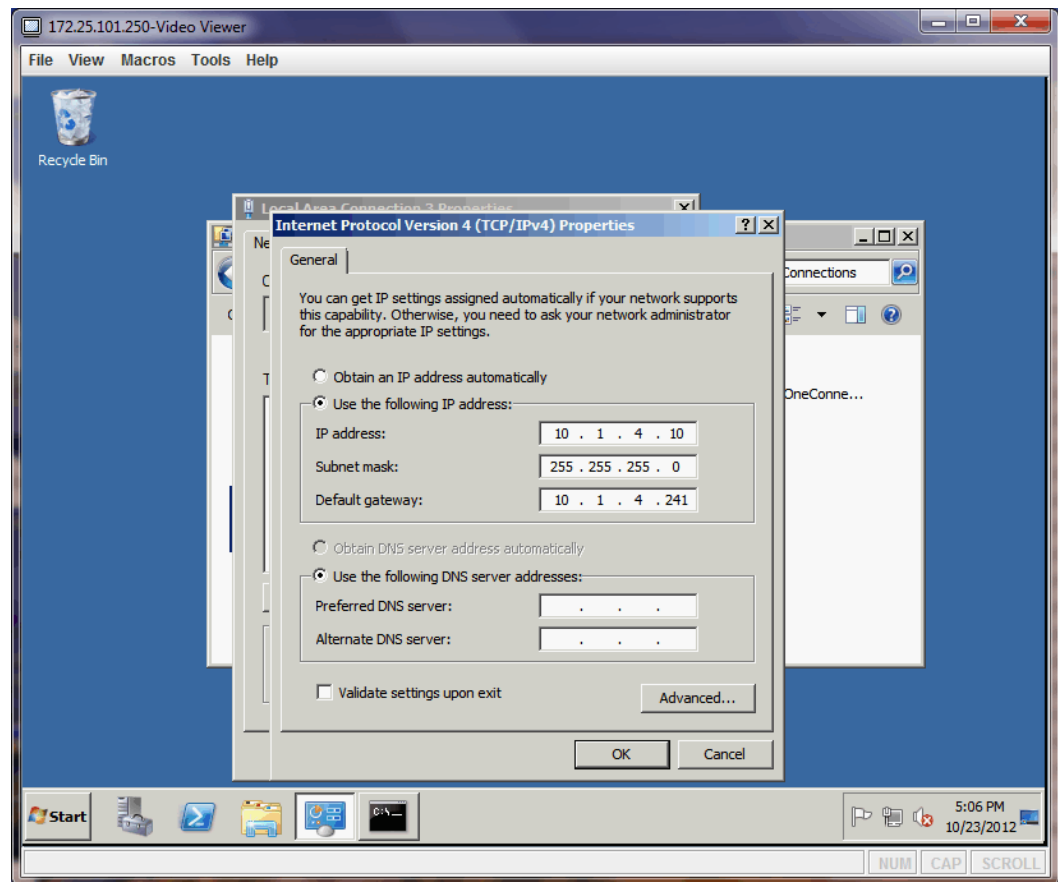


Figure 5-11 Inputting the IPv4 management address on the NIC Team adapter

11. Although usually not necessary, stop and restart the Emulex host bus adapter (HBA) management service that is shown in Figure 5-12 to ensure that your new configuration is working properly and will be active after the device reboots.

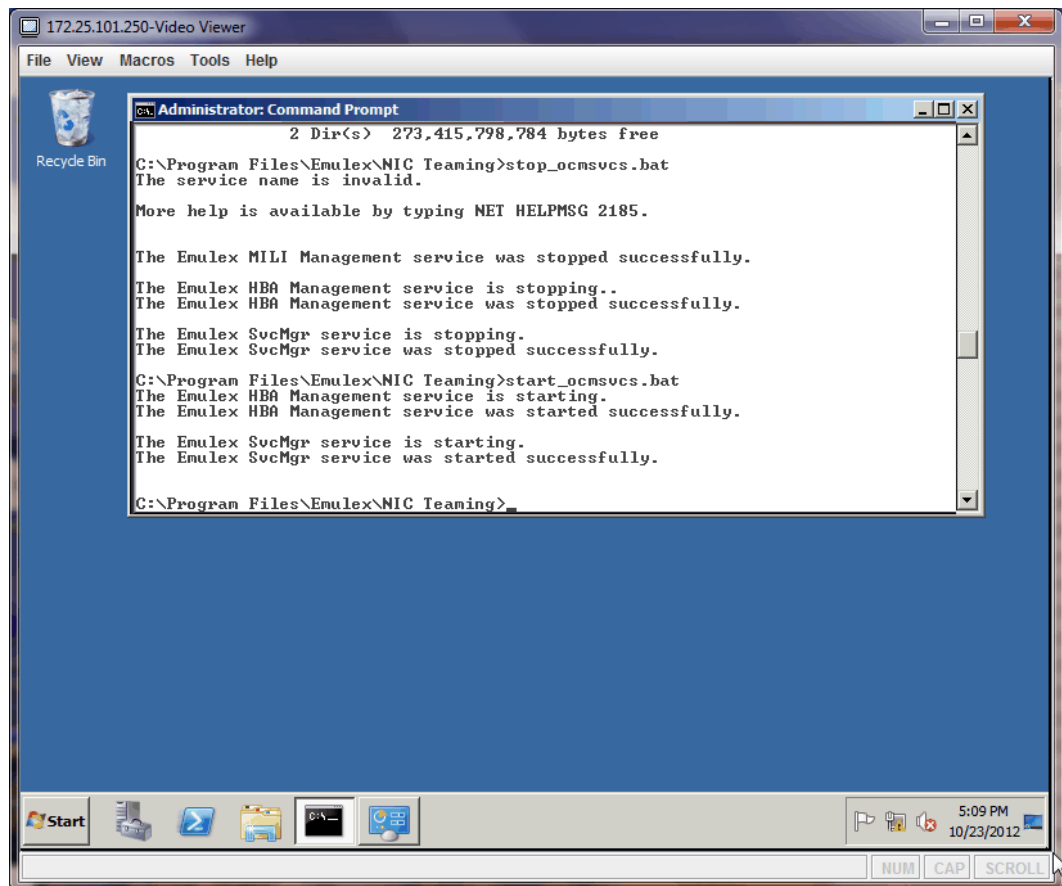


Figure 5-12 Restarting the OneCommand background process to verify that changes work across device reboots

You have now configured a Windows 2008 R2 Server for active-backup NIC teaming.

5.4 Red Hat Enterprise Linux Server 6

Configuring Red Hat Enterprise Linux for NIC teaming can be done through the standard CLI through editing a series of configuration files and restarting network services. The **NetworkManager** utility is installed by default and supposedly can configure “bonds” in the OS itself by the means of an interactive GUI window. However, we found that using the CLI was more straightforward in teaming the NIC cards together in the compute node and functioned more reliably than NetworkManager.

Important: This chapter was framed to illustrate how to set up the networking stack on the various operating systems with 802.1Q tagging required on VLAN 4092. However, we discovered an issue with the 802.1Q support in Red Hat Enterprise Linux (RHEL) 6.3. We found that the compute node lost network connectivity if *either* of the physical interfaces were shut down via the EN4093/EN4093R switch in an attempt to simulate a network or switch outage.

Given this scenario, the Linux implementation section described below illustrates how to set up networking *without* enabling 802.1Q tagging. Failover works as it should in this scenario.

5.4.1 Implementation

1. Depending on the options selected at installation by the reader, either a DHCP or statically assigned IP address will be active on one of the Ethernet interfaces. To achieve high availability, we must bundle the physical interfaces together in a *bond* in order to protect against link or switch failure. All the subsequent steps should be executed as the root user.
2. Begin by first stopping and disabling the NetworkManager service, as shown in Example 5-1.

Example 5-1 Stopping NetworkManager service daemon

```
[root@flexnode1 ~]# /sbin/service NetworkManager stop
[root@flexnode1 ~]# /sbin/chkconfig NetworkManager off
```

3. In order for the channel bonding interface to be persistent across reboots, the kernel module named *bonding* must be associated with a new virtual interface that we call *bond0*. Create a file called *bonding.conf* in the */etc/modprobe.d/* directory with the contents that are shown in Example 5-2.

Example 5-2 Creating module association with bond0 interface

```
alias bond0 bonding
```

4. Next, create the channel bonding interface configuration file in the */etc/sysconfig/network-scripts/* directory, as shown in Example 5-3.

Example 5-3 bond0 interface creation

```
DEVICE=bond0
IPADDR=10.1.4.10
NETMASK=255.255.255.0
GATEWAY=10.1.4.241
ONBOOT=yes
BOOTPROTO=None
USERCTL=no
BONDING_OPTS="mode=active-backup miimon=100"
```

5. Reconfigure the physical interfaces to be bound together in our newly created and consolidated bond0 interface that is shown in Example 5-4.

Note: The commission of the physical Media Access Control (MAC) addresses in this step was done to provide more flexibility if the burned-in MAC changes on either eth0 or eth1, which can occur during hardware replacement or user modification.

Example 5-4 eth0 and eth1 interface reconfiguration

```
[root@flexnode1 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE=eth0
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
USERCTL=no
TYPE=Ethernet
```

```
[root@flexnode1 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth1
```

```
DEVICE=eth1
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
TYPE=Ethernet
USERCTL=no
```

6. Restart the network service to pick up the changes, as indicated in Example 5-5.

Example 5-5 Restarting network services

```
[root@flexnode1 ~]# /sbin/service network restart
```

```
Bringing down interface eth0:           [ OK ]
Bringing down interface eth1:           [ OK ]
Shutting down loopback interface:       [ OK ]
Bringing up loopback interface:         [ OK ]
Bringing up interface bond0:            [ OK ]
```

7. Start the **ifconfig** utility to verify that the IPv4 address is on the bond0 interface and that the physical interfaces are running in slave mode, as shown in Example 5-6. The same MAC address on bond0, eth0, and eth1 is expected because of the nature of how the bonding driver works while in active-backup mode.

Example 5-6 ifconfig output and verification

```
bond0    Link encap:Ethernet  HWaddr 00:00:C9:FB:DB:A6
         inet addr:10.1.4.10  Bcast:10.1.4.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
         RX packets:532 errors:0 dropped:0 overruns:0 frame:0
         TX packets:391 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:48954 (47.8 KiB)  TX bytes:38700 (37.7 KiB)

eth0     Link encap:Ethernet  HWaddr 00:00:C9:FB:DB:A6
         UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
```

```

RX packets:520 errors:0 dropped:0 overruns:0 frame:0
TX packets:391 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:46516 (45.4 KiB) TX bytes:38700 (37.7 KiB)
Interrupt:90 Memory:da000000-da012800

eth1    Link encap:Ethernet  HWaddr 00:00:C9:FB:DB:A6
        UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
        RX packets:12 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:2438 (2.3 KiB) TX bytes:0 (0.0 b)
        Interrupt:98 Memory:d8000000-d8012800

lo       Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:116057917 errors:0 dropped:0 overruns:0 frame:0
        TX packets:116057917 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:1880463508 (1.7 GiB) TX bytes:1880463508 (1.7 GiB)

```

8. To list more pertinent details about the bond itself, including information regarding slave interface real MAC addresses, speed/duplex, and which physical interface is active, issue the command that is shown in Example 5-7.

Example 5-7 proc filesystem output for bond0

```
[root@flexnode1 ~]# cat /proc/net/bonding/bond0
```

```

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

```

```

Slave Interface: eth0
MII Status: up
Speed: 10000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:00:c9:fb:db:a6

```

```

Slave Interface: eth1
MII Status: up
Speed: 10000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:00:c9:fb:db:aa

```

Configuring a Red Hat Enterprise Linux 6.3 machine for active-backup NIC teaming is now complete.

5.5 VMware ESXi 5

Administering VMware's ESXi operating system is performed through the vSphere client, downloadable as an executable file from a web-service that runs on the VMware ESXi installation on the compute node. It is recommended to set an accessible IPv4 or IPv6 address on the ESXi machine's direct console, accessible through the compute node's IMM connection.

- More instructions on how to configure the management network from the console are on the following website:

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1006710

When installation is finished and you can access the machine, suggestions for the network implementation are in the following sections.

Important: Although we are illustrating how to implement active-backup NIC teaming in this section, the adapters themselves are placed into an active-active configuration in the **administration** utility. The VMware default load balance algorithm for virtual machines is port-based in that each VM chooses a vmnic interface and uses that during its life, moving to the other vmnic only if a problem is encountered. This behavior is similar to Windows and Linux in that only one physical interface is used on a per VM basis, adhering to an active-backup design.

5.5.1 Implementation

1. Log in using the vSphere client that is shown in Figure 5-13.

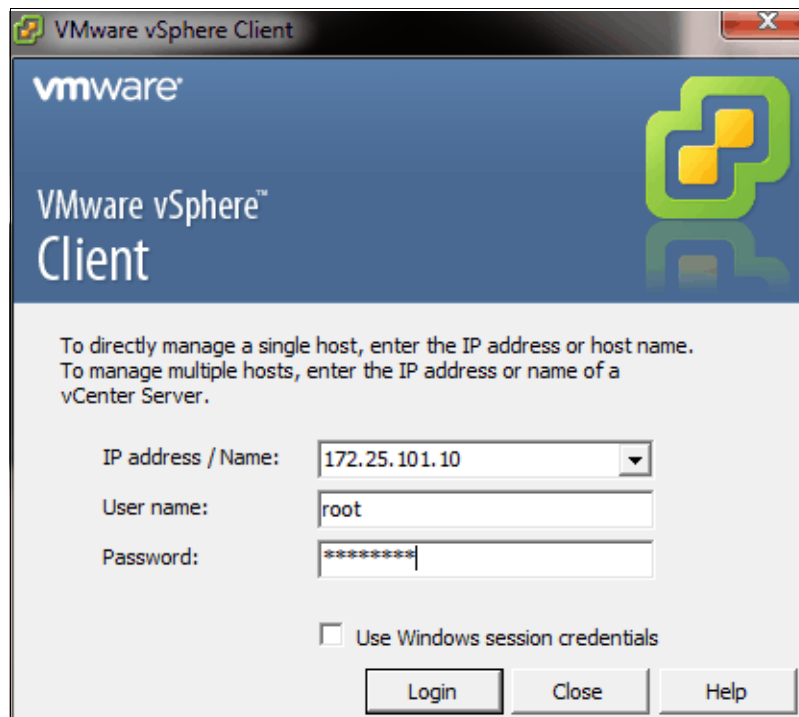


Figure 5-13 Logging in to vSphere client

2. After logging in, click the **Configuration** tab then **Networking** under the Hardware panel. Notice that one of our NICs is in a standby mode, which we adjust to a more appropriate active-active mode. Click the **Properties** field for vSwitch0, as shown in Figure 5-14.

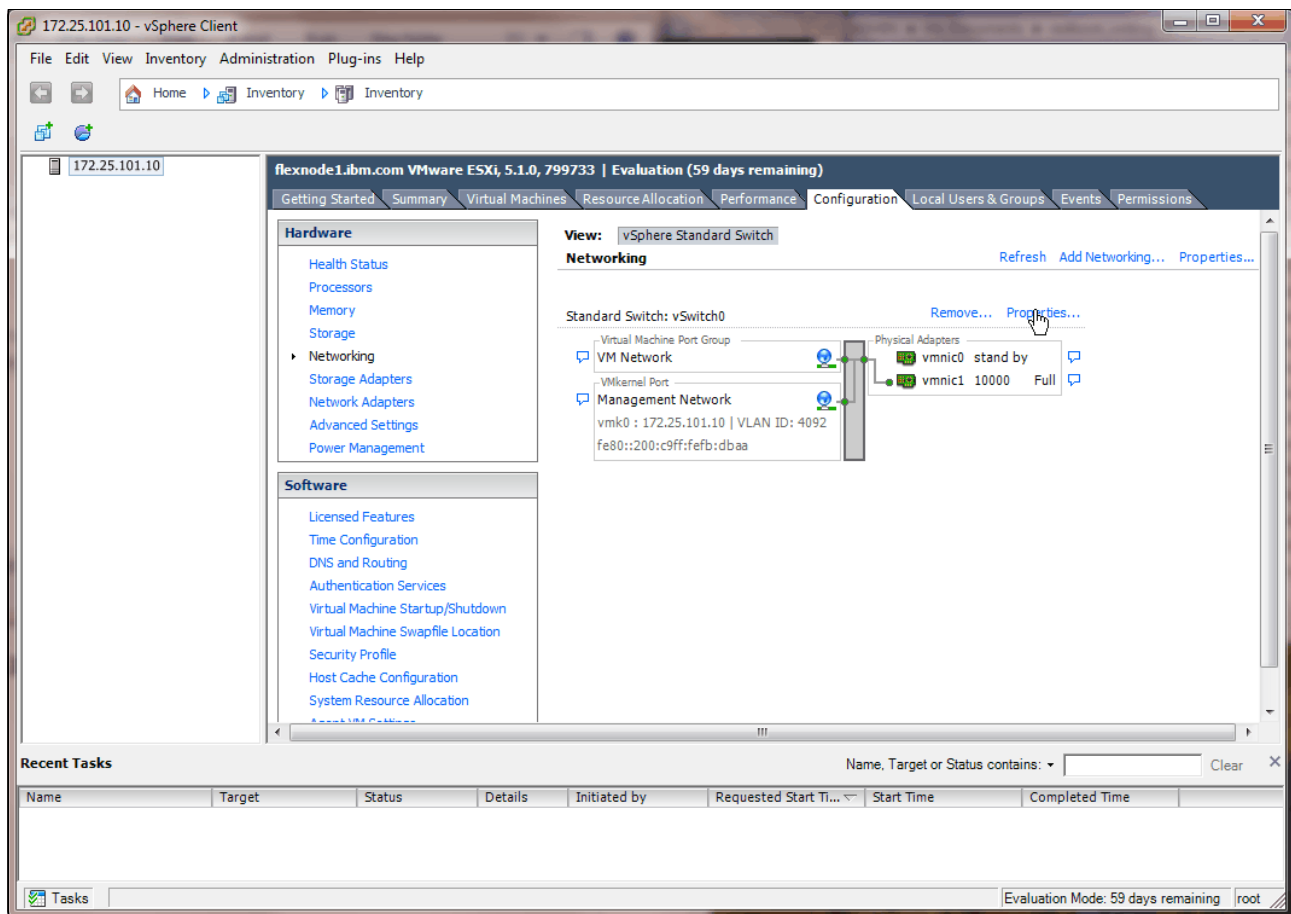


Figure 5-14 Configuration pane showing NIC card status

3. Open the configuration page for vSwitch0 by clicking **Edit**, as shown in Figure 5-15. We want both vmnic0 and vmnic1 to be active adapters in a team.

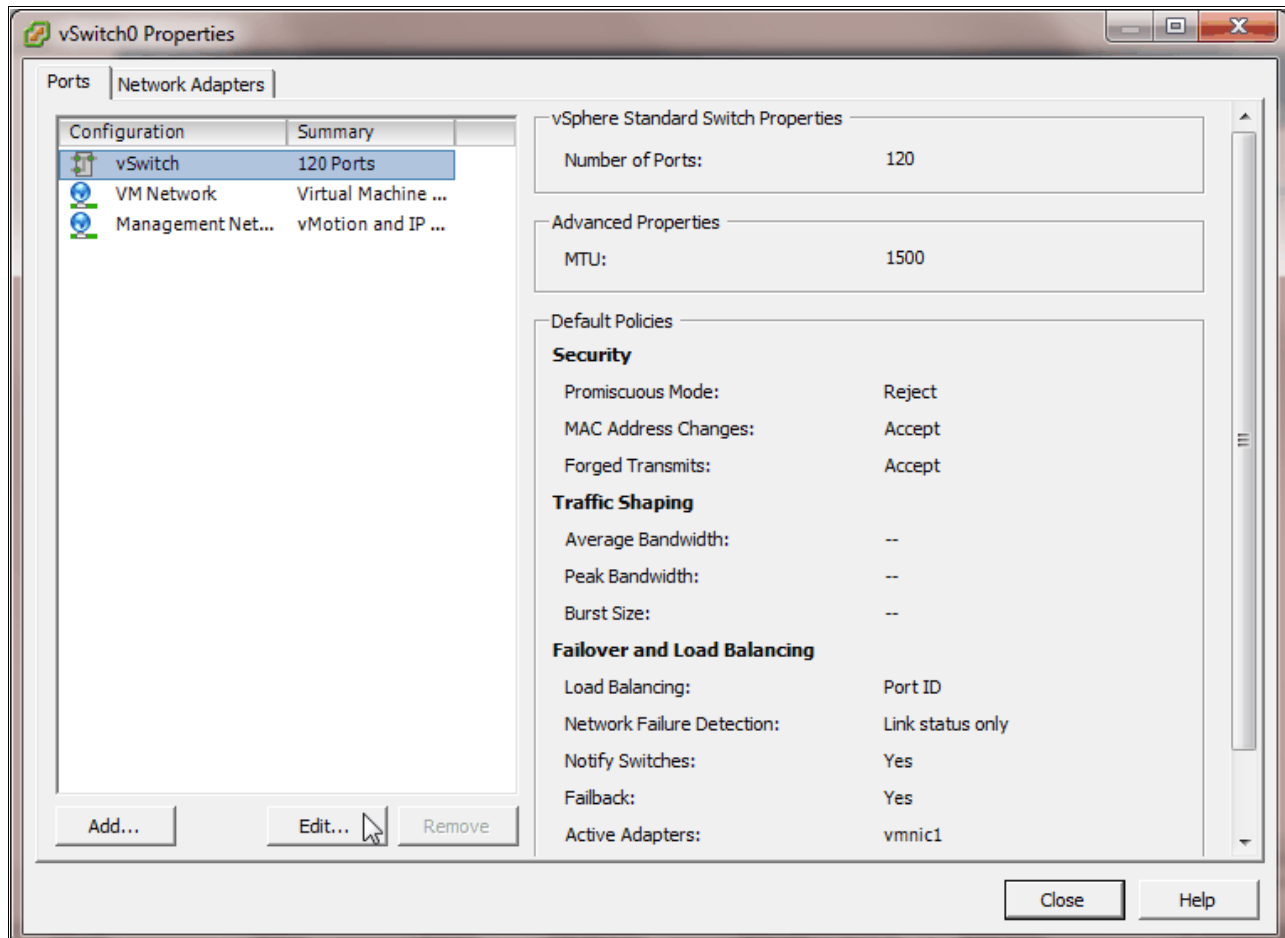


Figure 5-15 Editing vSwitch0 properties

4. Move the NIC currently in the standby position up to the Active Adapters category and then click **OK**, as shown in Figure 5-16.

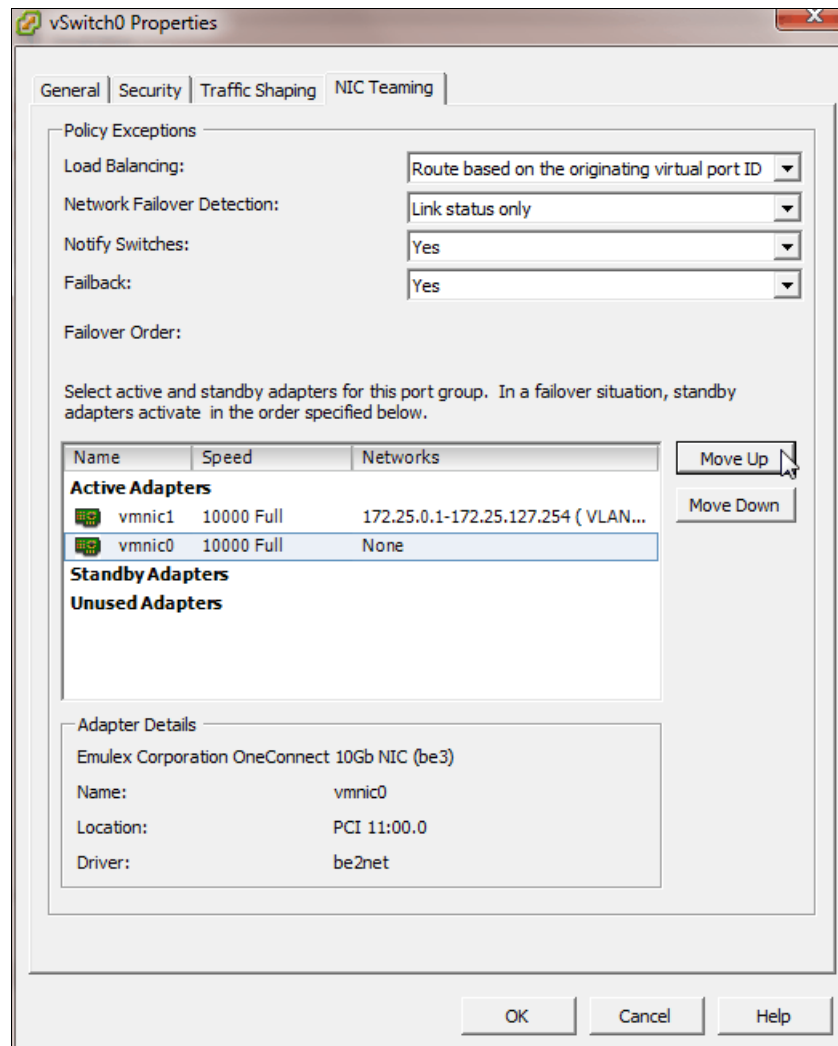


Figure 5-16 Moving one of the vmnic's to active to become an active adapter

- Go back to the Configuration panel and notice that both physical adapters are now active and connected to vSwitch0, as shown in Figure 5-17. Click the **Properties** field underneath the vSphere Standard Switch view to enable IPv6.

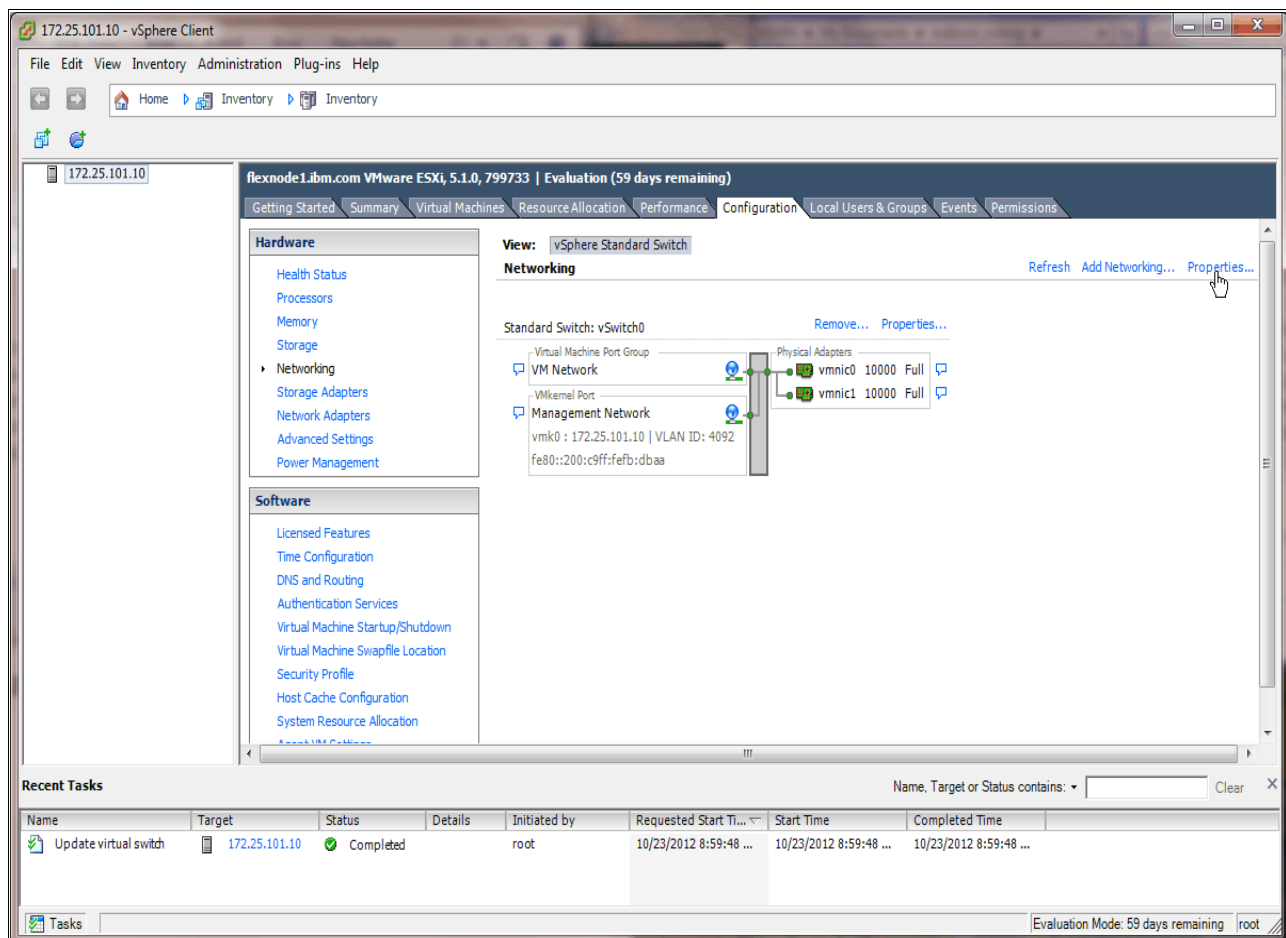


Figure 5-17 Verifying adapter configuration, ready to turn on IPv6

- Select **Enable IPv6 support on this host system**, and click **OK**, as shown in Figure 5-18.

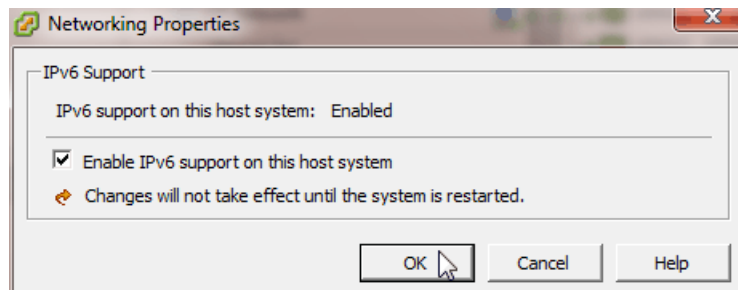


Figure 5-18 Enabling IPv6

7. Go back into editing vSwitch0 by clicking **Properties...** once more. We can set the Management IP addresses for the hypervisor for both IPv4 and IPv6. Click **Management Network** and then **Edit...**, as shown in Figure 5-19.

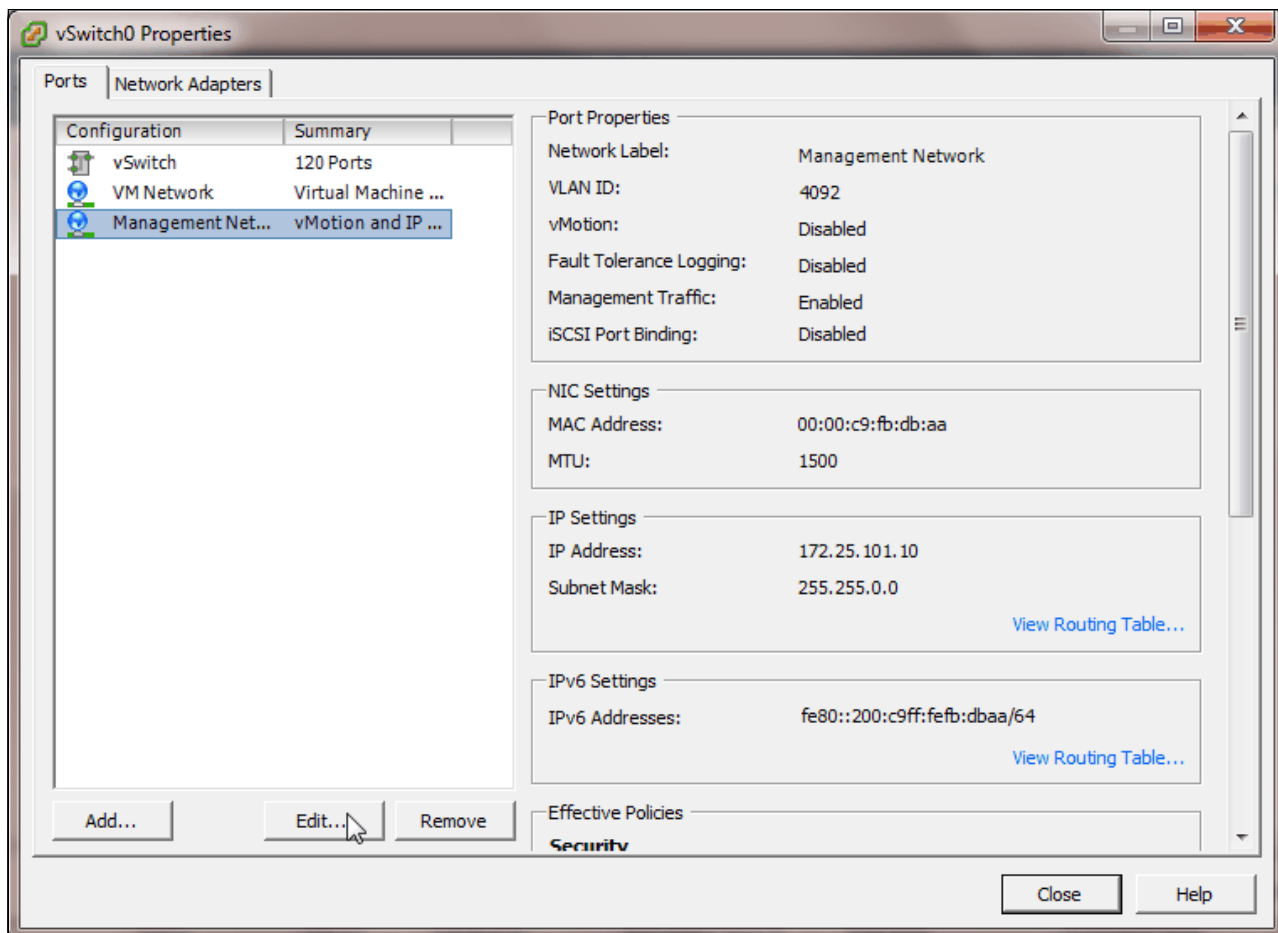


Figure 5-19 Modifying management network properties for vSwitch0

8. Set the IPv4 and IPv6 addresses in the following window, as indicated in Figure 5-20. Ensure that you also tag VLAN 4092 by inputting “4092” into the VLAN ID field on the General tab of Management Network Properties, not shown directly in Figure 5-20.

Caution: Ensure that you use a different VLAN for management traffic in VMWare. Do not use this management VLAN (4092) for data traffic going to the compute nodes because these should always be on separate VLANs, per best practice guidelines.

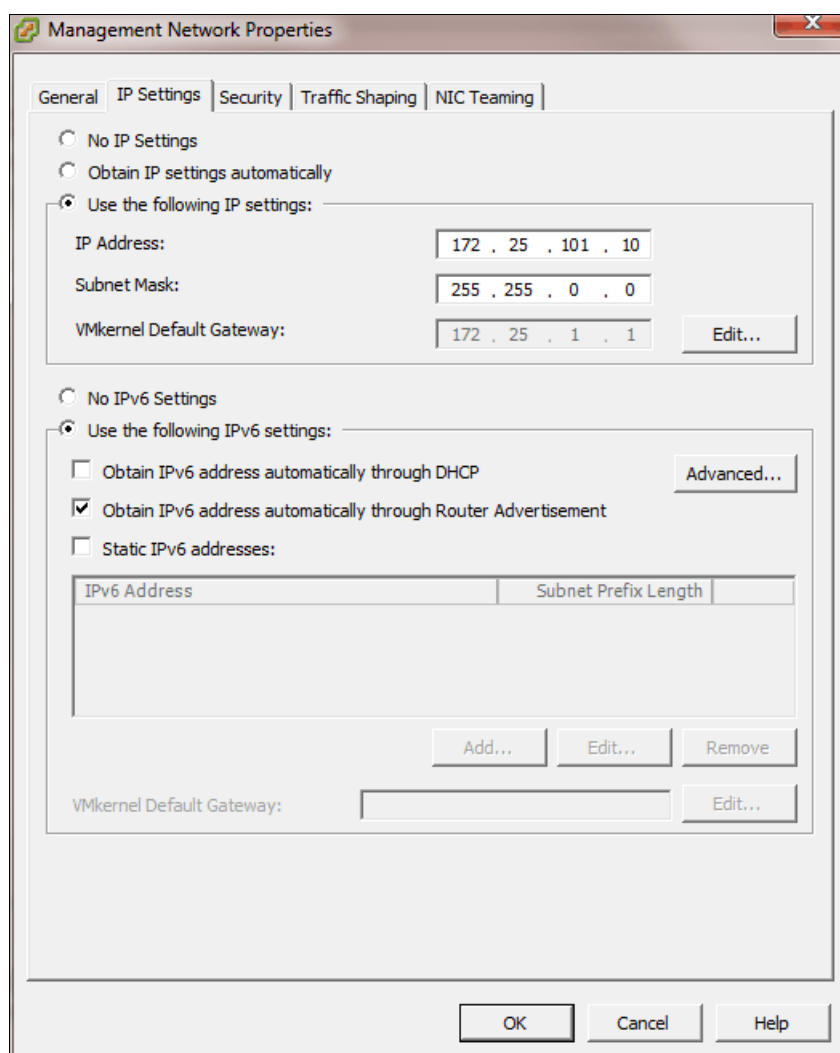


Figure 5-20 Management network IP address configuration

You are now done configuring a VMWare ESXi 5 hypervisor for active-backup NIC teaming.



Implementation of IBM PureFlex Systems and IBM System Networking connectivity

This chapter is a result of a practical configuration and implementation of the connectivity between the IBM Flex System Enterprise Chassis and the IBM System Networking switches portfolio.

This is a best practice recommendation.

6.1 Introduction

We introduced the concepts that have prepared you to take the step of the practical implementation. We describe a straightforward way of connecting System Networking switches and the PureFlex chassis to ensure implementation success.

Figure 6-1 is a representation of the environment that we are setting up.

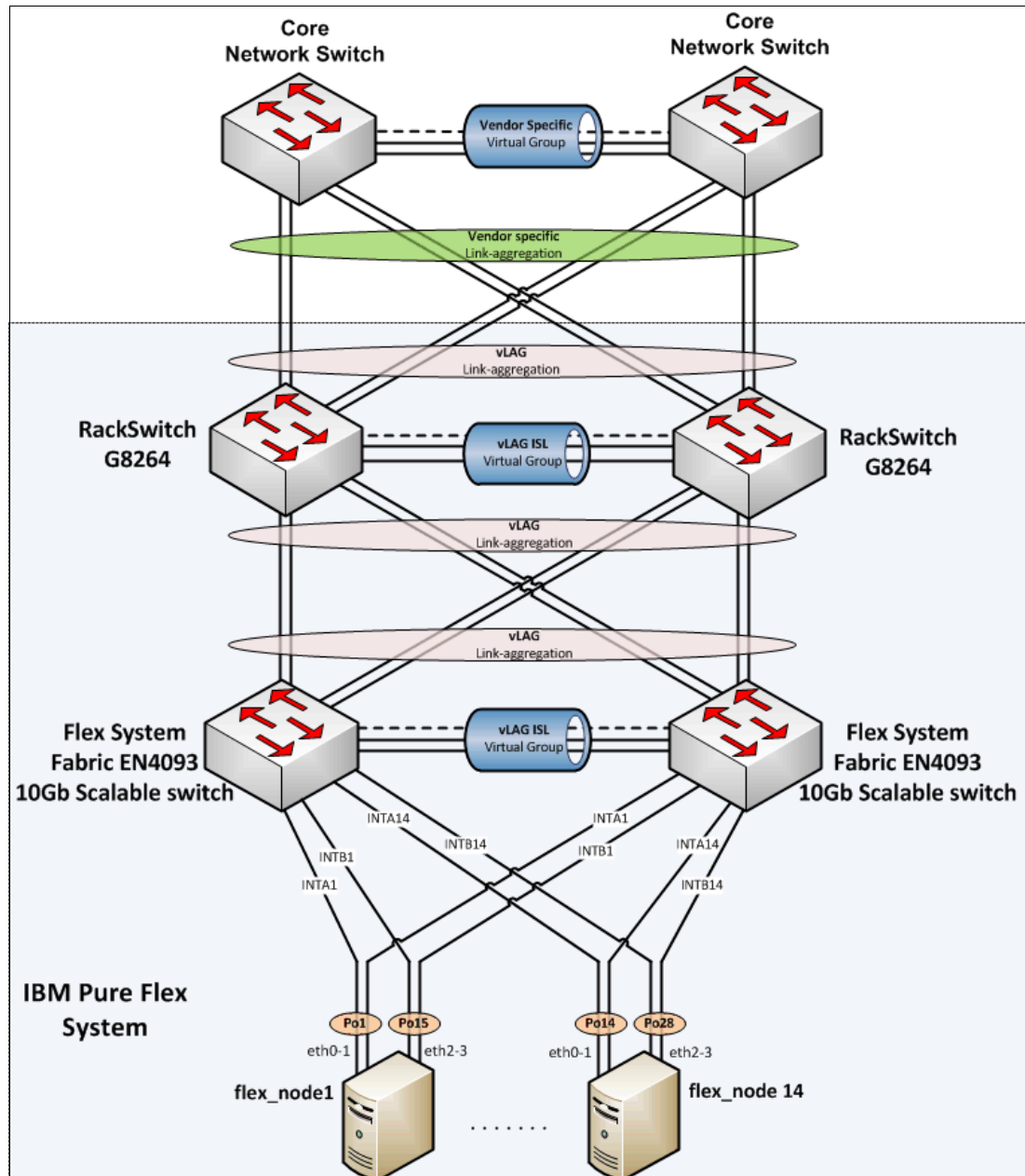


Figure 6-1 Our intended environment

The upstream network is not considered in this book.

6.2 IBM System Networking components used

- ▶ IBM G8264 RackSwitch (quantity 2)
- ▶ IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch (quantity 2)

6.3 Physical setup

Start by verifying the physical cabling between the EN4093/EN4093R and G8264 switches. In our lab environment, we used four IBM QSFP+ DAC Break Out Cables from the EN4093/EN4093R switches to the upstream G8264 switches. This requires the EN4093/EN4093R switches to be licensed for these particular features so that the ports can be used:

- ▶ Four 1-m IBM QSFP+-to-QSFP+ Cables were used to form the 160-Gb inter-switch link (ISL) between the G8264 switches.
- ▶ 10 Gb enhanced small form-factor pluggable (SFP+) DAC cables were used for all other connections in the diagram.

6.4 EN4093flex_1 configuration

We begin the implementation of this scenario on the IBM Flex System Fabric EN4093/EN4093R switches, working our way northward on the diagram in Figure 6-1 on page 112. Each step provides the commands necessary and is reflective of the numbering schema in the diagram to aid the user in what is being configured.

General configuration

1. Begin by creating the ISL Health Check, ISL data, and data VLANs, as shown in Example 6-1, giving them descriptive names, assigning them to spanning-tree groups, and enabling them. You can elect to let the switch itself create Spanning Tree Protocol (STP) instances for you. We have chosen to manually create them instead.

