Junos Space High Availability Deployment Guide

Release
14.1

Modified: 2016-07-04

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos Space High Availability Deployment Guide
14.1

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- JA1500
- JA2500

Documentation Conventions

Table 1 on page x defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page x defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.

- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

Junos Space High Availability Architecture

Junos Space High Availability Overview

Junos Space is designed as a carrier-grade system that provides a complete fault tolerant solution. The set of topics describing Junos Space high availability (HA) provide an overview of the Junos Space high availability design and implementation, as well as all the steps that are required to deploy a high availability solution, from ordering your appliances and preparing a Junos Space high availability cluster, to final deployment.

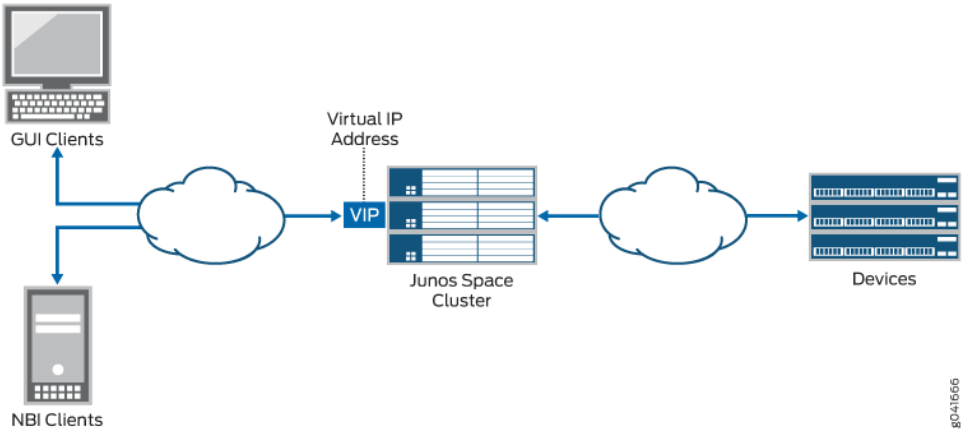
In order to gain an understanding of the Junos Space high availability solution, we recommend that you read all the Junos Space high availability topics. However, if you are primarily interested in setting up high availability, including the prerequisite steps, see the "Configuring the Junos Space Cluster for High Availability Overview" on page 24 topic. If you are interested in high availability monitoring, see "High Availability for Network Monitoring" on page 21. A set of frequently asked questions about Junos Space high availability are also answered in the *Junos® Space Frequently Asked Questions*.

Junos Space Network Management Platform is available in two form factors:

- JA1500 or JA2500 carrier-grade hardware appliance
- Virtual appliance for VMware ESX server environment

Both the Junos Space hardware appliance and virtual appliance use the same software build with identical features to provide the complete package including OS, databases, load balancers and JBoss engines. You can cluster multiple appliances together to form a Junos Space cluster, as shown in Figure 1 on page 4.

Figure 1: Deployment of Junos Space Cluster



A Junos Space fabric (cluster) can contain only hardware appliances (JA1500, JA2500, or both), only virtual appliances, or a combination of both hardware and virtual appliances. Each appliance in the cluster is called a *node*. Junos Space cluster architecture also incorporates load balancing across all nodes in the cluster, which becomes the corner stone for providing scalability for a Junos Space deployment.

A Junos Space high availability solution comprises the following key components:

- Junos Space cluster architecture allows multiple Junos Space appliances (hardware or virtual) to be connected together to form a single cluster. All services within the cluster are provided through a single virtual IP address that GUI and Northbound Interface (NBI) clients can use. This architecture provides protection against any single point of failure (SPOF) in the cluster. If any node in the cluster fails, all services will continue to be available, albeit with reduced capacity.

  Three logical clusters are formed within the single physical cluster when Junos Space appliances are connected together. For more information, see "Understanding the Logical Clusters Within a Junos Space Cluster" on page 15.

- The Junos Space Appliance (JA1500 or JA2500) is a carrier-grade hardware appliance designed to ensure hardware-level reliability and incorporates several fault tolerance features to eliminate single point of failure and minimize its downtime. The Junos Space Appliance contributes significantly to the availability of the overall cluster. For more information, see the "High Availability Characteristics of Junos Space Appliances" on page 24 topic.