Example 6-1 Create ISL hlthchk, DATA, and ISL vlans on EN4093flex_1

```
configure terminal
vlan 4000
    enable
    name "ISL hlthchk"
    stg 125
    exit
vlan 4092
    enable
    name "DATA"
    stg 126
    exit
vlan 4094
    enable
    name "ISL"
    stg 127
    exit
```

2. Assign IP addresses for both the ISL Health Check and data VLANs that are shown in Example 6-2 so that we can verify connectivity between the various pieces of equipment when verifying the configuration. In this example, interface ip 40 represents the virtual Link Aggregation Group (vLAG) Health Check IP address, and interface ip 92 represents an address on the data VLAN that uses the prefix 10.1.4, with the last octet being borrowed from the network diagram's management address to quickly aid in the identification of which piece of equipment we are verifying connectivity to.

Example 6-2 Create IP interfaces and assign vlans and IP addresses on EN4093flex_1

```
configure terminal
interface ip 40
    ip address 1.1.1.1 255.255.255.0
    vlan 4000
    enable
    exit
interface ip 92
    ip address 10.1.4.238 255.255.255.0
    vlan 4092
    enable
    exit
```

Step [1]: Configure ISL between EN4093flex switches

3. Configure the eventual ISL in Example 6-3 between the EN4093/EN4093R switches by configuring them to have a default (untagged) VLAN of 4094, Link Aggregation Control Protocol (LACP) key of 1000 to bundle the ports together in an aggregation, with 802.1q tagging enabled so that L2 VLAN traffic can traverse the ISL. Carry data VLAN 4092 over these links.

Example 6-3 Initial ISL configuration on EN4093flex_1, step [1]

```
configure terminal
interface port ext7-ext10
    pvid 4094
    tagging
    exit
vlan 4092
    member ext7-ext10
    exit
interface port ext7-ext10
    lacp key 1000
    lacp mode active
    exit
```

4. Create the dedicated Health Check VLAN and physical interface in Example 6-4 to be used for heartbeats between the EN4093/EN4093R switches. We have chosen EXT4 as a dedicated interface and VLAN 4000 to serve as the Health Check for the ISL.

Example 6-4 Create vLAG Health Check on EN4093flex_1, step [1] continued

```
configure terminal
vlan 4000
    name "ISL hlthchk"
    enable
    exit
interface port ext4
```



```
pvid 4000
exit
```

5. Disable STP between the EN4093/EN4093R switches and activate a vLAG between them so that they appear as a single entity to upstream and downstream infrastructure, as shown in Example 6-5, referencing the LACP key configured in the previous step.

Example 6-5 Disable STP and activate ISL vLAG on EN4093flex_1, step [1] continued

```
configure terminal
no spanning-tree stp 127 enable
vlag tier-id 1
vlag isl vlan 4094
vlag isl adminkey 1000
vlag hlthchk peer-ip 1.1.1.2
vlag enable
```

Step [2]: Configure downstream internal node ports

6. Configure downstream node interfaces in Example 6-6 to have a default (untagged) VLAN of 4092, with 802.1q tagging enabled. Add the ability for all member ports to be on VLAN 4092.

Example 6-6 Downstream internal node port configuration on EN4093flex, step [2]

```
configure terminal
interface port inta1-intb14
    pvid 4092
    tagging
    spanning-tree edge
    exit
vlan 4092
    member inta1-intb14
exit
```

7. For redundancy, we have created two port-channels on each of the 14 nodes. Each port-channel aggregates two ports, one from each EN4093flex switch: port-channels 1 - 14 to match the “A” internally labeled ports, and port-channels 15 - 28 to match the “B” ports. See Example 6-7.

Example 6-7 Node-facing port channel creation and vLAG activation on EN4093flex_1, step [2] continued

```
configure terminal
portchannel 1 port inta1
portchannel 1 enable
vlag portchannel 1 enable
portchannel 15 port intb1
portchannel 15 enable
vlag portchannel 15 enable
portchannel 2 port inta2
portchannel 2 enable
vlag portchannel 2 enable
portchannel 16 port intb2
portchannel 16 enable
vlag portchannel 16 enable
portchannel 3 port inta3
```

```
portchannel 3 enable
vlag portchannel 3 enable
portchannel 17 port intb3
portchannel 17 enable
vlag portchannel 17 enable
portchannel 4 port inta4
portchannel 4 enable
vlag portchannel 4 enable
portchannel 18 port intb4
portchannel 18 enable
vlag portchannel 18 enable
portchannel 5 port inta5
portchannel 5 enable
vlag portchannel 5 enable
portchannel 19 port intb5
portchannel 19 enable
vlag portchannel 19 enable
portchannel 6 port inta6
portchannel 6 enable
vlag portchannel 6 enable
portchannel 20 port intb6
portchannel 20 enable
vlag portchannel 20 enable
portchannel 7 port inta7
portchannel 7 enable
vlag portchannel 7 enable
portchannel 21 port intb7
portchannel 21 enable
vlag portchannel 21 enable
portchannel 8 port inta8
portchannel 8 enable
vlag portchannel 8 enable
portchannel 22 port intb8
portchannel 22 enable
vlag portchannel 22 enable
portchannel 9 port inta9
portchannel 9 enable
vlag portchannel 9 enable
portchannel 23 port intb9
portchannel 23 enable
vlag portchannel 23 enable
portchannel 10 port inta10
portchannel 10 enable
vlag portchannel 10 enable
portchannel 24 port intb10
portchannel 24 enable
vlag portchannel 24 enable
portchannel 11 port inta11
portchannel 11 enable
vlag portchannel 11 enable
portchannel 25 port intb11
portchannel 25 enable
vlag portchannel 25 enable
portchannel 12 port inta12
portchannel 12 enable
```

```
vlag portchannel 12 enable
portchannel 26 port intb12
portchannel 26 enable
vlag portchannel 26 enable
portchannel 13 port inta13
portchannel 13 enable
vlag portchannel 13 enable
portchannel 27 port intb13
portchannel 27 enable
vlag portchannel 27 enable
portchannel 14 port inta14
portchannel 14 enable
vlag portchannel 14 enable
portchannel 28 port intb14
portchannel 28 enable
vlag portchannel 28 enable
```

Step [3]: Configure upstream, G8264tor facing ports and layer 2 failover

8. Configure the upstream ports with a default (untagged) VLAN of 4092 (data VLAN), tag the Port VLAN ID (PVID), and use an LACP key of 2000 to bundle the ports together. See Example 6-8.

Example 6-8 Upstream G8264tor facing ports configuration on EN4093flex_1, step [3]

```
configure terminal
interface port ext15-ext22
    pvid 4092
    tagging
    tag-pvid
    exit
vlan 4092
    member ext15-ext22
    exit
interface port ext15-ext22
    lacp key 2000
    lacp mode active
    exit
```

9. Activate the vLAG feature for the upstream EN4093/EN4093R ports so that the G8264s see the EN4093/EN4093Rs as a single, virtualized entity, as shown in Example 6-9. Use adminkey 2000, which represents the LACP key bundling ports EXT15-22 together as one.

Example 6-9 Activating the upstream G8264tor-facing vLAG on EN4093flex_1, step [3] continued

```
configure terminal
vlag adminkey 2000 enable
```

10. Enable layer 2 failover as shown in Example 6-10, which effectively shuts down the links to the compute nodes if the uplinks for the EN4093/EN4093R switch fail. This guarantees that the downstream node is aware of the upstream failure and can fail traffic over to the other network interface card (NIC) in the node. In our case, it is connected to the other EN4093/EN4093R switch in the Enterprise Chassis, ensuring that redundancy is maintained.

Example 6-10 Enable layer 2 failover for the compute nodes on EN4093flex_1, step [3] continued

```
configure terminal
failover trigger 1 mmon monitor admin-key 2000
failover trigger 1 mmon control member INTA1-INTB14
failover trigger 1 enable
failover enable
```

Now, repeat this configuration for EN4093_flex2 on the other I/O module. The only difference between the EN4093flex_1 switch and EN4093flex_2 switch is the vLAG Health Check peer address, and the data and ISL hlthchk vlan ip addresses. To verify the EN4093flex switch configuration, run the show commands outlined in detail in section 6.6, “Verification and show command output” on page 120.

6.5 G8264tor_1 configuration

Next, is the configuration of the RackSwitch G8264.

General configuration

1. Begin by creating the ISL Health Check, ISL data, and data VLANs, as shown in Example 6-11, giving them descriptive names, assigning them to spanning-tree groups, and enabling them.

Example 6-11 Create ISL hlthchk, data, and ISL VLANs on G8264tor_1

```
configure terminal
vlan 4000
    enable
    name "ISL hlthchk"
    stg 125
    exit
vlan 4092
    enable
    name "Data"
    stg 126
    exit
vlan 4094
    enable
    name "ISL"
    stg 127
    exit
```

2. Assign IP addresses for the ISL Health Check, data VLANs, and management VLAN, as shown in Example 6-12 on page 119. Interface ip 128 represents the management IP address that is referenced in the Network Topology diagram that is shown in Figure 6-1 on page 112. IP gateway 4 is the upstream router interface for the 172 management network.

Example 6-12 Create IP interfaces and assign VLANs and IP addresses; configure the management interface on G8264tor_1

```
configure terminal
interface ip 40
    ip address 1.1.1.1 255.255.255.0
    vlan 4000
    enable
    exit
interface ip 92
    ip address 10.1.4.243 255.255.255.0
    vlan 4092
    enable
    exit
interface ip 128
    ip address 172.25.101.243 255.255.0.0
    enable
    exit
ip gateway 4 address 172.25.1.1
ip gateway 4 enable
```

Step [4]: Configure ISL between G8264tor switches

3. Configure the ISL between the G8264 switches. See Example 6-13. Make the default (untagged) VLAN 4094 (ISL hlthchk). Assign LACP key of 1000 to bundle the ports together in an aggregation, with 802.1q tagging enabled so that L2 VLAN traffic can traverse the ISL. Allow VLAN 4092 (data VLAN) over these links.

Example 6-13 Initial ISL configuration on G8264tor_1, step [4]

```
configure terminal
interface port 1-16
    pvid 4094
    tagging
    exit
vlan 4092
    member 1-16
    exit
interface port 1-16
    lacp key 1000
    lacp mode active
    exit
```

4. Disable STP between the G8264 switches and activate a vLAG between them so that they appear as a single entity to upstream and downstream infrastructure, as shown in Example 6-14, referencing the LACP key configured in the previous step.

Example 6-14 Disable STP and activate ISL vLAG on G8264tor_1, step [4] continued

```
configure terminal
no spanning-tree stp 127 enable
vlag tier-id 2
vlag isl vlan 4094
vlag isl adminkey 1000
vlag hlthchk peer-ip 1.1.1.2
vlag enable
```

Step [5]: Configure downstream EN4093flex facing ports

5. Configure the downstream EN4093flex facing ports. See Example 6-15. Make the default (untagged) VLAN 4092 (data VLAN), with 802.1q tagging enabled. Add the ability for all member ports to be on VLAN 4092.

Example 6-15 Configure downstream EN4093flex facing ports on G8264tor_1, step [5]

```
configure terminal
interface port 25-28,37-40
    pvid 4092
    tagging
    tag-pvid
    exit
vlan 4092
    member 25-28,37-40
    exit
interface port 25-28,37-40
    lacp key 2002
    lacp mode active
    exit
```

6. Activate the vLAG feature for the downstream EN4093flex facing ports so that the EN4093/EN4093Rs see the G8264s as a single, virtualized entity, as shown in Example 6-16. Use adminkey 2002, which represents the LACP key bundling ports 25 - 28, and 37 - 40 together as one.

Example 6-16 Activate downstream EN4093flex facing vLAG on G8264tor_1, step [5] continued

```
configure terminal
vlag adminkey 2002 enable
```

Now, repeat this configuration for G8264tor_2. The only difference between the G8264tor_1 switch and the G8264tor_2 switch is the vLAG Health Check peer address and the data, management, and ISL hlthchk vlan ip addresses. To verify the G8264tor switch configuration, run the **show** commands outlined in detail in section 6.6, “Verification and show command output” on page 120.

6.6 Verification and show command output

The following section lists output from common **show** commands that might aid the network architect in the implementation of the preceding scenario. Ping verification of the various IP addresses configured on the equipment for the data VLAN is also done to show that all of the devices can reach each other successfully.

As illustrated in the implementation section, we begin by showing helpful commands from the EN4093/EN4093R switches, working our way up the Network Topology diagram that is shown in Figure 6-1 on page 112, all the way to the IBM G8264 RackSwitch pair.

EN4093/EN4093R output

Here, we list output from the switch with host name, EN4093flex_1. Similar or identical output exists for the switch with host name, EN4093flex_2.

Show version output

Example 6-17 command output shows information regarding the switch that we used, and the associated code and firmware level at the time.

Example 6-17 EN4093flex_1 show version output

```
System Information at 23:04:56 Fri Oct 12, 2012
Time zone: No timezone configured
Daylight Savings Time Status: Disabled

IBM Flex System Fabric EN4093 10Gb Scalable Switch

Switch has been up for 1 day, 2 hours, 1 minute and 21 seconds.
Last boot: 21:05:54 Thu Oct 11, 2012 (reset from Telnet/SSH)

MAC address: 6c:ae:8b:bf:6d:00    IP (If 40) address: 1.1.1.1
Internal Management Port MAC Address: 6c:ae:8b:bf:6d:ef
Internal Management Port IP Address (if 128): 172.25.101.238
External Management Port MAC Address: 6c:ae:8b:bf:6d:fe
External Management Port IP Address (if 127):
Software Version 7.3.1.0          (FLASH image1), active configuration.


Hardware Part Number      : 49Y4272
Hardware Revision        : 02
Serial Number            : Y250VT24M099
Manufacturing Date (WWYY) : 1712
PCBA Part Number         : BAC-00072-01
PCBA Revision            : 0
PCBA Number              : 00
Board Revision           : 02
PLD Firmware Version     : 1.5


Temperature Warning       : 32 C (Warn at 60 C/Recover at 55 C)
Temperature Shutdown     : 32 C (Shutdown at 65 C/Recover at 60 C)
Temperature Inlet        : 27 C
Temperature Exhaust       : 33 C


Power Consumption         : 54.300 W (12.244 V,  4.435 A)

Switch is in I/O Module Bay 1
```

Show vlan output

Example 6-18 shows output regarding VLAN assignment for all the various ports on the switch.

Example 6-18 EN4093flex_1 show vlan output

VLAN	Name	Status	MGT	Ports
----	-----	-----	-----	-----
1	Default VLAN	ena	dis	INTA2-INTA14 INTB2-INTB14 EXT1-EXT3 EXT5 EXT6
4000	ISL hlthchk	ena	dis	EXT4
4092	DATA	ena	dis	INTA1 INTB1 EXT7-EXT10 EXT15-EXT22

4094	ISL	ena	dis	EXT7-EXT10
4095	Mgmt VLAN	ena	ena	EXTM MGT1

Show interface status

Because we have only one compute node in the chassis (in slot 1), this explains why all the other internal ports are listed as down from a link perspective in the following output shown in Example 6-19.

Example 6-19 EN4093flex_1 show interface status output

Alias	Port	Speed	Duplex	Flow Ctrl		Link	Name
				--TX--	--RX--		
INTA1	1	1000	full	no	no	up	INTA1
INTA2	2	1G/10G	full	yes	yes	down	INTA2
INTA3	3	1G/10G	full	yes	yes	down	INTA3
INTA4	4	1G/10G	full	yes	yes	down	INTA4
INTA5	5	1G/10G	full	yes	yes	down	INTA5
INTA6	6	1G/10G	full	yes	yes	down	INTA6
INTA7	7	1G/10G	full	yes	yes	down	INTA7
INTA8	8	1G/10G	full	yes	yes	down	INTA8
INTA9	9	1G/10G	full	yes	yes	down	INTA9
INTA10	10	1G/10G	full	yes	yes	down	INTA10
INTA11	11	1G/10G	full	yes	yes	down	INTA11
INTA12	12	1G/10G	full	yes	yes	down	INTA12
INTA13	13	1G/10G	full	yes	yes	down	INTA13
INTA14	14	1G/10G	full	yes	yes	down	INTA14
INTB1	15	1000	full	no	no	up	INTB1
INTB2	16	1G/10G	full	yes	yes	down	INTB2
INTB3	17	1G/10G	full	yes	yes	down	INTB3
INTB4	18	1G/10G	full	yes	yes	down	INTB4
INTB5	19	1G/10G	full	yes	yes	down	INTB5
INTB6	20	1G/10G	full	yes	yes	down	INTB6
INTB7	21	1G/10G	full	yes	yes	down	INTB7
INTB8	22	1G/10G	full	yes	yes	down	INTB8
INTB9	23	1G/10G	full	yes	yes	down	INTB9
INTB10	24	1G/10G	full	yes	yes	down	INTB10
INTB11	25	1G/10G	full	yes	yes	down	INTB11
INTB12	26	1G/10G	full	yes	yes	down	INTB12
INTB13	27	1G/10G	full	yes	yes	down	INTB13
INTB14	28	1G/10G	full	yes	yes	down	INTB14
EXT1	43	10000	full	no	no	up	EXT1
EXT2	44	10000	full	no	no	up	EXT2
EXT3	45	10000	full	no	no	up	EXT3
EXT4	46	10000	full	no	no	up	ISL h1thchk
EXT5	47	1G/10G	full	no	no	down	EXT5
EXT6	48	1G/10G	full	no	no	down	EXT6
EXT7	49	10000	full	no	no	up	ISL
EXT8	50	10000	full	no	no	up	ISL
EXT9	51	10000	full	no	no	up	ISL
EXT10	52	10000	full	no	no	up	ISL
EXT15	57	10000	full	no	no	up	Link to g8264tor_1
EXT16	58	10000	full	no	no	up	Link to g8264tor_1
EXT17	59	10000	full	no	no	up	Link to g8264tor_1
EXT18	60	10000	full	no	no	up	Link to g8264tor_1

EXT19	61	10000	full	no	no	up	Link to g8264tor_2
EXT20	62	10000	full	no	no	up	Link to g8264tor_2
EXT21	63	10000	full	no	no	up	Link to g8264tor_2
EXT22	64	10000	full	no	no	up	Link to g8264tor_2
EXTM	65	1000	half	yes	yes	down	EXTM
MGT1	66	1000	full	yes	yes	up	MGT1

Show lldp remote-device output

Example 6-20 command output illustrates the physical topology and verifies that cables are plugged into the ports that we have specified in both our Network Topology diagram (Figure 6-1 on page 112), and the configuration specified in Appendix A, “An integration guide to the IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch” on page 225.

Example 6-20 EN4093flex_1 show lldp remote-device output

LLDP Remote Devices Information

LocalPort	Index	Remote Chassis ID	Remote Port	Remote System Name
EXT16	3	08 17 f4 33 9d 00	25	G8264TOR-1
EXT15	4	08 17 f4 33 9d 00	26	G8264TOR-1
EXT18	5	08 17 f4 33 9d 00	27	G8264TOR-1
EXT17	6	08 17 f4 33 9d 00	28	G8264TOR-1
EXT21	7	08 17 f4 33 75 00	25	G8264TOR-2
EXT19	8	08 17 f4 33 75 00	26	G8264TOR-2
EXT22	9	08 17 f4 33 75 00	27	G8264TOR-2
EXT20	10	08 17 f4 33 75 00	28	G8264TOR-2
EXT4	12	6c ae 8b bf fe 00	46	en4093flex_2
EXT7	13	6c ae 8b bf fe 00	49	en4093flex_2
EXT8	14	6c ae 8b bf fe 00	50	en4093flex_2
EXT9	15	6c ae 8b bf fe 00	51	en4093flex_2
EXT10	16	6c ae 8b bf fe 00	52	en4093flex_2

Show vlag isl output

Example 6-21 shows command output regarding the status of the ISL between the EN4093/EN4093R switches, and the ports comprising the ISL itself.

Example 6-21 EN4093flex_1 show vlag isl output

ISL_ID	ISL_Vlan	ISL_Trunk	ISL_Members	Link_State	Trunk_State
65	4094	Adminkey 1000	EXT7	UP	UP
			EXT8	UP	UP
			EXT9	UP	UP
			EXT10	UP	UP

Show vlag information output

Example 6-22 command output shows that the vLAG between the EN4093/EN4093R switches and G8264 switches is up and operational as referenced by the LACP admin key of 2000. Our ISL between the EN4093/EN4093R switches is up as well.

EN4093flex_1 is acting as the admin and operational role of PRIMARY. For centralized vLAG functions, such as vLAG STP, one of the vLAG switches must control the protocol operations. To select the switch that controls the centralized vLAG function, role election is performed. The switch with the primary role controls the centralized operation. Role election is non-preemptive. That is, if there exists a *primary*, another switch coming up remains as *secondary*, even if it can become primary based on the role election logic.

Role election is determined by comparing the local vLAG system priority and local system Media Access Control (MAC) address. The switch with the smaller priority value is the vLAG primary switch. If priority is the same, the switch with the smaller system MAC address is the vLAG primary switch. It is possible to configure vLAG priority to anything between 0 - 65535. Priority was left at the default value of 0 in all examples.

Example 6-22 EN4093flex_1 show vlag information output

```
vLAG Tier ID: 1
vLAG system MAC: 08:17:f4:c3:dd:00
Local MAC 6c:ae:8b:bf:6d:00 Priority 0 Admin Role PRIMARY (Operational Role
PRIMARY)
Peer MAC 6c:ae:8b:bf:fe:00 Priority 0
Health local 1.1.1.1 peer 1.1.1.2 State UP
ISL trunk id 65
ISL state Up
Startup Delay Interval: 120s (Finished)
```

```
vLAG 65: config with admin key 2000, associated trunk 66, state formed
```

Show vlag adminkey 2000 output

Example 6-23 output shows that the vLAG is formed and enabled by using LACP reference key 2000.

Example 6-23 EN4093flex_1 show vlag adminkey 2000 output

```
vLAG is enabled on admin key 2000
Current LACP params for EXT15: active, Priority 32768, Admin Key 2000, Min-Links 1

Current LACP params for EXT16: active, Priority 32768, Admin Key 2000, Min-Links 1

Current LACP params for EXT17: active, Priority 32768, Admin Key 2000, Min-Links 1

Current LACP params for EXT18: active, Priority 32768, Admin Key 2000, Min-Links 1

Current LACP params for EXT19: active, Priority 32768, Admin Key 2000, Min-Links 1

Current LACP params for EXT20: active, Priority 32768, Admin Key 2000, Min-Links 1

Current LACP params for EXT21: active, Priority 32768, Admin Key 2000, Min-Links 1

Current LACP params for EXT22: active, Priority 32768, Admin Key 2000, Min-Links 1
```

Show lacp information state up

Example 6-24 command output shows which ports are participating in an LACP aggregation, and which reference keys are used on those specific interfaces.

Example 6-24 EN4093flex_1 show lacp information state up

port	mode	adminkey	operkey	selected	prio	aggr	trunk	status	minlinks
EXT7	active	1000	1000	yes	32768	49	65	up	1
EXT8	active	1000	1000	yes	32768	49	65	up	1
EXT9	active	1000	1000	yes	32768	49	65	up	1
EXT10	active	1000	1000	yes	32768	49	65	up	1
EXT15	active	2000	2000	yes	32768	57	66	up	1
EXT16	active	2000	2000	yes	32768	57	66	up	1
EXT17	active	2000	2000	yes	32768	57	66	up	1
EXT18	active	2000	2000	yes	32768	57	66	up	1
EXT19	active	2000	2000	yes	32768	57	66	up	1
EXT20	active	2000	2000	yes	32768	57	66	up	1
EXT21	active	2000	2000	yes	32768	57	66	up	1
EXT22	active	2000	2000	yes	32768	57	66	up	1

Show failover trigger 1 output

Failover output showing which ports are monitored, and which ports will be shut down if an issue is encountered, is shown in Example 6-25. In our case, our upstream to G8264 links is monitored with LACP reference key 2000. Our control ports are the downstream internal I/O module ports that are used by the compute nodes.

Example 6-25 EN4093flex_1 show failover output

```
Failover: On
VLAN Monitor: OFF

Trigger 1 Manual Monitor: Enabled
Trigger 1 limit: 0
Monitor State: Up
Member      Status
-----
adminkey 2000
EXT15      Operational
EXT16      Operational
EXT17      Operational
EXT18      Operational
EXT19      Operational
EXT20      Operational
EXT21      Operational
EXT22      Operational
Control State: Auto Controlled
Member      Status
-----
INTA1      Operational
INTA2      Operational
INTA3      Operational
INTA4      Operational
INTA5      Operational
INTA6      Operational
INTA7      Operational
```

INTA8	Operational
INTA9	Operational
INTA10	Operational
INTA11	Operational
INTA12	Operational
INTA13	Operational
INTA14	Operational
INTB1	Operational
INTB2	Operational
INTB3	Operational
INTB4	Operational
INTB5	Operational
INTB6	Operational
INTB7	Operational
INTB8	Operational
INTB9	Operational
INTB10	Operational
INTB11	Operational
INTB12	Operational
INTB13	Operational
INTB14	Operational

Trigger 2: Disabled

Trigger 3: Disabled

Trigger 4: Disabled

Trigger 5: Disabled

Trigger 6: Disabled

Trigger 7: Disabled

Trigger 8: Disabled

Ping output for equipment on VLAN 4092

To verify connectivity, we issued **ping** commands to devices in the lab infrastructure on VLAN 4092 (data VLAN) in Example 6-26. IP address 10.4.1.10 represents a compute node with an operating system installed (flex_node1) on the Network Topology diagram (Figure 6-1 on page 112).

Example 6-26 Ping verification for equipment on VLAN 4092

```
en4093flex_1#ping 10.1.4.10 data-port
Connecting via DATA port.
[host 10.1.4.10, max tries 5, delay 1000 msec, length 0, ping source N/S, ttl 255,
tos 0]
10.1.4.10: #1 ok, RTT 1 msec.
10.1.4.10: #2 ok, RTT 0 msec.
10.1.4.10: #3 ok, RTT 1 msec.
10.1.4.10: #4 ok, RTT 0 msec.
10.1.4.10: #5 ok, RTT 0 msec.
Ping finished.
```

```
en4093flex_1#ping 10.1.4.239 data-port
Connecting via DATA port.
[host 10.1.4.239, max tries 5, delay 1000 msec, length 0, ping source N/S, ttl
255, tos 0]
10.1.4.239: #1 ok, RTT 4 msec.
10.1.4.239: #2 ok, RTT 1 msec.
10.1.4.239: #3 ok, RTT 2 msec.
10.1.4.239: #4 ok, RTT 3 msec.
10.1.4.239: #5 ok, RTT 1 msec.
Ping finished.
```

```
en4093flex_1#ping 10.1.4.243 data-port
Connecting via DATA port.
[host 10.1.4.243, max tries 5, delay 1000 msec, length 0, ping source N/S, ttl
255, tos 0]
10.1.4.243: #1 ok, RTT 1 msec.
10.1.4.243: #2 ok, RTT 1 msec.
10.1.4.243: #3 ok, RTT 2 msec.
10.1.4.243: #4 ok, RTT 8 msec.
10.1.4.243: #5 ok, RTT 6 msec.
Ping finished.
```

```
en4093flex_1#ping 10.1.4.244 data-port
Connecting via DATA port.
[host 10.1.4.244, max tries 5, delay 1000 msec, length 0, ping source N/S, ttl
255, tos 0]
10.1.4.244: #1 ok, RTT 1 msec.
10.1.4.244: #2 ok, RTT 2 msec.
10.1.4.244: #3 ok, RTT 1 msec.
10.1.4.244: #4 ok, RTT 2 msec.
10.1.4.244: #5 ok, RTT 0 msec.
Ping finished.
```

```
en4093flex_1#ping 10.1.4.249 data-port
Connecting via DATA port.
[host 10.1.4.241, max tries 5, delay 1000 msec, length 0, ping source N/S, ttl
255, tos 0]
10.1.4.241: #1 ok, RTT 2 msec.
10.1.4.241: #2 ok, RTT 1 msec.
10.1.4.241: #3 ok, RTT 2 msec.
10.1.4.241: #4 ok, RTT 1 msec.
10.1.4.241: #5 ok, RTT 3 msec.
Ping finished.
```

```
en4093flex_1#ping 10.1.4.200 data-port
Connecting via DATA port.
[host 10.1.4.241, max tries 5, delay 1000 msec, length 0, ping source N/S, ttl
255, tos 0]
10.1.4.241: #1 ok, RTT 2 msec.
10.1.4.241: #2 ok, RTT 2 msec.
10.1.4.241: #3 ok, RTT 2 msec.
10.1.4.241: #4 ok, RTT 1 msec.
10.1.4.241: #5 ok, RTT 3 msec.
Ping finished
```

G8264 output

Here, we list output from the switch with host name G8264tor_1. Similar or identical output exists for the switch with host name G8264tor_2.

Show version output

Example 6-27 output shows information regarding the switch that we used, and the associated code and firmware level at the time.

Example 6-27 G8264tor_1 show version output

System Information at 20:30:07 Thu Oct 18, 2012

Time zone: No timezone configured

Daylight Savings Time Status: Disabled

IBM Networking Operating System RackSwitch G8264

Switch has been up for 1 day, 20 hours, 28 minutes and 18 seconds.

Last boot: 6:05:44 Thu Feb 7, 2001 (reset from console)

MAC address: 08:17:f4:33:9d:00 IP (If 20) address: 10.10.20.2

Management Port MAC Address: 08:17:f4:33:9d:fe

Management Port IP Address (if 128): 172.25.101.243

Hardware Revision: 0

Hardware Part No: BAC-00065-00

Switch Serial No: US71120007

Manufacturing date: 11/13

Software Version 7.4.1.0

(FLASH image1), active configuration.

Temperature Mother Top: 26 C

Temperature Mother Bottom: 32 C

Temperature Daughter Top: 26 C

Temperature Daughter Bottom: 30 C

Warning at 75 C and Recover at 90 C

Fan 1 in Module 1: RPM= 8463 PWM= 15(5%) Front-To-Back

Fan 2 in Module 1: RPM= 3976 PWM= 15(5%) Front-To-Back

Fan 3 in Module 2: RPM= 8667 PWM= 15(5%) Front-To-Back

Fan 4 in Module 2: RPM= 4115 PWM= 15(5%) Front-To-Back

Fan 5 in Module 3: RPM= 7894 PWM= 15(5%) Front-To-Back

Fan 6 in Module 3: RPM= 4195 PWM= 15(5%) Front-To-Back

Fan 7 in Module 4: RPM= 8852 PWM= 15(5%) Front-To-Back

Fan 8 in Module 4: RPM= 3976 PWM= 15(5%) Front-To-Back

System Fan Airflow: Front-To-Back

Power Supply 1: OK

Power Supply 2: OK

Power Faults: ()

Fan Faults: ()

Service Faults: ()

Show vlan output

Example 6-28 output shows VLAN assignment for all the various ports on the switch.

Example 6-28 G8264tor_1 show vlan output

VLAN	Name	Status	Ports
1	Default VLAN	ena	17-63
4000	ISL h1thchk	ena	64
4092	DATA	ena	1-16 18 20 22 24-28 37-40
4094	ISL	ena	1-16
4095	Mgmt VLAN	ena	MGT

Show interface status output

Because we have only one compute node in our chassis (in slot 1), this explains why all the other internal ports are listed as down from a link perspective in the following output in Example 6-29.

Example 6-29 G8264tor_1 show interface status output

Alias	Port	Speed	Duplex	Flow Ctrl	Link	Name
				--TX--RX--		
1	1	10000	full	no no	up	ISL
2	2	10000	full	no no	up	ISL
3	3	10000	full	no no	up	ISL
4	4	10000	full	no no	up	ISL
5	5	10000	full	no no	up	ISL
6	6	10000	full	no no	up	ISL
7	7	10000	full	no no	up	ISL
8	8	10000	full	no no	up	ISL
9	9	10000	full	no no	up	ISL
10	10	10000	full	no no	up	ISL
11	11	10000	full	no no	up	ISL
12	12	10000	full	no no	up	ISL
13	13	10000	full	no no	up	ISL
14	14	10000	full	no no	up	ISL
15	15	10000	full	no no	up	ISL
16	16	10000	full	no no	up	ISL
17	17	1G/10G	full	no no	down	17
18	18	10000	full	no no	down	18
19	19	1G/10G	full	no no	down	19
20	20	10000	full	no no	down	20
21	21	1G/10G	full	no no	down	21
22	22	10000	full	no no	down	22
23	23	1G/10G	full	no no	down	23
24	24	10000	full	no no	down	24
25	25	10000	full	no no	up	Link to EN4093-1
26	26	10000	full	no no	up	Link to EN4093-1
27	27	10000	full	no no	up	Link to EN4093-1
28	28	10000	full	no no	up	Link to EN4093-1
29	29	1G/10G	full	no no	down	29
30	30	1G/10G	full	no no	down	30
31	31	1G/10G	full	no no	down	31
32	32	1G/10G	full	no no	down	32
33	33	1G/10G	full	no no	down	33

34	34	1G/10G	full	no	no	down	34
35	35	1G/10G	full	no	no	down	35
36	36	1G/10G	full	no	no	down	36
37	37	10000	full	no	no	up	Link to EN4093-2
38	38	10000	full	no	no	up	Link to EN4093-2
39	39	10000	full	no	no	up	Link to EN4093-2
40	40	10000	full	no	no	up	Link to EN4093-2
41	41	1G/10G	full	no	no	down	41
42	42	1G/10G	full	no	no	down	42
43	43	1G/10G	full	no	no	down	43
44	44	1G/10G	full	no	no	down	44
45	45	1G/10G	full	no	no	down	45
46	46	1G/10G	full	no	no	down	46
47	47	1G/10G	full	no	no	down	47
48	48	1G/10G	full	no	no	down	48
49	49	1G/10G	full	no	no	down	49
50	50	1G/10G	full	no	no	down	50
51	51	1G/10G	full	no	no	down	51
52	52	1G/10G	full	no	no	down	52
53	53	1G/10G	full	no	no	down	53
54	54	1G/10G	full	no	no	down	54
55	55	1G/10G	full	no	no	down	55
56	56	1G/10G	full	no	no	down	56
57	57	1G/10G	full	no	no	down	57
58	58	1G/10G	full	no	no	down	58
59	59	1G/10G	full	no	no	down	59
60	60	1G/10G	full	no	no	down	60
61	61	1G/10G	full	no	no	down	61
62	62	1G/10G	full	no	no	down	62
63	63	1G/10G	full	no	no	down	63
64	64	10000	full	no	no	up	ISL h1thchk
MGT	65	1000	full	yes	yes	up	MGT

show lldp remote-device output

Example 6-30 command output illustrates our physical topology and verifies that cables are plugged into the ports that we have specified in both our Network Topology diagram (Figure 6-1 on page 112), and the configuration specified in the appendix.

Example 6-30 G8264tor_1 show lldp remote-device output

LocalPort	Index	Remote Chassis ID	Remote Port	Remote System Name
MGT	1	fc cf 62 40 a6 00	22	BNT-AS-PM
1	2	08 17 f4 33 75 00	1	G8264TOR-2
2	3	08 17 f4 33 75 00	2	G8264TOR-2
3	4	08 17 f4 33 75 00	3	G8264TOR-2
4	5	08 17 f4 33 75 00	4	G8264TOR-2
5	6	08 17 f4 33 75 00	5	G8264TOR-2
6	7	08 17 f4 33 75 00	6	G8264TOR-2
26	8	6c ae 8b bf 6d 00	57	en4093flex_1
25	10	6c ae 8b bf 6d 00	58	en4093flex_1
7	11	08 17 f4 33 75 00	7	G8264TOR-2
28	12	6c ae 8b bf 6d 00	59	en4093flex_1
27	13	6c ae 8b bf 6d 00	60	en4093flex_1
8	14	08 17 f4 33 75 00	8	G8264TOR-2

37	15	6c ae 8b bf fe 00	57	en4093flex_2
39	16	6c ae 8b bf fe 00	58	en4093flex_2
9	17	08 17 f4 33 75 00	9	G8264TOR-2
38	19	6c ae 8b bf fe 00	59	en4093flex_2
10	20	08 17 f4 33 75 00	10	G8264TOR-2
40	21	6c ae 8b bf fe 00	60	en4093flex_2
11	24	08 17 f4 33 75 00	11	G8264TOR-2
12	25	08 17 f4 33 75 00	12	G8264TOR-2
13	26	08 17 f4 33 75 00	13	G8264TOR-2
14	27	08 17 f4 33 75 00	14	G8264TOR-2
15	28	08 17 f4 33 75 00	15	G8264TOR-2
16	29	08 17 f4 33 75 00	16	G8264TOR-2
64	30	08 17 f4 33 75 00	64	G8264TOR-2

Show vlag isl output

Example 6-31 command output shows the status of the ISL between the G8264 switches, and the ports comprising the ISL itself.

Example 6-31 G8264tor_1 show vlag isl output

ISL_ID	ISL_Vlan	ISL_Trunk	ISL_Members	Link_State	Trunk_State
67	4094	Adminkey 1000	1	UP	UP
			2	UP	UP
			3	UP	UP
			4	UP	UP
			5	UP	UP
			6	UP	UP
			7	UP	UP
			8	UP	UP
			9	UP	UP
			10	UP	UP
			11	UP	UP
			12	UP	UP
			13	UP	UP
			14	UP	UP
			15	UP	UP
			16	UP	UP

Show vlag information output

Example 6-32 on page 132 output shows that the downstream vLAG between the G8264 and EN4093/EN4093R switches is up and operational as referenced by the LACP admin key of 2002. Our ISL between the G8264 switches is up too.

G8264tor_1 is acting as the admin and operational role of SECONDARY. For centralized vLAG functions, such as vLAG STP, one of the vLAG switches must control the protocol operations. To select the switch that controls the centralized vLAG function, role election is performed. The switch with the primary role controls the centralized operation. Role election is non-preemptive. That is, if there exists a *primary*, another switch coming up remains as *secondary*, even if it can become primary based on the role election logic.

Role election is determined by comparing the local vLAG system priority and local system MAC address. The switch with the smaller priority value is the vLAG primary switch. If priority is the same, the switch with the smaller system MAC address is the vLAG primary switch. It is

possible to configure vLAG priority to anything between 0 - 65535. Priority was left at the default value of 0 in all examples.

Example 6-32 G8264tor_1 show vlag information output

```
vLAG Tier ID: 2
vLAG system MAC: 08:17:f4:c3:dd:01
Local MAC 08:17:f4:33:9d:00 Priority 0 Admin Role SECONDARY (Operational Role
SECONDARY)
Peer MAC 08:17:f4:33:75:00 Priority 0
Health local 1.1.1.1 peer 1.1.1.2 State UP
ISL trunk id 67
ISL state Up
Startup Delay Interval: 120s (Finished)
```

vLAG 65: config with admin key 2000, associated trunk 65, state formed

vLAG 66: config with admin key 2002, associated trunk 66, state formed

Show vlag adminkey 2002 output

Example 6-33 output shows that the downstream vLAG towards the EN4093/EN4093R switches is formed and enabled by using LACP reference key 2002.

Example 6-33 G8264tor_1 show vlag adminkey 2002 output

```
vLAG is enabled on admin key 2002
Current LACP params for 25: active, Priority 32768, Admin Key 2002, Min-Links 1

Current LACP params for 26: active, Priority 32768, Admin Key 2002, Min-Links 1

Current LACP params for 27: active, Priority 32768, Admin Key 2002, Min-Links 1

Current LACP params for 28: active, Priority 32768, Admin Key 2002, Min-Links 1

Current LACP params for 37: active, Priority 32768, Admin Key 2002, Min-Links 1

Current LACP params for 38: active, Priority 32768, Admin Key 2002, Min-Links 1

Current LACP params for 39: active, Priority 32768, Admin Key 2002, Min-Links 1

Current LACP params for 40: active, Priority 32768, Admin Key 2002, Min-Links 1
```

Show lacp information state up

Example 6-34 shows which ports are participating in an LACP aggregation, and which reference keys are used on those specific interfaces.

Example 6-34 G8264tor_1 show lacp information state up

port	mode	adminkey	operkey	selected	prio	aggr	trunk	status	minlinks
<hr/>									
1	active	1000	1000	yes	32768	1	67	up	1
2	active	1000	1000	yes	32768	1	67	up	1
3	active	1000	1000	yes	32768	1	67	up	1
4	active	1000	1000	yes	32768	1	67	up	1
5	active	1000	1000	yes	32768	1	67	up	1
6	active	1000	1000	yes	32768	1	67	up	1

7	active	1000	1000	yes	32768	1	67	up	1
8	active	1000	1000	yes	32768	1	67	up	1
9	active	1000	1000	yes	32768	1	67	up	1
10	active	1000	1000	yes	32768	1	67	up	1
11	active	1000	1000	yes	32768	1	67	up	1
12	active	1000	1000	yes	32768	1	67	up	1
13	active	1000	1000	yes	32768	1	67	up	1
14	active	1000	1000	yes	32768	1	67	up	1
15	active	1000	1000	yes	32768	1	67	up	1
16	active	1000	1000	yes	32768	1	67	up	1
18	active	2000	2000	yes	32768	20	65	up	1
20	active	2000	2000	yes	32768	20	65	up	1
22	active	2000	2000	yes	32768	20	65	up	1
24	active	2000	2000	yes	32768	20	65	up	1
25	active	2002	2002	yes	32768	26	66	up	1
26	active	2002	2002	yes	32768	26	66	up	1
27	active	2002	2002	yes	32768	26	66	up	1
28	active	2002	2002	yes	32768	26	66	up	1
37	active	2002	2002	yes	32768	26	66	up	1
38	active	2002	2002	yes	32768	26	66	up	1
39	active	2002	2002	yes	32768	26	66	up	1
40	active	2002	2002	yes	32768	26	66	up	1

Ping output for equipment on VLAN 4092

To verify connectivity, we issued several **ping** commands to devices in the lab infrastructure on VLAN 4092 (Data VLAN) in Example 6-35. IP address 10.4.1.10 represents a compute node with an operating system installed (flex_node1) on the Network Topology diagram (Figure 6-1 on page 112).

Example 6-35 Ping verification for equipment on VLAN 4092

```
G8264TOR-1#ping 10.1.4.10 data-port
Connecting via DATA port.
[host 10.1.4.10, max tries 5, delay 1000 msec, length 0, ping source N/S, ttl 255,
tos 0]
10.1.4.10: #1 ok, RTT 1 msec.
10.1.4.10: #2 ok, RTT 0 msec.
10.1.4.10: #3 ok, RTT 0 msec.
10.1.4.10: #4 ok, RTT 0 msec.
10.1.4.10: #5 ok, RTT 0 msec.
Ping finished.

G8264TOR-1#ping 10.1.4.249 data-port
Connecting via DATA port.
[host 10.1.4.249, max tries 5, delay 1000 msec, length 0, ping source N/S, ttl
255, tos 0]
10.1.4.249: #1 ok, RTT 1 msec.
10.1.4.249: #2 ok, RTT 0 msec.
10.1.4.249: #3 ok, RTT 1 msec.
10.1.4.249: #4 ok, RTT 0 msec.
10.1.4.249: #5 ok, RTT 0 msec.
Ping finished.

G8264TOR-1#ping 10.1.4.238 data-port
Connecting via DATA port.
```

```
[host 10.1.4.238, max tries 5, delay 1000 msec, length 0, ping source N/S, ttl
255, tos 0]
10.1.4.238: #1 ok, RTT 4 msec.
10.1.4.238: #2 ok, RTT 1 msec.
10.1.4.238: #3 ok, RTT 1 msec.
10.1.4.238: #4 ok, RTT 1 msec.
10.1.4.238: #5 ok, RTT 0 msec.
Ping finished.
```

6.7 Full configuration files

In this section, we display the configuration on all of the devices in the Network Topology diagram that is shown in Figure 6-1 on page 112.