- The Watchdog service provides process-level high availability. In the event of any software services failure on a Junos Space appliance, the watchdog service will automatically restart the service.

Related Documentation

- Junos Space High Availability Software Architecture Overview on page 5
- Software Components for Junos Space Nodes on page 9
- Understanding the Logical Clusters Within a Junos Space Cluster on page 15
- Understanding High Availability Management of DMI Connections on page 13

Junos Space High Availability Software Architecture Overview

The Junos Space platform is designed to ensure five-nines availability with a clustered, multi-tiered, distributed architecture comprising the following features:

- Standard browser-based Web 2.0 GUI clients and REST/HTTPS-based NBI clients
- Apache Load Balancer as a top-level load balancer
- JBoss Application Server based on J2EE technology to provide application framework
- MySQL database to manage persistent data

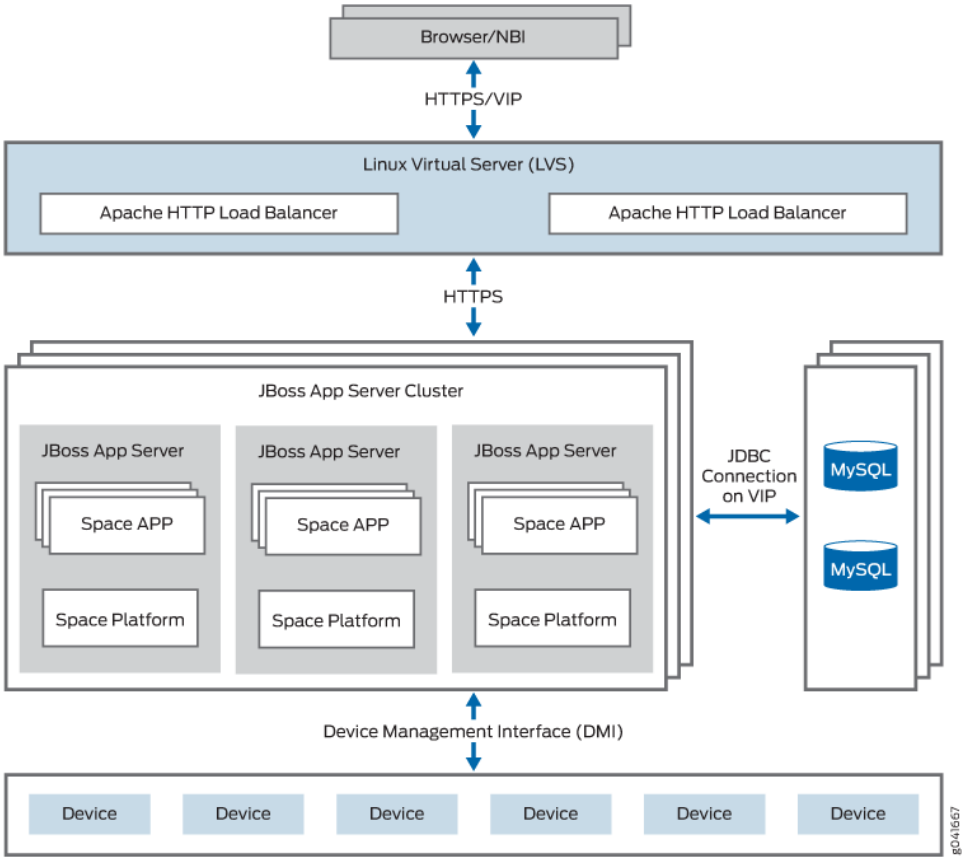
The following sections describe the Junos Space architecture and identify the basic requirements for communication between nodes in a Junos Space cluster:

- Junos Space Software Architecture on page 5
- Load-Balancing Architecture on page 7
- Database Architecture on page 7
- Inter-Node Communication Among Nodes in a Junos Space Cluster on page 8

Junos Space Software Architecture

Figure 2 on page 6 provides a high-level view of the Junos Space software architecture. Junos Space services are accessible to GUI and NBI clients by means of a single virtual IP address for the cluster.