EN4093flex-1

Example 6-36 lists the configuration for the EN4093flex-1 switch.

Example 6-36 EN4093flex-1 switch configuration file

```
version "7.3.1"
switch-type "IBM Flex System Fabric EN4093 10Gb Scalable Switch"
!
!

snmp-server name "en4093flex_1"
!
!
hostname "en4093flex_1"
!
!
interface port INTA1
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port INTB1
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT4
    name "ISL hlthchk"
    pvid 4000
    exit
!
interface port EXT7
    name "ISL"
    tagging
    pvid 4094
    exit
```

```

!
interface port EXT8
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port EXT9
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port EXT10
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port EXT15
    name "Link to g8264tor_1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT16
    name "Link to g8264tor_1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT17
    name "Link to g8264tor_1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT18
    name "Link to g8264tor_1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT19
    name "Link to g8264tor_2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT20

```

```

        name "Link to g8264tor_2"
        tagging
        tag-pvid
        pvid 4092
        exit
    !
interface port EXT21
    name "Link to g8264tor_2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT22
    name "Link to g8264tor_2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
vlan 1
    member INTA2-INTA14,INTB2-INTB14,EXT1-EXT3,EXT5-EXT6
    no member INTA1,INTB1,EXT4,EXT7-EXT10,EXT15-EXT22
!
vlan 4000
    enable
    name "ISL hlthchk"
    member EXT4
!
vlan 4092
    enable
    name "DATA"
    member INTA1,INTB1,EXT7-EXT10,EXT15-EXT22
!
vlan 4094
    enable
    name "ISL"
    member EXT7-EXT10
!
!
spanning-tree stp 125 vlan 4000
!
spanning-tree stp 126 vlan 4092
!
no spanning-tree stp 127 enable
spanning-tree stp 127 vlan 4094
!
!
interface port EXT7
    lacp mode active
    lacp key 1000
!
interface port EXT8
    lacp mode active
    lacp key 1000

```

```

!
interface port EXT9
    lacp mode active
    lacp key 1000
!
interface port EXT10
    lacp mode active
    lacp key 1000
!
interface port EXT15
    lacp mode active
    lacp key 2000
!
interface port EXT16
    lacp mode active
    lacp key 2000
!
interface port EXT17
    lacp mode active
    lacp key 2000
!
interface port EXT18
    lacp mode active
    lacp key 2000
!
interface port EXT19
    lacp mode active
    lacp key 2000
!
interface port EXT20
    lacp mode active
    lacp key 2000
!
interface port EXT21
    lacp mode active
    lacp key 2000
!
interface port EXT22
    lacp mode active
    lacp key 2000
!
failover enable
failover trigger 1 mmon monitor admin-key 2000
failover trigger 1 mmon control member INTA1-INTB14
failover trigger 1 enable
!
!
!
vlag enable
vlag tier-id 1
vlag isl vlan 4094
vlag hlthchk peer-ip 1.1.1.2
vlag isl adminkey 1000
vlag adminkey 2000 enable
!

```

```

!
!
!
!
!
!
!
!
lldp enable
!
interface ip 40
    ip address 1.1.1.1 255.255.255.0
    vlan 4000
    enable
    exit
!
interface ip 92
    ip address 10.1.4.238 255.255.255.0
    vlan 4092
    enable
    exit
!
!
!
!
!
ntp enable
ntp ipv6 primary-server fe80::211:25ff:fec3:9b69 MGT
ntp interval 15
ntp authenticate
ntp primary-key 8811
!
ntp message-digest-key 8811 md5-ekey
1e389d20083088209635f6e3cb802bd2b52a41c0125c9904874d06d2a3af9d16341b4054daa0d14523
ca25ad2e9ec7d8ef2248b85c18a59a2436918a0ee41cea
!
ntp trusted-key 8811
!
end

```

EN4093flex_2

Example 6-37 lists the configuration for the EN4093flex_2 switch.

Example 6-37 EN4093flex_2 switch configuration

```

version "7.3.1"
switch-type "IBM Flex System Fabric EN4093 10Gb Scalable Switch"
!
!

snmp-server name "en4093flex_2"
!
!
hostname "en4093flex_2"
!

```



```

!
interface port INTA1
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port INTB1
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT4
    name "ISL hlthchk"
    pvid 4000
    exit
!
interface port EXT7
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port EXT8
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port EXT9
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port EXT10
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port EXT15
    name "Link to g8264tor_1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT16
    name "Link to g8264tor_1"
    tagging
    tag-pvid
    pvid 4092
    exit

```

```

!
interface port EXT17
    name "Link to g8264tor_1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT18
    name "Link to g8264tor_1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT19
    name "Link to g8264tor_2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT20
    name "Link to g8264tor_2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT21
    name "Link to g8264tor_2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port EXT22
    name "Link to g8264tor_2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
vlan 1
    member INTA2-INTA14,INTB2-INTB14,EXT1-EXT3,EXT5-EXT6
    no member INTA1,INTB1,EXT4,EXT7-EXT10,EXT15-EXT22
!
vlan 4000
    enable
    name "ISL hlthchk"
    member EXT4
!
vlan 4092
    enable
    name "DATA"

```

```

    member INTA1,INTB1,EXT7-EXT10,EXT15-EXT22
!
vlan 4094
    enable
    name "ISL"
    member EXT7-EXT10
!
!
spanning-tree stp 125 vlan 4000
!
spanning-tree stp 126 vlan 4092
!
no spanning-tree stp 127 enable
spanning-tree stp 127 vlan 4094
!
!
no logging console
!
interface port EXT7
    lacp mode active
    lacp key 1000
!
interface port EXT8
    lacp mode active
    lacp key 1000
!
interface port EXT9
    lacp mode active
    lacp key 1000
!
interface port EXT10
    lacp mode active
    lacp key 1000
!
interface port EXT15
    lacp mode active
    lacp key 2000
!
interface port EXT16
    lacp mode active
    lacp key 2000
!
interface port EXT17
    lacp mode active
    lacp key 2000
!
interface port EXT18
    lacp mode active
    lacp key 2000
!
interface port EXT19
    lacp mode active
    lacp key 2000
!
interface port EXT20

```

```

    lacp mode active
    lacp key 2000
!
interface port EXT21
    lacp mode active
    lacp key 2000
!
interface port EXT22
    lacp mode active
    lacp key 2000
!
failover enable
failover trigger 1 mmon monitor admin-key 2000
failover trigger 1 mmon control member INTA1-INTB14
failover trigger 1 enable
!
!
!
vlag enable
vlag tier-id 1
vlag isl vlan 4094
vlag hlthchk peer-ip 1.1.1.1
vlag isl adminkey 1000
vlag adminkey 2000 enable
!
!
!
!
!
!
!
!
!
lldp enable
!
interface ip 40
    ip address 1.1.1.2 255.255.255.0
    vlan 4000
    enable
    exit
!
interface ip 92
    ip address 10.1.4.239 255.255.255.0
    vlan 4092
    enable
    exit
!
!
!
!
!
ntp enable
ntp ipv6 primary-server fe80::211:25ff:fec3:9b69 MGT
ntp interval 15
ntp authenticate

```

```

ntp primary-key 8811
!
ntp message-digest-key 8811 md5-ekey
ef9d8bb6cf808aa2b6b6e2f70c3029501c9b293eb41d60e5ebbd0fbbd72171ed3c867d24b9976e2052
771345e26681dc63a675b9033673c9923707f9d0f1c078
!
ntp trusted-key 8811
!
end

```

G8264tor_1

Example 6-38 lists the configuration for the G8264tor_1 switch.

Example 6-38 G8264tor_1 switch configuration

```

version "7.4.1"
switch-type "IBM Networking Operating System RackSwitch G8264"
!
!
ssh enable
!

!
!
no system dhcp
no system default-ip mgt
hostname "G8264TOR-1"
!
!
interface port 1
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 2
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 3
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 4
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 5
    name "ISL"

```

```

        tagging
        pvid 4094
        exit
    !
interface port 6
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 7
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 8
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 9
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 10
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 11
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 12
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 13
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 14
    name "ISL"
    tagging

```

```

        pvid 4094
        exit
    !
interface port 15
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 16
    name "ISL"
    tagging
    pvid 4094
    exit
!
interface port 25
    name "Link to EN4093-1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 26
    name "Link to EN4093-1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 27
    name "Link to EN4093-1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 28
    name "Link to EN4093-1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 37
    name "Link to EN4093-2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 38
    name "Link to EN4093-2"
    tagging
    tag-pvid
    pvid 4092

```

```

    exit
!
interface port 39
    name "Link to EN4093-2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 40
    name "Link to EN4093-2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 64
    name "ISL hlthchk"
    pvid 4000
    exit
!
vlan 1
    member 17-63
    no member 1-16,64
!
vlan 4000
    enable
    name "ISL hlthchk"
    member 64
!
vlan 4092
    enable
    name "DATA"
    member 1-16,18,20,22,24-28,37-40
!
vlan 4094
    enable
    name "ISL"
    member 1-16
!
!
!
spanning-tree stp 125 vlan 4000
!
spanning-tree stp 126 vlan 4092
!
no spanning-tree stp 127 enable
spanning-tree stp 127 vlan 4094
!
!
interface port 1
    lacp mode active
    lacp key 1000
!
interface port 2

```



```

        lacp mode active
        lacp key 1000
    !
interface port 3
    lacp mode active
    lacp key 1000
    !
interface port 4
    lacp mode active
    lacp key 1000
    !
interface port 5
    lacp mode active
    lacp key 1000
    !
interface port 6
    lacp mode active
    lacp key 1000
    !
interface port 7
    lacp mode active
    lacp key 1000
    !
interface port 8
    lacp mode active
    lacp key 1000
    !
interface port 9
    lacp mode active
    lacp key 1000
    !
interface port 10
    lacp mode active
    lacp key 1000
    !
interface port 11
    lacp mode active
    lacp key 1000
    !
interface port 12
    lacp mode active
    lacp key 1000
    !
interface port 13
    lacp mode active
    lacp key 1000
    !
interface port 14
    lacp mode active
    lacp key 1000
    !
interface port 15
    lacp mode active
    lacp key 1000
    !

```

```

interface port 16
    lacp mode active
    lacp key 1000
!
interface port 18
    lacp mode active
    lacp key 2000
!
interface port 20
    lacp mode active
    lacp key 2000
!
interface port 22
    lacp mode active
    lacp key 2000
!
interface port 24
    lacp mode active
    lacp key 2000
!
interface port 25
    lacp mode active
    lacp key 2002
!
interface port 26
    lacp mode active
    lacp key 2002
!
interface port 27
    lacp mode active
    lacp key 2002
!
interface port 28
    lacp mode active
    lacp key 2002
!
interface port 37
    lacp mode active
    lacp key 2002
!
interface port 38
    lacp mode active
    lacp key 2002
!
interface port 39
    lacp mode active
    lacp key 2002
!
interface port 40
    lacp mode active
    lacp key 2002
!
!
!
vlag enable

```

```

vlag tier-id 2
vlag isl vlan 4094
vlag hlthchk peer-ip 1.1.1.2
vlag isl adminkey 1000
vlag adminkey 2000 enable
vlag adminkey 2002 enable
!
!
!
!
!
!
!
!
!
!
!
interface ip 40
    ip address 1.1.1.1 255.255.255.0
    vlan 4000
    enable
    exit
!
interface ip 92
    ip address 10.1.4.243 255.255.255.0
    vlan 4092
    enable
    exit
!
interface ip 128
    ip address 172.25.101.243 255.255.0.0
    enable
    exit
!
ip gateway 4 address 172.25.1.1
ip gateway 4 enable
!
!
!
!
!
!
end

```

G8264tor_2

Example 6-39 lists the configuration for the G8264tor_2 switch.

Example 6-39 G8264tor_2 switch configuration

```
version "7.4.1"
switch-type "IBM Networking Operating System RackSwitch G8264"
!
!
ssh enable
!

!
!
no system dhcp
no system default-ip mgt
hostname "G8264TOR-2"
!
!
interface port 1
    name "ISL"
    tagging
    exit
!
interface port 2
    name "ISL"
    tagging
    exit
!
interface port 3
    name "ISL"
    tagging
    exit
!
interface port 4
    name "ISL"
    tagging
    exit
!
interface port 5
    name "ISL"
    tagging
    exit
!
interface port 6
    name "ISL"
    tagging
    exit
!
interface port 7
    name "ISL"
    tagging
    exit
!
interface port 8
    name "ISL"
```

```

        tagging
        exit
    !
interface port 9
    name "ISL"
    tagging
    exit
    !
interface port 10
    name "ISL"
    tagging
    exit
    !
interface port 11
    name "ISL"
    tagging
    exit
    !
interface port 12
    name "ISL"
    tagging
    exit
    !
interface port 13
    name "ISL"
    tagging
    exit
    !
interface port 14
    name "ISL"
    tagging
    exit
    !
interface port 15
    name "ISL"
    tagging
    exit
    !
interface port 16
    name "ISL"
    tagging
    exit
    !
interface port 25
    name "Link to EN4093-1"
    tagging
    tag-pvid
    pvid 4092
    exit
    !
interface port 26
    name "Link to EN4093-1"
    tagging
    tag-pvid
    pvid 4092

```

```

    exit
!
interface port 27
    name "Link to EN4093-1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 28
    name "Link to EN4093-1"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 37
    name "Link to EN4093-2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 38
    name "Link to EN4093-2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 39
    name "Link to EN4093-2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 40
    name "Link to EN4093-2"
    tagging
    tag-pvid
    pvid 4092
    exit
!
interface port 64
    name "ISL hlthchk"
    pvid 4000
    exit
!
vlan 1
    member 1-63
    no member 64
!
vlan 4000
    enable

```

```

        name "ISL h1thchk"
        member 64
    !
vlan 4092
    enable
    name "DATA"
    member 1-16,18,20,22,24-28,37-40
!
vlan 4094
    enable
    name "ISL"
    member 1-16
!
!
!
spanning-tree stp 125 vlan 4000
!
spanning-tree stp 126 vlan 4092
!
no spanning-tree stp 127 enable
spanning-tree stp 127 vlan 4094
!
!
interface port 1
    lacp mode active
    lacp key 1000
!
interface port 2
    lacp mode active
    lacp key 1000
!
interface port 3
    lacp mode active
    lacp key 1000
!
interface port 4
    lacp mode active
    lacp key 1000
!
interface port 5
    lacp mode active
    lacp key 1000
!
interface port 6
    lacp mode active
    lacp key 1000
!
interface port 7
    lacp mode active
    lacp key 1000
!
interface port 8
    lacp mode active
    lacp key 1000
!

```

```

interface port 9
    lacp mode active
    lacp key 1000
!
interface port 10
    lacp mode active
    lacp key 1000
!
interface port 11
    lacp mode active
    lacp key 1000
!
interface port 12
    lacp mode active
    lacp key 1000
!
interface port 13
    lacp mode active
    lacp key 1000
!
interface port 14
    lacp mode active
    lacp key 1000
!
interface port 15
    lacp mode active
    lacp key 1000
!
interface port 16
    lacp mode active
    lacp key 1000
!
interface port 18
    lacp mode active
    lacp key 2000
!
interface port 20
    lacp mode active
    lacp key 2000
!
interface port 22
    lacp mode active
    lacp key 2000
!
interface port 24
    lacp mode active
    lacp key 2000
!
interface port 25
    lacp mode active
    lacp key 2002
!
interface port 26
    lacp mode active
    lacp key 2002

```



```

!
interface port 27
    lacp mode active
    lacp key 2002
!
interface port 28
    lacp mode active
    lacp key 2002
!
interface port 37
    lacp mode active
    lacp key 2002
!
interface port 38
    lacp mode active
    lacp key 2002
!
interface port 39
    lacp mode active
    lacp key 2002
!
interface port 40
    lacp mode active
    lacp key 2002
!
!
!
vlag enable
vlag tier-id 2
vlag isl vlan 4094
vlag hlthchk peer-ip 1.1.1.1
vlag isl adminkey 1000
vlag adminkey 2000 enable
vlag adminkey 2002 enable
!
!
!
!
!
!
!
!
!
!
interface ip 40
    ip address 1.1.1.2 255.255.255.0
    vlan 4000
    enable
    exit
!
interface ip 92
    ip address 10.1.4.244 255.255.255.0
    vlan 4092
    enable
    exit

```

```
!  
interface ip 128  
  ip address 172.25.101.244 255.255.0.0  
  enable  
  exit  
!  
ip gateway 4 address 172.25.1.1  
ip gateway 4 enable  
!  
!  
!  
!  
!  
!  
end
```

Our setup is now complete.



System management

Management components and tools such as Chassis Management Module (CMM) and IBM Flex System Manager are designed to help you get the most out of your IBM Flex System installation while automating repetitive tasks. These management interfaces can significantly reduce the number of manual navigational steps for typical management tasks. They offer simplified system setup procedures using wizards and built-in expertise for consolidated monitoring of physical and virtual resources.

We describe the following aspects of IBM Flex System management:

- ▶ Management network
- ▶ Chassis Management Module (CMM)
- ▶ Security
- ▶ Compute node management
- ▶ I/O modules management
- ▶ IBM Flex System Manager (FSM)
- ▶ IBM System Networking Element Manager component (SNEM-C)
- ▶ IBM System Networking Element Manager (SNEM) solution

7.1 Management network

In an IBM Flex System Enterprise Chassis, you can configure separate management and data networks.

The management network is a private and secure Gigabit Ethernet (GbE) network used to complete management-related functions throughout the chassis, including management tasks related to the compute nodes, switches, and the chassis itself.

The management network is shown in Figure 7-1 as the blue line. It connects the CMM to the compute nodes, the switches in I/O bays, and the IBM Flex System Manager (FSM). The FSM connection to the management network is via a special Broadcom 5718-based management network adapter (Eth0). The management networks in multiple chassis can be connected together via the external ports of the CMMs in each chassis via a GbE top-of-rack switch.

The yellow line in Figure 7-1 shows the production data network. The FSM also connects to the production network (Eth1) so that it can access the Internet for product updates and other related information.

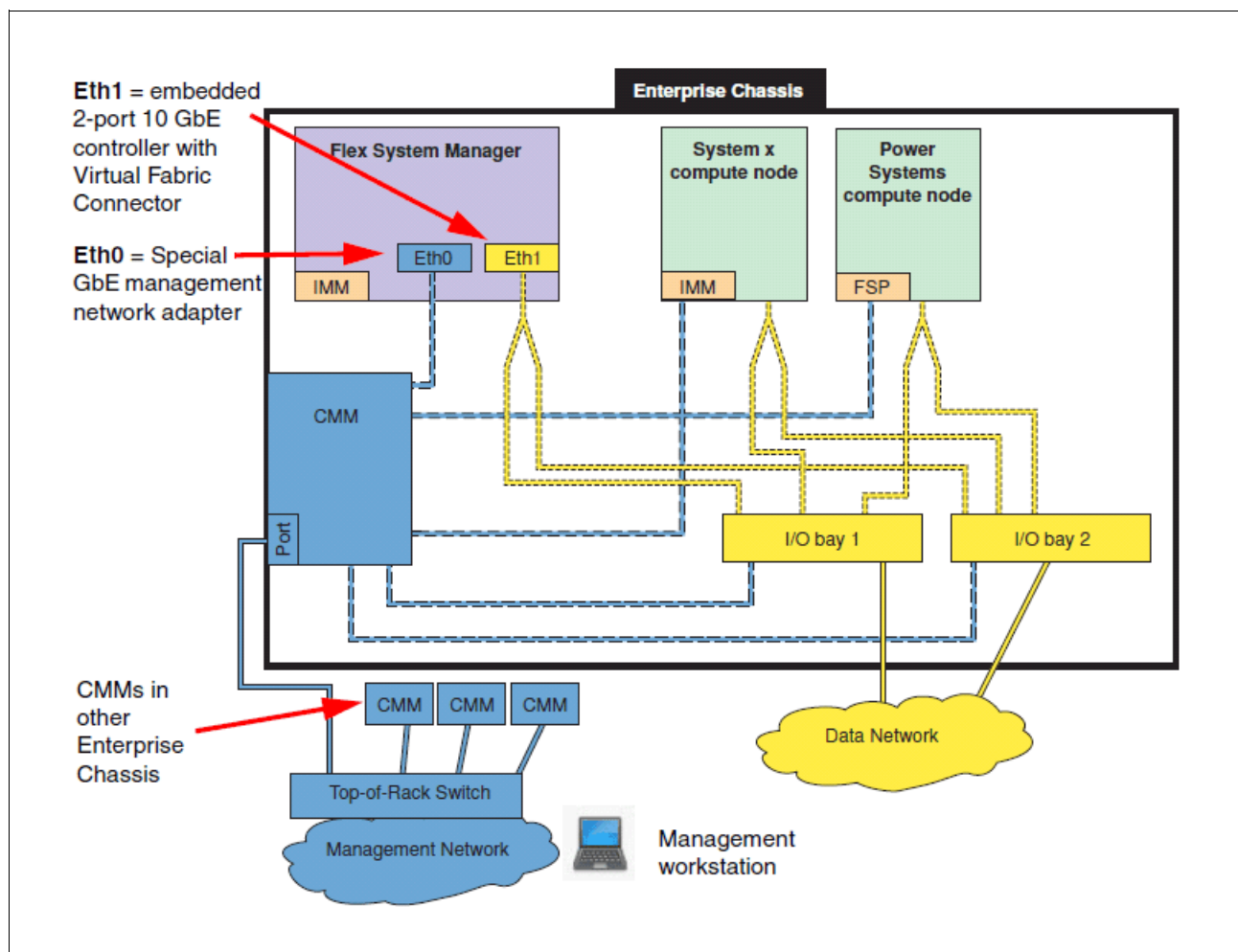


Figure 7-1 Separate management and production data networks

Note: If wanted, the management node console can be connected to the data network for convenience of access.

One of the key functions that the data network supports is discovery of operating systems on compute nodes. Discovery of operating systems by the FSM is required to support software updates on compute nodes. The FSM Checking and Updating Compute Nodes wizard assists you in discovering operating systems as part of the initial setup.

7.2 Chassis Management Module

The Chassis Management Module (CMM) provides single-chassis management. The CMM is used to communicate with the management controller in each compute node to provide system monitoring, event recording, and alerts, and to manage the chassis, its devices, and the compute nodes. The chassis supports up to two CMMs. If one CMM fails, the second CMM can detect its inactivity and activate itself and take control of the system without any disruption. The CMM is central of the management of the chassis and is required in the Enterprise Chassis.

The following section describes the usage models of the CMM and its features.

7.2.1 Overview

The *CMM* is a hot-swap module that provides basic system management functions for all devices installed in the Enterprise Chassis. An Enterprise Chassis comes with at least one CMM and supports CMM redundancy.

The CMM is shown in Figure 7-2.

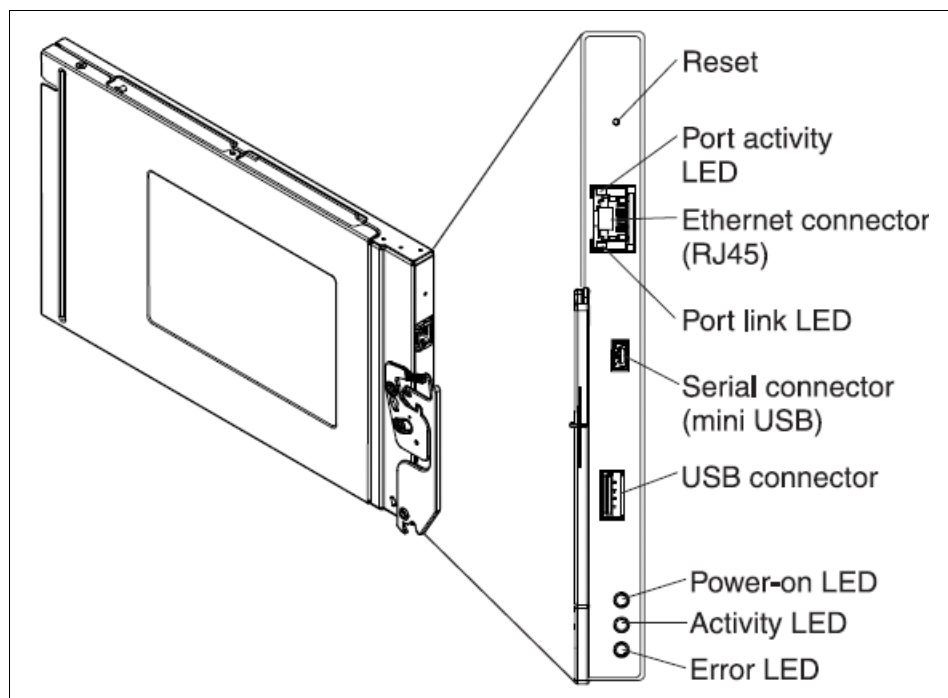


Figure 7-2 Chassis Management Module (CMM)

Through an embedded firmware stack, the CMM implements functions to monitor, control, and provide external user interfaces to manage all chassis resources. Following are some of the available functions:

- ▶ Define login IDs and passwords
- ▶ Configure security settings such as data encryption and user account security
- ▶ Select recipients for alert notification of specific events
- ▶ Monitor the status of the compute nodes and other components
- ▶ Find chassis component information
- ▶ Discover other chassis in the network and enable access to them
- ▶ Control the chassis, compute nodes, and other components
- ▶ Access the I/O modules to configure them
- ▶ Change the startup sequence in a compute node
- ▶ Set the date and time
- ▶ Use a remote console for the compute nodes
- ▶ Enable multi-chassis monitoring
- ▶ Set power policies and view power consumption history for chassis components

7.2.2 Chassis Management Module user interfaces

The Chassis Management Module (CMM) supports a web-based graphical user interface that provides a way to perform chassis management functions within a supported web browser. You can also perform management functions through the CMM command-line interface (CLI). Both the web-based and CLI interfaces are accessible via the single RJ45 Ethernet connector on the CMM or from any other system that is connected to the same management network.

The CMM has the following default IPv4 settings:

- ▶ IP address: 192.168.70.100
- ▶ Subnet: 255.255.255.0
- ▶ User ID: USERID (all capital letters)
- ▶ Password: PASSW0RD (all capital letters, with a zero instead of the letter O)

The CMM does not have a fixed static IPv6 IP address by default. Initial access to the CMM in an IPv6 environment can be done by either using the IPv4 IP address or the IPv6 link-local address. The IPv6 link-local address is automatically generated based on the Media Access Control (MAC) address of the CMM.

By default, the CMM is configured to respond to Dynamic Host Configuration Protocol (DHCP) first before using its static IPv4 address. In environments where this is not the wanted operation, it is suggested that the user connect locally to the CMM (through a notebook, for example), and change the IP settings accordingly.

The web-based graphical user interface (GUI) brings together all the functionality that is needed to manage the chassis elements in an easy-to-use fashion with consistency across all System x integrated management module v2 (IMMv2) based platforms. The CMM login panel is shown in Figure 7-3.



Figure 7-3 CMM login pane

Figure 7-4 shows an example of the CMM front page after login.

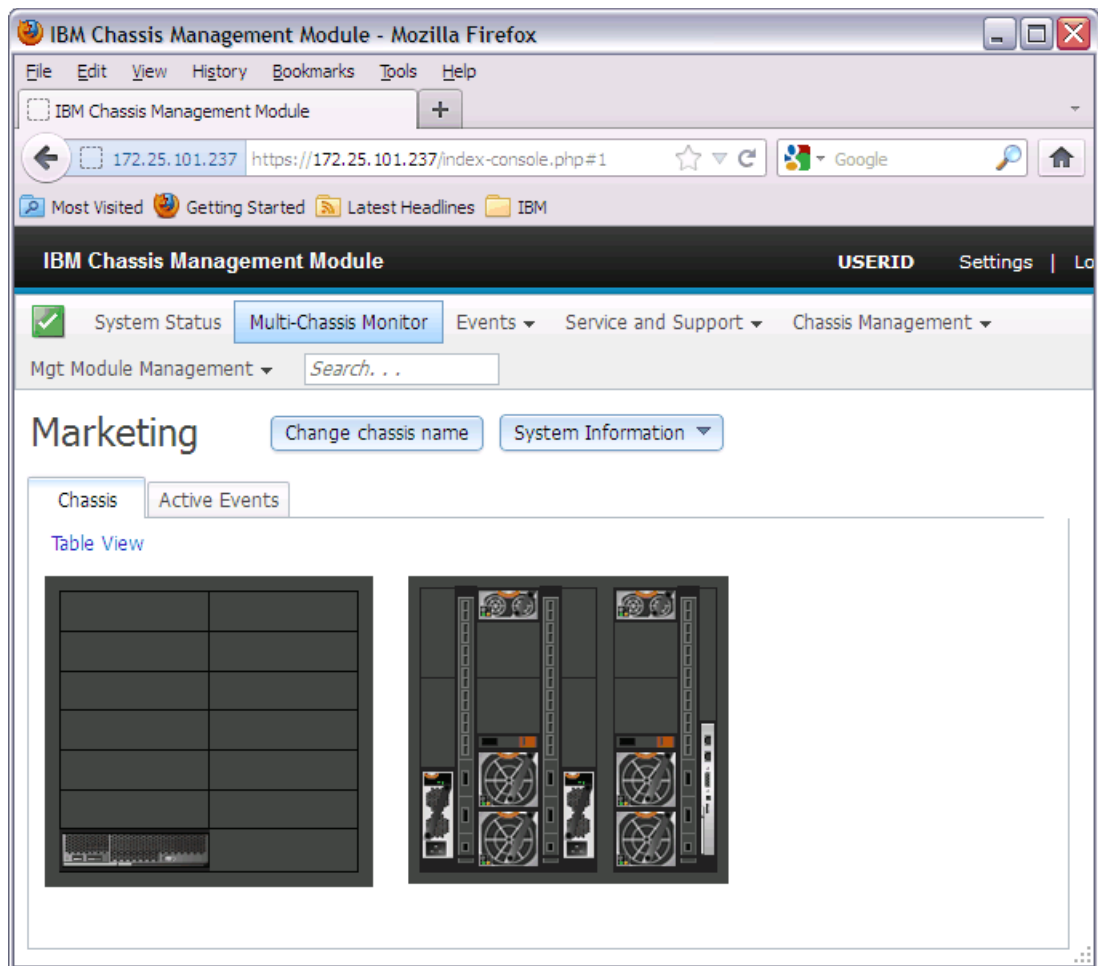


Figure 7-4 Initial view of CMM after login

You are now ready to manage, configure, or troubleshoot any component in the IBM Flex System chassis.

7.3 Security

The focus of IBM on smarter computing is evident in the improved security measures implemented in IBM Flex System Enterprise Chassis. Today's world of computing demands tighter security standards and native integration with computing platforms. For example, the push towards virtualization has increased the need for high degrees of security as more mission critical workloads are consolidated on fewer and more powerful servers. The IBM Flex System Enterprise Chassis takes a new approach to security with a ground-up chassis management design to meet new security standards.

Listed below are the security enhancements and features provided in the chassis:

- ▶ Single sign-on (central user management)
- ▶ End-to-end audit logs
- ▶ Secure boot: Trusted Platform Module (TPM) and Core Root of Trust for Measurement (CRTM)

- ▶ Intel TXT technology (Intel Xeon based compute nodes)
- ▶ Signed firmware updates to ensure authenticity
- ▶ Secure communications
- ▶ Certificate authority and management
- ▶ Chassis and compute node detection and provisioning
- ▶ Role-based access control
- ▶ Security policy management
- ▶ Same management protocols supported on BladeCenter advanced management module (AMM) for compatibility with an earlier version
- ▶ Insecure protocols come disabled by default in CMM, with locking mechanism to prevent user from inadvertently or maliciously enabling
- ▶ Supports up to 84 local CMM user accounts
- ▶ Supports up to 32 simultaneous sessions
- ▶ Planned support for Dynamic Root for Trusted Measurement (DRTM)

The Enterprise Chassis ships secure with built-in components, and supports two security policy settings:

- ▶ Secure: Default setting to ensure a secure chassis infrastructure
 - Strong password policies with automatic validation and verification checks
 - Updated passwords that replace the manufacturing default passwords after the initial setup
 - Only secure communication protocols such as Secure Shell (SSH), Secure Sockets Layer (SSL), and SSH File Transfer Protocol (SFTP)
 - Certificates to establish secure, trusted connections for applications that run on the management processors
- ▶ Legacy: Flexibility in chassis security
 - Weak password policies with minimal controls
 - Manufacturing default passwords that do not have to be changed
 - Unencrypted communication protocols such as Telnet, SNMPv1, TCP Command Mode, CIM-XML, FTP Server, and TFTP Server

The centralized security policy makes Enterprise Chassis easy to configure. In essence, all components run with the same security policy provided by the CMM. This ensures that all I/O modules run with a hardened attack surface.

7.4 Compute node management

Each node in the Enterprise Chassis has a management controller that communicates upstream via the CMM enabled 1 GbE private management network enabling management capability. Different chassis components supported in the Enterprise Chassis might implement different management controllers.

Table 7-1 details the different management controllers implemented in the chassis components.

Table 7-1 Chassis components and their respective management controllers

Chassis components	Management controller
Intel Xeon processor-based compute nodes	Integrated management module II (IMMv2)
Power Systems compute nodes	Flexible service processor (FSP)
Chassis Management Module	Integrated management module v2 (IMMv2)

The management controllers for the various Enterprise Chassis components have the following default IPv4 addresses:

- ▶ CMM: 192.168.70.100
- ▶ Compute nodes: 192.168.70.101-114 (corresponding to the slots in the chassis 1 - 14)
- ▶ I/O modules: 192.168.70.120-123 (sequentially corresponding to chassis bay numbering)

In addition to the IPv4 address, all I/O modules also support link-local IPv6 addresses and configurable external IPv6 addresses.

7.4.1 Integrated management module II

The integrated management module II (IMMv2) is the next generation of IMM (first released in the Intel Xeon Nehalem-EP-based servers). It is present on all Intel Xeon Romley based platforms with complete rework of hardware and firmware. The IMMv2 enhancements include a more responsive user interface, faster power-on, and increased remote presence performance.

The IMMv2 incorporates a new web user interface that provides a common look and feel across all IBM System x products. In addition to the new interface, the following list shows other major enhancements over IMM Version 1:

- ▶ Faster CPU and memory
- ▶ IMMv2 manageable *northbound* from outside the chassis; enables consistent management and scripting with System x rack servers
- ▶ Remote presence
 - Increased color depth and resolution for more detailed server video
 - Active X client in addition to Java client
 - Increased memory capacity (~50 MB) provides convenience for remote software installations
- ▶ No IMMv2 reset required on configuration changes; changes become effective immediately without IMMv2 reboot
- ▶ Hardware management of non-volatile storage
- ▶ Faster Ethernet over USB
- ▶ 1 Gb Ethernet management capability
- ▶ Improved system power-on and boot time
- ▶ More detailed information for Unified Extensible Firmware Interface (UEFI) detected events enables easier problem determination and fault isolation
- ▶ User interface meets accessibility standards (CI-162 compliant)

- ▶ Separate audit and event logs
- ▶ *Trusted* IMM with significant security enhancements (CRTM and TPM, signed updates, authentication policies, and so on)
- ▶ Simplified update and flashing mechanism
- ▶ Addition of syslog alerting mechanism provides users with an alternative to email and Simple Network Management Protocol (SNMP) traps
- ▶ Support for IBM Features On Demand (FoD) enablement of server functions, option card features, and System x solutions and applications
- ▶ First Failure Data Capture: One button web press initiates data collection and download

For more information about IMMv2, see the following documentation:

- ▶ *Integrated Management Module II User's Guide*
<http://ibm.com/support/entry/portal/docdisplay?ln docid=MIGR-5086346>
- ▶ *IMM and IMM2 Support on IBM System x and BladeCenter Servers*, TIPS0849
<http://www.redbooks.ibm.com/abstracts/tips0849.html>

7.4.2 Flexible service processor

There are several advanced system management capabilities built into IBM POWER7® based compute nodes. A flexible service processor (FSP) handles most of the server-level system management. The FSP used in Enterprise Chassis-compatible IBM POWER® based nodes is the same service processor used on POWER rack servers. It has features, such as system alerts and Serial-over-LAN (SOL) capability.

The FSP provides out-of-band system management capabilities, such as system control, runtime error detection, configuration, and diagnostics. Generally, you do not interact with the FSP directly but, rather, using tools, such as IBM Flex System Manager and CMM.

Both the p260 and p460 have one FSP each.

The flexible service processor provides an SOL interface, which is available using the CMM and the console command. The POWER7 based compute nodes do not have an onboard video chip and do not support keyboard, video, and mouse (KVM) connections. Server console access is obtained by an SOL connection only.

SOL provides a means to manage servers remotely by using a CLI over a Telnet or Secure Shell (SSH) connection. SOL is required to manage servers that do not have KVM support or that are attached to the FSM. SOL provides console redirection for both system management services (SMS) and the server operating system.

The SOL feature redirects server serial-connection data over a LAN without requiring special cabling by routing the data using the CMM network interface. The SOL connection enables POWER7 based compute nodes to be managed from any remote location with network access to the CMM.

SOL offers the following functionality:

- ▶ Remote administration without KVM (headless servers)
- ▶ Reduced cabling and no requirement for a serial concentrator
- ▶ Standard Telnet/SSH interface, eliminating the requirement for special client software

The CMM CLI provides access to the text-console command prompt on each server through an SQL connection, enabling the POWER7 based compute nodes to be managed from a remote location.

7.5 I/O modules management

The I/O modules supported in the Enterprise Chassis do not standardize on a specific management controller (such as IMM2 or FSP on compute nodes). Instead, there are functional requirements they must provide over the 1 Gb and I2C interfaces to the CMM. The base functionality that is required is listed here:

- ▶ Initialization
- ▶ Configuration
- ▶ Diagnostics (both power-on and concurrent)
- ▶ Status reporting

In addition, the following set of protocols and software features are supported on the I/O modules:

- ▶ Supports configuration method over the Ethernet management port.
- ▶ A scriptable SSH CLI, a web server with SSL support, SNMPv3 agent with alerts, and an SFTP client.
- ▶ Server ports used for Telnet, HTTP, SNMPv1 agents, TFTP, FTP, and any other insecure protocols are disabled by default.
- ▶ Lightweight Directory Access Protocol (LDAP) authentication protocol support for user authentication.
- ▶ For Ethernet I/O modules, 802.1x enabled with policy enforcement point (PEP) capability to allow support of Trusted Network Connect (TNC).
- ▶ The ability to capture and apply a switch configuration file and the ability to capture a first-failure data capture (FFDC) data file.
- ▶ Ability to transfer files via URL update methods (HTTP, HTTPS, FTP, TFTP, and SFTP).
- ▶ Various methods for firmware updates are supported, including FTP, SFTP, and TFTP. In addition, firmware updates via a URL that include protocol support for HTTP, HTTPS, FTP, SFTP, and TFTP are supported.
- ▶ Support Service Location Protocol (SLP) discovery in addition to SNMPv3.
- ▶ Ability to detect firmware and hardware hangs and the ability to pull a crash-failure dump file to an FTP (SFTP) server.
- ▶ Supports selectable primary and backup firmware banks as the current operational firmware.
- ▶ Ability to send events, SNMP traps, and event logs to the CMM, including security audit logs.
- ▶ IPv4 and IPv6 on by default.
- ▶ The CMM management port supports IPv4 and IPv6 (IPv6 support includes the use of link local addresses).
- ▶ The following list provides port mirroring capabilities:
 - Port mirroring of CMM ports to both internal and external ports.
 - For security reasons, the ability to mirror the CMM traffic is hidden and is available only to development and service personnel.

- Management VLANs for Ethernet switches: A configurable management 802.1q tagged VLAN (in the standard VLAN range of 1 - 4094) that includes the CMM's internal management ports and the I/O modules' internal ports that are connected to the nodes.

7.5.1 I/O module management in Chassis Management Module web interface

To manage an I/O module (such as EN4093/EN4093R 10Gb Ethernet switch) in Chassis Management Module (CMM) user interface, begin at the CMM initial view (shown in Figure 7-5). We show how to manage I/O module 1, which is highlighted.

Switches: Throughout this book, we use EN4093/EN4093R to denote that either switch can be used.

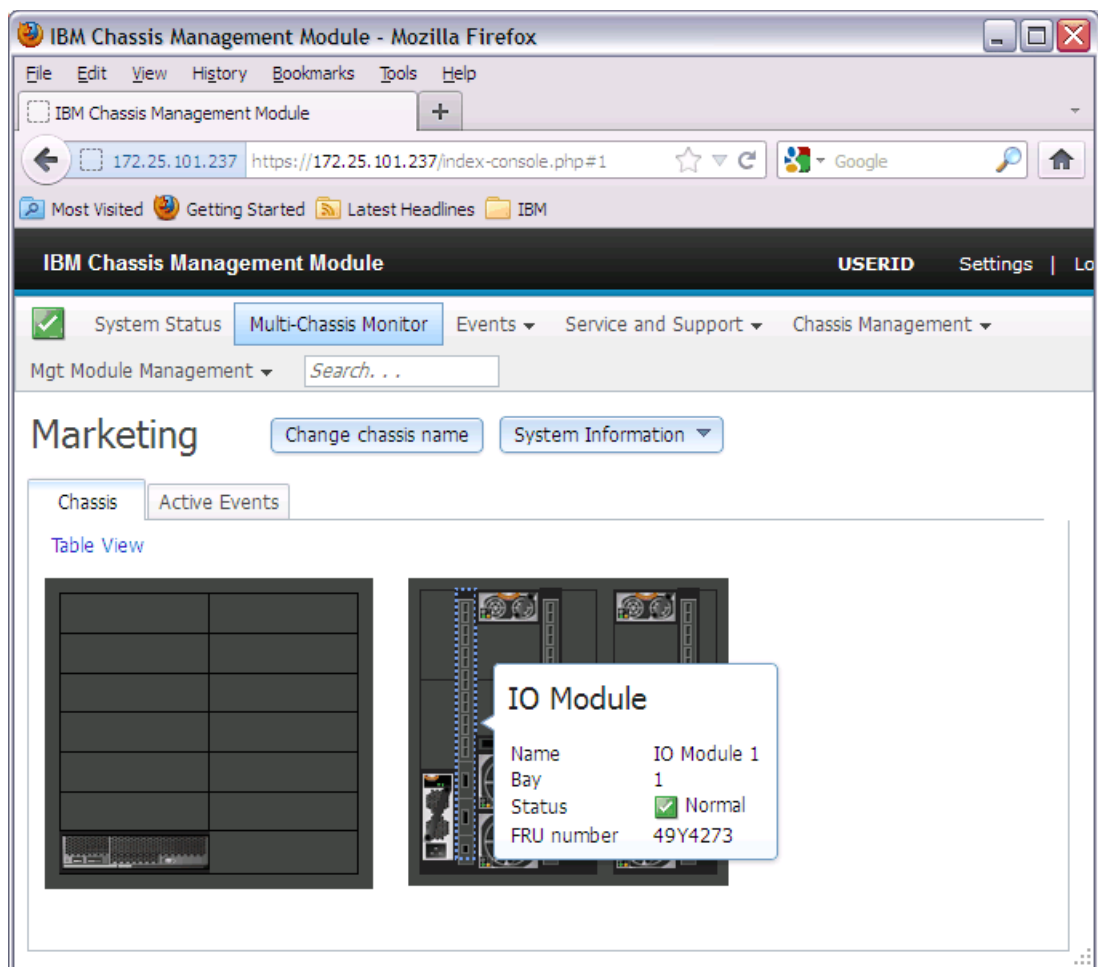


Figure 7-5 CMM initial view with I/O module 1 highlighted

Next, select **Chassis Management** → **I/O Modules** (see Figure 7-6).

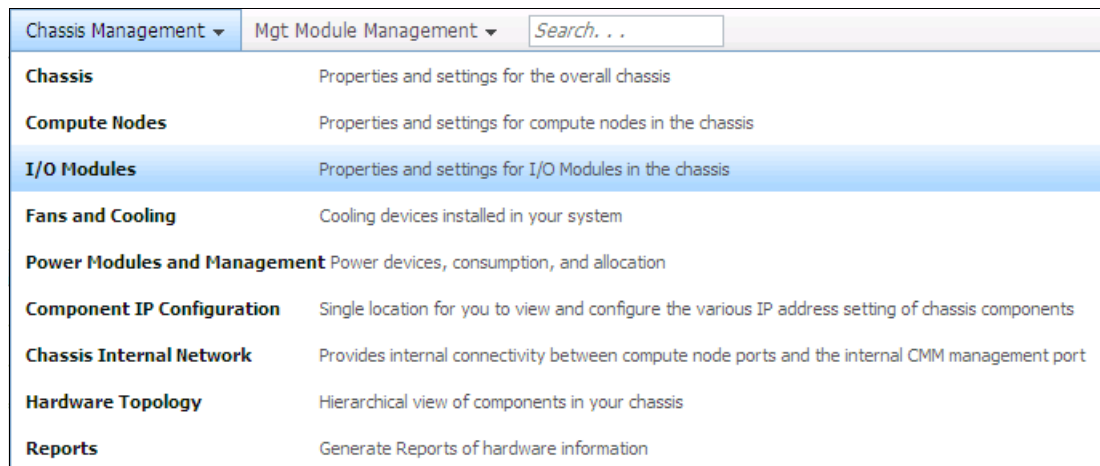


Figure 7-6 Chassis Management: I/O Modules

This displays a list of I/O modules that are installed in the chassis, as shown in Figure 7-7.

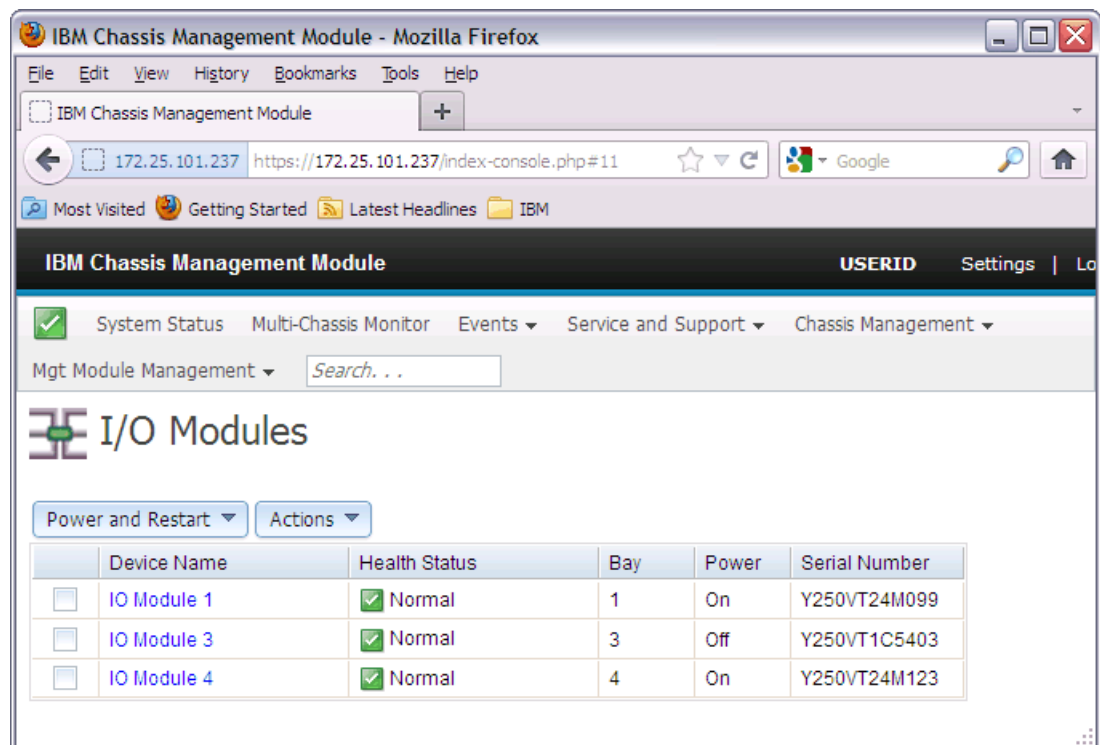


Figure 7-7 I/O modules management

You can see the I/O module properties by clicking it. Figure 7-8 shows an example of properties for I/O Module 1. As shown, EN4093 Ethernet switch is installed in this I/O bay.

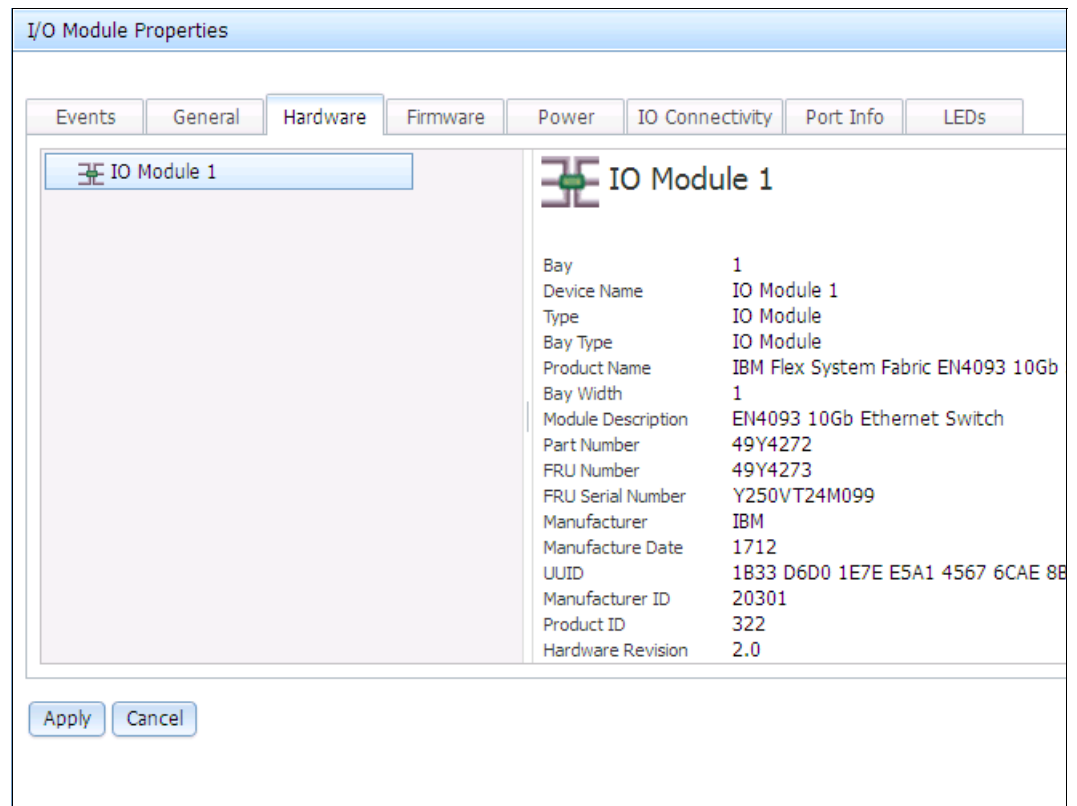


Figure 7-8 I/O Module Properties window

You can use the two buttons to manage the I/O module:

- **Power and Restart** option allows you to perform the actions shown in Figure 7-9.

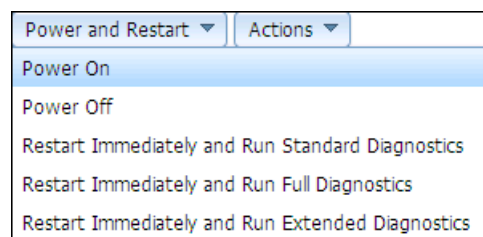


Figure 7-9 Power and Restart button

- **Actions** option allows you to do the tasks shown in Figure 7-10.

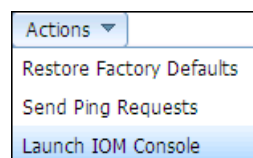


Figure 7-10 Actions button

For example, if you select the **Launch IOM Console** action, this starts the Browser-Based Interface (BBI) for the EN4093/EN4093R Ethernet switch (as shown in Figure 7-11).

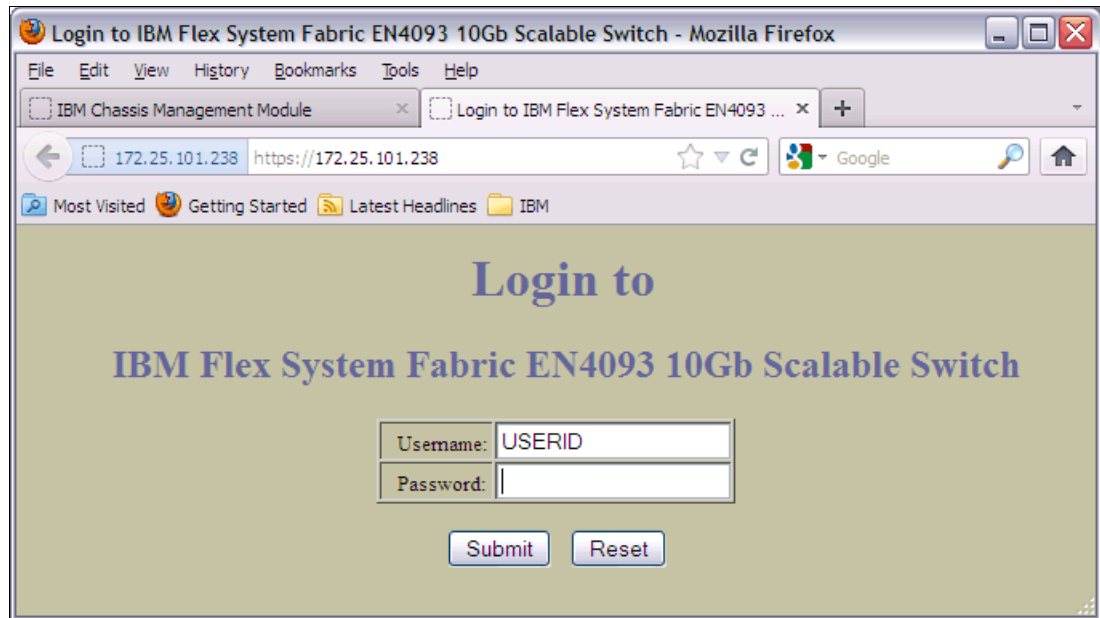


Figure 7-11 BBI: Login to EN4093/EN4093R Ethernet switch

After a successful login, the EN4093/EN4093R switch dashboard displays (see Figure 7-12 on page 171), and you can manage and configure the switch.

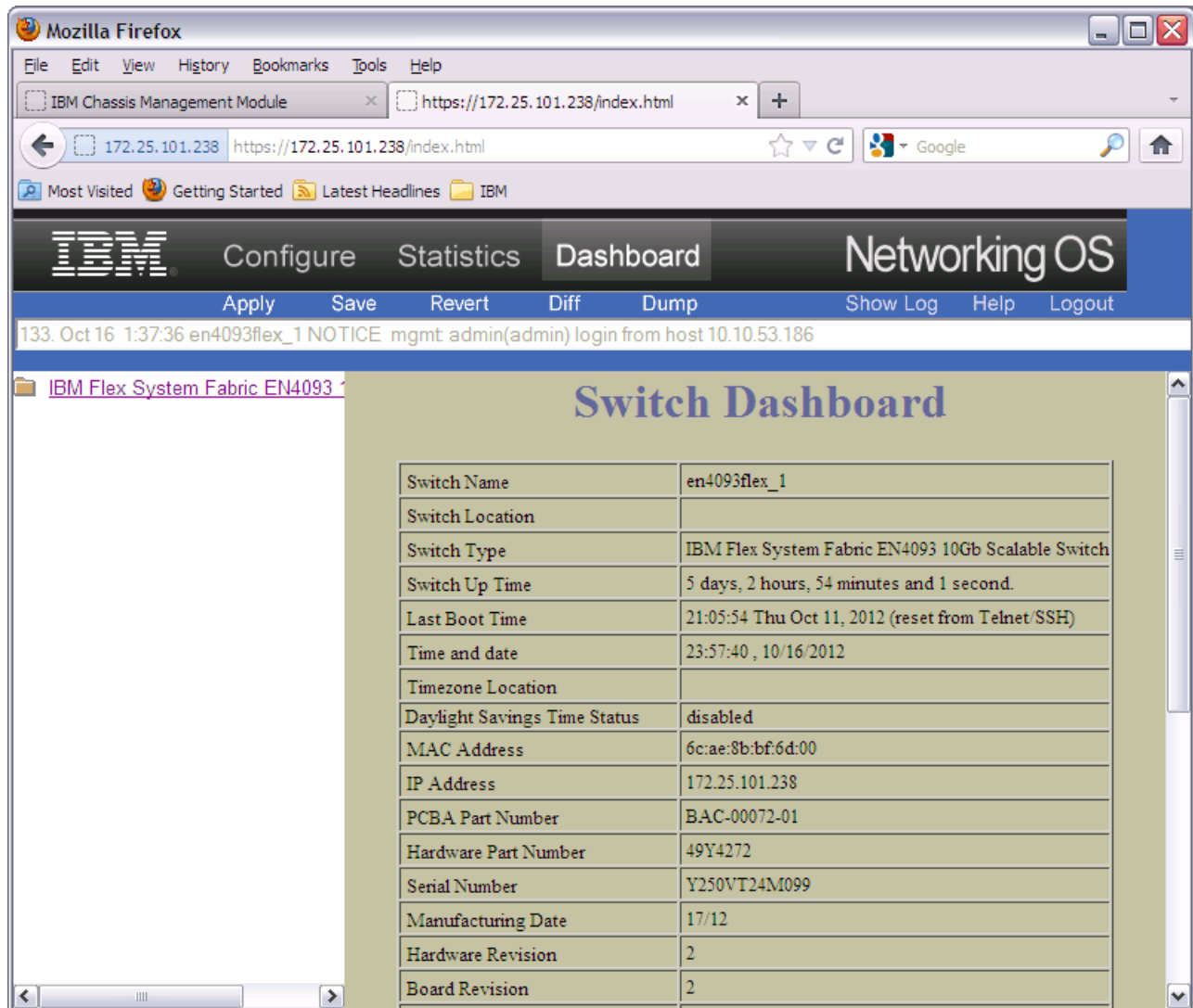


Figure 7-12 BBI: EN4093/EN4093R switch dashboard

You can now use BBI to manage, configure, monitor, and troubleshoot the EN4093/EN4093R Ethernet switch module.

7.6 IBM Flex System Manager

The IBM Flex System Manager (FSM) is a high performance scalable system management appliance based on the IBM Flex System x240 Compute Node. The FSM hardware comes preinstalled with systems management software, which enables you to configure, monitor, and manage IBM Flex System resources in up to four chassis.

Note: Support for management of greater than four chassis with a single FSM might be added later.

The following list describes the high-level features and functions of the IBM Flex System Manager:

- ▶ Supports a comprehensive, pre-integrated system that is configured to optimize performance and efficiency
- ▶ Automated processes that are triggered by events simplify management and reduce manual administrative tasks
- ▶ Centralized management reduces the skills and the number of steps it takes to manage and deploy a system
- ▶ Enables comprehensive management and control of energy usage and costs
- ▶ Automates responses for a reduced need for manual tasks: Custom actions/filters, configure, edit, relocate, automation plans
- ▶ Full integration with server views, including virtual server views, enables efficient management of resources

The preinstallation contains a set of software components that are responsible for performing certain management functions. These components must be activated by using the available FoD software entitlement licenses, and they are licensed on a per-chassis basis. That is, you need one license for each chassis that you plan to manage. The management node comes standard without any entitlement licenses. Therefore, you must purchase a license to enable the required FSM functionality.

The part number to order the management node is shown in Table 7-2.

Table 7-2 Ordering information for IBM Flex System Manager node

Part number	Description
8731A1x ^a	IBM Flex System Manager node

a. x in the part number represents a country-specific letter (for example, the EMEA part number is 8731A1G, and the US part number is 8731A1U). Ask your local IBM representative for specifics.

The part numbers to order FoD software entitlement licenses are shown in Table 7-3 (for United States, Canada, Asia Pacific, and Japan) and Table 7-4 on page 173 (for Latin America and Europe/Middle East/Africa). The part numbers for the same features are different across geographies. Ask your local IBM representative for specifics.

Table 7-3 Ordering information for FoD licenses (United States, Canada, Asia Pacific, and Japan)

Part number	Description
Base feature set	
90Y4217	IBM Flex System Manager per managed chassis with 1 Year IBM Software Subscription and Support (SW S&S)
90Y4222	IBM Flex System Manager per managed chassis with 3 Year SW S&S
Advanced feature set	
90Y4249	IBM Flex System Manager, Advanced Upgrade, per managed chassis with 1 Year SW S&S
00D7554	IBM Flex System Manager, Advanced Upgrade, per managed chassis with 3 Year SW S&S
Fabric Manager	

Part number	Description
00D7550	IBM Fabric Manager, per managed chassis with 1 Year SW S&S
00D7551	IBM Fabric Manager, per managed chassis with 3 Year SW S&S

Table 7-4 Ordering information for FoD licenses (Latin America and Europe/Middle East/Africa)

Part number	Description
Base feature set	
95Y1174	IBM Flex System Manager Per Managed Chassis with 1 Year IBM Software Subscription and Support (SW S&S)
95Y1179	IBM Flex System Manager Per Managed Chassis with 3 Year SW S&S
Advanced feature set	
94Y9219	IBM Flex System Manager, Advanced Upgrade, Per Managed Chassis with 1 Year SW S&S
94Y9220	IBM Flex System Manager, Advanced Upgrade, Per Managed Chassis with 3 Year SW S&S
Fabric Manager	
00D4692	IBM Fabric Manager, Per Managed Chassis with 1 Year SW S&S
00D4693	IBM Fabric Manager, Per Managed Chassis with 3 Year SW S&S

IBM Flex System Manager base feature set offers the following functionality:

- ▶ Support for up to four managed chassis
- ▶ Support for up to 5,000 managed elements
- ▶ Auto-discovery of managed elements
- ▶ Overall health status
- ▶ Monitoring and availability
- ▶ Hardware management
- ▶ Security management
- ▶ Administration
- ▶ Network management (Network Control)
- ▶ Storage management (Storage Control)
- ▶ Virtual machine lifecycle management (VMControl Express)

IBM Flex System Manager advanced feature set offers all capabilities of the base feature set plus the following functions:

- ▶ Image management (VMControl Standard)
- ▶ Pool management (VMControl Enterprise)

IBM Fabric Manager offers the following features:

- ▶ Manage assignments of Ethernet MAC and Fibre Channel worldwide name (WWN) addresses
- ▶ Monitor the health of compute nodes, and automatically without user intervention, replace a failed compute node from a designated pool of spare compute nodes
- ▶ Preassign MAC and WWN addresses, and storage boot targets, for up to 256 chassis or 3584 compute nodes

- By using an enhanced GUI, you can create addresses for compute nodes, save the address profiles, and deploy the addresses to the slots in the same chassis or in up to 256 different chassis

IBM Flex System Manager management software version 1.2.0 provides many new features and enhancements. To see the highlights, refer to the following web page:

http://publib.boulder.ibm.com/infocenter/flexsys/information/topic/com.ibm.acc.8731.doc/whats_new_120.html

7.6.1 Hardware overview

From a hardware point of view, the FSM is a locked-down compute node with a specific hardware configuration designed for optimal performance of the preinstalled software stack. The FSM looks similar to the Intel based x240 compute node. However, there are slight differences between the system board designs making these two hardware nodes not interchangeable. Figure 7-13 shows a front view of the FSM.



Figure 7-13 IBM Flex System Manager

Figure 7-14 shows the internal layout and major components of the FSM.

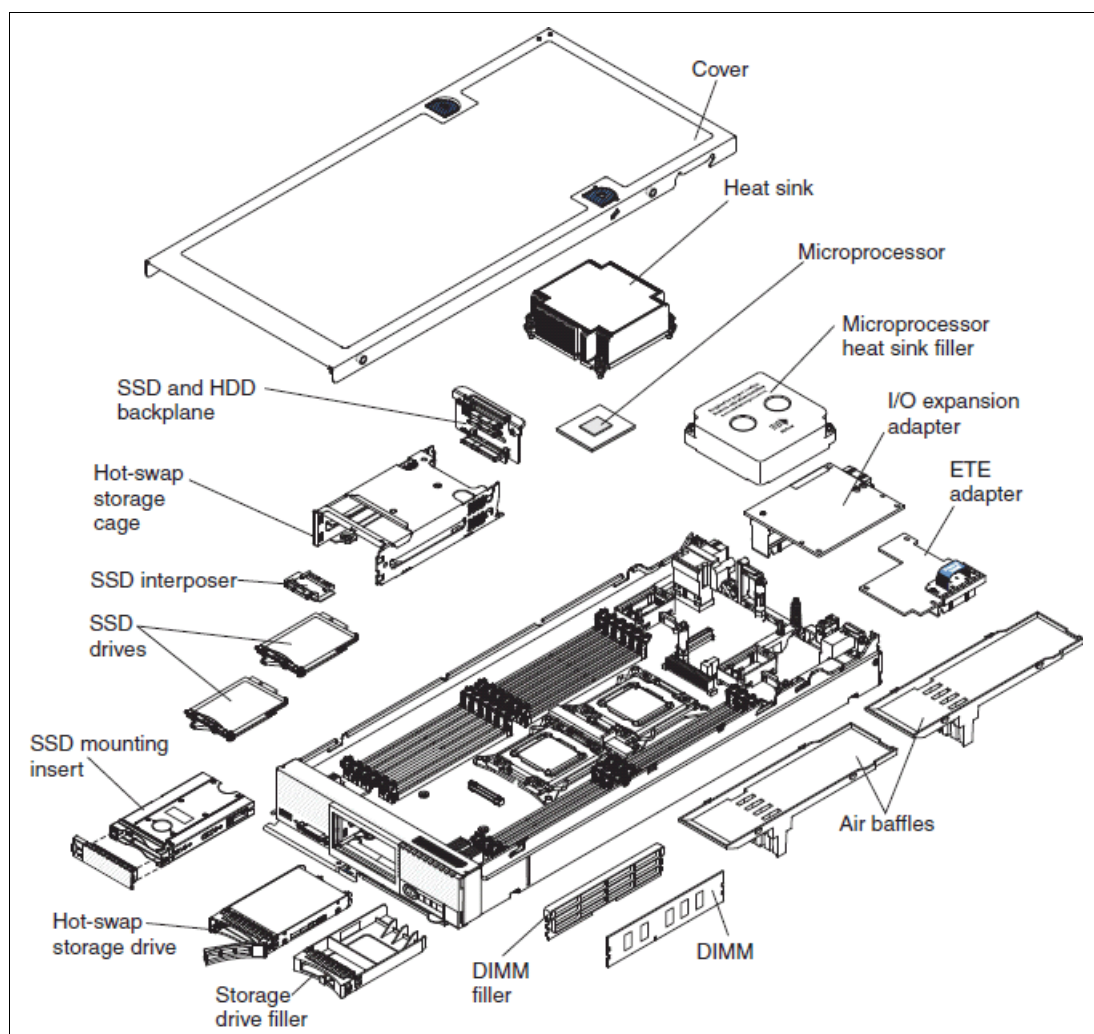


Figure 7-14 Exploded view of the IBM Flex System Manager node, showing major components

Additionally, the FSM comes preconfigured with the components described in Table 7-5.

Table 7-5 Features of the IBM Flex System Manager node (8731)

Feature	Description
Processor	1x Intel Xeon Processor E5-2650 8C 2.0GHz 20MB Cache 1600MHz 95W
Memory	8 x 4GB (1x4GB, 1Rx4, 1.35 V) PC3L-10600 CL9 ECC DDR3 1333MHz LP RDIMM
SAS Controller	One LSI 2004 SAS Controller
Disk	1 x IBM 1TB 7.2K 6Gbps NL SATA 2.5" SFF HS HDD 2 x IBM 200GB SATA 1.8" MLC SSD (configured in a RAID-1)
Integrated network interface card (NIC)	Embedded dual-port 10 Gb Virtual Fabric Ethernet controller (Emulex BE3) Dual-port 1 GbE Ethernet controller on a management adapter (Broadcom 5718)
Systems Management	Integrated Management Module v2 (IMMv2) Management network adapter

Figure 7-15 shows the internal layout of the FSM.

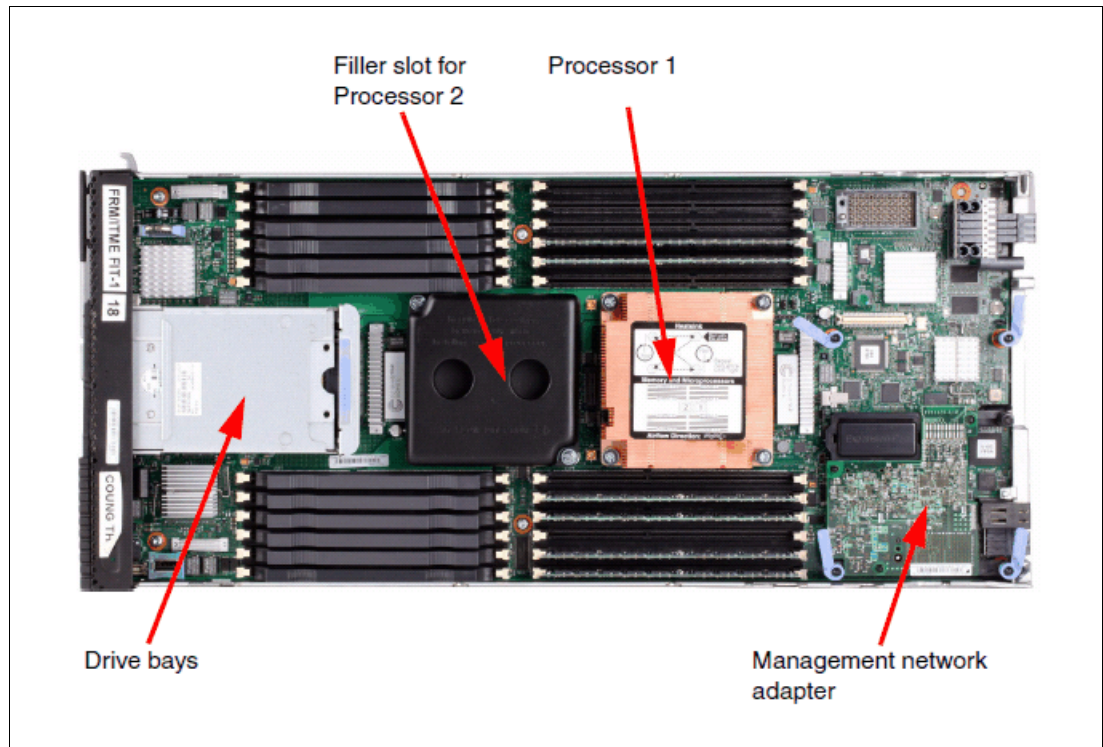


Figure 7-15 Internal view showing major components of IBM Flex System Manager

Front controls

The FSM has similar controls and light-emitting diodes (LEDs) as the IBM Flex System x240 Compute Node. Figure 7-16 shows the front of an FSM with the location of controls and LEDs.

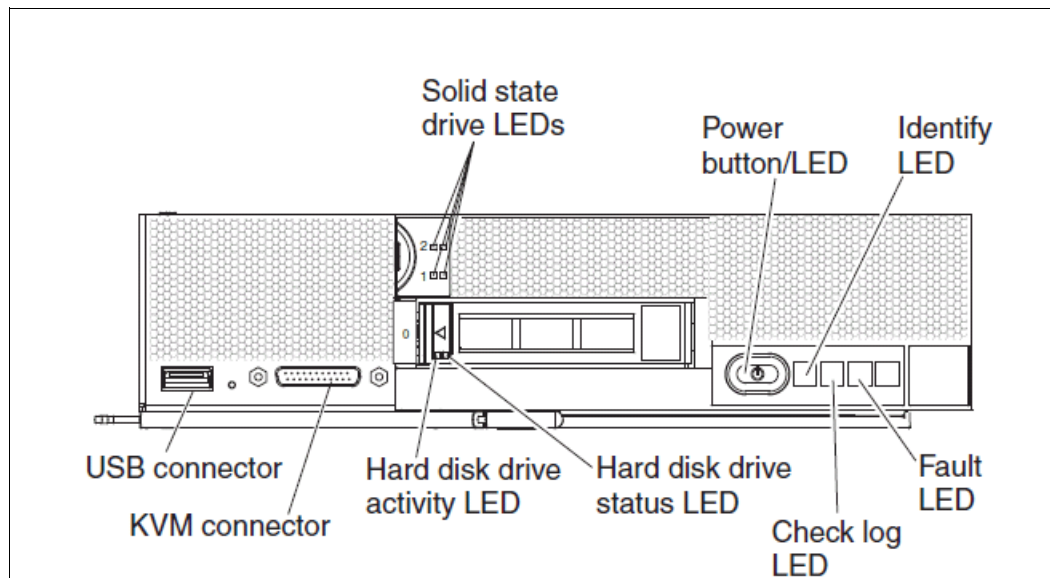


Figure 7-16 FSM front panel: Controls and LEDs

Storage

The FSM ships with 2 x IBM 200 GB SATA 1.8" MLC SSD and 1 x IBM 1 TB 7.2 K 6 Gbps NL Serial Advanced Technology Attachment (SATA) 2.5" SFF HS HDD drives. The 200 GB solid-state drive (SSD) drives are configured as a Redundant Array of Independent Disks 1 (RAID 1) pair providing roughly 200 GB of usable space. The 1 TB SATA drive is not part of a RAID group.

The partitioning of the disks is listed in Table 7-6.

Table 7-6 Detailed SSD and HDD disk partitioning

Physical disk	Virtual disk size	Description
SSD	50 MB	Boot disk
SSD	60 GB	OS/Application disk
SSD	80 GB	Database disk
HDD	40 GB	Update repository
HDD	40 GB	Dump space
HDD	60 GB	Spare disk for OS/Application
HDD	80 GB	Spare disk for database
HDD	30 GB	Service Partition

Management network adapter

The management network adapter is a standard feature of the FSM and provides a physical connection into the private management network of the chassis. The adapter is shown in Figure 7-14 on page 175 as the everything-to-everything (ETE) adapter.

The management network adapter contains a Broadcom 5718 Dual 1 GbE adapter and a Broadcom 5389 8-port L2 switch. This card is one of the features that makes the FSM unique compared to all other nodes supported by the Enterprise Chassis. The management network adapter provides a physical connection into the private management network of the chassis such that the software stack has visibility into both the data and management networks. The L2 switch on this card is automatically setup by the IMMv2 and connects the FSM and the onboard IMMv2 into the same internal private network.

All other nodes supported by the Enterprise Chassis only have a connection into the management network via the management controller (IMMv2 for System x nodes; FSP for POWER nodes), which is not accessible through the operating system.

7.6.2 Software features

The IBM Flex System Manager management software provides the following main features:

- Monitoring and problem determination:
 - A real-time multichassis view of hardware components with overlays for additional information
 - Automatic detection of issues in your environment through event setup that triggers alerts and actions
 - Identification of changes that might affect availability
 - Server resource usage by virtual machine or across a rack of systems

- ▶ Hardware management:
 - Automated discovery of physical and virtual servers and interconnections, applications, and supported third-party networking
 - Inventory of hardware components
 - Chassis and hardware component views
 - Hardware properties
 - Component names and hardware identification numbers
 - Firmware levels
 - Utilization rates
- ▶ Network management:
 - Management of network switches from various vendors
 - Discovery, inventory, and status monitoring of switches
 - Graphical network topology views
 - Support for kernel-based virtual machine (KVM), IBM POWER Hypervisor™ (PHYP), VMware virtual switches, and physical switches
 - VLAN configuration of switches
 - Integration with server management
 - Per-virtual machine network usage and performance statistics provided to VMControl
 - Logical views of servers and network devices grouped by subnet and VLAN
- ▶ Storage management:
 - Discovery of physical and virtual storage devices
 - Support for virtual images on local storage across multiple chassis
 - Inventory of physical storage configuration
 - Health status and alerts
 - Storage pool configuration
 - Disk sparing and redundancy management
 - Virtual volume management
 - Support for virtual volume discovery, inventory, creation, modification, and deletion
- ▶ Virtualization management (base feature set):
 - Support for VMware, Hyper-V, KVM, and IBM PowerVM
 - Create virtual servers
 - Edit virtual servers
 - Manage virtual servers
 - Relocate virtual servers
 - Discover virtual server, storage, and network resources, and visualize the physical-to-virtual relationships
- ▶ Virtualization management (advanced feature set):
 - Create new image repositories for storing virtual appliances and discover existing image repositories in your environment
 - Import external, standards-based virtual appliance packages into your image repositories as virtual appliances

- Capture a running virtual server that is configured just the way you want, complete with guest operating system, running applications, and virtual server definition
- Import virtual appliance packages that exist in the Open Virtualization Format (OVF) from the Internet or other external sources
- Deploy virtual appliances quickly to create new virtual servers that meet the demands of your ever-changing business needs
- Create, capture, and manage workloads
- Create server system pools, which enable you to consolidate your resources and workloads into distinct and manageable groups
- Deploy virtual appliances into server system pools
- Manage server system pools, including adding hosts or additional storage space and monitoring the health of the resources and the status of the workloads in them
- Group storage systems together using storage system pools to increase resource utilization and automation
- Manage storage system pools by adding storage, editing the storage system pool policy, and monitoring the health of the storage resources
- Additional features:
 - Resource-oriented chassis map provides an instant graphical view of chassis resources including nodes and I/O modules:
 - Fly-over provides instant view of individual server (node) status and inventory
 - Chassis map provides inventory view of chassis components, a view of active status requiring administrative attention, and a compliance view of server (node) firmware
 - Actions can be taken on nodes such as working with server-related resources, showing and installing updates, submitting service requests, and launching into the remote access tools
 - Remote console:
 - Open video sessions and mount media such as DVDs with software updates to the servers from local workstation
 - Remote Keyboard, Video, Mouse (KVM) connections
 - Remote Virtual Media connections (mount CD/DVD/ISO/USB media)
 - Power operations against servers (Power On/Off/Restart)
 - Hardware detection and inventory creation
 - Firmware compliance and updates
 - Automatic detection of hardware failures:
 - Provides alerts
 - Takes corrective action
 - Notifies IBM of problems to escalate problem determination
 - Health status (such as CPU utilization) on all hardware devices from a single chassis view
 - Administrative capabilities, such as setting up users within profile groups, assigning security levels, and security governance

7.6.3 Supported agents, hardware, operating systems and tasks

IBM Flex System Manager provides four tiers of agents for managed systems. For each managed system, you must choose the tier that provides the amount and level of capabilities that you need for that managed system. Depending on the type of managed system and the management tasks that you need to perform, you can choose the level of agent capabilities that best fits your needs.

IBM Flex System Manager has four agent tiers:

- ▶ **Agentless in-band**
Managed systems without any FSM client software installed. FSM communicates with the managed system through the operating system.
- ▶ **Agentless out-of-band**
Managed systems without any FSM client software installed. FSM communicates with the managed system through something other than the operating system, such as a service processor or a Hardware Management Console.
- ▶ **Platform Agent**
Managed systems with Platform Agent installed. FSM communicates with the managed system through the Platform Agent.
- ▶ **Common Agent**
Managed systems with Common Agent installed. FSM communicates with the managed system through the Common Agent.

Table 7-7 lists the agent tier support for the IBM Flex System managed compute nodes. Managed nodes include x240 compute nodes supporting Windows, Linux, and VMware, and p260 and p460 compute nodes supporting IBM AIX, IBM i, and Linux.

Table 7-7 Agent tier support by management system type

Managed system type	Agent tier	Agentless in-band	Agentless out-of-band	Platform Agent	Common Agent
Compute nodes running AIX		Yes	Yes	No	Yes
Compute nodes running IBM i		Yes	Yes	Yes	Yes
Compute nodes running Linux		No	Yes	Yes	Yes
Compute nodes running Linux and supporting SSH		Yes	Yes	Yes	Yes
Compute nodes running Windows		No	Yes	Yes	Yes
Compute nodes running Windows and supporting SSH or DCOM		Yes	Yes	Yes	Yes
Compute nodes running VMware		Yes	Yes	Yes	Yes
Other managed resources supporting SSH or SNMP		Yes	Yes	No	No

Table 7-8 summarizes the management tasks supported by the compute nodes depending on agent tier.

Table 7-8 Compute node management tasks supported by the agent tier

Managed system type	Agent tier	Agentless in-band	Agentless out-of-band	Platform Agent	Common Agent
Command Automation	No	No	No	No	Yes
Hardware alerts	No	Yes	Yes	Yes	Yes
Platform alerts	No	No	No	Yes	Yes
Health and status monitoring	No	No	No	Yes	Yes
File Transfer	No	No	No	No	Yes
Inventory (hardware)	No	Yes	Yes	Yes	Yes
Inventory (software)	Yes	No	Yes	Yes	Yes
Problems (hardware status)	No	Yes	Yes	Yes	Yes
Process Management	No	No	No	No	Yes
Power Management	No	Yes	No	No	Yes
Remote Control	No	Yes	No	No	No
Remote Command Line	Yes	No	Yes	Yes	Yes
Resource Monitors	No	No	Yes	Yes	Yes
Update Manager	No	No	Yes	Yes	Yes

Table 7-9 shows supported virtualization environments and their management tasks.

Table 7-9 Supported virtualization environments and management tasks

Virtualization environment	AIX and Linux^a	IBM i	VMware vSphere	Microsoft Hyper-V	Linux KVM
Deploy virtual servers	Yes	Yes	Yes	Yes	Yes
Deploy virtual farms	No	No	Yes	No	Yes
Relocate virtual servers	Yes	No	Yes	No	Yes
Import virtual appliance packages	Yes	Yes	No	No	Yes
Capture virtual servers	Yes	Yes	No	No	Yes
Capture workloads	Yes	Yes	No	No	Yes
Deploy virtual appliances	Yes	Yes	No	No	Yes
Deploy workloads	Yes	Yes	No	No	Yes
Deploy server system pools	Yes	No	No	No	Yes
Deploy storage system pools	Yes	No	No	No	No

a. Linux on Power Systems compute nodes

Table 7-10 shows supported I/O switches and their management tasks.

Table 7-10 Supported I/O switches and management tasks

I/O module Management task	EN2092 1 Gb Ethernet	EN4093/ EN4093R 10 Gb Ethernet	FC3171 8 Gb SAN	FC5022 16 Gb SAN
Discovery	Yes	Yes	Yes	Yes
Inventory	Yes	Yes	Yes	Yes
Monitoring	Yes	Yes	Yes	Yes
Alerts	Yes	Yes	Yes	Yes
Configuration	Yes	Yes	Yes	No

Table 7-11 shows IBM StorWize V7000 supported management tasks.

Table 7-11 IBM Storwize® V7000 supported management tasks

Management task	Storage system V7000
Storage device discovery	Yes
Integrated physical and logical topology views	Yes
Show relationships between storage and server resources	Yes
Perform logical and physical configuration	Yes
View controller and volume status and to set notification alerts	Yes

For more information, see the following IBM Flex System Manager product publications, available from the IBM Flex System Information Center:

<http://publib.boulder.ibm.com/infocenter/flexsys/information/index.jsp>

7.7 IBM System Networking Element Manager Component

IBM System Networking Element Manager Component (SNEM-C, formerly BladeHarmony Manager) is an application developed specifically to manage the IBM System Networking portfolio including BladeCenter, PureFlex, 5000v Distributed Virtual, and Top-of-Rack Ethernet switches. It helps network administrators monitor, configure, and view summary reports for IBM System Networking switches.

The SNEM home page gives a quick summary of the discovered devices. It provides a graphical representation of health status, panic dump, events, save pending, running software version, and device discovery time stamp, grouped into separate panes along with the device counts.

SNEM-C provides several useful tools for managing and monitoring VMready and Edge Virtual Bridging (IEEE 802.1Qbg) configuration using the 5000v Distributed Virtual Switch. For more information, see the IBM Redbooks publication *Implementing a VM-Aware Network Using VMready*, SG24-7985.

A sample System Networking Element Manager home page is shown in Figure 7-17.

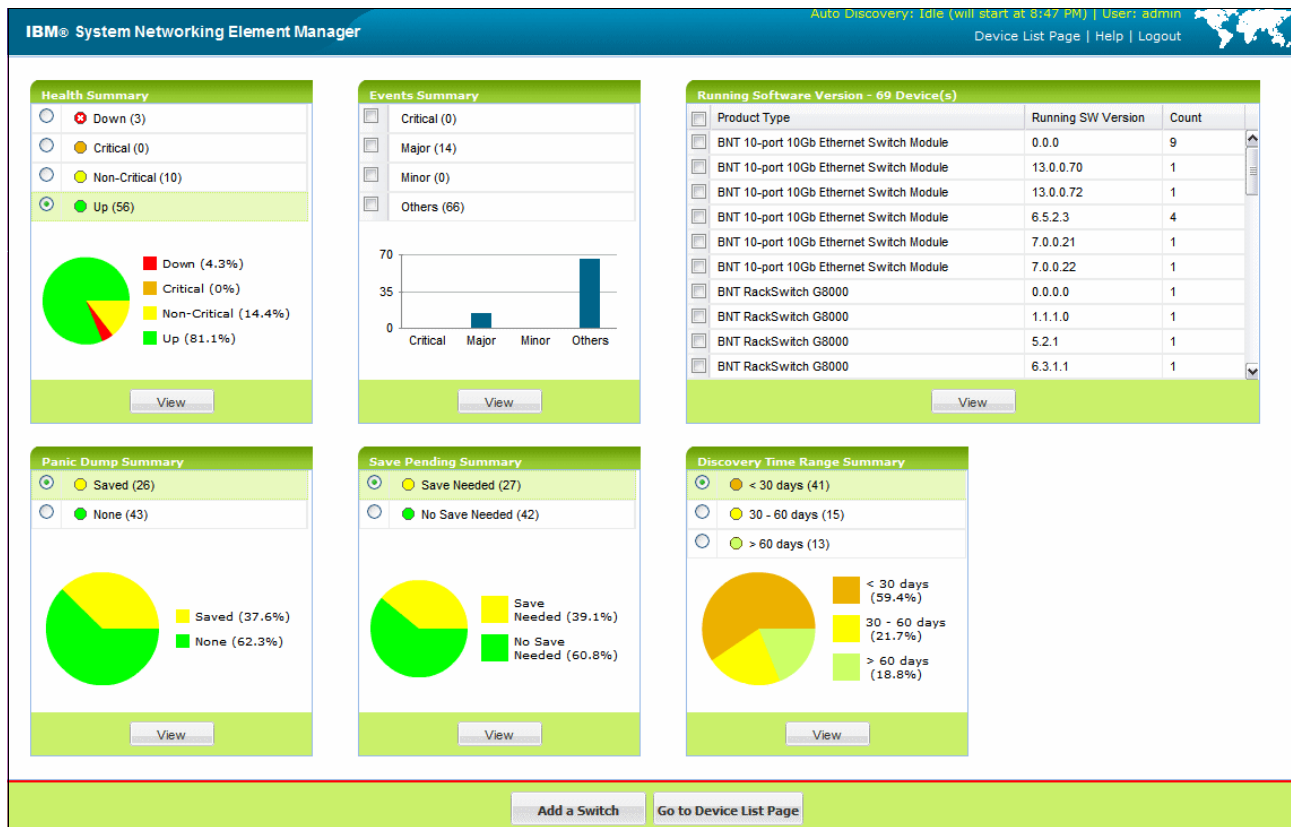


Figure 7-17 System Networking Element Manager home page example

The information is updated periodically to provide the actual counts and status of managed devices. It provides an option for the user to filter the devices available on the device list page based on the selection made here.