Figure 2: Junos Space Software Architecture



The requests from clients are load-balanced between multiple nodes in the cluster through the Apache HTTP Load Balancer, which is deployed in an active-hot standby configuration on two nodes in the cluster. The load balancer on the node which owns the virtual IP (VIP) address acts as the active instance. If the node which currently owns the VIP address goes down, the other node in the Linux Virtual Server (LVS) cluster will detect this failure and automatically take over the VIP address. The HTTP requests are load-balanced across all active JBoss servers in the cluster using a round-robin algorithm.

Active JBoss servers within the cluster provide the application framework for Junos Space applications, including the following services:

- Hosting the applications and associated business logic
- Application-level load balancing within the cluster
- Application monitoring and automatic recovery
- Cluster node monitoring and automatic recovery
- Database services with direct access to MySQL DB through JDBC
- Hosting Device Mediation Logic

Load-Balancing Architecture

A Junos Space cluster is presented with two kinds of loads:

- Incoming requests from GUI and NBI clients
- Communication with managed devices

Junos Space is designed to load-balance incoming requests across all active nodes in the cluster. Requests from GUI and NBI clients arrive as HTTP requests serviced by the active instance of the Apache HTTP load balancer. The load balancer distributes the requests to all active JBoss servers in the cluster using a round-robin algorithm. Sticky sessions are utilized to ensure that all HTTP requests associated with a specific GUI session are served by the same JBoss server during the lifetime of that session. For the purpose of application-level load balancing, JBoss business logic processes complex requests as a set of sub-jobs, which are distributed across multiple nodes in the cluster. For example, a single request to a four-node Space cluster to resynchronize 100 devices is divided into four sub-jobs that are executed on four different nodes, with each node resynchronizing 25 devices. For a detailed overview of load balancing, see the topic "Understanding the Logical Clusters Within a Junos Space Cluster" on page 15.

To perform device-level load balancing, Junos Space employs logic in the Device Mediation Layer (DML) so that device connections are equally distributed across all active nodes in the cluster. Device-level load balancing is performed during device discovery by comparing the number of device connections served by individual nodes and selecting the least loaded node. If any node goes down, all associated device connections are distributed to the remaining active nodes in the cluster, thus preventing a node outage from affecting device connectivity. For a detailed overview of device connectivity management, see the topic "Understanding High Availability Management of DMI Connections" on page 13.

Database Architecture

MySQL Enterprise Edition is used to provide database services for managing persistent data for both platform and applications. MySQL DB servers are running on two nodes in the cluster in active-standby configuration. Database transactions are replicated between the two MySQL servers in near real time. For information about the MySQL cluster that is formed within each Junos Space cluster, see "Understanding the Logical Clusters Within a Junos Space Cluster" on page 15.

Junos Space platform also incorporates network monitoring for fault and performance management, which uses the PostgreSQL relational database service for storing fault and performance related data. The PostgreSQL server runs on two nodes in the Space cluster in active-active configuration with real-time replication to ensure that fault and performance data continues to be available even if one of these nodes fail. For more information, see "High Availability for Network Monitoring" on page 21.

Inter-Node Communication Among Nodes in a Junos Space Cluster

In order to facilitate seamless communication between the nodes in a Space cluster and to achieve optimum performance of the cluster, you need to ensure the following:

- All nodes in a Junos Space cluster are configured with IP addresses inside the same subnet. This is important for the VIP switchover mechanism to work correctly.
- All nodes in a Space cluster are connected by means of a 1-Gbps or 100-Mbps local network with negligible latency.
- JBoss servers within a Junos Space cluster communicate by means of a UDP multicast to form logical clusters.

NOTE: **UDP multicast traffic must be allowed within the nodes in the cluster, which also means that you should disable IGMP snooping on the switches that interconnect the cluster or configure them explicitly to allow UDP multicast between the nodes.**

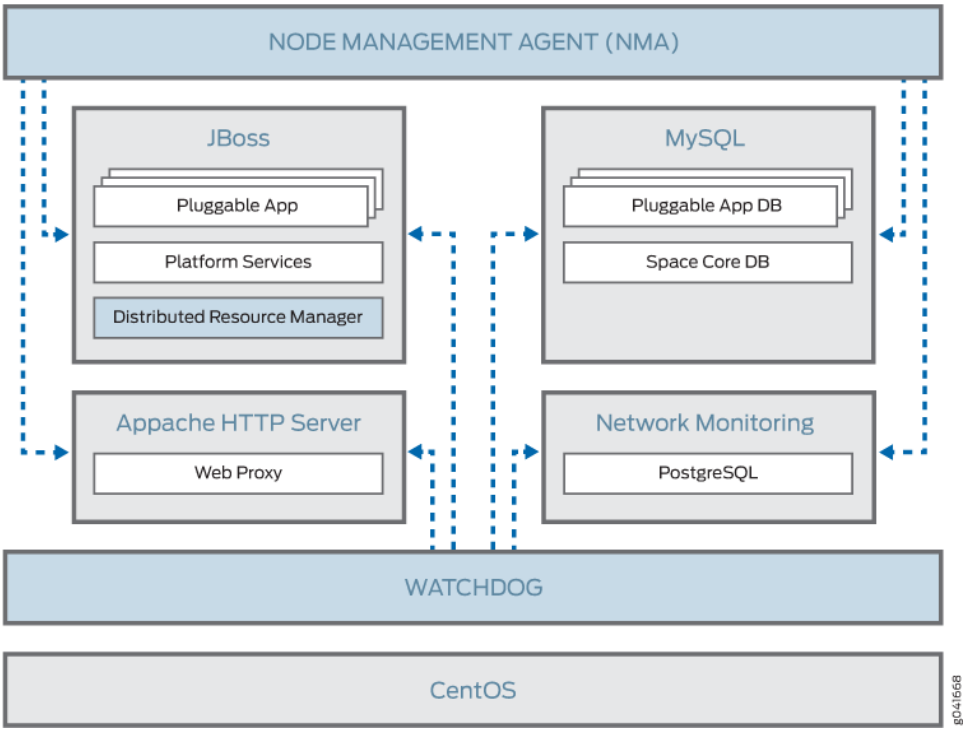
Related Documentation

- Junos Space High Availability Overview on page 3
- Software Components for Junos Space Nodes on page 9
- Understanding the Logical Clusters Within a Junos Space Cluster on page 15

Software Components for Junos Space Nodes

The Junos Space Appliance (JA1500 or JA2500) and Junos Space virtual appliance both run the same software stack, as shown in Figure 3 on page 9.

Figure 3: Software Stack on a Junos Space Appliance



The Junos Space software architecture is based on a combination of the following mature and proven software components:

- CentOS 5.9 distribution is used as the underlying OS of the appliance. CentOS distribution is binary compatible with Red Hat Enterprise Linux (RHEL). Services that are required for Junos Space are leveraged from this distribution, with all other services removed. Junos Space administrators do not need to directly access the Linux components because all operations, administration, and management (OAM) of the platform is performed from the Junos Space user interface or CLI. At the same time, it is important to note that the underlying operating system is an industry-standard distribution with a strong heritage of reliability and security.
- The MySQL Enterprise Edition 5.5 relational database service provides persistent storage for the Junos Space Network Management Platform and all hosted applications. A common database instance stores all persistent data that the Network Management Platform requires. As shown in the preceding illustration, each pluggable application that is installed on the platform has its own unique database instance. All database instances are contained within a single MySQL server, which runs on two nodes in the cluster to form an active-standby cluster. The remaining the nodes in the cluster do not run a MySQL server.

- JBoss 7.1 Application Server is the container that hosts the presentation layer, business logic layer, and data access layer of Junos Space platform as well as the hosted applications. One JBoss server runs on each node in the cluster and they all work together as a single load-sharing cluster.
- Apache HTTP Server (version 2.2.21) is the front-end load balancer for all requests coming from GUI and NBI clients. This server runs on two nodes in the cluster which together form an active-standby cluster.
- Network monitoring services are provided using OpenNMS, which is an award winning, enterprise-grade network monitoring platform developed under the open source model. OpenNMS is integrated into the Junos Space Network Management Platform **Network Monitoring** workspace and provides fault monitoring and performance monitoring features. Junos Space uses PostgreSQL as the relational database server for persisting fault and performance data.