7.7.1 Health Summary pane

The Health Summary pane shows the individual count of devices discovered that are Down (red), Critical (orange), Non-Critical (yellow), and Up (green). It also provides a pie chart that indicates the percentages of Down, Critical, Non-Critical, and Up devices. A sample Health Summary pane is shown in Figure 7-18.

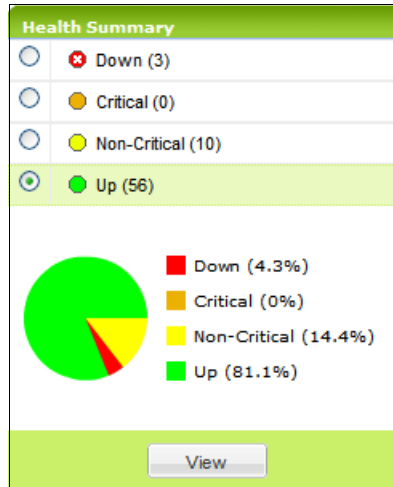


Figure 7-18 Health Summary pane

You can filter out the devices of the Health Summary by selecting the appropriate choice and clicking **View**, which takes you to the device list page.

7.7.2 Viewing Health Status

The Health Status page shows CPU and Memory Utilization, ARP and Routing Table Utilization, Power Supply status, Panic Dump status, Temperature Sensors reading, and Fan Speed. A sample Health Status window is shown in Figure 7-19 on page 185.

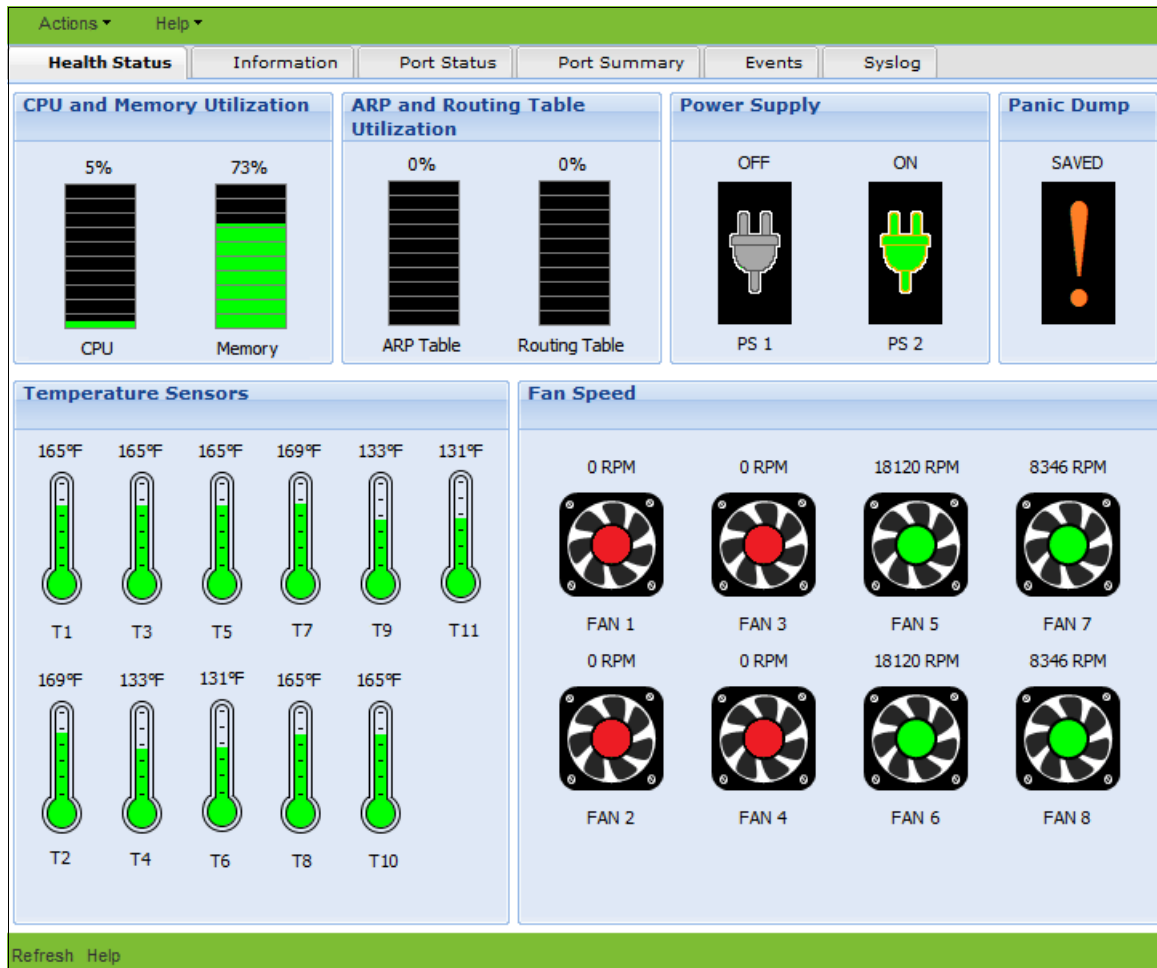


Figure 7-19 Health Status window

7.7.3 Viewing reports

You can view various reports associated with all the discovered switches by choosing the items under the Reports menu in SNEM.

The list of reports include:

- ▶ Event List Report
- ▶ Syslog List Report
- ▶ SNEM Alerts Report
- ▶ Switch Version Report
- ▶ Transceiver Information Report
- ▶ VM Data Center Report
- ▶ VMready VM Report
- ▶ VMready VM Report – Port Groups Report

7.8 IBM System Networking Element Manager Solution

IBM System Networking Element Manager Solution is a web-based application for monitoring and management of IBM System Networking switches and connected devices. This solution allows for automation of basic network tasks, including remote monitoring and management of Ethernet switches. The different software components of the SNEM solution are shown in Figure 7-20.

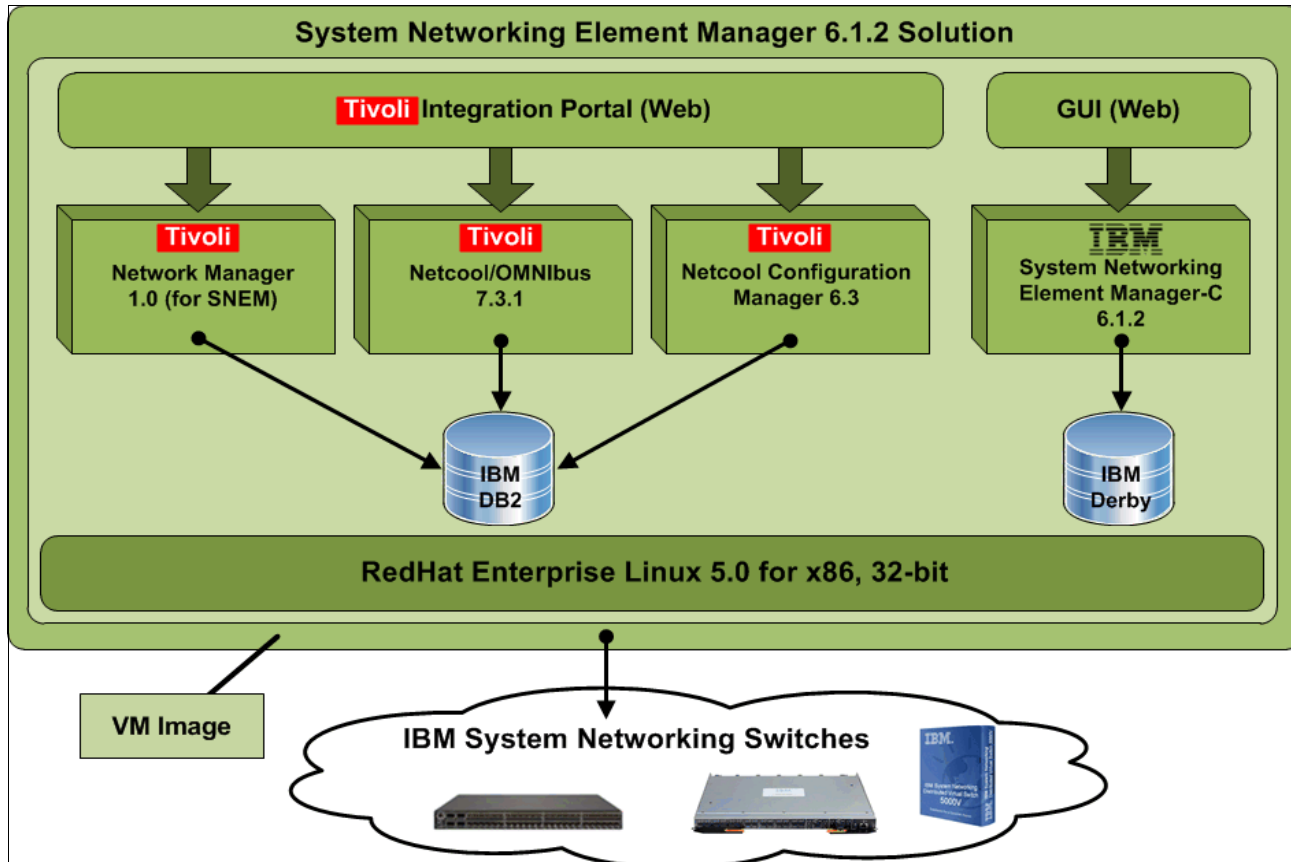


Figure 7-20 System Networking Element Manager 6.1.2 Solution

The SNEM solution is a VM package that contains the following network management software packages:

- ▶ IBM System Networking Element Manager Component (SNEM-C) V6.1.2
- ▶ IBM Tivoli Network Manager V1.0 for System Networking Element Manager
- ▶ IBM Tivoli Netcool Configuration Manager V6.3
- ▶ IBM Tivoli Netcool/OMNIBus V7.3.1

IBM Tivoli Network Manager V1 for SNEM

Tivoli Network Manager provides the features necessary to manage complex networks. These features include network discovery, device polling, including storage of polled SNMP and ICMP data for reporting and analysis, and topology visualization. Tivoli Network Manager can display network events, perform root-cause analysis of network events, and enrich network events with topology and other network data. Tivoli Network Manager integrates with other IBM products, such as Tivoli Business Service Manager, Tivoli Application Dependency Discovery Manager, and Systems Director.

By using Tivoli Network Manager, you can perform the following tasks:

- ▶ Manage complex networks.
- ▶ View the network in multiple ways.
- ▶ Apply ready-to-use device and interface polling capabilities.
- ▶ Use built-in root-cause analysis capabilities.
- ▶ Troubleshoot network problems using right-click tools.
- ▶ Generate richer network visualization and event data.
- ▶ Discover increasingly bigger networks.
- ▶ Run reports to retrieve essential network data.
- ▶ Build custom multi-portlet pages.

Tivoli Netcool Configuration Manager

Tivoli Netcool Configuration Manager provides configuration management support for network devices, including extensive configuration policy threshold capabilities.

Tivoli Netcool/OMNIBus

The Tivoli Netcool/OMNIBus software collects and manages network event information, and delivers real-time, centralized monitoring of complex networks and IT domains.

Tivoli Netcool/OMNIBus tracks alert information in a high-performance, in-memory database, and presents information of interest to specific users through filters and views that can be configured individually. Tivoli Netcool/OMNIBus has automation functions that can perform intelligent processing on managed alerts.

7.8.1 IBM System Networking Element Manager solution requirements

IBM System Networking Element Manager V6.1.2 is a software product that is distributed as a virtual appliance. A virtual software appliance requires a hypervisor to enable it to run. SNEM V6.1.2 supports the following hypervisors:

- ▶ Linux kernel-based virtual machine (KVM)
- ▶ VMware ESX/ESXi

Installation of the ESX version and KVM are independent and mutually exclusive and you must choose the version based on the type of hypervisor that you are using.

More information

For more information about SNEM, see the following publications:

- ▶ *IBM SNEM 6.1.1 Solution Getting Started Guide:*
<http://www.ibm.com/support/docview.wss?uid=isg3T7000564>
- ▶ *IBM SNEM 6.1.2 Application User Guide:*
<http://www.ibm.com/support/docview.wss?uid=isg3T7000561>
- ▶ *IBM System Networking Element Manager Solution Device Support List (6.1):*
<http://www.ibm.com/support/docview.wss?uid=isg3T7000474>
- ▶ *Quick Start Guide for installing and running KVM:*
http://pic.dhe.ibm.com/infocenter/lnxinfo/v3r0m0/topic/liaai.kvminstall/kvminstall_pdf.pdf



Troubleshooting and maintenance

In this chapter, we explain the troubleshooting and maintenance steps on IBM PureFlex System switches, with an emphasis on the EN4093/EN4093R switch. We describe the following topics:

- ▶ Troubleshooting
- ▶ Configuration management
- ▶ Firmware management
- ▶ Logging and reporting

8.1 Troubleshooting

In this section, we show the basic troubleshooting tools and techniques. We describe various troubleshooting steps, such as inspecting light-emitting diodes (LEDs) on the switch, troubleshooting network connectivity, port mirroring for capturing data traffic, and the use of serial connection.

8.1.1 Basic troubleshooting procedures

This section contains basic troubleshooting information to help resolve problems that might occur during the installation and operation of your EN4093/EN4093R switch. We recommend downloading and using the EN4093/EN4093R documentation, available on the IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch Information Center web page:

http://publib.boulder.ibm.com/infocenter/flexsys/information/topic/com.ibm.acc.net.workdevices.doc/Io_module_compassplus.html

LEDs on EN4093/EN4093R

The EN4093/EN4093R switch contains the following LEDs for easy identification of switch and port status:

- System status LEDs (shown in Figure 8-1).

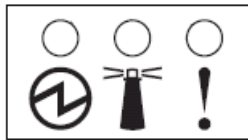


Figure 8-1 System status LEDs: OK, Identify, and Error (left to right)

The meaning of three system status LEDs (OK, Identify, and Error) is as follows:

- OK (green):
 - When this LED is lit, it indicates that the switch is powered on.
 - When this LED is not lit, but yellow Error LED is lit, it indicates a critical alert.
 - When both LEDs are off, this indicates that the switch is off.
- Identify (blue):

You can use this LED to identify the location of the switch in the chassis. Use the Chassis Management Module (CMM) web interface to change the state of this LED:

- i. Click **Chassis Management** → **I/O Modules** in the CMM web graphical user interface (GUI).

The window that is shown in Figure 8-2 on page 191 displays.

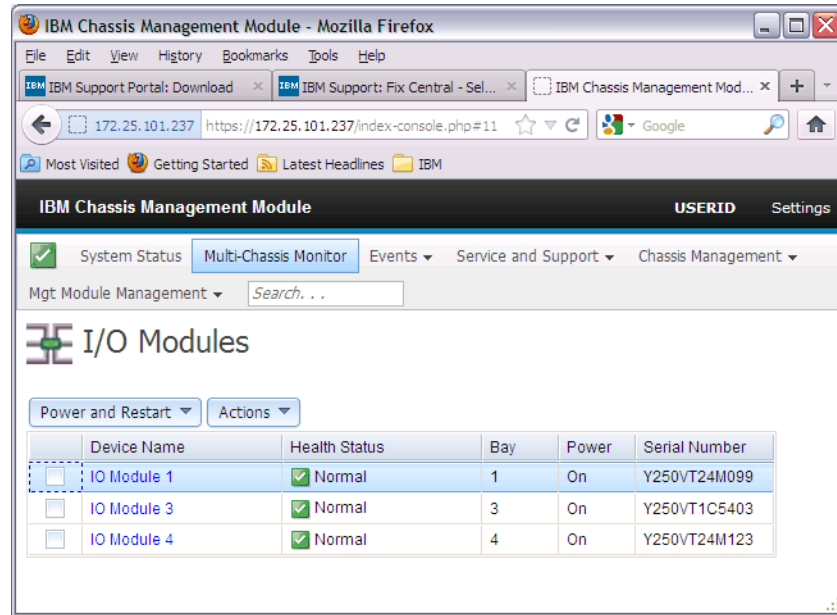


Figure 8-2 Select I/O module

- ii. Click the I/O module that you want to identify. In our case, we click **IO Module 1**. This opens the window that is shown in Figure 8-3.

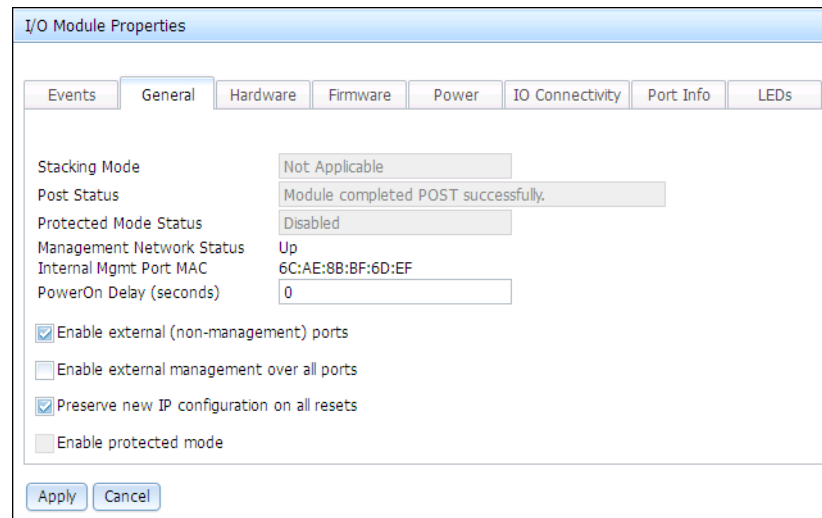


Figure 8-3 I/O module properties

- iii. Click the **LEDs** tab to display the window that is shown in Figure 8-4.

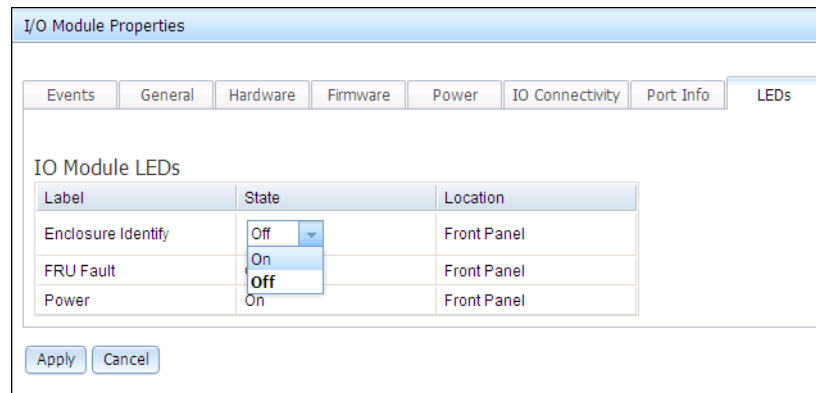


Figure 8-4 Toggle the Identify LED state

You can now toggle the Identify LED state for easy identification of the switch in the chassis.

- Error (yellow)

When this LED is lit, it indicates a critical alert or power-on self-test (POST) failure.

- ▶ Enhanced small form-factor pluggable (SFP+) and quad small form-factor pluggable (QSFP+) module port LEDs (shown in Figure 8-5 and Figure 8-6 on page 193).

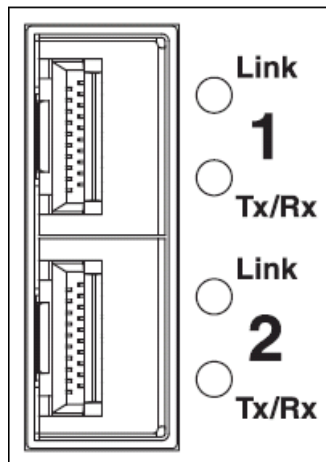


Figure 8-5 SFP+ port LEDs

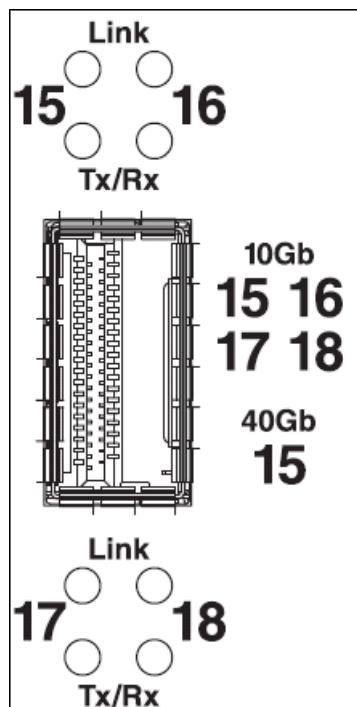


Figure 8-6 QSFP+ port LEDs

The Link and Tx/Rx LEDs function as explained here:

- Link (green):
 - When this LED is lit, there is an active connection between the port and the connected device.
 - When the LED is not lit, there is no signal on the port, or the link is down.
- Tx/Rx (green):

When this LED is flashing, link activity is occurring on the port.

Port link LED does not light

Symptom: The port link LED does not light.

Solution 1: Check the port configuration. If the port is configured with a specific speed or duplex mode, check the other device to verify that it is set to the same configuration. If the switch port is set to autonegotiate, verify that the other device is set to autonegotiate.

Solution 2: Check the cables that connect the port to the other device. Make sure that they are connected. Verify that you are using the correct cable type.

Switch does not boot

Symptom: All the switch LEDs stay on, and the command prompt is not displayed on the console.

Solution: The switch firmware might be damaged. Use the console port to perform a serial upgrade of the switch firmware, as documented in section 8.3.3, “Recovering from a failed firmware upgrade” on page 212.

8.1.2 Connectivity troubleshooting

In this section, you find basic information about how to troubleshoot the IP connectivity in a network built on IBM System Networking switches. IBM switches come with a set of simple tools that can be helpful for troubleshooting IP connectivity issues.

Ping

The **ping** command is a simple tool, which is based on a request-response mechanism, to verify connectivity to a remote network node. The **ping** command is based on Internet Control Message Protocol (ICMP). The request is an ICMP echo packet and the reply is an ICMP echo reply. Like a regular IP packet, an ICMP packet is forwarded based on the intermediate routers' routing table until it reaches the destination. After it reaches the destination, the ICMP echo reply packet is generated and forwarded back to the originating node.

Important: In IBM switches, **ping** sends an ICMP echo packet on the management interface first. If you want to change that option, you must add the **data-port** keyword to a command as a parameter.

Example 8-1 shows the use of the **ping** command to verify connectivity between the switch and IP address 172.25.101.237.

Example 8-1 Ping command example

```
en4093flex_1#ping 172.25.101.237
Connecting via MGT port.
[host 172.25.101.237, max tries 5, delay 1000 msec, length 0, ping

source N/S, ttl 255, tos 0]
172.25.101.237: #1 ok, RTT 1 msec.
172.25.101.237: #2 ok, RTT 2 msec.
172.25.101.237: #3 ok, RTT 2 msec.
172.25.101.237: #4 ok, RTT 1 msec.
172.25.101.237: #5 ok, RTT 2 msec.
Ping finished.
```

You can see in the output that all five ICMP echo requests received the replies. There is also more information about the Round Trip Time (RTT), that is, the time that it took for the switch to receive a response.

Traceroute

You can use the **traceroute** command to not only verify connectivity to a remote network node, but to track the responses from intermediate nodes as well. This action is done by using the time to live (TTL) field in IP packets. The **traceroute** command sends a User Datagram Protocol (UDP) packet to a port that is likely to not be used on a remote node with a TTL of 1. After the packet reaches the intermediate router, the TTL is decremented, and the ICMP time-exceeded message is sent back to the originating node, which increments the TTL to 2, and the process repeats. After the UDP packet reaches a destination host, an ICMP port-unreachable message is sent back to the sender. This action provides the sender with information about all intermediate routers on the way to the destination.

The command that is shown in Example 8-2 verifies which hops are on the way from the switch to the system with IP address 10.0.100.1.

Example 8-2 Traceroute command example

```
ACC-2#traceroute 10.0.100.1 data-port
Connecting via DATA port.
[host 10.0.100.1, max-hops 32, delay 2048 msec]
 1  10.0.100.1      0 ms
Trace host responded.
```

From the output, you see that there is only one hop on the way from the switch to the destination. We use Open Shortest Path First (OSPF) in our network, which selects this path as the shortest one.

For test purposes, we shut down the direct link between the switch and target system and run **traceroute** again. The output is shown in Example 8-3.

Example 8-3 Traceroute command example

```
ACC-2#traceroute 10.0.100.1 data-port
Connecting via DATA port.
[host 10.0.100.1, max-hops 32, delay 2048 msec]
 1  10.0.104.1      0 ms
 2  10.0.100.1      1 ms
Trace host responded.
```

Now you can see that to reach the destination, the switch uses the 10.0.104.1 system as the intermediate router.

8.1.3 Port mirroring

You can use the IBM System Networking switches port mirroring feature to mirror (copy) the packets of a target port, and forward them to a monitoring port. Port mirroring functions for all Layer 2 and Layer 3 traffic on a port. This feature can be used as a troubleshooting tool or to enhance the security of your network.

For example, an intrusion detection system (IDS) server or other traffic sniffer device or analyzer can be connected to the monitoring port to detect intruders that attack the network.

IBM System Networking switches support a “many to one” mirroring model. As shown in Figure 8-7, selected traffic for ports 1 and 2 is being monitored by port 3. In the example, both ingress traffic and egress traffic on port 2 are copied and forwarded to the monitor. However, port 1 mirroring is configured so that only ingress traffic is copied and forwarded to the monitor. A device that is attached to port 3 can capture and analyze the resulting mirrored traffic.

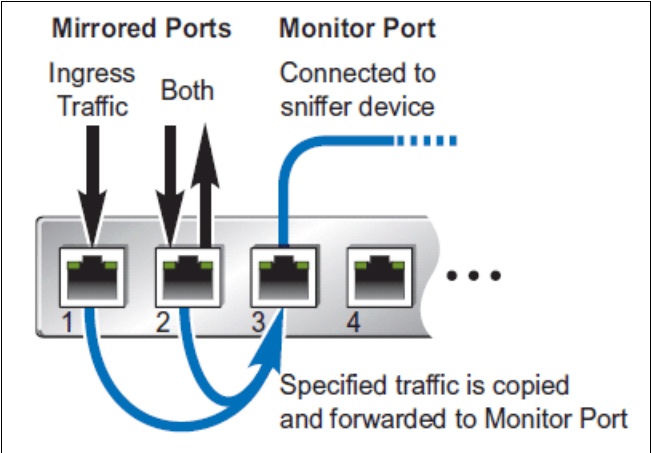


Figure 8-7 Mirroring ports

The composition of monitored packets in the EN4093/EN4093R, based on the configuration of the ports, works like this:

- ▶ Packets that are mirrored at port egress are mirrored before VLAN tag processing and might have a different Port VLAN ID (PVID) than packets that egress the port toward their actual network destination.
- ▶ Packets that are mirrored at port ingress are not modified.

In Example 8-4, we show the Industry Standard CLI (ISCLI) commands to enable port mirroring and to mirror ingress and egress traffic on ports EXT1 - EXT4 to monitoring port EXT6.

Example 8-4 Port mirroring ISCLI commands

```
en4093flex_1(config)#port-mirroring enable
en4093flex_1(config)#port-mirroring monitor-port EXT6 mirroring-port EXT1-EXT4
both
```

You can check the port mirroring configuration with the **show port-mirroring** ISCLI command. As shown in Example 8-5, both ingress and egress traffic on ports EXT1 - EXT4 will be mirrored to monitoring port EXT6.

Example 8-5 Port mirroring configuration verification

```
en4093flex_1(config)#show port-mirroring
Port Mirroring is enabled

Monitoring port  Mirrored ports
INTA1            none
INTA2            none
INTA3            none
...
Lines deleted for clarity
```

```

...
EXT5          none
EXT6          (EXT1,both) (EXT2,both) (EXT3,both) (EXT4,both)
EXT7          none
...
Lines deleted for clarity
...

```

8.1.4 Serial cable troubleshooting procedures

When all else fails, you can use the serial cable that is delivered with EN4093/EN4093R to connect to the switch and investigate the problem. A terminal emulation utility must run on management system (such as Windows Hyperterminal or PuTTY). Use the following serial connection parameters:

- ▶ Speed: 9600 bps
- ▶ Data Bits: 8
- ▶ Stop Bits: 1
- ▶ Parity: None
- ▶ Flow Control: None

When the serial session is established, you must reboot the EN4093/EN4093R switch to start the Boot Management Menu with recovery options. In the CMM web GUI, you can either power-cycle the affected EN4093/EN4093R switch, or restart it.

When you see the memory test run in the terminal window, press **Shift+B** to display the menu with recovery options. Example 8-6 shows the Boot Management Menu.

Example 8-6 Boot Management Menu

```

Resetting the System ...
Memory Test .....
Boot Management Menu
    1 - Change booting image
    2 - Change configuration block
    3 - Boot in recovery mode (tftp and xmodem download of images to recover
switch)
    4 - Xmodem download (for boot image only - use recovery mode for
application images)
    5 - Reboot
    6 - Exit
Please choose your menu option:

```

By using the Boot Management Menu, you can perform the following tasks:

- ▶ Change the active boot image from image1 to image2 or vice versa. Section “Changing the boot image using serial interface” on page 207 shows how to do this task.
- ▶ Change the active configuration block. You can select between active, backup, and factory default configuration blocks. This option can be used to restore the EN4093/EN4093R switch to factory defaults, as shown in section “Resetting with no terminal access to the switch” on page 205.
- ▶ Download new firmware to the switch. This option can be helpful if you must recover the switch after a failed firmware upgrade. We show an example of firmware recovery in section 8.3.3, “Recovering from a failed firmware upgrade” on page 212.

8.2 Configuration management

This section describes how to manage configuration files and how to save and restore a configuration in the switch.

8.2.1 Configuration files

The switch stores its configuration in two files:

- ▶ `startup-config` is the configuration that the switch uses when it is reloaded.
- ▶ `running-config` is the configuration that reflects all the changes that you made from the CLI. It is stored in memory and is lost after the reload of the switch.

8.2.2 Configuration blocks

The switch stores its configuration in one of two configuration blocks:

- ▶ `active-config` is stored in the active configuration block.
- ▶ `backup-config` is stored in the backup configuration block.

When you save the running configuration (`copy running-config startup-config`), the new configuration is placed into the active configuration block. The previous configuration is copied into the backup configuration block.

In addition, there is also a factory configuration block. This block holds the factory default configuration, allowing you to restore the switch to factory defaults if needed.

This setup has the flexibility that you need to manage the configuration of the switch and perform a possible configuration rollback.

Use the following command to select the configuration block that the switch will load on the next reboot:

```
Switch# boot configuration-block {active|backup|factory}
```

8.2.3 Managing configuration files

This section describes the different ways of managing the configuration files.

Managing the configuration using ISCLI

You can manage the configuration files by using several commands:

- ▶ Run the following command to display the current configuration file:

```
Switch#show running-config
```

- ▶ Run the following command to copy the current (running) configuration from switch memory to the `startup-config` partition:

```
Switch#copy running-config startup-config
```

The following command also copies running configuration to the startup configuration:

```
Switch#write memory
```

- ▶ Run the following command to copy the current (running) configuration from switch memory to the `backup-config` block:

```
Switch#copy running-config backup-config
```

- Run the following command to back up the current configuration to a file on an FTP/TFTP server:

```
Switch#copy running-config {ftp|tftp}
```

- Run the following command to restore the current configuration from an FTP/TFTP server.

```
Switch#copy {ftp|tftp} running-config
```

Managing the configuration through SNMP

This section describes how to use Management Information Base (MIB) calls to work with switch configuration files.

You can use a standard SNMP tool to perform the actions, by using the MIBs listed in Table 8-1. For information about how to set up your switch to use Simple Network Management Protocol (SNMP), see 8.4.2, “Simple Network Management Protocol” on page 217.

Table 8-1 SNMP MIBs for managing switch configuration and firmware

MIB name	MIB OID
agTransferServer	1.3.6.1.4.1872.2.5.1.1.7.1.0
agTransferImage	1.3.6.1.4.1872.2.5.1.1.7.2.0
agTransferImageFileName	1.3.6.1.4.1872.2.5.1.1.7.3.0
agTransferCfgFileName	1.3.6.1.4.1872.2.5.1.1.7.4.0
agTransferDumpFileName	1.3.6.1.4.1872.2.5.1.1.7.5.0
agTransferAction	1.3.6.1.4.1872.2.5.1.1.7.6.0
agTransferLastActionStatus	1.3.6.1.4.1872.2.5.1.1.7.7.0
agTransferUserName	1.3.6.1.4.1872.2.5.1.1.7.9.0
agTransferPassword	1.3.6.1.4.1.1872.2.5.1.1.7.10.0
agTransferTSDumpFileName	1.3.6.1.4.1.1872.2.5.1.1.7.11.0

The following configuration-related SNMP actions can be performed by using the MIBs listed in Table 8-1:

- Load a previously saved switch configuration from an FTP/TFTP server.
- Save the switch configuration to an FTP/TFTP server.

You can also use the SNMP MIBs in Table 8-1 to perform other functions, such as upgrading the switch firmware and saving the switch dump to an FTP/TFTP server.

Loading a saved configuration

To load a saved switch configuration with the name `MyRunningConfig.cfg` into the switch, complete the following steps. This example shows a TFTP server at IPv4 address `172.25.101.200` (although IPv6 is also supported) where the previously saved configuration is available for download:

1. Set the FTP/TFTP server address where the switch configuration file is located:

```
Set agTransferServer.0 "172.25.101.200"
```

2. Set the name of the configuration file:

```
Set agTransferCfgFileName.0 "MyRunningConfig.cfg"
```

3. If you are using an FTP server, enter a user name:

```
Set agTransferUserName.0 "MyName"
```

4. If you are using an FTP server, enter a password:

```
Set agTransferPassword.0 "MyPassword"
```

5. Initiate the transfer. To restore a running configuration, use transfer action 3:

```
Set agTransferAction.0 "3"
```

Saving the configuration

To save the switch configuration to an FTP/TFTP server, complete the following steps. This example shows an FTP/TFTP server at IPv4 address 172.25.101.200, although IPv6 is also supported:

1. Set the FTP/TFTP server address where the configuration file is saved:

```
Set agTransferServer.0 "172.25.101.200"
```

2. Set the name of the configuration file:

```
Set agTransferCfgFileName.0 "MyRunningConfig.cfg"
```

3. If you are using an FTP server, enter a user name:

```
Set agTransferUserName.0 "MyName"
```

4. If you are using an FTP server, enter a password:

```
Set agTransferPassword.0 "MyPassword"
```

5. Initiate the transfer. To save a running configuration file, use transfer action 4.

```
Set agTransferAction.0 "4"
```

Other tasks: Saving a switch dump

SNMP MIBs are not only useful to save and load switch configurations. You can also perform other tasks, such as saving a switch dump. To save a switch dump to an FTP/TFTP server, complete the following steps. This example shows an FTP/TFTP server at 172.25.101.200, although IPv6 is also supported:

1. Set the FTP/TFTP server address where the configuration is saved:

```
Set agTransferServer.0 "172.25.101.200"
```

2. Set the name of the dump file:

```
Set agTransferDumpFileName.0 "MyDumpFile.dmp"
```

3. If you are using an FTP server, enter a user name:

```
Set agTransferUserName.0 "MyName"
```

4. If you are using an FTP server, enter a password:

```
Set agTransferPassword.0 "MyPassword"
```

5. Initiate the transfer. To save a dump file, use transfer action 5.

```
Set agTransferAction.0 "5"
```

8.2.4 Resetting to factory defaults

You might need to reset the switch to factory defaults in certain situations, for example, when redeploying the switch for use in a different scenario, or when troubleshooting a configuration issue. To reset the switch to factory defaults, you need to perform one of the following procedures.

Resetting EN4093/EN4093R to factory defaults via CMM

Follow these steps to reset EN4093/EN4093R to factory defaults via CMM:

1. Point your web browser to the CMM IP address, and log in. See Figure 8-8.

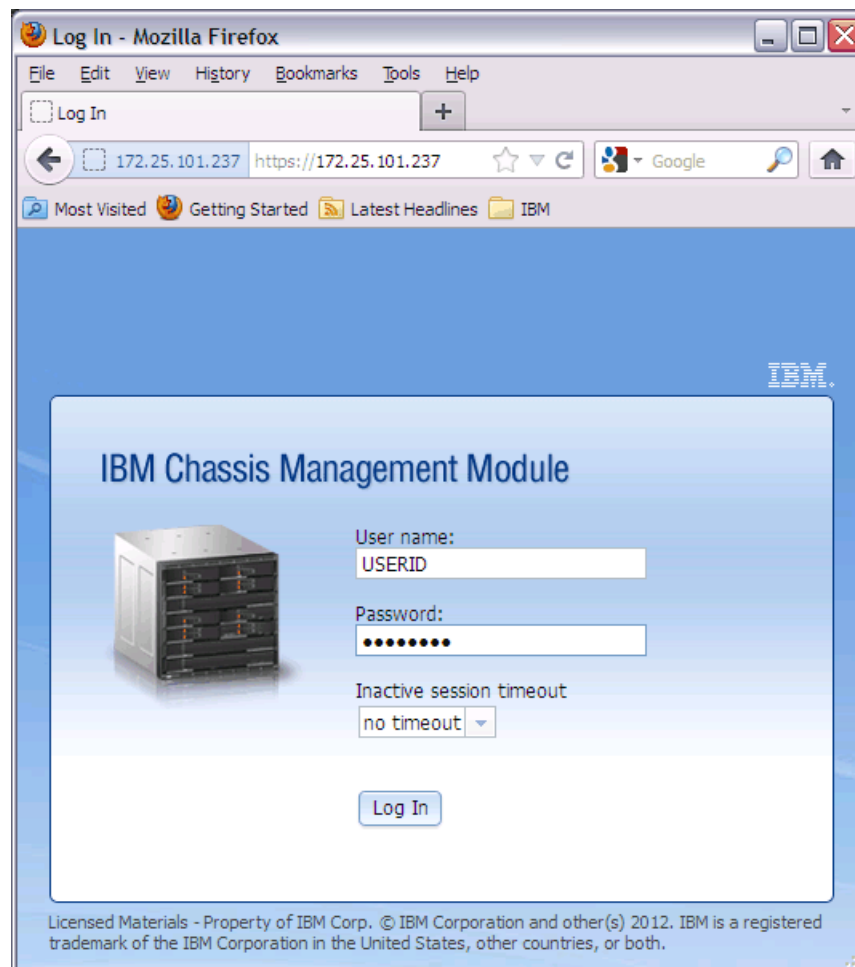


Figure 8-8 Log in to CMM

2. After successfully logging in, the CMM GUI displays (see Figure 8-9).

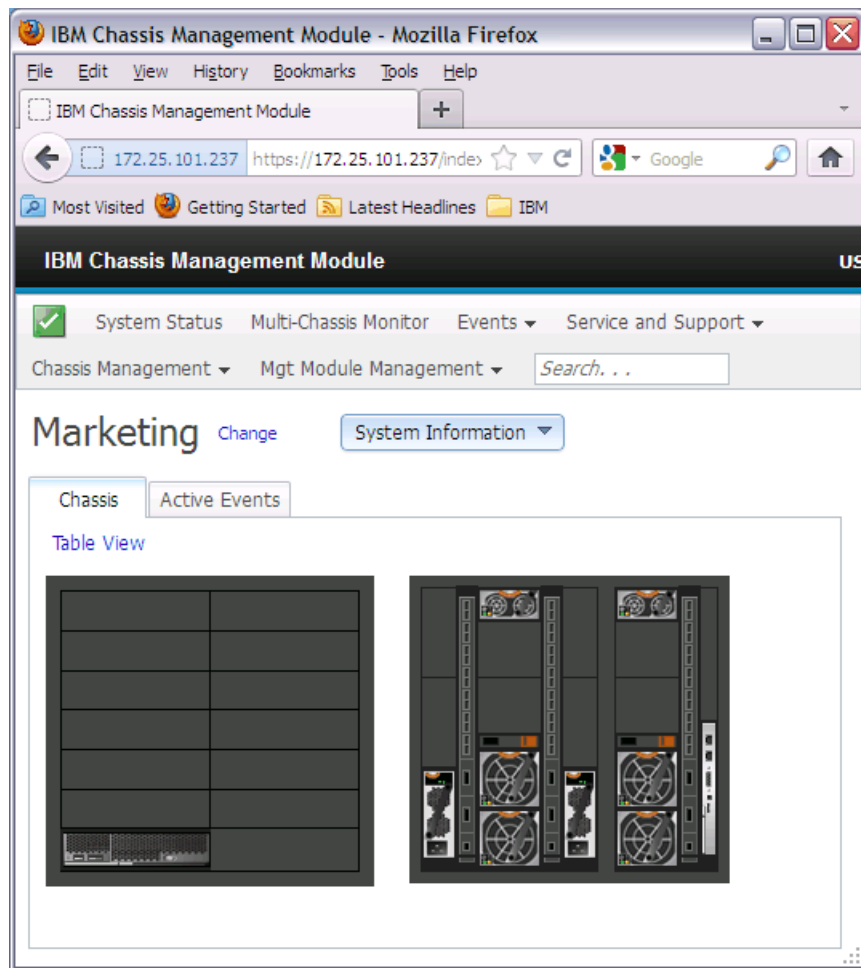


Figure 8-9 CMM GUI

3. Select **Chassis Management** → **I/O Modules** (as shown in Figure 8-10).

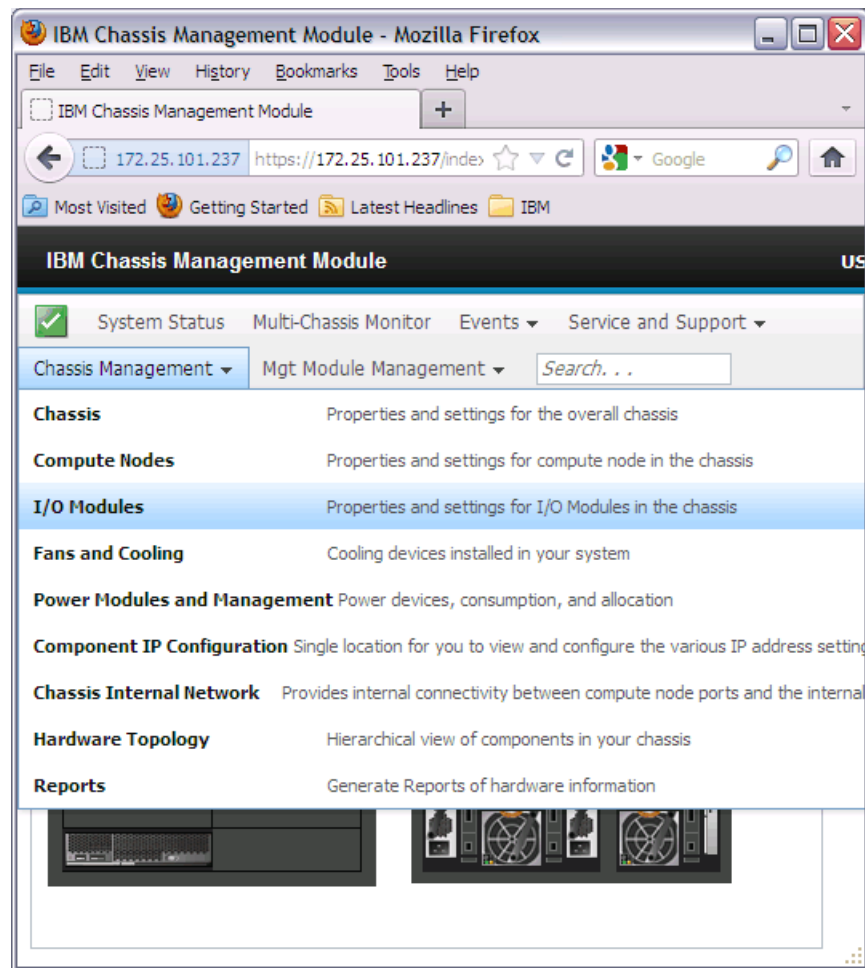


Figure 8-10 Select I/O modules management

- As shown in Figure 8-11, select the I/O module that needs to be reset to factory defaults, and click **Actions** → **Restore Factory Defaults**.

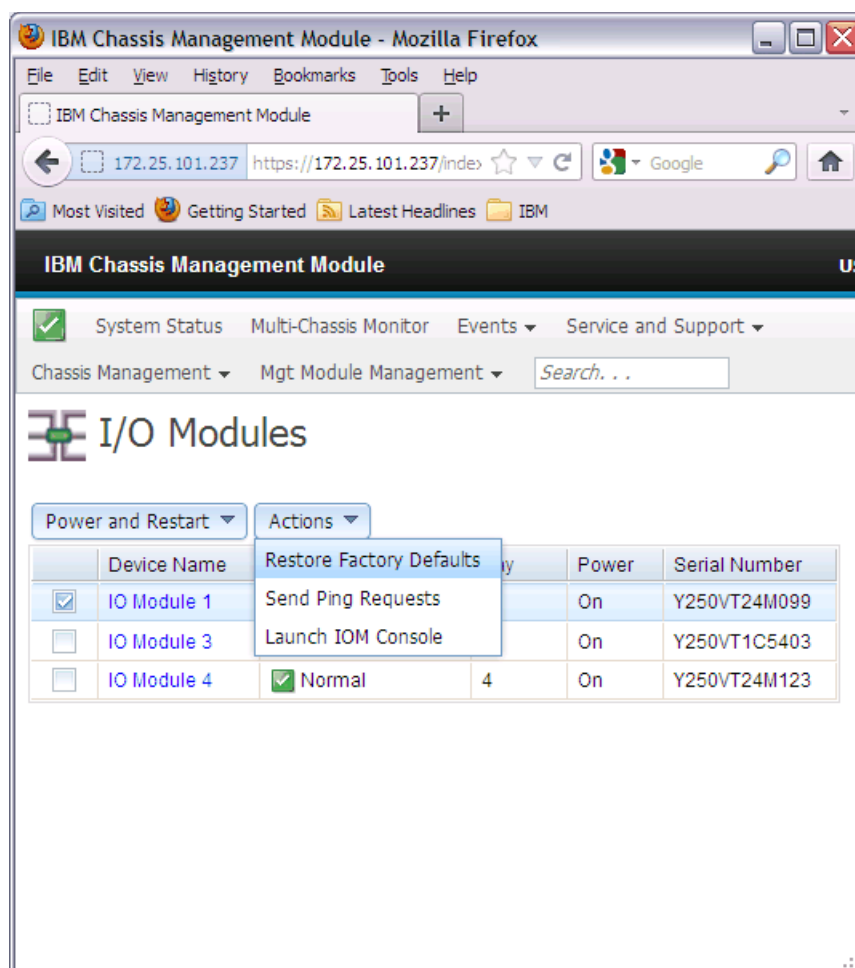


Figure 8-11 I/O Module 1 restore factory defaults

Resetting with terminal access to the switch

If you have terminal access to the switch, and you would like to reset the switch to factory defaults, use the **boot configuration-block factory** ISCLI command and then reload the switch (as shown in Example 8-7).

Example 8-7 Reset to factory defaults using ISCLI

```
compass-2(config)#boot configuration-block factory
Next boot will use factory default config block instead of active.
```

```
compass-2(config)#reload
```

Reset will use software "image2" and the factory default config block.

>> Note that this will RESTART the Spanning Tree,

>> which will likely cause an interruption in network service.

Confirm reload (y/n) ? y

The switch reloads with the factory default configuration.

Resetting with no terminal access to the switch

If you want to reset the switch to factory defaults and have no terminal access, you can use the serial console port. Complete the following steps:

1. Connect the management system to the serial port on the switch. Run a terminal emulation utility (such as Windows Hyperterminal or PuTTY) and use the following communication parameters to establish a session:
 - Speed: 9600 bps
 - Data Bits: 8
 - Stop Bits: 1
 - Parity: None
 - Flow Control: None
2. You must restart the switch by powering it off and back on, or by restarting it in the CMM web interface.
3. You can interrupt the boot process and enter the Boot Management Menu from the serial console port. When the system shows Memory Test, press **Shift+B**. The Boot Management Menu opens, as shown in example Example 8-8.

Example 8-8 Boot Management Menu

```
Boot Management Menu
  1 - Change booting image
  2 - Change configuration block
  3 - Boot in recovery mode (tftp and xmodem download of images to
recover switch)
  4 - Xmodem download (for boot image only - use recovery mode for
application images)
  5 - Reboot
  6 - Exit
Please choose your menu option:
```

4. Enter 2 to change the configuration block (see Example 8-9).

Example 8-9 Change configuration block

```
Please choose your menu option: 2

Unknown current config block 255
Enter configuration block: a, b or f (active, backup or factory):
```

5. As displayed in Example 8-10, enter f to use the factory defaults configuration block.

Example 8-10 Use factory defaults configuration block

```
Enter configuration block: a, b or f (active, backup or factory): f
```

6. You see the initial menu once again. Enter 6 to exit and reset the switch with the default configuration, as shown in Example 8-11.

Example 8-11 Exit from Boot Management Menu

```
Boot Management Menu
  1 - Change booting image
  2 - Change configuration block
  3 - Boot in recovery mode (tftp and xmodem download of images to
recover switch)
```

```
4 - Xmodem download (for boot image only - use recovery mode for
application images)
5 - Reboot
6 - Exit
Please choose your menu option: 6
```

The switch resets to the factory default configuration.

Important: If you set the configuration block to factory, do not forget to change it back to active configuration by running the following command:

```
Switch(config)#boot configuration-block active
```

8.2.5 Password recovery

To perform password recovery, you must set the switch to the factory default by using one of the procedures described in 8.2.4, “Resetting to factory defaults” on page 200.

After you reset the switch, run the following command:

```
Switch#copy active-config running-config
```

After the command finishes running, the switch is in enable mode without a password. Change the password by running **password** in the configuration mode:

```
Switch(config)#password
```

8.3 Firmware management

The switch firmware is the executable code that runs on the switch. The device comes preinstalled with certain firmware levels. As new firmware versions are released, we suggest upgrading the code that runs on your switch. You can find the latest version of firmware that is supported for your switch on IBM Fix Central, at the following website:

<http://www.ibm.com/support/fixcentral>

8.3.1 Firmware images

IBM switches can store up to two different IBM Networking OS images (called *image1* and *image2*) and a special boot image (called *boot*). When you load new firmware, make sure that you upgrade both the OS and boot image.

Run the **show boot** ISCLI command to see what images are installed. The output is shown in Example 8-12.

Example 8-12 Showing the current version of boot and OS images on the switch

```
compass-2#show boot
Currently set to boot software image1, active config block.
NetBoot: disabled, NetBoot tftp server: , NetBoot cfgfile:
Current CLI mode set to IBMNOS-CLI with selectable prompt enabled.
Current FLASH software:
  image1: version 7.2.2.2, downloaded 14:55:26 Mon Jun 18, 2012
  image2: version 7.3.1, downloaded 22:55:05 Mon Oct 1, 2012
```

```
boot kernel: version 7.3.1
Currently scheduled reboot time: none
```

In Example 8-12 on page 206, you can see that the system has two OS images:

- ▶ image1: Version 7.2.2.2
- ▶ image2: Version 7.3.1

The boot image version is 7.3.1. But the switch is set to boot from OS image1, which is at version 7.2.2.2. We want to ensure that the switch uses the same version for the boot image and the OS image. To boot from OS image2, run the **boot image image2** command, as shown in Example 8-13.

Example 8-13 Change to boot from image2

```
compass-2(config)#boot image image2
Next boot will use switch software image2 instead of image1.
```

Changing the boot image using serial interface

You can use the serial connection and Boot Management Menu to change the boot image. Follow these steps:

1. Connect the serial cable to the switch serial management port and the management system, then start the terminal emulation utility on the management system.
2. Use the following set of parameters to establish a terminal emulation session:
 - Speed: 9600 bps
 - Data Bits: 8
 - Stop Bits: 1
 - Parity: None
 - Flow Control: None
3. When the system shows Memory Test, press **Shift+B**. The Boot Management Menu is displayed (see Example 8-14).

Example 8-14 Boot Management Menu

```
Boot Management Menu
  1 - Change booting image
  2 - Change configuration block
  3 - Boot in recovery mode (tftp and xmodem download of images to
recover switch)
  4 - Xmodem download (for boot image only - use recovery mode for
application images)
  5 - Reboot
  6 - Exit
Please choose your menu option: 1

Current boot image is 1. Enter image to boot: 1 or 2: 2
Booting from image 2
```

4. As shown in Example 8-14, we select menu option 1 to change the boot image from image1 to image2.

8.3.2 Upgrading the firmware with ISCLI

In this section, we show how to upgrade firmware of the Flex System embedded switch EN4093/EN4093R. The latest firmware version at the time of writing is 7.3.1.0. This code level is available on Fix Central and at the following link:

<http://www.ibm.com/support/entry/portal/docdisplay?lnodocid=migr-5090394>

1. First, we must download the code update package (either from IBM Fix Central or from the preceding link) and unpack it. The update package contains two image files:

- Boot image file: *GbScSE-10G-7.3.1.0_Boot.img*
- OS image file: *GbScSE-10G-7.3.1.0_OS.img*

For convenience, we renamed these files as follows:

- Boot image file: *7310boot.img*
- OS image file: *7310os.img*

2. Next, we put the two files onto an FTP or SFTP server. In our example, we use the CMM built-in TFTP server. Figure 8-12 on page 209 shows the two files on the CMM TFTP server.

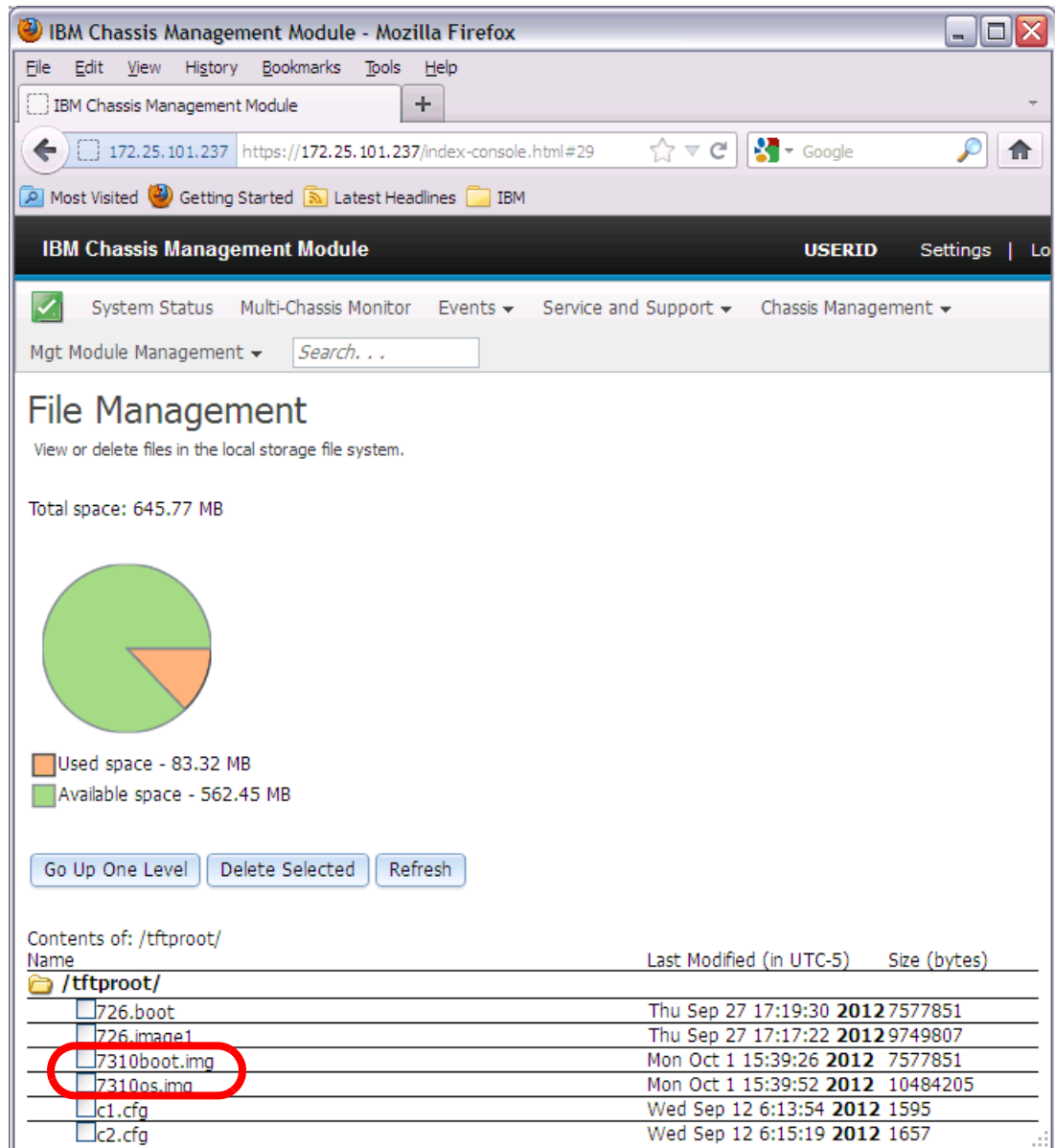


Figure 8-12 Firmware v7.3.1.0 image files on the CMM TFTP server

3. We are now ready to download the image files to EN4093/EN4093R. First, we must log in to EN4093/EN4093R as *administrator*, as shown in Example 8-15. When prompted to select the CLI mode, we choose **iscli**.

Example 8-15 Log in to EN4093/EN4093R

```
login as: admin
Using keyboard-interactive authentication.
Enter password:
```

IBM Flex System Fabric EN4093 10Gb Scalable Switch.

```
Select Command Line Interface mode (ibmnos-cli/iscli): iscli
System Information at 14:41:22 Mon Oct 1, 2012
Time zone: America/US/Pacific
```

Daylight Savings Time Status: Disabled

IBM Flex System Fabric EN4093 10Gb Scalable Switch

Switch has been up for 2 days, 23 hours, 22 minutes and 43 seconds.
Last boot: 15:20:45 Fri Sep 28, 2012 (reset from Telnet/SSH)

MAC address: 6c:ae:8b:bf:fe:00 IP (If 10) address: 10.10.10.239
Internal Management Port MAC Address: 6c:ae:8b:bf:fe:ef
Internal Management Port IP Address (if 128): 172.25.101.239
External Management Port MAC Address: 6c:ae:8b:bf:fe:fe
External Management Port IP Address (if 127):
Software Version 7.2.2.2 (FLASH image1), active configuration.

Hardware Part Number	: 49Y4272
Hardware Revision	: 02
Serial Number	: Y250VT24M123
Manufacturing Date (WWYY)	: 1712
PCBA Part Number	: BAC-00072-01
PCBA Revision	: 0
PCBA Number	: 00
Board Revision	: 02
PLD Firmware Version	: 1.5
Temperature Warning	: 29 C (Warn at 60 C/Recover at 55 C)
Temperature Shutdown	: 30 C (Shutdown at 65 C/Recover at 60 C)
Temperature Inlet	: 24 C
Temperature Exhaust	: 30 C
Power Consumption	: 43.530 W (12.184 V, 3.572 A)

Switch is in I/O Module Bay 4

4. Next, we enable the privileged EXEC mode (**enable** command) and download the boot image file. As shown in Example 8-16, we use the **copy tftp boot-image** command to download the boot image file.

Example 8-16 Enable privileged EXEC mode and download boot image

```
compass-2>enable
```

```
Enable privilege granted.  
compass-2#copy tftp boot-image  
Port type ["DATA"/"MGT"/"EXTM"]: MGT  
Address or name of remote host: 172.25.101.237  
Source file name: 7310boot.img
```

```
boot kernel currently contains Software Version 7.2.2.2  
New download will replace boot kernel with file "7310boot.img"  
from FTP/TFTP server 172.25.101.237.  
Connecting via MGT port.  
Confirm download operation (y/n) ? y  
Starting download...
```

```
File appears valid  
Download in progress
```


.....
.....
.....
.....
.....

Boot image (FS, 7577851 bytes) download complete.
Writing to flash...This can take up to 90 seconds. Please wait
FS Sector now contains Software Version 7.3.1
Boot image (Kernel, 7577851 bytes) download complete.
Writing to flash...This can take up to 90 seconds. Please wait
Kernel Sector now contains Software Version 7.3.1
Boot image (DFT, 7577851 bytes) download complete.
Writing to flash...This can take up to 90 seconds. Please wait
DFT Sector now contains Software Version 7.3.1
Boot image (Boot, 7577851 bytes) download complete.
Writing to flash...This can take up to 90 seconds. Please wait
Boot Sector now contains Software Version 7.3.1

5. As displayed in Example 8-17, we download the OS image file into image2 and set the switch to boot from image2. We use the **copy tftp image2** command.

Example 8-17 Download OS image file

```
compass-2#copy tftp image2
Port type ["DATA"/"MGT"/"EXTM"]: MGT
Address or name of remote host: 172.25.101.237
Source file name: 7310os.img
```

```
image2 currently contains Software Version 7.2.2.2
that was downloaded at 6:57:31 Mon Jun 18, 2012.
New download will replace image2 with file "7310os.img"
from FTP/TFTP server 172.25.101.237.
Connecting via MGT port.
Confirm download operation (y/n) ? y
Starting download...
```

```
File appears valid
Download in progress
```

.....
.....
.....
.....
.....
.....

```
Image download complete (10484205 bytes)
Writing to flash...This takes about 10 seconds. Please wait
Write complete (10484205 bytes), now verifying FLASH...
Verification of new image2 in FLASH successful.
image2 now contains Software Version 7.3.1
Switch is currently set to boot software image1.
Do you want to change that to the new image2? [y/n]
Oct 1 14:55:05 compass-2 INFO mgmt: image2 downloaded from host
```

```
172.25.101.237, file '7310os.img', software version 7.3.1
y
```

Next boot will use new software image2.

6. We must reboot the switch to activate the new code (see Example 8-18).

Example 8-18 Reboot the switch

```
compass-2#reload
```

```
Reset will use software "image2" and the active config block.  
>> Note that this will RESTART the Spanning Tree,  
>> which will likely cause an interruption in network service.  
Confirm reload (y/n) ? y
```

7. When the switch reloads, you can use the **show boot** command to verify that the new firmware, 7.3.1.0, is installed and running (shown in Example 8-19).

Example 8-19 New firmware verification

```
compass-2#show boot  
Currently set to boot software image2, active config block.  
NetBoot: disabled, NetBoot tftp server: , NetBoot cfgfile:  
Current CLI mode set to IBMNOS-CLI with selectable prompt enabled.  
Current FLASH software:  
  image1: version 7.2.2.2, downloaded 14:55:26 Mon Jun 18, 2012  
  image2: version 7.3.1, downloaded 22:55:05 Mon Oct 1, 2012  
  boot kernel: version 7.3.1  
Currently scheduled reboot time: none
```

This completes the EN4093/EN4093R firmware upgrade procedure.

8.3.3 Recovering from a failed firmware upgrade

Although extremely unlikely, the firmware upgrade process might fail. If this situation occurs, you can still recover the EN4093/EN4093R switch. Connect a PC running a terminal emulation utility to the serial port of your switch while the switch is off, and access the switch as described in the user's guide. Use the following communication parameters to establish a terminal emulation session:

- ▶ Speed: 9600 bps
- ▶ Data Bits: 8
- ▶ Stop Bits: 1
- ▶ Parity: None
- ▶ Flow Control: None

Important: The procedure described in this section might also be useful when you boot the switch and the boot and OS versions are not equal.

Then, power on the switch. From your terminal window, press **Shift + B** while the memory tests are processing and dots are showing the progress. A menu opens, as shown in Example 8-20.

Example 8-20 Boot Management Menu

```
Boot Management Menu  
  1 - Change booting image  
  2 - Change configuration block
```

- 3 - Boot in recovery mode (tftp and xmodem download of images to recover switch)
- 4 - Xmodem download (for boot image only - use recovery mode for application images)
- 5 - Reboot
- 6 - Exit

Please choose your menu option:

Select 4 for Xmodem download of boot image. Change the serial connection speed as follows:
Switch baudrate to 115200 bps and press ENTER ...

Change the settings of your terminal to meet the 115200 bps requirement and press Enter. The system switches to download accept mode. You see a series of C characters on the panel that prompt you when the switch is ready. Start an Xmodem terminal to push the boot code you want to restore into the switch. Select the boot code for your system, and the switch starts the download. You should see a panel similar to Example 8-21.

Example 8-21 Xmodem boot image download

xyzModem - CRC mode, 62106(SOH)/0(STX)/0(CAN) packets, 3 retries

```
Extracting images ... Do *NOT* power cycle the switch.
**** RAMDISK ****
Un-Protected 33 sectors
Erasing Flash...
..... done
Erased 33 sectors
Writing to Flash...9....8....7....6....5....4....3....2....1....0done
Protected 33 sectors
**** KERNEL ****
Un-Protected 25 sectors
Erasing Flash...
..... done
Erased 25 sectors
Writing to Flash...9....8....7....6....5....4....3....2....1....done
Protected 25 sectors
**** DEVICE TREE ****
Un-Protected 1 sectors
Erasing Flash...
. done
Erased 1 sectors
Writing to Flash...9....8....7....6....5....4....3....2....1....done
Protected 1 sectors
**** BOOT CODE ****
Un-Protected 4 sectors
Erasing Flash...
.... done
Erased 4 sectors
Writing to Flash...9....8....7....6....5....4....3....2....1....done
Protected 4 sectors
```

When this process is finished, you are prompted to reconfigure your terminal to the speed of 9600 bps:

Change the baud rate back to 9600 bps, hit the <ESC> key

Change the speed of your serial connection, and then press Esc. The Boot Management Menu opens again. Select option 3 now, and change the speed to 115000 bps when the following message is displayed, to start pushing the OS image:

```
## Switch baudrate to 115200 bps and press ENTER ...
```

When the speed is changed to 115200 bps, press Enter to continue the download. Select the OS image that you want to upload to the switch. The Xmodem client starts sending the image to the switch. When the upload is complete, you see a panel similar to the one in Example 8-22.

Example 8-22 OS image upgrade

```
xyzModem - CRC mode, 27186(SOH)/0(STX)/0(CAN) packets, 6 retries
Extracting images ... Do *NOT* power cycle the switch.
**** Switch OS ****
Please choose the Switch OS Image to upgrade [1|2|n] :
```

You are prompted to select the image space in the switch that you want to upgrade. After selecting the OS image bank, you see a panel similar to the one in Example 8-23.

Example 8-23 Upgrading the OS image

```
Switch OS Image 1 ...
Un-Protected 27 sectors
Erasing Flash..... done
Writing to Flash.....done
Protected 27 sectors
```

When this process is done, you are prompted to reconfigure your terminal to the speed of 9600 bps again:

Change the baud rate back to 9600 bps, hit the <ESC> key

Press Esc to show the Boot Management Menu, and choose option 6 to exit and boot the new image.

8.4 Logging and reporting

This section focuses on the following topics:

- ▶ Managing and configuring system logs
- ▶ Configuring an SNMP agent and SNMP traps
- ▶ Remote monitoring
- ▶ sFlow

8.4.1 System logs

IBM Networking OS can provide valuable maintenance and troubleshooting information through a system log (syslog) that uses the following fields in log entries: Date, time, switch name, criticality level, and message.

You can view the latest system logs by running **show logging messages** (Example 8-24).

Example 8-24 Example of syslog output

```
Oct 17 22:30:47 en4093flex_1 NOTICE mgmt: admin(admin) login from host
10.10.53.121
Oct 17 22:30:53 en4093flex_1 INFO mgmt: new configuration saved from ISCLI
Oct 17 22:32:27 en4093flex_1 INFO telnet/ssh-1: Current config successfully
tftp'd to 10.10.53.121:en4093flex_1-OSPF
Oct 17 22:32:29 en4093flex_1 NOTICE mgmt: admin(admin) connection closed from
Telnet/SSH
Oct 17 22:35:16 en4093flex_1 NOTICE ntp: System clock updated
Oct 17 22:49:06 en4093flex_1 NOTICE mgmt: USERID(Admin) login from BBI.
Oct 17 22:50:16 en4093flex_1 NOTICE ntp: System clock updated
Oct 17 23:25:08 en4093flex_1 NOTICE mgmt: USERID(Admin) logout from BBI.
Oct 17 23:35:23 en4093flex_1 NOTICE ntp: System clock updated
Oct 17 23:45:18 en4093flex_1 NOTICE mgmt: admin(admin) login from host
10.10.53.121
Oct 17 23:45:45 en4093flex_1 ALERT vlag: vLAG on portchannel 1 is up
Oct 17 23:45:46 en4093flex_1 ALERT vlag: vLAG on portchannel 15 is up
Oct 17 23:46:26 en4093flex_1 INFO cfgchg: Configured from SSHv2 by admin on
host 10.10.53.121
```

Each syslog message has a criticality level that is associated with it, included in text form as a prefix to the log message. Depending on the condition that the administrator is being notified of, one of eight different prefixes is used:

- ▶ Level 0 - EMERG: Indicates that the system is unusable.
- ▶ Level 1 - ALERT: Indicates that action should be taken immediately.
- ▶ Level 2 - CRIT: Indicates critical conditions.
- ▶ Level 3 - ERR: Indicates error conditions or operations in error.
- ▶ Level 4 - WARNING: Indicates warning conditions.
- ▶ Level 5 - NOTICE: Indicates a normal but significant condition.
- ▶ Level 6 - INFO: Indicates an information message.
- ▶ Level 7 - DEBUG: Indicates a debug-level message.

Information logged

You can selectively choose what information should be logged by the syslog. You have a number of options:

all	All
bgp	Border Gateway Protocol (BGP)
cfg	Configuration
cli	Command-line interface
console	Console
dcbx	DCB Capability Exchange
difftrak	Configuration difference tracking
failover	Failover
fcoe	Fibre Channel over Ethernet
hotlinks	Hot Links
ip	Internet Protocol
ipv6	IPv6
lACP	Link Aggregation Control Protocol
link	System port link
lldp	LLDP
management	Management

mld	Media library device (MLD)
netconf	NETCONF Configuration Protocol
ntp	Network time protocol
ospf	Open Shortest Path First (OSPF)
ospfv3	OSPFv3
rmon	Remote monitoring
server	Syslog server
spanning-tree-group	Spanning Tree Group
ssh	Secure Shell
system	System
vlag	Virtual Link Aggregation Group
vlan	VLAN
vm	Virtual machine
vnic	VNIC
vrrp	Virtual Router Redundancy Protocol
web	Web

Use the following ISCLI command syntax:

```
[no] logging log [<feature>]
```

For example, the following command enables syslog messages generation for SSH:

```
logging log ssh
```

The following command disables syslog messages generation for LACP:

```
no logging log lacp
```

The following command displays a list of features for which syslog messages are generated:

```
show logging
```

Logging destinations

You can set up to two destinations for reporting. A destination of 0.0.0.0 means logs are stored locally on the switch. Another instance of a log destination host can be a remote logging server. In this case, the logs are sent to the server through the syslog. For each of the two destinations, you can define many parameters, including the severity of logs to be sent to that particular destination.

In Example 8-25, we set a configuration to log locally the messages with ALERT (Level 1) severity and to send all critical (severity CRIT and Level 2) events to 172.25.101.200.

Example 8-25 Example of syslog configuration

```
en4093flex_1(config)#logging host 1 address 0.0.0.0
en4093flex_1(config)#logging host 1 severity 1
en4093flex_1(config)#logging host 2 address 172.25.101.200
```

```
Oct 18 0:54:32 en4093flex_1 NOTICE mgmt: second syslog host changed to
172.25.101.200 via MGT port
en4093flex_1(config)#logging host 2 severity 2
```

You can also use the **logging host** command to specify the interface that is used for logging. There are three options:

- ▶ data-port
- ▶ extm-port
- ▶ mgt-port

For example, to send the logs to a second destination from a data port, run the command shown in Example 8-26.

Example 8-26 Changing the logging interface

```
en4093flex_1(config)#logging host 2 data-port
```

```
Oct 18 0:57:13 en4093flex_1 NOTICE mgmt: second syslog host changed to 0.0.0.0 via Data port
```

Logging console

To make logging output visible on the console, run **logging console**. You can select the severity level of messages to be logged with the following syntax:

```
logging console severity <0-7>
```

8.4.2 Simple Network Management Protocol

IBM Networking OS provides Simple Network Management Protocol (SNMP) version 1, version 2, and version 3 support for access through any network management software, such as IBM Systems Director. Default SNMP version support is for SNMPv3 only.

Important: SNMP read and write functions are enabled by default. If SNMP is not needed for your network, it is best practice that you disable these functions before connecting the switch to the network.

SNMP versions 1 and 2

To access the SNMP agent on the EN4093/EN4093R, the read and write community strings on the SNMP manager should be configured to match the community strings on the switch. The default read community string on the switch is public and the default write community string is private.

The read and write community strings on the switch can be changed by running the following commands:

```
en4093flex_1(config)# snmp-server read-community <1-32 characters>
en4093flex_1(config)# snmp-server write-community <1-32 characters>
```

The SNMP manager should be able to reach the management interface or any of the IP interfaces on the switch.

For the SNMP manager to receive the SNMPv1 traps sent out by the SNMP agent on the switch, configure the trap host on the switch by running the following command:

```
en4093flex_1(config)# snmp-server trap-src-if <trap source IP interface>
en4093flex_1(config)# snmp-server host <IPv4 address> <trap host community string>
```

SNMP version 3

SNMP version 3 (SNMPv3) is an enhanced version of the Simple Network Management Protocol, approved by the Internet Engineering Steering Group in March 2002. SNMPv3 contains more security and authentication features that provide data origin authentication, data integrity checks, timeliness indicators, and encryption to protect against threats, such as masquerade, modification of information, message stream modification, and disclosure.

Using SNMPv3, your clients can query the MIBs securely.

Default configuration

IBM Networking OS has two SNMPv3 users by default. Both of the following users have access to all the MIBs supported by the switch:

- ▶ User 1 name is adminmd5 (password adminmd5). The authentication used is MD5.
- ▶ User 2 name is adminsha (password adminsha). The authentication used is Secure Hash Algorithm (SHA).

Up to 16 SNMP users can be configured on the switch. To modify an SNMP user, run the following command:

```
en4093flex_1(config)# snmp-server user <1-16> name <1-32 characters>
```

Users can be configured to use the authentication and privacy options. The EN4093/EN4093R switch supports two authentication algorithms, MD5 and SHA, as specified in the following command:

```
en4093flex_1(config)# snmp-server user <1-16> authentication-protocol  
{md5|sha} authentication-password
```

User configuration example

To configure a user, complete the following steps:

1. To configure a user with the name admin, the authentication type MD5, the authentication password of admin, and the privacy option Data Encryption Standard (DES) with a privacy password of admin, run the commands shown in Example 8-27.

Example 8-27 SNMP v3 user configuration example

```
en4093flex_1(config)# snmp-server user 5 name admin  
en4093flex_1(config)# snmp-server user 5 authentication-protocol md5  
authentication-password  
Changing authentication password; validation required:  
Enter current admin password: <admin. password>  
Enter new authentication password: <auth. password>  
Re-enter new authentication password: <auth. password>  
New authentication password accepted.  
en4093flex_1(config)# snmp-server user 5 privacy-protocol des  
privacy-password  
Changing privacy password; validation required:  
Enter current admin password: <admin. password>  
Enter new privacy password: <privacy password>  
Re-enter new privacy password: <privacy password>  
New privacy password accepted.
```

2. Configure a user access group, along with the views the group might access, by running the commands shown in Example 8-28. Use the access table to configure the group's access level.

Example 8-28 SNMPv3 group and view configuration example

```
en4093flex_1(config)# snmp-server access 5 name admingrp  
en4093flex_1(config)# snmp-server access 5 level authpriv  
en4093flex_1(config)# snmp-server access 5 read-view iso  
en4093flex_1(config)# snmp-server access 5 write-view iso  
en4093flex_1(config)# snmp-server access 5 notify-view iso
```

Because the read view, write view, and notify view are all set to iso, the user type has access to all private and public MIBs.

3. Assign the user to the user group by running the commands shown in Example 8-29. Use the group table to link the user to a particular access group.

Example 8-29 SNMPv3 user assignment configuration

```
en4093flex_1(config)# snmp-server group 5 user-name admin
en4093flex_1(config)# snmp-server group 5 group-name admingrp
```

Configuring Simple Network Management Protocol traps

Here, we describe the steps for configuring the Simple Network Management Protocol (SNMP) traps.

SNMPv2 trap configuration

To configure the SNMPv2 trap, complete the following steps:

1. Configure a user with no authentication and password, as shown in Example 8-30.

Example 8-30 SNMP user configuration example

```
en4093flex_1(config)#snmp-server user 10 name v2trap
```

2. Configure an access group and group table entries for the user. Use the menu shown in Example 8-31 to specify which traps can be received by the user.

Example 8-31 SNMP group configuration

```
en4093flex_1(config)#snmp-server group 10 security snmpv2
en4093flex_1(config)#snmp-server group 10 user-name v2trap
en4093flex_1(config)#snmp-server group 10 group-name v2trap
en4093flex_1(config)#snmp-server access 10 name v2trap
en4093flex_1(config)#snmp-server access 10 security snmpv2
en4093flex_1(config)#snmp-server access 10 notify-view iso
```

3. Configure an entry in the notify table, as shown in Example 8-32.

Example 8-32 SNMP notify entry configuration

```
en4093flex_1(config)#snmp-server notify 10 name v2trap
en4093flex_1(config)#snmp-server notify 10 tag v2trap
```

4. Specify the IPv4 address and other trap parameters in the targetAddr and targetParam tables. Use the commands shown in Example 8-33 to specify the user name associated with the targetParam table.

Example 8-33 SNMP trap destination and trap parameters configuration

```
en4093flex_1(config)#snmp-server target-address 10 name v2trap address
100.10.2.1
en4093flex_1(config)#snmp-server target-address 10 taglist v2trap
en4093flex_1(config)#snmp-server target-address 10 parameters-name v2param
en4093flex_1(config)#snmp-server target-parameters 10 name v2param
en4093flex_1(config)#snmp-server target-parameters 10 message snmpv2c
en4093flex_1(config)#snmp-server target-parameters 10 user-name v2trap
en4093flex_1(config)#snmp-server target-parameters 10 security snmpv2
```

5. Use the community table to specify which community string is used in the trap, as shown in Example 8-34.

Example 8-34 SNMP community configuration

```
en4093flex_1(config)#snmp-server community 10 index v2trap
en4093flex_1(config)#snmp-server community 10 user-name v2trap
```

SNMPv3 trap configuration

To configure a user for SNMPv3 traps, you can choose to send the traps with both privacy and authentication, with authentication only, or without privacy or authentication.

You can configure these settings in the access table by running the following commands:

- ▶ en4093flex_1(config)#snmp-server access <1-32> level
- ▶ en4093flex_1(config)#snmp-server target-parameters <1-16>

Configure the user in the user table.

It is not necessary to configure the community table for SNMPv3 traps because the community string is not used by SNMPv3.

Example 8-35 shows how to configure an SNMPv3 user v3trap with authentication only.

Example 8-35 SNMPv3 trap configuration

```
en4093flex_1(config)#snmp-server user 11 name v3trap
en4093flex_1(config)#snmp-server user 11 authentication-protocol md5
authentication-password
Changing authentication password; validation required:
Enter current admin password: <admin. password>
Enter new authentication password: <auth. password>
Re-enter new authentication password: <auth. password>
New authentication password accepted.
en4093flex_1(config)#snmp-server access 11 notify-view iso
en4093flex_1(config)#snmp-server access 11 level authnopriv
en4093flex_1(config)#snmp-server group 11 user-name v3trap
en4093flex_1(config)#snmp-server group 11 tag v3trap
en4093flex_1(config)#snmp-server notify 11 name v3trap
en4093flex_1(config)#snmp-server notify 11 tag v3trap
en4093flex_1(config)#snmp-server target-address 11 name v3trap address
172.25.101.200
en4093flex_1(config)#snmp-server target-address 11 taglist v3trap
en4093flex_1(config)#snmp-server target-address 11 parameters-name v3param
en4093flex_1(config)#snmp-server target-parameters 11 name v3param
en4093flex_1(config)#snmp-server target-parameters 11 user-name v3trap
en4093flex_1(config)#snmp-server target-parameters 11 level authNoPriv
```

8.4.3 Remote Monitoring

The IBM switches provide a Remote Monitoring (RMON) interface that allows network devices to exchange network monitoring data. RMON allows the switch to perform the following functions:

- ▶ Track events and trigger alarms when a threshold is reached.
- ▶ Notify administrators by issuing a syslog message or SNMP trap.

The RMON MIB provides an interface between the RMON agent on the switch and an RMON management application. The RMON MIB is described in Request for Comments (RFC) 1757:

<http://www.ietf.org/rfc/rfc1757.txt>

The RMON standard defines objects that are suitable for the management of Ethernet networks. The RMON agent continuously collects statistics and proactively monitors switch performance. You can use RMON to monitor traffic that flows through the switch.

The switch supports the following RMON groups, as described in RFC 1757:

- ▶ Group 1: Statistics
- ▶ Group 2: History
- ▶ Group 3: Alarms
- ▶ Group 9: Events

RMON Group 1: Statistics

The switch supports collection of Ethernet statistics as outlined in the RMON statistics MIB, referring to etherStatsTable. You can configure RMON statistics on a per-port basis. RMON statistics are sampled every second, and new data overwrites any old data on a port.

Important: RMON port statistics must be enabled for the port before you can view them.

Example configuration

Here is an example configuration:

1. Enable RMON on a port. To enable RMON on a port, run **interface** and **rmon**:
 - en4093flex_1(config)# interface port 1
 - en4093flex_1(config-if)# rmon
2. To view the RMON statistics, run **interface**, run **rmon**, and run **show** to show the interface, as shown in Example 8-36.

Example 8-36 View of the RMON statistics

```
en4093flex_1(config)# interface port INTA1
en4093flex_1(config-if)# rmon
en4093flex_1(config-if)# show interface port INTA1 rmon-counters
```

```
-----
RMON statistics for port INTA1:
etherStatsDropEvents: NA
etherStatsOctets: 7305626
etherStatsPkts: 48686
etherStatsBroadcastPkts: 4380
etherStatsMulticastPkts: 6612
etherStatsCRCAlignErrors: 0
etherStatsUndersizePkts: 0
etherStatsOversizePkts: 0
etherStatsFragments: 2
etherStatsJabbers: 0
etherStatsCollisions: 0
etherStatsPkts64Octets: 27445
etherStatsPkts65to127Octets: 12253
etherStatsPkts128to255Octets: 1046
etherStatsPkts256to511Octets: 619
etherStatsPkts512to1023Octets: 7283
```

RMON Group 2: History

You can use the RMON History Group to sample and archive Ethernet statistics for a specific interface during a specific time interval. History sampling is done per port.

Important: RMON port statistics must be enabled for the port before an RMON History Group can monitor the port.

Data is stored in buckets, which store data gathered during discreet sampling intervals. At each configured interval, the History index takes a sample of the current Ethernet statistics, and places them into a bucket. History data buckets are in dynamic memory. When the switch is rebooted, the buckets are emptied.

Requested buckets are the number of buckets, or data slots, requested by the user for each History Group. Granted buckets are the number of buckets granted by the system, based on the amount of system memory available. The system grants a maximum of 50 buckets.

You can use an SNMP browser to view History samples.

History MIB Object ID

The type of data that can be sampled must be of an ifIndex object type, as described in RFC 1213 and RFC 1573:

- ▶ <http://www.ietf.org/rfc/rfc1213.txt>
- ▶ <http://www.ietf.org/rfc/rfc1573.txt>

The most common data type for the History sample is as follows:

1.3.6.1.2.1.2.2.1.1.<x>

The last digit (x) represents the number of the port to monitor.

8.4.4 Using sFlow to monitor traffic

IBM System Networking switches support sFlow technology for monitoring traffic in data networks. The switch includes an embedded sFlow agent that can be configured to provide continuous monitoring information of IPv4 traffic to a central sFlow analyzer.

The switch is responsible only for forwarding sFlow information. A separate sFlow analyzer is required elsewhere in the network to interpret sFlow data.

Use the following commands to enable and configure sFlow:

- ▶ Enable sFlow on the switch:
`sflow enable`
- ▶ Set sFlow analyzer IP address:
`sflow server <IP address>`
- ▶ Optionally, set UDP port for sFlow analyzer (default is 6343):
`sflow port <1-65535>`
- ▶ Display sFlow configuration settings:
`show sflow`

sFlow statistical counters

IBM System Networking switch can be configured to send network statistics to an sFlow analyzer at regular intervals. For each port, a polling interval of 5 - 60 seconds can be configured, or 0 (the default) can be set to disable this feature.

Use the following command to set the sFlow port polling interval:

```
sflow polling <5-60>
```

When polling is enabled, at the end of each configured polling interval, the switch reports general port statistics and port Ethernet statistics.

sFlow network sampling

In addition to statistical counters, IBM System Networking switches can be configured to collect periodic samples of the traffic data received on each port. For each sample, 128 bytes are copied, UDP-encapsulated, and sent to the configured sFlow analyzer.

For each port, the sFlow sampling rate can be configured to occur every 256 - 65536 packets, or set to 0 to disable (the default) this feature. A sampling rate of 256 means that one sample is taken for approximately every 256 packets received on the port. The sampling rate is statistical, however. It is possible to have more or fewer samples sent to the analyzer for any specific group of packets (especially under low traffic conditions). The actual sample rate becomes the most accurate over time, and under higher traffic flow.

Use the following command to set the sFlow port sampling rate:

```
sflow sampling <256-65536>
```

sFlow sampling has the following restrictions:

- ▶ Sample rate: The fastest sFlow sample rate is 1 out of every 256 packets.
- ▶ Access control lists (ACLs): sFlow sampling is performed before ACLs are processed. For ports configured both with sFlow sampling and one or more ACLs, sampling occurs regardless of the action of the ACL.
- ▶ Port mirroring: sFlow sampling does not occur on mirrored traffic. If sFlow sampling is enabled on a port that is configured as a port monitor, the mirrored traffic is not sampled.

sFlow sampling: Although sFlow sampling is not generally a processor-intensive operation, configuring fast sampling rates (such as once every 256 packets) on ports under heavy traffic loads can cause switch processor utilization to reach maximum. Use larger rate values for ports that experience heavy traffic.