The following software components or services also play a significant role in the overall management of a Junos Space cluster:

- Distributed Resource Manager (DRM)—DRM is deployed as a service inside the JBoss application server, just like all other services provided by Network Management Platform and the hosted applications. You can think of DRM as the server-side component that you interact with when you navigate to the **Network Management Platform > Administration > Fabric** workspace in the Junos Space user interface. DRM works together with the Node Management Agent to fulfill the following responsibilities:
  - Managing the Junos Space cluster—DRM implements the business logic for adding and removing nodes in the cluster and monitors the overall health of the cluster.
  - Managing the logical clusters in the cluster—The logical clusters within the physical cluster formed by the Junos Space nodes include the Apache Load Balancer cluster, JBoss cluster, and Database cluster. DRM implements the business logic to add and remove nodes in these logical clusters and monitors their status. The logical clusters are described in detail in "Understanding the Logical Clusters Within a Junos Space Cluster" on page 15.
- Node Management Agent (NMA)—NMA runs on each node in the cluster and is deployed as a set of CGI scripts run by an Apache HTTP daemon. NMA has the following responsibilities:
  - Monitor system resource usage on the node and the health of various services running on the node.
  - Start and stop services on the node based on requests from DRM.
  - Manage the configuration files for various services running on the node.
  - Manage installation, uninstallation, and upgrades of pluggable applications as well as upgrade of the Network Management Platform software on the node.
- Watchdog—The watchdog service (jmp-watchdog) runs on each node in the cluster to ensure that required services on the node are running. Every second, the watchdog checks that the required services are running and if the watchdog detects that a service is down, it restarts the service.

Related Documentation

- Junos Space High Availability Overview on page 3
- Junos Space High Availability Software Architecture Overview on page 5
- Understanding the Logical Clusters Within a Junos Space Cluster on page 15

CHAPTER 2

Junos Space High Availability Management

Understanding High Availability Management of DMI Connections

Junos Space maintains a persistent device management interface (DMI) connection with each managed device and supports the following types of DMI connections:

- Space-initiated (default)—A TCP connection from a JBoss server process on a node to the SSH port (22 by default) on the device.
- Device-initiated—A TCP connection from the device to port 7804 on a JBoss server process on a node.

To load balance DMI connections, all connections are distributed across all the nodes in a Junos Space cluster. A device keepalive monitor sends a heartbeat message to devices every 40 seconds. If there is no reply for 15 minutes, the device keepalive monitor marks the connection status of the device as Down.

A device connection monitor scans the connection status of all devices with space-initiated connections. If the monitor detects that the connection status of a device is Down, it attempts to reconnect to the device. If this first attempt fails, a second attempt is made after 30 minutes. Because each reconnect attempt is performed from a node in the cluster that is the least loaded in terms of the number of devices managed, the device might get reconnected from a different node in the cluster after a connection failure.

When devices are discovered using device-initiated connection mode, the device management IP address of all nodes in the Junos Space cluster gets configured in the outbound SSH stanza on the device. The device will keep trying to connect to one of these IP addresses until one succeeds. The device is responsible for detecting any failures on the connection and for reconnecting to another node in the cluster. For more information, see the *Junos XML Management Protocol Guide*.

If a JBoss server process crashes or is stopped, or if the node running the process is shut down, all the DMI connections that it maintains are migrated to another node in the cluster. When this JBoss server comes up, these DMI connections are not automatically migrated back to the JBoss server because it is available for any new devices that are

being discovered. At present, there is no way to migrate DMI connections back to this original JBoss server, which can result in poor load balancing of DMI connections if there are not many new devices to be discovered.