An integration guide to the IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch

This appendix presents a straightforward way to configure and connect an IBM Flex System Fabric EN4093/EN4093R 10Gb Scalable Switch with a Cisco Nexus 5000 series upstream switch (including its configuration).

Following are the topics covered on this implementation case:

- ▶ Overview of the Flex factory network configuration
- ▶ Description of the EN4093/EN4093R configuration
- ▶ Consolidating the VLANs across the Link Aggregation Group
- ▶ Consolidating the VLANs across a single link
- ▶ Cisco Nexus switch configuration
- ▶ Changing the VLAN IDs
- ▶ Showing running configuration
- ▶ A script to add a second EN4093/EN4093R switch to the setup

Overview of the factory network configuration

This section describes an IBM PureFlex System implementation. Components in this implementation come preconfigured for certain virtual local area networks (VLANs) and connections that might not be obvious or intuitive at first glance. We provide insight into these configurations, and hints and tips, to aid an implementor with the installation process. It is possible for the implementor to extensively adapt the pre-configuration to be used in different designs, but that is beyond the scope of this document.

The IBM PureFlex System comes with three networks, each of which is isolated from the others by VLAN assignments within the Ethernet switches. Your view to each network is through one or more uplink ports, which are untagged. Figure A-1 shows the network design for this implementation.

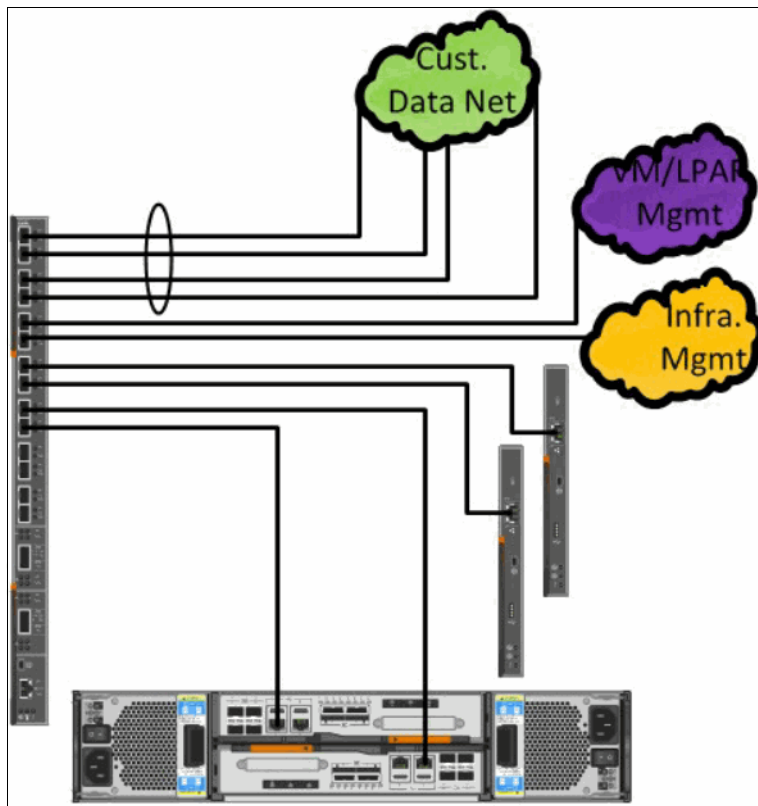


Figure A-1 PureFlex implementation network design

Figure A-2 shows the logical network diagram including the network colors that we discuss.

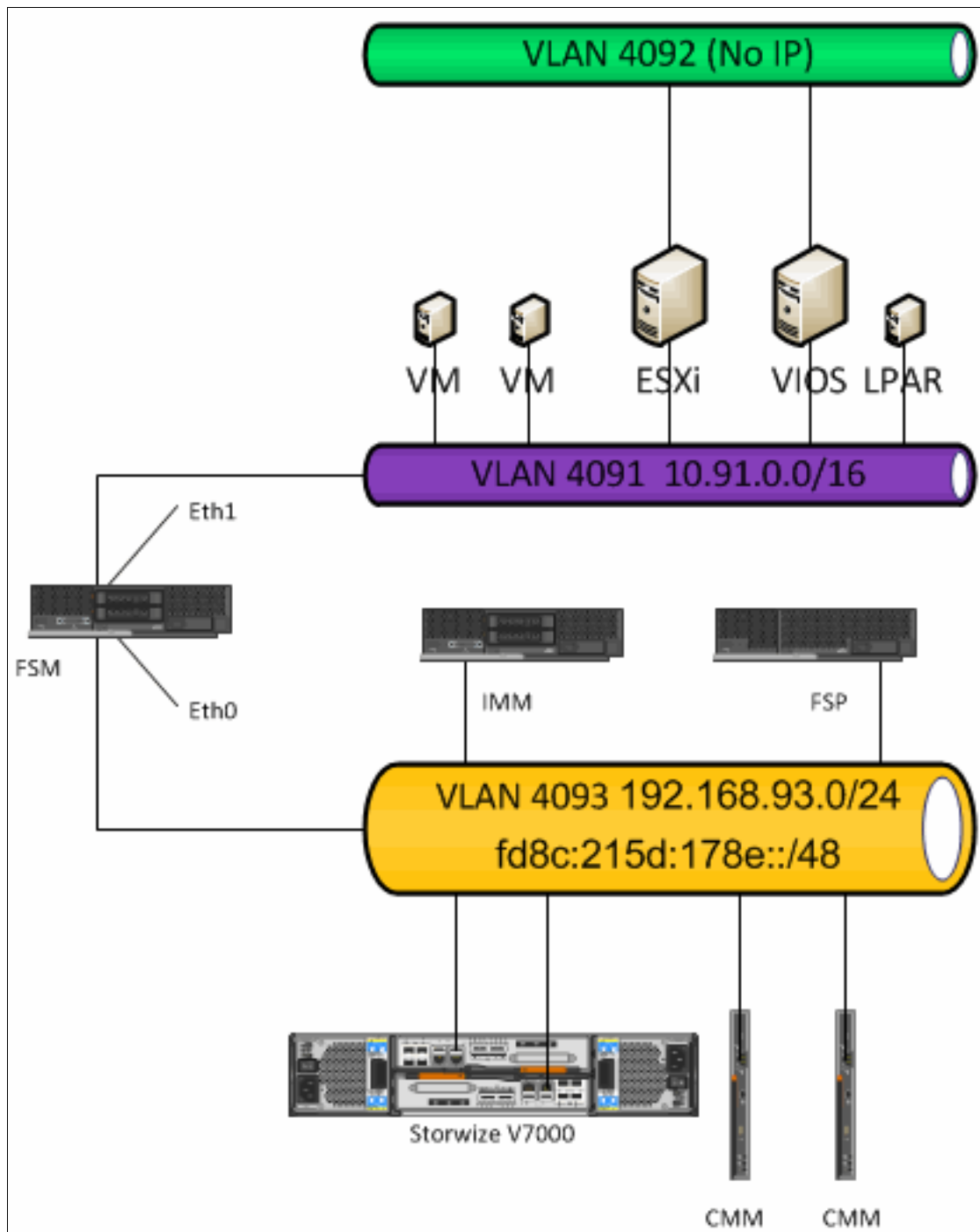


Figure A-2 The network logical diagram with network colors

No IPv4 gateway address is assigned to the devices in the Flex chassis and there is no DNS server in the environment. DNS is required for the Flex System Manager. IP gateway and DNS functions must be provided by the upstream network.

Purple network

The purple network is assigned to VLAN 4091. All entities on the purple network are assigned IPv4 addresses within the 10.91.0.0/16 network (that is, 10.91.*.* with netmask 255.255.0.0). This is the network where you access the IBM Flex System Manager node on its external Ethernet (eth1) interface. It also enables you to directly access the following:

- ▶ The Virtual I/O Server (VIOS) on Power Systems compute nodes
- ▶ The ESXi hypervisor on IBM X-Architecture® compute nodes

In addition, any guest operating systems that you want to manage using the IBM Flex System Manager must have an IP address on this network.

Green network

The green network is for your data and it is assigned to VLAN 4092. There are no IP addresses assigned to this network in IBM Manufacturing, so you can assign them to meet your needs.

Gold network

The gold internal device management network is assigned to VLAN 4093 on single chassis systems without any top-of-rack switches, and is entirely isolated on the top-of-rack switches, if present. This network contains the following components:

- ▶ The IBM Flex System Manager node's internal Ethernet (eth0) interface
- ▶ Chassis Management Modules (CMMs)
- ▶ The management interfaces for any Ethernet or storage area network (SAN) switches
- ▶ The flexible service processors of Power Systems compute nodes and integrated management modules (IMMs)
- ▶ IMMs of X-Architecture compute nodes
- ▶ The IBM Storwize V7000 management interfaces

Each of these endpoints is assigned a fixed IPv4 address on the 192.168.93.0/24 network (that is, 192.168.93.* addresses with netmask 255.255.255.0), and an IPv6 Unique Local Address (ULA) that is guaranteed not to conflict with any address on the client's network. The IBM Flex System Manager node manages all of these endpoints using the IPv6 addresses. Therefore, you can choose to reassign the IPv4 addresses as wanted to fit into your network.

EN4093/EN4093R VLAN configuration

There is a difference in how IBM switches and Cisco switches configure VLANs on a port. In a Cisco device, a multi-VLAN link is called a *trunk*, and by default, if the VLAN is configured in the switch, a trunk allows the VLAN to use this link.

In an IBM network device, a multi-VLAN link is called a *tagging* port, and the VLAN must be explicitly configured to allow the port to transport the VLAN.

The second difference is in terminology for an untagged VLAN. The concept is the same in both environments. This is the VLAN ID that does not have 802.1Q VLAN tags applied to the packets. In IBM terminology, the untagged VLAN is the *Port VLAN ID (PVID)*. In an IBM switch, PVID 100 indicates that untagged packets are treated as members of VLAN 100. In a

Cisco configuration, the single untagged VLAN is called the *Access VLAN*. If port 5 is set for *switchport access VLAN 100*, untagged packets on the port are treated as belonging to VLAN 100.

To define a port to transport multiple VLANs with VLAN tags and allow a single VLAN to not use tags, an IBM switch configuration would have the keyword, *tagging*, and an assigned PVID. In a Cisco configuration, the configuration would be *switchport trunk native VLAN x*.

Also, IBM allows and disallows ports in the VLAN configurations; Nexus configures it per port.

You can see more options and details for PVID usage in the *IBM RackSwitch G8264 Application Guide* that presents useful sections for terminology and tagging explanations. The guide can be found at the following website:

<http://www-01.ibm.com/support/docview.wss?uid=isg3T7000599>

Table A-1 presents the configuration commands and comments about the EN4093/EN4093R setup.

Table A-1 IBM EN4093/EN4093R configuration

Configuration	Description
interface port INTA1-INTA141 tagging pvid 4091 exit	These configuration commands set the first 14 internal interfaces to accept multiple VLANs (tagging), and define VLAN ID 4091 as the untagged VLAN. Cisco equivalent would be “mode trunk” and “native vlan 4091”.
interface port EXT1-EXT4 pvid 4092 exit	These configuration statements set the first four EXTERNAL ports to pass only VLAN 4092, and accept only untagged packets. Cisco equivalent “mode access” and “access vlan 4092”. (These four ports are aggregated together in later commands).
interface port EXT5 pvid 4091 exit	This assigns external port 5 as an untagged member of VLAN 4091.
interface port EXT6-10 pvid 4093 exit	External ports 6 - 10 are configured to accept external management interfaces of the CMM and redundant CMM, and the v7000 management interfaces: - Single VLAN port (untagged) - Member of VLAN 4093 - Gold network
interface port EXT1 lacp mode active lacp key 3000 interface port EXT2 lacp mode active lacp key 3000 interface port EXT3 lacp mode active lacp key 3000 interface port EXT4 lacp mode active lacp key 3000	Ports EXT1 through EXT4 are aggregated into a single logical link. Using “Active” mode of Link Aggregation Control Protocol (LACP) will actively try to create LAG with connected switch.

Configuration	Description
<pre> vlan 1 member INTB1-INTB14,EXT15-EXT22 no member INTA1-INTA14,EXT1-EXT10 vlan 4091 enable name "OS Mgmt" member INTA1-INTA14,EXT5 vlan 4092 enable name "Data" member INTA1-INTA14,EXT1-EXT4 vlan 4093 enable name "Device Mgmt" member EXT6-EXT10 </pre>	<p>Defines and enables each VLAN.</p> <p>Identifies which ports can carry traffic in the VLAN.</p> <p>Also defines which ports are explicitly removed from the VLAN.</p> <p>Names can be assigned to VLANs.</p>
<pre> spanning-tree stp 123 vlan 4091 spanning-tree stp 124 vlan 4092 spanning-tree stp 125 vlan 4093 </pre>	<p>Per VLAN Rapid Spanning Tree (PVRST). Each VLAN is running its own, separate instance of the Spanning-Tree Protocol. Same concept as Cisco devices. STP number defines the instance. This is created and assigned automatically when VLAN is enabled.</p>
<pre> ntp enable ntp ipv6 primary-server fe80::211:25ff:fec3:5ded MGT ntp interval 15 ntp authenticate ntp primary-key 32051 ntp message-digest-key 32051 md5-ekey (encrypted password) ntp trusted-key 32051 </pre>	<p>Enables NTP Using IPv6. Defines the primary NTP server.</p> <p>This is the CMM's link-local IPv6 address. Automatically pushed to switch from CMM.</p> <p>Interval 15 is how often it updates its clock (15 mins.).</p> <p>The next four lines contain the authentication information for secure NTP.</p>
<pre> system idle 60 no logging console </pre>	<p>Idle time defines the "auto-logout" time; this is set to 60 minutes.</p> <p>Stops the system from echoing syslog messages to the console.</p>

Consolidating VLANs across Link Aggregation Groups

One of the possibilities to consolidate VLANs is by using Link Aggregation Groups. Figure A-3 on page 231 presents a physical linking to allow that.

This setup shows how to configure the EN4093/EN4093R and the Nexus-OS switch to consolidate all traffic across the 4-port port-channel. All three VLANs: 4091, 4092, and 4093, are configured to use the <port-channel> uplink.

Important: Although not typical practice, the management interfaces from the CMMs and Storwize V7000 are still connected through the EN4093/EN4093R.

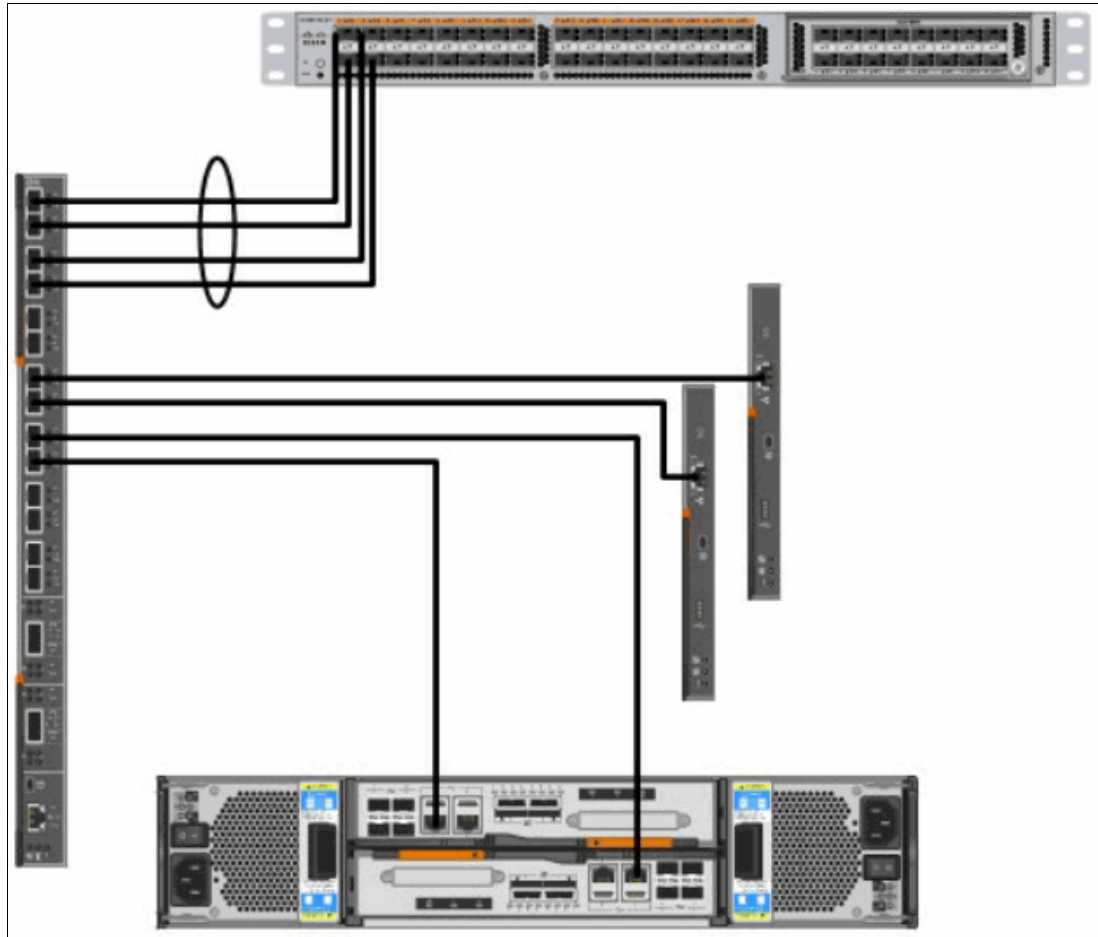


Figure A-3 Consolidating the VLAN across the Link Aggregation Group

In this section, we configure the EN4093/EN4093R to send all traffic from VLANs 4091, 4092, and 4093 across the 4-port Link-Aggregation.

Looking at the VLAN configuration from the proof of concept (POC) configuration, see the ports assigned to VLANs as shown in Example A-1.

Example: A-1 Show VLAN

Router#show vlan					
VLAN	Name	Status	MGT	Ports	
1	Default VLAN	ena	dis	INTB1-INTC14 EXT15-EXT22	
4091	OS Mgmt	ena	dis	INTA1-INTA14 EXT5	
4092	Data	ena	dis	INTA1-INTA14 EXT1-EXT4	
4093	Device Mgmt	ena	dis	EXT6-EXT10	
4095	Mgmt VLAN	ena	ena	EXTM MGT1	

We want to allow the three VLANs to use the link aggregation (ports EXT1 through EXT4) to connect to the site network. Therefore, we need to accomplish a couple of things:

- ▶ Enable the ports to transport multiple VLANs: enable tagging
- ▶ Add the uplinks to all three VLANs

It is important to enable tagging on the port before adding the port to more VLANs. If the port is still non-tagging, and you add the port to an additional VLAN, the switch changes the PVID for you. This is not what you want to happen in this case.

To avoid an accidental network loop, best practices dictate that there should be no links active while the following process is followed. Either disconnect any cables between the EN4093/EN4093R and the site network switch, or disable (shutdown) any ports that are undergoing configuration.

On the EN4093/EN4093R, the command would look like Example A-2.

Example: A-2 EN4093/EN4093R “conf t” command output

```
Router#conf t
Enter configuration commands, one per line. End with Ctrl/Z.
Router(config)#int port EXT1-EXT4
Router(config-if)#shut
```

On the Nexus switch, the command would look like Example A-3.

Example: A-3 Nexus “conf t” command output

```
nx5020# conf t
Enter configuration commands, one per line. End with CNTL/Z.
nx5020(config)# int port-channel 1
nx5020(config-if)# shut
nx5020(config-if)#
```

The first action now is to create the VLANs on the Nexus device. The result is shown in Example A-4.

Example: A-4 Show the created VLANs

nx5020#show vlan		Status	Ports
VLAN	Name		

4091	PoC-OS_Mgmt_Purple	active	
4092	PoC-data_Green	active	
4093	PoC-Mgt_Gold	active	

Second, configure the port-channel to be a trunk by using the command presented in Example A-5.

Example: A-5 Nexus “conf t” command output

```
nx5020(config-if)# switchport mode trunk
```

It is important now to determine if any VLANs will be used in an untagged fashion (native VLAN or PVID). Ensure that both the IBM switch and the Cisco switch are using the same VLAN number for this function.

For this example, VLAN 4091 – the OS-Mgmt VLAN will be configured as the untagged VLAN. Make appropriate changes to the configurations for the local environment.

This will be a change from the POC configuration.

Example A-6 presents the Nexus configuration.

Example: A-6 Nexus configuration

```

nx5020(config-if)# switchport trunk native vlan 4091
nx5020(config-if)# show run int port-channel 1
interface port-channel1
    switchport mode trunk
    switchport trunk native vlan 4091

```

Example A-7 presents the EN4093/EN4093R configuration.

Example: A-7 EN4093/EN4093R configuration

```

Router(config)#int port ext1-ext4
Router(config-if)#tagging

```

Now, we add the external ports to the other two VLANs, as shown in Example A-8.

Example: A-8 Adding external ports to VLANs

```

Router(config)#vlan 4091
Router(config-vlan)#member ext1-ext4
Router(config-vlan)#vlan 4093
Router(config-vlan)#member ext1-ext4
Router(config-vlan)#

```

Ports EXT1 - EXT4 are now members of VLAN 4092. After this change, the output should look like Example A-9.

Example: A-9 Showing configuration until this step

```

Router(config-vlan)#exit
Router(config)#sh vlan

```

VLAN	Name	Status	MGT	Ports
1	Default VLAN	ena	dis	INTA1-INTC14 EXT11-EXT22
4091	OS Mgmt	ena	dis	INTA1-INTA14 EXT1-EXT5
4092	Data	ena	dis	INTA1-INTA14 EXT1-EXT4
4093	Device Mgmt	ena	dis	EXT1-EXT4 EXT6-EXT10
4095	Mgmt VLAN	ena	ena	EXTM MGT1

Next, change the PVID on the four external ports to VLAN 4091 to match the native VLAN on the Nexus switch, like shown in Example A-10.

Example: A-10 Changing PVID to VLAN 4091

```

Router(config)#int port ext1-ext4
Router(config-if)#pvid 4091

```

The configuration on the EN4093/EN4093R ports EXT1 – EXT4 should now look like what is shown in Example A-11.

Example: A-11 Showing configuration until this step

```

interface port EXT1
    shutdown
    tagging
    pvid 4091
    exit
!
interface port EXT2
    shutdown
    tagging
    pvid 4091
    exit
!
interface port EXT3
    shutdown
    tagging
    pvid 4091
    exit
!
interface port EXT4
    shutdown
    tagging
    pvid 4091
    exit

```

Start up the Link-Aggregation, again one port at a time. We re-enable the four ports on the Nexus together with commands shown in Example A-12.

Example: A-12 Bringing Link-Aggregation

```

nx5020(config)# int port-channel 1
nx5020(config-if)# no shut

```

Then, start the ports on the EN4093/EN4093R one at a time, like in Example A-13.

Example: A-13 Start the EN4093/EN4093R ports

```

Router(config)#int port ext1
Router(config-if)#no shut

```

Check to ensure that one link starts with the command in Example A-14.

Example: A-14 Check lacp link

```

Router(config)#show lacp information state up
port    mode  adminkey  operkey  selected  prio  aggr  trunk  status  minlinks
-----
EXT1    active  3000     3000     yes       32768  43    65     up      1

nx5020(config-if)# show port-channel summary
Flags:  D - Down          P - Up in port-channel (members)
        I - Individual    H - Hot-standby (LACP only)
        s - Suspended     r - Module-removed
        S - Switched      R - Routed
        U - Up (port-channel)

```

Group	Port-Channel	Type	Protocol	Member Ports
1	Po1(SU)	Eth	LACP	Eth1/17(P) Eth1/18(D) Eth1/19(D) Eth1/20(D)

If everything looks good, start the remaining links with the commands in Example A-15.

Example: A-15 Start the remaining links

```
Router(config)#interface port ext2-ext4
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#show lacp information state up
port    mode  adminkey  operkey  selected  prio  aggr  trunk  status  minlinks
-----
--
EXT1    active  3000      3000     yes       32768  43    65     up      1
EXT2    active  3000      3000     yes       32768  43    65     up      1
EXT3    active  3000      3000     yes       32768  43    65     up      1
EXT4    active  3000      3000     yes       32768  43    65     up      1

nx5020(config-if)# show port-channel summary
Flags:  D - Down          P - Up in port-channel (members)
        I - Individual    H - Hot-standby (LACP only)
        s - Suspended     r - Module-removed
        S - Switched      R - Routed
        U - Up (port-channel)
-----
-
Group Port-Channel    Type    Protocol  Member Ports
-----
-
1      Po1(SU)           Eth     LACP      Eth1/17(P) Eth1/18(P) Eth1/19(P)
                                   Eth1/20(P)
```

Consolidating VLANs across a single link

Another possibility is to consolidate VLANs across a single link, as shown in Figure A-4 on page 236.

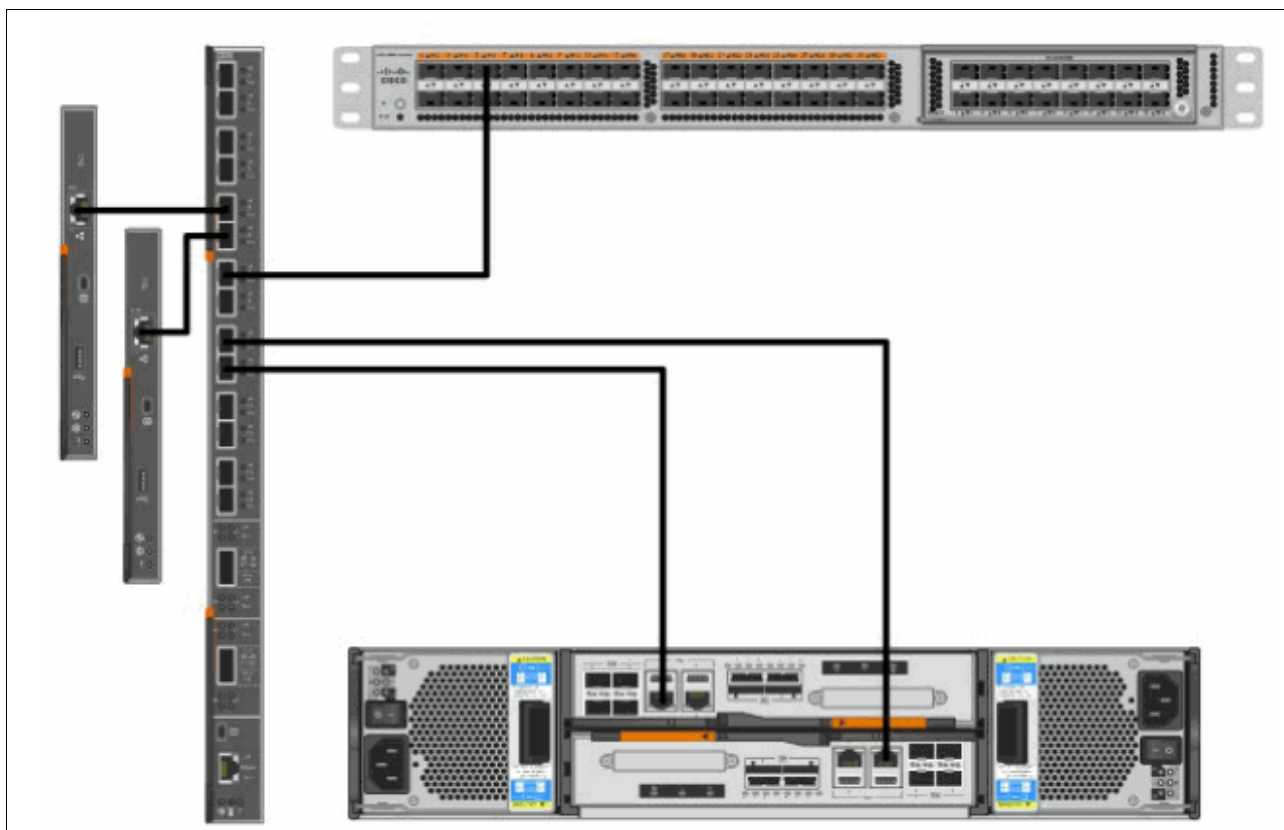


Figure A-4 Consolidating the VLAN across a single link

In this configuration, we set up the EN4093/EN4093R to use a single uplink connection to the site network. All three of the VLANs configured in the EN4093/EN4093R will be configured to cross the single link.

We use port EXT6 on the EN4093/EN4093R, and port E1/21 on a Nexus 5020.

There are two things that must be changed on the EN4093/EN4093R for this to work:

- ▶ Enable tagging on the external port of the EN4093/EN4093R – to allow multiple VLANs to use the link
- ▶ Add the external port to the VLAN configurations

Nexus configuration

The Nexus switch must also be configured to make this work. Start with the Nexus switch configuration, that you can check on Example A-16.

Example: A-16 Nexus configuration

```

nx5020# conf t
Enter configuration commands, one per line. End with CNTL/Z.
nx5020(config)# vlan 4091
nx5020(config-vlan)# name OS-Mgmt
nx5020(config-vlan)# vlan 4092
nx5020(config-vlan)# name Data
nx5020(config-vlan)# vlan 4093
  
```

```
nx5020(config-vlan)# name Device_Mgmt
```

Now show the created VLANs, as in Example A-17.

Example: A-17 Show the created VLANs

```
nx5020# show vlan brief
```

VLAN Name	Status	Ports
1002 VLAN1002	active	
4091 OS-Mgmt	active	
4092 Data	active	
4093 Device_Mgmt	active	

Set the uplink port to be trunk as in Example A-18.

Example: A-18 Set uplink ports as trunk

```
nx5020# conf t
Enter configuration commands, one per line. End with CNTL/Z.
nx5020(config)# int e1/21
nx5020(config-if)# switchport mode trunk
```

As the port is configured now, VLAN 1 is the native VLAN, this might, or might not be what is needed. If one of the VLANs from the EN4093/EN4093R is untagged (PVID), set the same VLAN number as the native VLAN on the Nexus switch, as shown in Example A-19.

Example: A-19 Set VLAN on Nexus

```
nx5020(config-if)# switchport access vlan 4091
```

EN4093/EN4093R configuration

First, set EXT6 to start tagging, as in Example A-20.

Example: A-20 Set tagging on EXT6 port

```
Router#conf t
Enter configuration commands, one per line. End with Ctrl/Z.
Router(config)#interface port ext6
Router(config-if)#tagging
```

Add port EXT6 to the VLANs, as in Example A-21.

Example: A-21 Add EXT6 port to the VLAN

```
Router(config)#vlan 4091
Router(config-vlan)#member ext6
Router(config-vlan)#vlan 4092
Router(config-vlan)#member ext6
Router(config-vlan)#vlan 4093
```

```
Router(config-vlan)#member ext6
```

Set the PVID (untagged, or native VLAN) of port EXT6 to match the configuration of the Nexus switch, as in Example A-22.

Example: A-22 Set PVID as on the Nexus configuration

```
Router(config-vlan)#exit
Router(config)#int port ext6
Router(config-if)#pvid 4091
```

The configuration of port EXT6 looks like Example A-23.

Example: A-23 EXT6 port configured

```
interface port EXT6
    tagging
    pvid 4091
exit
```

The VLAN configuration should look like Example A-24.

Example: A-24 VLAN configuration

```
Router(config)#show vlan
VLAN          Name                Status MGT          Ports
-----
1      Default VLAN      ena    dis    INTB1-INTC14 EXT15-EXT22
4091   OS Mgmt           ena    dis    INTA1-INTA14 EXT5  EXT6
4092   Data              ena    dis    INTA1-INTA14 EXT1-EXT4 EXT6
4093   Device Mgmt        ena    dis    EXT6-EXT10
4095   Mgmt VLAN          ena    ena    EXTM MGT1
```

Now check the link on both sides, as in Example A-25 and Example A-26.

Example: A-25 Nexus side link check

```
nx5020# show interface status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Eth1/21	--	connected	trunk	full	10G	10g

Example: A-26 EN4093/EN4093R side link check

```
Router(config)#show interface status
```

Alias	Port	Speed	Duplex	Flow Ctrl	Link	Name
-----	----	-----	-----	--TX-----RX--	-----	-----
EXT6	48	10000	full	no	no	up
						EXT6

Using EXT1 as the uplink port

To use port EXT1 as the uplink, first you must remove port EXT1 from the 4-port link aggregation, as shown in Example A-27.

Example: A-27 Remove port EXT1 from the 4-port link aggregation

```
interface port EXT1
    lacp mode active
    lacp key 3000

Router(config)#no lacp ?
<1-65535> admin key
Router(config)#no lacp 3000
```

The configuration lines indicating the Link Aggregation Control Protocol (LACP) key are still active in the configuration, but the LACP mode is gone. Therefore, LACP is no longer active on any of the four ports (EXT1 - EXT4). To cause this configuration statement to disappear from the configuration, set the values back to the default value (43).

From this point, you can continue with the instructions, but substitute EXT1 instead of EXT6.

Cisco Nexus switch configuration

This section shows how a Nexus-OS device can be configured to integrate with the Flex POC switch configuration.

In this example, the 4-port data-network port-channel is connected to the Nexus switch, and individual ports are used for the VM/LPAR management and Customer Infrastructure Management VLANs.

This example uses the POC VLAN IDs, but the configuration can be changed later.

Figure A-5 shows how a Nexus-OS device can be configured to integrate with the Flex implementation switch configuration.

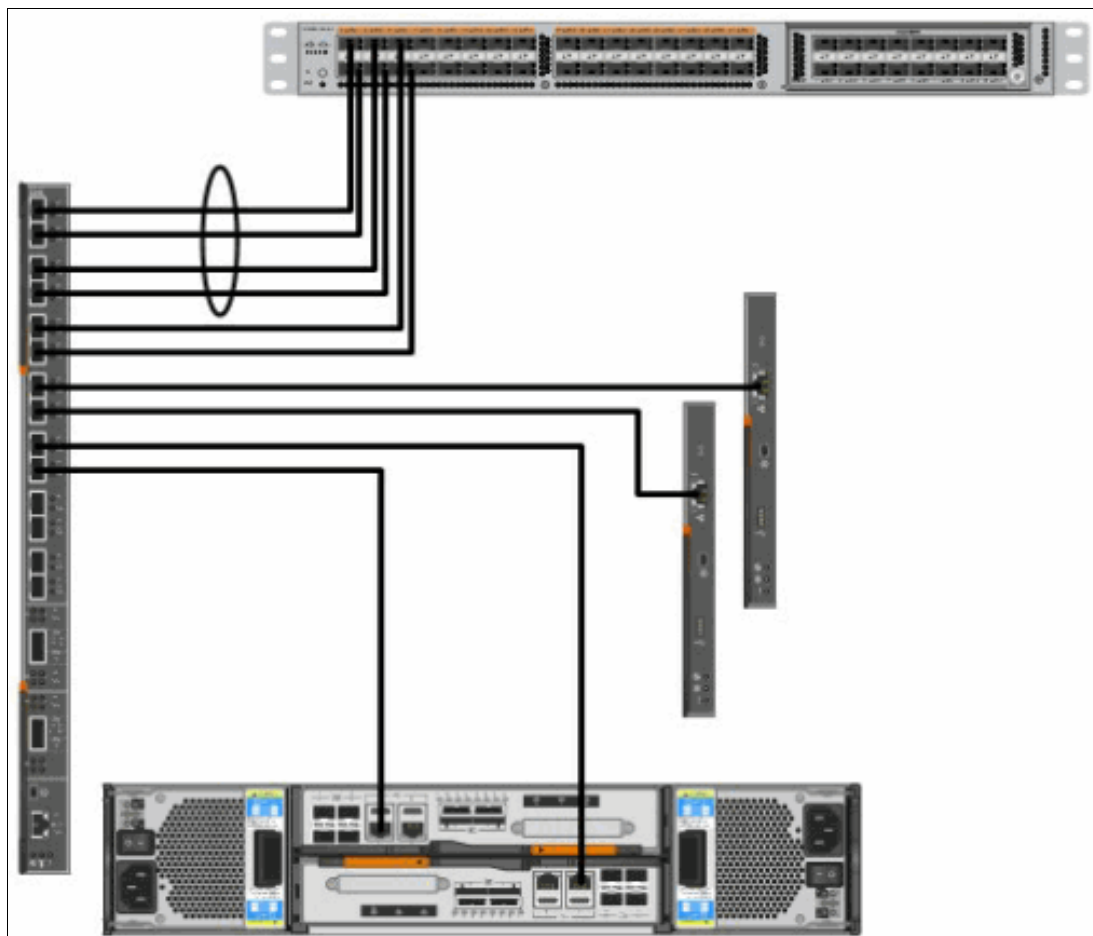


Figure A-5 Nexus configured to integrate with the Flex implementation switch configuration

To enable the 4-port link aggregation, first you must enable the LACP feature (if that is not already done), as shown in Example A-28.

Example: A-28 Enable LACP feature

```
feature lacp
```

In this example, we are using ports 17 - 20 on a Nexus 5020 for the port-channel, as shown in Example A-29.

Example: A-29 Port-channels in use

```
nx5020(config)#conf t
nx5020(config)#int e1/17-20
nx5020(config)#channel-group mode 1 active
!change the channel-group number to meet local requirements!
```

Because we are currently using only a single VLAN (4092), the port-channel can be an access mode, with native VLAN 4092, as shown in Example A-30.

Example: A-30 Native VLAN 4092 access

```
nx5020(config)# int port-channel 1
nx5020(config-if)# switchport access vlan 4092
```

The next step is to connect one port from the EN4093/EN4093R to the Nexus switch to ensure that the link comes up as expected. Example A-31 shows the commands.

Example: A-31 Connect EN4093/EN4093R to the Nexus switch

```
nx5020(config)# show port-channel summary
```

Flags: D - Down P - Up in port-channel (members)
 I - Individual H - Hot-standby (LACP only)
 s - Suspended r - Module-removed
 S - Switched R - Routed
 U - Up (port-channel)

Group	Port-Channel	Type	Protocol	Member Ports
1	Pol(SU)	Eth	LACP	Eth1/17(P) Eth1/18(D) Eth1/19(D) Eth1/20(D)

As shown in Example A-32, check from the EN4093/EN4093R to see that the link is up as expected.

Example: A-32 Check status on EN4092 side

```
Router#show etherchannel summary
```

PortChannel 65: Enabled
 Protocol - LACP
 Port State:
 EXT1: STG 28 forwarding
 ---- non-relevant output omitted ----
 =====

port	mode	adminkey	operkey	selected	prio	aggr	trunk	status	minlinks
EXT1	active	3000	3000	yes	32768	43	65	up	1
EXT2	active	3000	3000	no	32768	--	--	down	1
EXT3	active	3000	3000	no	32768	--	--	down	1
EXT4	active	3000	3000	no	32768	--	--	down	1

Add more links as required (the EN4093/EN4093R is configured for four ports), as shown in Example A-33.

Example: A-33 Add additional link

```
nx5020(config)# show port-channel summary
```

Flags: D - Down P - Up in port-channel (members)
 I - Individual H - Hot-standby (LACP only)
 s - Suspended r - Module-removed

S - Switched R - Routed
U - Up (port-channel)

Group	Port-Channel	Type	Protocol	Member Ports
1	Po1(SU)	Eth	LACP	Eth1/17(P) Eth1/18(P) Eth1/19(P) Eth1/20(P)

Router#**show etherchannel summary**

PortChannel 65: Enabled

Protocol - LACP

Port State:

EXT1: STG 28 forwarding

EXT2: STG 28 forwarding

EXT3: STG 28 forwarding

EXT4: STG 28 forwarding

port	mode	adminkey	operkey	selected	prio	aggr	trunk	status
minlinks								
EXT1	active	3000	3000	yes	32768	43	65	up 1
EXT2	active	3000	3000	yes	32768	43	65	up 1
EXT3	active	3000	3000	yes	32768	43	65	up 1
EXT4	active	3000	3000	yes	32768	43	65	up 1

Link Layer Discovery Protocol

Enable feature Link Layer Discovery Protocol (LLDP) on the Nexus device, as shown in Example A-34, in case it is not already done.

Example: A-34 Enable LLDP

nx5020# **show lldp neighbors**

Capability codes:

(R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device

(W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID	Local Intf	Hold-time	Capability	Port ID
ToR	mgmt0	120	BR	24
Flex-SW1	Eth1/17	120	BR	43
Flex-SW1	Eth1/18	120	BR	44
Flex-SW1	Eth1/19	120	BR	45
Flex-SW1	Eth1/20	120	BR	46

Total entries displayed: 5

Now enable Enable LLDP on the EN4093/EN4093R, as shown in Example A-35.

Example: A-35 Enabling LLDP on the EN4093/EN4093R side

Router(config)# LLDP en

Change the hostname on the EN4093 to have the name propagated through LLDP

Router(config)# hstname Flex-SW1

Now show the LLDP remote-host information, as shown in Example A-36.

Example: A-36 Show LLDP remote-host information

```
Flex-SW1#show lldp remote-device
```

LLDP Remote Devices Information

LocalPort	Index	Remote Chassis ID	Remote Port	Remote System Name
EXT1	1	00 0d ec a3 0f 58	Eth1/17	nx5020
EXT2	2	00 0d ec a3 0f 59	Eth1/18	nx5020
EXT3	3	00 0d ec a3 0f 5a	Eth1/19	nx5020
EXT4	4	00 0d ec a3 0f 5b	Eth1/20	nx5020
INTA3	5	5c f3 fc 7f 17 99	5c-f3-fc-7f-17-99	
INTA1	6	5c f3 fc 5f 44 5b	5c-f3-fc-5f-44-5b	
INTA4	7	5c f3 fc 7f 97 99	5c-f3-fc-7f-97-99	

Management network

In this example, port E1/21 on the Nexus 5020 will be attached to port EXT6 on the EN4093/EN4093R. Example A-37 shows how to create VLAN 4093, and give it a name (optional).

Example: A-37 Create VLAN 4093, and give it a name (optional)

```
nx5020(config)# vlan 4093
nx5020(config-vlan)# name PoC_Mgt_Gold
```

Example A-38 adds the VLAN to port E1/21.

Example: A-38 Add the VLAN to port E1/21

```
nx5020(config)# int e1/21
nx5020(config-if)# description to EN4093-EXT6
nx5020(config-if)# switchport access vlan 4093
```

Last, Example A-39 shows how to make the connection from port E1/21 to port EXT6.

Example: A-39 Connect port E1/21 to port EXT6

```
nx5020(config)# show interface status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Eth1/21	to EN4093-EXT6	connected	4093	full	10G	10g

Double check if everything is accurate on the EN4093/EN4093R side, as shown in Example A-40.

Example: A-40 Checking status on EB4093 after Nexus configuration

```
Router#show interface status
```

Alias	Port	Speed	Duplex	Flow Ctrl	Link	Name
EXT6	48	10000	full	no no	up	EXT6

Changing the VLAN IDs in the Flex

If in the site network, VLANs 4091, 4092, and 4093 are not in use (99% of the cases), we must change the VLAN IDs used in the EN4093/EN4093R to work with the site network.

Important: This procedure will not change the IP addresses used in the Flex.

Example A-41 shows the VLAN configuration from manufacturing.