Related Documentation

- Disaster Recovery Solution and Connectivity Requirements on page 44
- Understanding High Availability Nodes in a Cluster on page 20

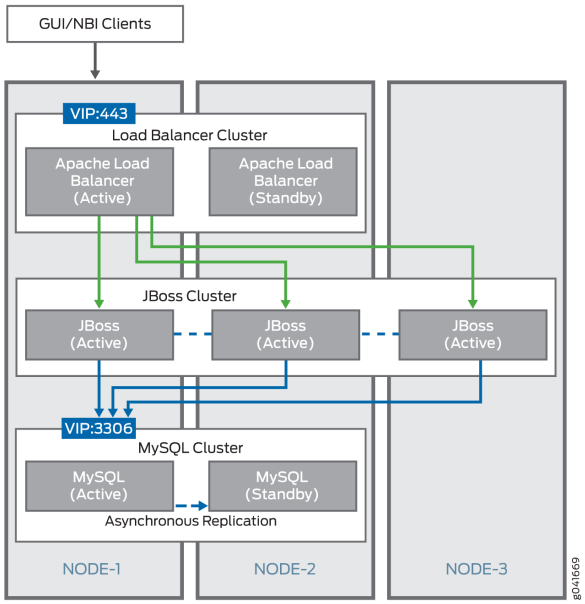
CHAPTER 3

Junos Space Clusters

Understanding the Logical Clusters Within a Junos Space Cluster

You can connect multiple Junos Space appliances (hardware or virtual) together to form a Junos Space cluster. Figure 4 on page 16 shows the logical clusters (Apache Load Balancer cluster, the JBoss cluster, and MySQL cluster) that are formed within each Junos Space cluster.

Figure 4: Junos Space Logical Clusters



- Apache Load-Balancer Cluster on page 16
- JBoss Cluster on page 17
- MySQL Cluster on page 17

Apache Load-Balancer Cluster

The Apache HTTP server, with the mod_proxy load-balancer module enabled, runs on two nodes in the cluster at any given time. These servers form an active-standby logical cluster. They both listen on the TCP port 443 for HTTP requests from GUI and NBI clients. All clients use the virtual IP (VIP) address of the cluster to access its services. At any time, the VIP address is owned by only one node in the cluster. Hence the Apache HTTP server on the node that owns the VIP address receives all HTTP requests from GUI and NBI clients and it acts as the active load-balancer server, and the other server acts as the standby. It uses a round-robin load-balancing algorithm to distribute requests to JBoss servers running on all nodes in the cluster. The load balancer also employs session-stickiness to ensure that all HTTP requests from a user session get sent to the same node in the cluster. To achieve this, the server sets a cookie named JSESSIONID. The value of this cookie identifies the specific node in the cluster that serves requests that belong to this user session. All additional requests will contain this cookie and the load balancer will forward the request to the JBoss server that runs on the node that this cookie identifies.

If the Apache HTTP server on a node goes down, the server is automatically restarted by the watchdog service on that node. If this node owns the VIP address, then the GUI and NBI clients might experience a brief service outage until the Apache HTTP server is restarted. However, this outage lasts only a few seconds (typically 2 seconds) and is hardly noticed by the clients. On the other hand, if the Apache HTTP server goes down on the node that does not currently own the VIP address, no side effects are noticed by

any clients or any other components. The watchdog service restarts the server and it comes back up in about 2 seconds.

JBoss Cluster

The JBoss application server runs on all nodes in the Junos Space cluster. The nodes form a single all-active logical cluster and the load-balancer server (described previously) distributes the load across all the nodes. Even if one or more of the JBoss servers in the cluster fails, the application logic still continues to be accessible from the surviving nodes. Jboss servers on all nodes are started with the same configuration, and use UDP multicast to detect each other and form the single cluster. JBoss also uses UDP multicast for session replication and caching services across all the nodes.

When the JBoss server on a node goes down, other nodes in the JBoss cluster will detect this change and automatically re-configure themselves to remove the failed node from the cluster. The time taken by other cluster members to detect a failed JBoss server depends on whether the JBoss server process crashed abnormally or is non-responsive. In the former case, cluster members will detect the failure immediately (around 2 seconds) because their TCP connections to the crashed JBoss server are closed by the operating system. In the latter case, cluster members will detect the failure in about 52 seconds. If a JBoss server crashes, it will be restarted automatically by the watchdog service (jmp-watchdog) running on the node. When the JBoss server comes back up, it will be automatically discovered by other cluster