Example: A-41 Show VLAN command output

Router#show vlan					
VLAN	Name	Status	MGT	Ports	
1	Default VLAN	ena	dis	INTB1-INTC14	EXT11-EXT22
4091	OS Mgmt	ena	dis	INTA1-INTA14	EXT5
4092	Data	ena	dis	INTA1-INTA14	EXT1-EXT4
4093	Device Mgmt	ena	dis	EXT6-EXT10	
4095	Mgmt VLAN	ena	ena	EXTM MGT1	

For this example, we use:

- ▶ VLAN 100 for Data
- ▶ VLAN 200 for Device Management
- ▶ VLAN 300 for OS Management

We set VLAN 300 as the default or native, untagged VLAN.

We then use a single uplink (EXT 6) to the site network to attach the POC into the site network.

Note: If using one of the EXternal ports that are pre-configured for LACP, you must also remove the LACP configuration.

First, we must create VLAN 100, as shown in Example A-42.

Example: A-42 Create VLAN 100

```
Router#conf t
Enter configuration commands, one per line. End with Ctrl/Z.
Router(config)#vlan 100

VLAN number 100 with name "VLAN 100" created.

Warning: VLAN 100 was assigned to STG 100.
```

You should notice that the switch automatically creates a new instance of the Spanning Tree Protocol (STG 100), and assigns it to VLAN 100. This is *Per VLAN Spanning Tree*, which matches the Cisco default. Each VLAN has its own instance of the Spanning Tree Protocol.

We can optionally give this new VLAN a name as shown in Example A-43.

Example: A-43 Give a name to the VLAN

```
Router(config-vlan)#name "New-Data"
```

Next, we must enable the VLAN as shown in Example A-44. That makes the VLAN active, and usable in the switch.

Example: A-44 Enable VLAN

```
Router(config-vlan)#en
```

In Example A-45, we add the ports to the VLAN configuration.

Example: A-45 Add ports to the VLAN

```
Router(config-vlan)#member inta1-inta14,ext6
Port EXT6 is an UNTAGGED port and its PVID is changed from 4093 to 100
```

Notice that we can add multiple ports to a VLAN with a single command. Also, note that because port EXT6 is not configured for tagging, its PVID was automatically changed. In the example, VLAN 300 is our untagged (PVID) VLAN.

The next step is to set port EXT6 to start tagging, as shown in Example A-46.

Example: A-46 Set port EXT6 to tag

```
Router(config-vlan)#exit
Router(config)#interface port EXT6
Router(config-if)#tagging
Router(config-if)#PVID 300
VLAN number 300 with name "VLAN 300" created.

PORT EXT6 added in VLAN number 300 with name "VLAN 300".

Warning: VLAN 300 was assigned to STG 46.
```

In this example, we created VLAN 300 by setting port EXT6 PVID to VLAN 300, which automatically enabled the VLAN, and gave it a spanning-tree instance. Example A-47 shows the current status.

Example: A-47 Check for the created VLAN 300

```
Router(config-if)#exit
Router(config)#show vlan
```

VLAN	Name	Status	MGT	Ports
1	Default VLAN	ena	dis	INTA1-INTC14 EXT11-EXT22
100	New-Data	ena	dis	INTA1-INTA14 EXT6
300	VLAN 300	ena	dis	EXT6
4091	OS Mgmt	ena	dis	INTA1-INTA14 EXT5
4092	Data	ena	dis	INTA1-INTA14 EXT1-EXT4
4093	Device Mgmt	ena	dis	EXT7-EXT10
4095	Mgmt VLAN	ena	ena	EXTM MGT1

We can add the internal ports to VLAN 300 with the commands shown in Example A-48.

Example: A-48 Adding internal ports to VLAN 300

```
Router(config)#vlan 300
Router(config-vlan)#member inta1-inta14
```

VLAN	Name	Status	MGT	Ports
300	VLAN 300	ena	dis	INTA1-INTA14 EXT6

Lastly, create VLAN 200 and add the ports as shown in Example A-49.

Example: A-49 Create VLAN 200 and add the ports

```
Router(config-vlan)#vlan 200

VLAN number 200 with name "VLAN 200" created.

Warning: VLAN 200 was assigned to STG 73.
Router(config-vlan)#name "Device Management"
Router(config-vlan)#en
Router(config-vlan)#member inta1-inta14,ext6
```

List of commands

Example A-50 shows the commands needed to create three VLANs: XXX, YYY, and ZZZ (substitute as required for your local environment), assign ports 1-14 (INTA1-INTA14), and one external port (EXTx, substitute as needed). The PVID of the EXT port can be changed if you substitute the **PVID 1** command to some other valid VLAN ID.

Example: A-50 A sequence of commands to create VLANs and assign ports, including an external one

```
conf t
int port EXTx !change to valid external port
tagging
vlan XXX    !change to valid VLAN ID
en
member INTA1-INTA14,EXTx !change to valid external port
vlan YYY    !change to valid VLAN ID
en
member INTA1-INTA14,EXTx !change to valid external port
vlan ZZZ    !change to valid VLAN ID
en
member INTA1-INTA14,EXTx !change to valid external port
int port EXTx !change to valid external port
pvid 1      !change to valid VLAN ID to set to different PVID
end
```

Show EN4093/EN4093R running configuration

In a PureFlex system, CMM security policy is set to Secure Mode, which means that the switch is set to allow Secure Shell (SSHv2), but not Telnet. Any SSH client can be used to connect to the switch.

The Browser-Based Interface (BBI) is also restricted by the security policy of the PureFlex System. HTTPS is allowed; HTTP is not allowed.

The network path to attach to the EN4093/EN4093R is through the CMM. The CMM performs a Proxy Address Resolution Protocol (ARP) for the IP addresses connected through its internal network including the Ethernet and SAN fabric switches in the chassis and the IMMs on the compute nodes.

To access the CMM network, the external interface of the CMM must be reachable on the network. This can be accomplished by attaching the CMM to the site network, or while the Flex POC is still in the initial configuration, attaching to an unused external port on the EN4093/EN4093R.

On an IBM switch, the **Show Running-Configuration** command shows only the configuration items that are changed from the factory default.

The factory default configuration can be summed up as follows:

- ▶ Internal ports are configured for tagging (multiple VLANs)
- ▶ External ports are non-tagging (single, untagged VLAN)
- ▶ VLAN 1 is the only configured VLAN
- ▶ All ports are a member of VLAN 1

The factory-default configuration differs from the manufacturing configuration, which is applied to the switch for the POC.

The EN4093/EN4093R (like many, but not all) switches have aliases assigned to the ports to aid in identification and configuration. When performing configuration, either the port number or port alias can be used to specify the port.

Interfaces beginning with INT in the name are internal ports, which are connected to the adapter interfaces on the nodes. The letter in the alias name indicates which port on the adapter the switch interface is attached to. For example, A indicates the first partition and the first adapter port, B is the second port on the adapter, and C is the third adapter. The number indicates the compute node bay that the port is attached to. For example, INTB3 is the port that attaches the second port of compute node bay 3.

Aliases beginning with EXT are external ports.

Example A-51 shows the running configuration on the EN4093/EN4093R side.

Example: A-51 EN4093/EN4093R configuration dump

```
version "7.2.2.2"
switch-type "IBM Flex System Fabric EN4093 10Gb Scalable Switch"
!
!
system idle 60
!
!
interface port INTA1
    tagging
    pvid 4091
    exit
!
interface port INTA2
    tagging
    pvid 4091
```

```

        exit
    !
    interface port INTA3
        tagging
        pvid 4091
        exit
    !
    interface port INTA4
        tagging
        pvid 4091
        exit
    !
    interface port INTA5
        tagging
        pvid 4091
        exit
    !
    interface port INTA6
        tagging
        pvid 4091
        exit
    !
    interface port INTA7
        tagging
        pvid 4091
        exit
    !
    interface port INTA8
        tagging
    pvid 4091
        exit
    !
    interface port INTA9
        tagging
        pvid 4091
        exit
    !
    interface port INTA10
        tagging
        pvid 4091
        exit
    !
    interface port INTA11
        tagging
        pvid 4091
        exit
    !
    interface port INTA12
        tagging
        pvid 4091
        exit
    !
    interface port INTA13
        tagging
        pvid 4091

```

```

        exit
    !
    interface port INTA14
        tagging
        pvid 4091
        exit
    !
    interface port EXT1
        pvid 4092
        exit
    !
    interface port EXT2
        pvid 4092
        exit
    !
    interface port EXT3
        pvid 4092
        exit
    !
    interface port EXT4
        pvid 4092
        exit
    !
    interface port EXT5
        pvid 4091
        exit
    !
    interface port EXT6
        pvid 4093
        no auto
        exit
    !
    interface port EXT7
        pvid 4093
        exit
    !
    interface port EXT8
        pvid 4093
        exit
    !
    interface port EXT9
        pvid 4093
        exit
    !
    interface port EXT10
        pvid 4093
        exit
    !
    interface port EXTM
        shutdown
        exit
    !
    vlan 1
        member INTB1-INTB14,EXT15-EXT22
        no member INTA1-INTA14,EXT1-EXT10

```

```

!
!
vlan 4091
    enable
    name "OS Mgmt"
    member INTA1-INTA14,EXT5
!
!
vlan 4092
    enable
    name "Data"
    member INTA1-INTA14,EXT1-EXT4
!
!
vlan 4093
    enable
    name "Device Mgmt"
member EXT6-EXT10
!
!
!
spanning-tree stp 123 vlan 4091
spanning-tree stp 124 vlan 4092
spanning-tree stp 125 vlan 4093
!
no logging console
!
interface port EXT1
    lacp mode active
    lacp key 3000
!
interface port EXT2
    lacp mode active
    lacp key 3000
!
interface port EXT3
    lacp mode active
    lacp key 3000
!
interface port EXT4
    lacp mode active
    lacp key 3000
!
!
!
!
!
!
!
!
!
ntp enable
ntp ipv6 primary-server fe80::211:25ff:fec3:5ded MGT
ntp interval 15

```



```
ntp authenticate
ntp primary-key 32051
!
ntp message-digest-key 32051 md5-ekey
748d815354058002bcc6e2b297b523f09761d882fcc378a439bd5557daaa6f01ab507c82b7bc3a9
641a
1d4cde03ff7517696bd90950e8feff1a933f80421c0f2
!
ntp trusted-key 32051
!
end
```

Adding a second IBM EN4093/EN4093R switch to the setup

Example A-52 shows how to use a script to add a second EN4093/EN4093R to the setup. It should be used only on a switch with the factory default configuration. The Industry Standard command-line interface (ISCLI) directive is assumed here.

Example: A-52 A script to add a second IBM EN4093/EN4093R LBS-compatible

```
! enter configuration mode
conf t
! settings for the user
system idle 60
no logging console
! set the internal ports for tagging (multiple vlans)
interface port INTA1-INTA14
tagging
exit
! create vlans, add member ports
vlan 1
no member INTA1-INTA14,EXT1-EXT10
vlan 4091
enable
name "OS Mgmt"
member INTA1-INTA14,EXT5
vlan 4092
enable
name "Data"
member INTA1-INTA14,EXT1-EXT4
vlan 4093
enable
name "Device Mgmt"
member EXT6-EXT10
exit
! Go back and set the internal ports "native"/"default"/"untagged" VLAN
interface port INTA1-INTA14
pvid 4091
exit
! Set up LACP on 4 external ports
interface port EXT1
lacp mode active
lacp key 3000
interface port EXT2
```

```
lacp mode active
lacp key 3000
interface port EXT3
lacp mode active
lacp key 3000
interface port EXT4
lacp mode active
lacp key 3000
exit
end
!Save our work
copy run start
!you may be prompted here, answer 'y'
```

This completes the setup.



Easy Connect

IBM Easy Connect is a simple configuration mode implemented on IBM System Networking Ethernet and converged switches that enables easy integration of IBM Flex and PureSystems with existing Cisco, Juniper, and other vendor data center networks. Easy Connect makes connecting to existing upstream networks simple while enabling advanced in-system connectivity at the network edge. It also allows administrators to allocate bandwidth and optimize performance. In short, it supports both your existing and future network.

The following topics are described:

- ▶ Introduction to IBM Easy Connect
- ▶ Single Mode
 - Example diagram and characteristics
 - Implementation
- ▶ Storage Mode
 - Example diagram and characteristics
 - Implementation
- ▶ Multi-Chassis Mode (PureFlex only)
 - Example diagram and characteristics
 - Implementation with CN/EN4093/R
 - Implementation with RackSwitch G8264
- ▶ Client examples
 - Telecommunications
 - State government
 - Medical center
- ▶ Limitations

8.5 Introduction to IBM Easy Connect

Easy Connect configuration mode enables IBM PureSystems to meet the primary selection criteria for adding new integrated systems to existing data center networks. Instead of requiring complex network configuration for each individual server, Easy Connect mode allows connection to a complete, integrated multiprocessor chassis or rack comprising PureSystems compute, storage, system management, and networking resources. It then manages this scalable resource with the simplicity of a single network node.

IBM System Networking Ethernet switches supporting the Easy Connect feature include the following components:

1. IBM Flex System Fabric EN4093/EN0493R and Virtual Fabric 10 Gb Scalable Switches
2. IBM Flex System Fabric CN4093 10 Gb Converged Scalable Switch
3. IBM System Networking RackSwitch G8264CS
4. IBM RackSwitch G8264 or G8124E
5. IBM RackSwitch G8214 (not in Fibre Channel over Ethernet (FCoE) mode)

Easy Connect mode provides transparent PureSystems connectivity to your existing Cisco, Juniper, or other vendor network. With Easy Connect enabled on the EN4093/R, CN4093, or G8264 switches, the core network sees a “big pipe” for compute traffic coming to and from the PureSystems chassis. The switch becomes a simple I/O module that connects servers and storage with the core network. It aggregates compute node ports, and behaves similarly to Cisco Fabric Extension (FEX) by appearing as a “dumb” device to the upstream network, with the main difference being that intra-chassis switching is supported. Unlike Cisco FEX, traffic does not have to be sent upstream if the network destination is housed in the same physical chassis.

The Spanning Tree Protocol is disabled on the supported IBM System Networking switch in all Easy Connect modes, eliminating the data center administrator's Spanning Tree concerns. This loop-free topology requires no additional configuration after setup, and helps to provide economical bandwidth use with prioritized pipes and network virtualization for both Intel and Power Systems compute nodes.

8.6 Easy Connect Single Mode

Easy Connect Single Mode allows the IBM Flex System EN4093/R switch to act as a Fabric Extension Module in a Cisco network. Clients that use active/passive network interface card (NIC) teaming with no NIC bonding (Link Aggregation Control Protocol (LACP) or static PortChannel) on the compute nodes are well suited with Single Mode, as illustrated in Figure B-1 on page 255.

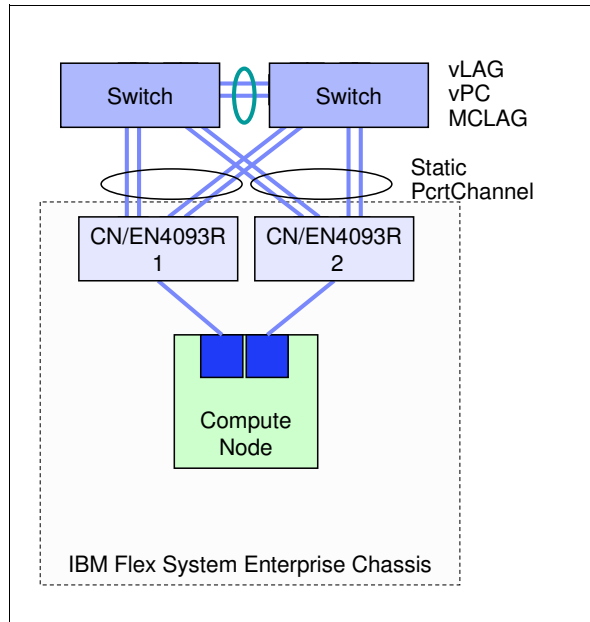


Figure B-1 IBM Easy Connect Single Mode diagram

Following are some important distinctions with Single Mode:

1. All local Layer-2 traffic pointing to the same I/O bay in the Enterprise Chassis remains within the same chassis.
2. The CN4093 or EN4093/EN4093R I/O modules are not connected together with a virtual Link Aggregation Group (vLAG). Therefore, traffic destined for compute nodes using different I/O bays within the same Enterprise Chassis will need to travel to the upstream switch, and then back down.
3. Each Enterprise Chassis appears as two separate devices to the upstream network when using two I/O modules.

8.6.1 Implementation

To configure the CN4093 or EN4093/EN4093R I/O modules for Easy Connect Single Mode, perform the following steps:

1. Connect to the I/O module's command-line interface (CLI) using Telnet or Secure Shell (SSH).
2. Change the configuration mode to the Industry Standard CLI (ISCLI), if it is not already configured to do so. See Example 8-37. Enable the CLI prompt in the last step if the Flex System Manager (FSM) is being used in the environment.

Example 8-37 Changing the I/O module to use the ISCLI

```
/boot/mode iscli
/boot/reset
/boot/prompt enable
```

3. If the I/O module is not already in a factory default configuration, do the steps shown in Example 8-38 after connecting to it via Telnet/SSH.

Example 8-38 Resetting the I/O module to a factory default configuration

```
EN4093> enable
EN4093# configure terminal
EN4093#(config) boot configuration-block factory
EN4093#(config) reload
```

4. When the I/O module returns to a factory default configuration, perform the steps that are shown in Example 8-39 to enable Easy Connect Single Mode.

Example 8-39 Implementing Easy Connect Single Mode

```
spanning-tree mode disable
portchannel 1 port ext1-ext10 enable
vnic enable
    vnic vnicgroup 1
    vlan 4091
    port INTA1-INTA14
    portchannel 1
    enable
    failover
    exit
write memory
```

5. Easy Connect Single Mode is now implemented.

Note: The IBM Virtual Fabric Switch Module (VFSM) for the IBM BladeCenter H or HT chassis is supported by Easy Connect Single and Storage Modes. Configuration steps are identical. This can also be done in a System x environment with rack servers by using G8124, G8264, or G8264CS.

Some important considerations and potential next steps now that Easy Connect Single Mode is enabled:

- ▶ Configure Spanning-Tree BPDU Guard and Edge on the upstream switch for more protection. These are enabled by default on Cisco Nexus 2000 Fabric Extender ports and cannot be disabled.
- ▶ Setting “spanning-tree type network” on an upstream Cisco Nexus port is not supported.

8.7 Storage Mode

Easy Connect Storage Mode allows the IBM Flex System EN4093/R switch to act as a Fabric Extension Module in a Cisco network running on Fibre Channel over Ethernet (FCoE) connections. Storage Mode is nearly identical to Single Mode from a configuration standpoint. The only difference is that Converged Enhanced Ethernet (CEE) must be enabled in order for FCoE to function. Storage Mode is illustrated in Figure B-2 on page 257.

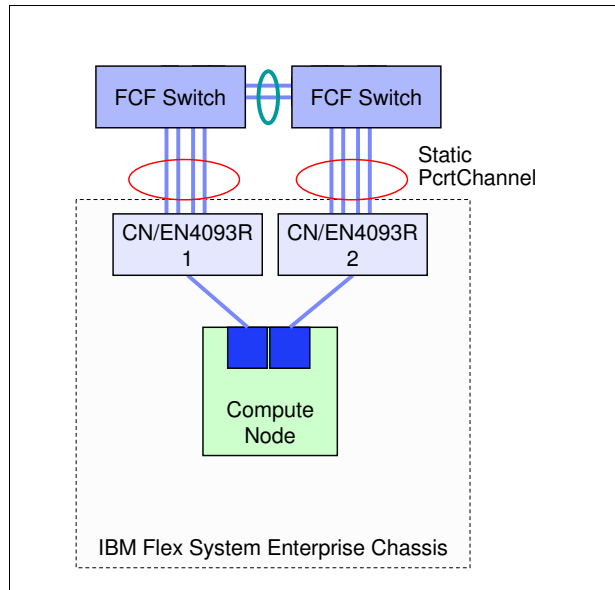


Figure B-2 IBM Easy Connect Storage Mode diagram

The distinctions listed for Single Mode are the same for Storage Mode.

8.7.1 Implementation

Perform the following steps to implement Easy Connect Storage Mode.

1. To configure the CN4093 or EN4093/EN4093R I/O modules for Easy Connect Storage Mode, first perform steps 1, 2, and 3 as listed in the Single Mode implementation in section 2 on page 255.
2. After you complete those steps, do the following on the I/O module that is shown in Example 8-40 to implement Storage Mode. The only difference is highlighted in bold text.

Example 8-40 Implementing Easy Connect Storage Mode

```
spanning-tree mode disable
portchannel 1 port ext1-ext10 enable
vnic enable
  vnic vnicgroup 1
  vlan 4091
  port intal-intal4
  portchannel 1
  enable
  failover
  exit
cee enable
write memory
```

3. Easy Connect Storage Mode is now implemented.

The same considerations listed for Single Mode and next steps apply similarly towards Storage Mode, with the exception of the following caveat:

IBM Networking OS 7.6 and earlier does not support FCoE traffic over multiple aggregated links, using either LACP or static PortChannels.

8.8 Easy Connect Multi-Chassis Mode

Easy Connect Multi-Chassis Mode allows IBM RackSwitch G8264 (acting as an aggregator for multiple chassis) and Flex System EN4093/R switches to act as Fabric Extension Modules in a Cisco network.

Clients that use active/active NIC teaming with either Link Aggregation Control Protocol (LACP, or IEEE 802.3ad), or Static IP Hash on the compute node will be best suited with Multi-Chassis Mode, as illustrated in Figure B-3.

Multiple chassis: Alternatively, this could be multiple chassis' connected to a pair of G8264s at the top-of-rack going out to a client's existing network.

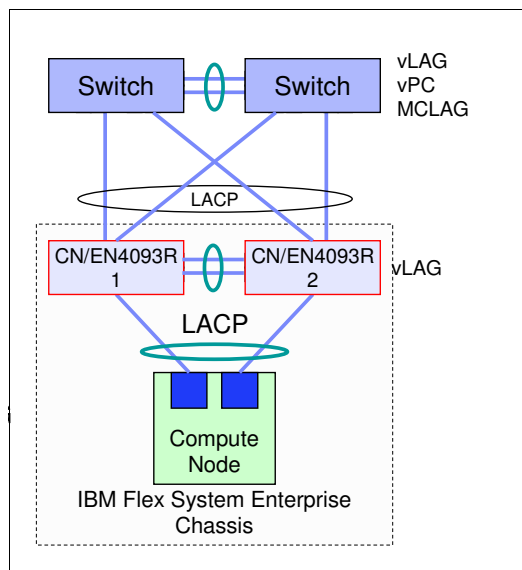


Figure B-3 IBM Easy Connect Multi-Chassis Mode diagram

Consider the following important distinctions with Multi-Chassis Mode:

1. Because the CN4093 or EN4093/EN4093R I/O modules are connected together with a virtual Link Aggregation Group (vLAG) inter-switch link (ISL), *all* layer-2 traffic destined for compute nodes using either the same, or different I/O Bays within the same Enterprise Chassis will never leave the chassis.
2. Each Enterprise Chassis appears as a single device to the upstream network when using two I/O modules.
3. All operating systems (AIX, Linux, Windows, VMWare, VIO) within the IBM Flex System Enterprise Chassis *must* TAG VLANs.

Exception: If the Flex System Manager (FSM) is used, the client *must* enable the top-of-rack port "Native VLAN ID" with the VLAN that the FSM is configured on because the FSM cannot TAG.

4. Multi-Chassis Mode allows for pNIC or switch-independent vNIC modes to be used on the compute node network adapters. If multiple vNIC groups are used for either traffic separation and using IBM Virtual Fabric Mode, each vNIC group requires its own uplink/PortChannel.

Note: IBM Flex System POWER Nodes support *pNIC mode only* as of this writing.

5. Multi-Chassis Mode allows for the eventual implementation of IBM Virtual Fabric Mode.

8.8.1 Implementation with CN/EN4093/R

To configure the CN4093 or EN4093/R I/O modules for Easy Connect Multi-Chassis Mode, perform the following generalized steps:

1. Restore the factory default configuration to the I/O module. Detailed steps for this step are described in Example 8-38 on page 256.
2. Disable the Spanning-Tree Protocol globally.
3. Configure all the internal (INT) and external (EXT) CN4093 or EN4093/R ports using the **tagpvid-ingress** keyword using VLAN 4091 as the Port VLAN ID (PVID).
4. Enable 802.1Q VLAN tagging on the external ports being used as the vLAG peer link between the I/O modules using VLAN 4090 (vLAG ISL VLAN) as the PVID. Add VLAN 4091 as a tagged member.
5. Configure all required LACP aggregations (vLAG Peer Link, EXT, and INT ports).
6. Configure a superfluous IP address to be used by the management EXT port vLAG **Health Check** parameter.
Consider using address 1.1.1.1 for the first I/O module, and 1.1.1.2 for the second I/O module.
7. Finally, configure the vLAG ISL, Health Check peer-ip, and all associated vLAG pairs.
8. Easy Connect Multi-Chassis Mode is now implemented on the CN/EN4093/R.

Note: The IBM Virtual Fabric Switch Module (VFSM) for the IBM BladeCenter H or HT chassis does not work in Multi-Chassis Mode because it does not support vLAG.

A sample script to enable Easy Connect Multi-Chassis Mode on the CN/EN4093/R I/O module is shown in Example 8-41.

Example 8-41 Sample script for Easy Connect Multi-Chassis Mode on CN/EN4093/R

```
spanning-tree mode disable
interface port ext9,ext10      --> ISL vLAG Peer-Link Ports
    pvid 4090
    tagging
    lacp key 1001
    lacp mode active
vlan 4090
    enable
    name Peer-Link
vlan 4091
    enable
    name Intel-Nodes
    member int1-int14,ext1-ext4,ext9,ext10
interface port int1-int14,ext1-ext4
    tagpvid-ingress
interface port ext1-ext4      --> uplink ports to AGG/Core
    lacp key 4091             --> use SAME key on both VFSM INTEL Uplinks (4091)
```

```

lacp mode active
interface port inat1          --> INTa1 on both Switches will be in same
PortChannel using vLAG (lacp key MUST match)
lacp key 101
lacp mode active
interface port inat2
lacp key 102
lacp mode active
interface ip 127              --> IP 127 is dedicated to the MGT Port used for
vLAG health check
ip address 1.1.1.1
enable
vlag ena
vlag isl peer-ip 1.1.1.2      --> other switch will use 1.1.1.1
vlag isl vlan 4090
vlag isl adminkey 1001
vlag tier-id 10              --> each pair of switches connecting to each
other should be a different Tier-ID
vlag adminkey 4091 enable
vlag adminkey 101 enable
vlag adminkey 102 enable     --> repeat for each Server using 802.3ad / LACP
write memory

```

8.8.2 Implementation with G8264

If the client is using a pair of IBM RackSwitch G8264 switches in the overall topology, as shown in Figure B-4, such as in a pre-racked, pre-cabled IBM PureFlex System Express, Standard, or Enterprise rack configuration, the following section describes how Easy Connect can be used.

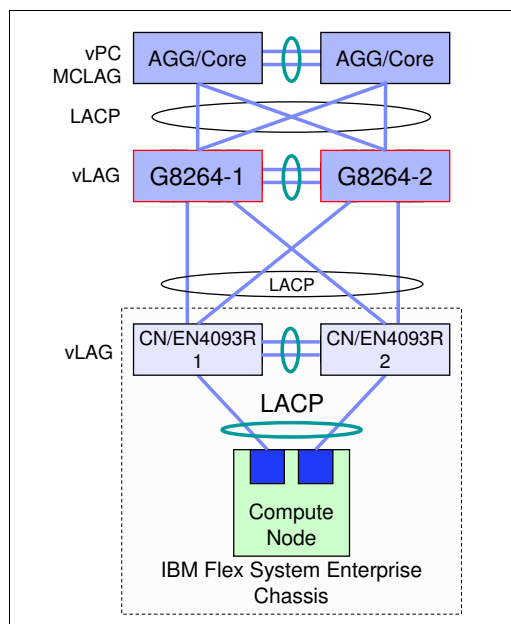


Figure B-4 IBM Easy Connect Multi-Chassis Mode with RackSwitch G8264

To configure the RackSwitch G8264 for Easy Connect Multi-Chassis Mode, perform the following generalized steps:

1. Restore the factory default configuration to the G8264. Generalized steps for the EN4093/R can be used and are described in Example 8-38 on page 256.
2. Disable the Spanning-Tree Protocol globally.
3. Configure all the upstream and downstream G8264 ports using the **tagpvid-ingress** keyword using VLAN 4091 as the PVID.
4. Enable 802.1Q VLAN tagging on the ports being used as the vLAG peer link between the G8264s using VLAN 4090 (vLAG ISL VLAN) as the PVID. Add VLAN 4091 as a tagged member.
5. Configure all required LACP aggregations (vLAG peer link, CN4093/EN4093/R facing ports).
6. Configure a superfluous IP address to be used by the management EXT port vLAG **Health Check** parameter.
Consider using address 1.1.1.1 for the first I/O module, and 1.1.1.2 for the second I/O module.
7. Finally, configure the vLAG ISL, Health Check peer-ip, and all associated vLAG pairs.
8. Easy Connect Multi-Chassis Mode is now implemented on the RackSwitch G8264.

A sample script to enable Easy Connect Multi-Chassis Mode on the RackSwitch G8264 is shown in Example 8-42.

Example 8-42 Sample script for Easy Connect Multi-Chassis Mode on RackSwitch G8264

```

spanning-tree mode disable          --> Optional
interface port 1,5                  --> 2x 40Gb ISL (e.g. between G8264's)
    tagging
    pvid 4090
    lacp key 4090
    lacp mode active
vlan 4090
    enable
    name Peer-Link
vlan 4091
    enable
    name "Transparent-Ports"
interface port 17-64                --> Uplinks and CN/EN4093/R facing Ports ONLY
    tagpvid-ingress
interface port 17,18                --> Uplink ports to AGG/Core
    lacp key 1001
    lacp mode active
interface port 19,20                --> Ports facing first PureFlex enclosure
    lacp key 1920
    lacp mode active
interface port 21,22                --> Ports facing second PureFlex enclosure
    lacp key 2122
    lacp mode active
vlag enable
vlag isl adminkey 4090
vlag tier-id 1
vlag adminkey 1001 ena              --> Uplink PortChannel to AGG/Core
vlag adminkey 1920 ena

```

```
vlag adminkey 2122 ena          --> Repeat for each Port-Channel to each  
CN/EN4093/R  
write memory
```

An important consideration and potential next step now that Easy Connect Multi-Chassis Mode is enabled, is to configure Spanning-Tree BPDU Guard and Edge on the upstream switch for additional protection.

8.9 Client examples with diagrams

The following section lists common implementation scenarios with Easy Connect for various industries that have purchased IBM PureFlex System hardware. Requirements are listed as dictated by the client, and a proceeding network diagram to fit those requirements is proposed.

8.9.1 Telecommunications client

This client requires the following components:

- ▶ No Spanning Tree or any other protocols seen by the network.
- ▶ Upstream connection must be in to a Cisco Nexus 2000 Fabric Extender that is not running virtual path connection (VPC).
- ▶ The EN4093/R I/O modules in the IBM Flex System Enterprise Chassis must be transparent devices that require no management by any group after initial setup.

Figure B-5 on page 263 shows how Easy Connect satisfies all of the telecommunications client's requirements.

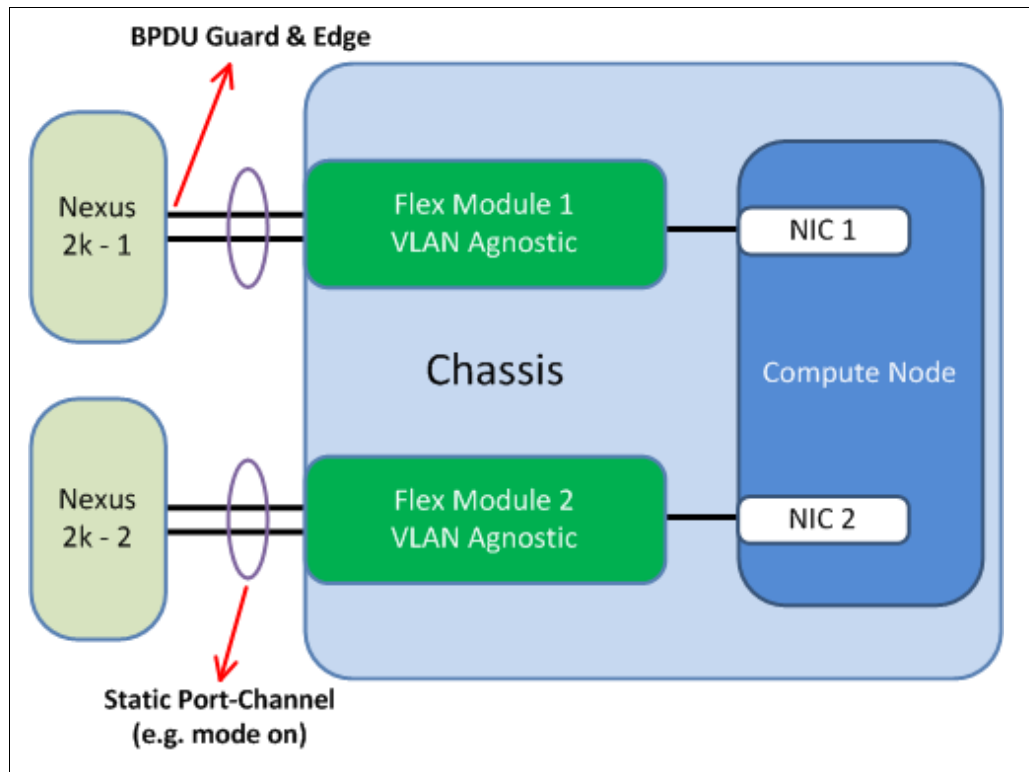


Figure B-5 Telecommunications client network diagram

8.9.2 State government client

This client requires the following components:

- ▶ Use LAN on Motherboard (LoM) in Virtual Fabric Mode so bandwidth can be adjusted as needed for each vNIC as required.
- ▶ Dedicated uplink vPC PortChannel from each EN4093/R for each vNIC group for separation of traffic.
- ▶ The EN4093/R I/O modules in the IBM Flex System Enterprise Chassis must be transparent devices that require no management by any group after initial setup.

Figure B-6 shows how Easy Connect satisfies all of the state government client's requirements.

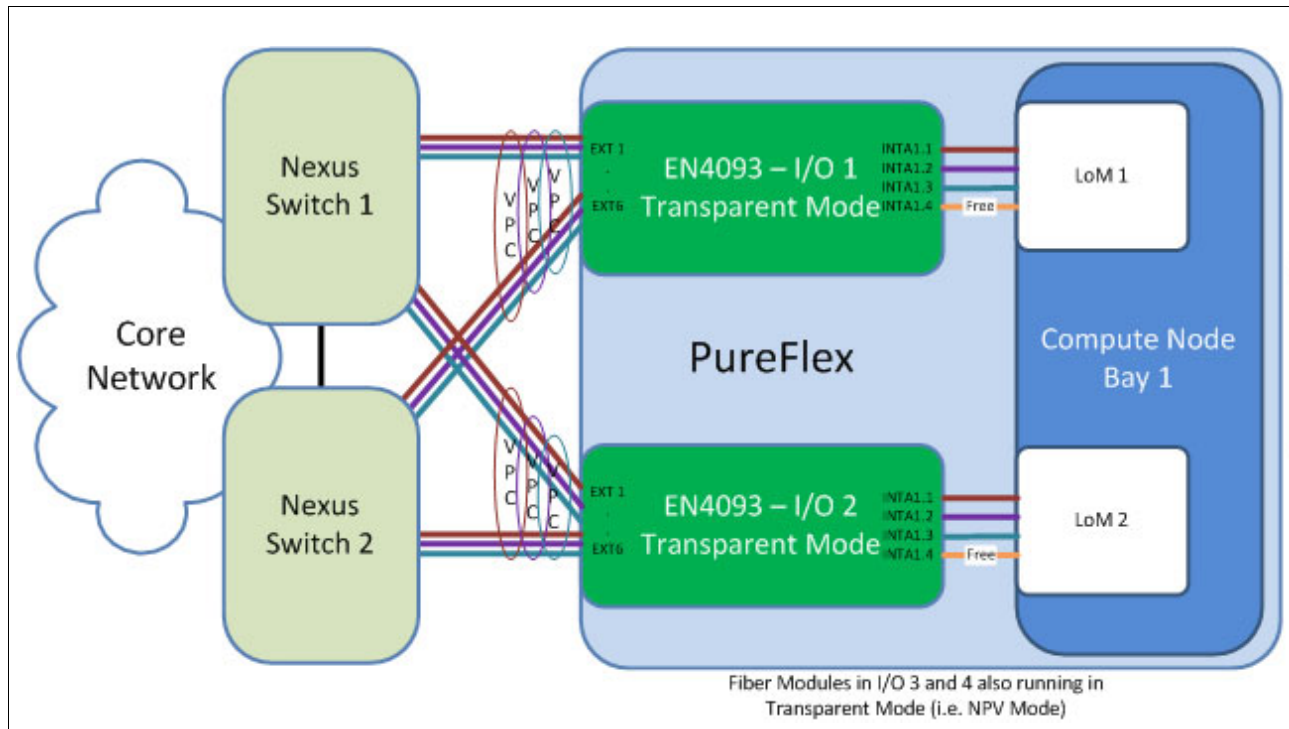


Figure B-6 State government client network diagram

8.9.3 Medical center client

This client requires the following components:

- ▶ Separation of and dedicated Fibre Channel and Ethernet from each compute node and IBM Flex System Enterprise Chassis.
- ▶ Total hardware redundancy including both NIC and application-specific integrated circuit (ASIC) on each compute node using the CN4054 mezzanine adapter.
- ▶ Transparency on both Ethernet (Easy Connect) and Fibre Channel (NPV).

Figure B-7 shows how Easy Connect satisfies all of the state government client's requirements.

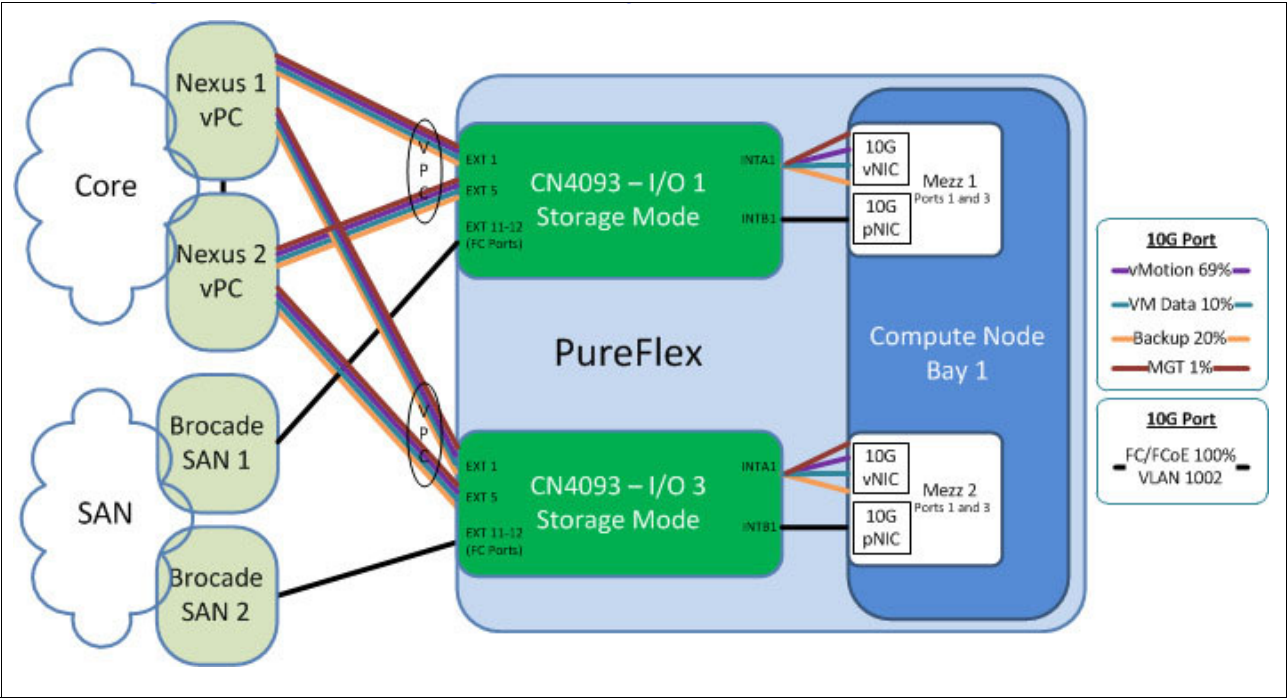


Figure B-7 Medical center client network diagram

8.10 Easy Connect limitations

When configured for any Easy Connect mode, the following stand-alone features are not supported:

- ▶ Basic routing
- ▶ Border Gateway Protocol (BGP)
- ▶ Edge Virtual Bridging/802.1QBG
- ▶ Internet Group Management Protocol (IGMP) Relay, IGMP Querier, IGMP Multicast Snooping, and IGMPv3
- ▶ Stacking
- ▶ OSPF and OSPFv3
- ▶ Policy-based routing
- ▶ Routing Information Protocol (RIP)
- ▶ Routed ports
- ▶ Virtual Router Redundancy Protocol (VRRP)
- ▶ VMReady across the data center

Additionally, if multi-tenant security is a concern within the same IBM Flex System Enterprise Chassis, Easy Connect might not be recommended because each vNIC group is a single broadcast domain.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only:

- ▶ *Moving to IBM PureFlex System x86-to-x86 Migration*, REDP-4887
- ▶ *IBM PureFlex System and IBM Flex System Products and Technology*, SG24-7984
- ▶ *IBM PureFlex System and IBM Flex System Products and Technology*, SG24-7984
- ▶ *Implementing Systems Management of IBM PureFlex System*, SG24-8060
- ▶ *IBM System Networking RackSwitch G8264/G8264T*, TIPS0815

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ IBM PureFlex Systems
<http://www.ibm.com/systems/pureflex/index.html>
- ▶ IBM System Networking
<http://www.ibm.com/systems/networking/>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



IBM Flex System and PureFlex System Network Implementation

(0.5" spine)
0.475" <-> 0.873"
250 <-> 459 pages



IBM Flex System and PureFlex System Network Implementation



Learn how to implement the IBM Flex System and PureFlex System

Understand the design and architecture

Learn troubleshooting techniques

To meet today's complex and ever-changing business demands, you need a solid foundation of server, storage, networking, and software resources that are simple to deploy and can quickly and automatically adapt to changing conditions. You also need access to, and the ability to take advantage of, broad expertise and proven best practices in systems management, applications, hardware maintenance, and more.

IBM PureFlex System, which is a part of the IBM PureSystems family of expert integrated systems, combines advanced IBM hardware and software along with patterns of expertise and integrates them into three optimized configurations that are simple to acquire and deploy so that you can achieve faster time to value.

If you want a preconfigured, preintegrated infrastructure with integrated management and cloud capabilities, factory tuned from IBM with x86 and IBM Power Systems hybrid solution, IBM PureFlex System is the answer.

In this IBM Redbooks publication, which is aimed at system and network administrators, we show the design and architecture, how to configure hosts and switches, maintain, and troubleshoot using the IBM Flex System Ethernet I/O modules (EN2092 1Gb Ethernet Scalable Switch and EN4093R 10Gb Scalable Switch).

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks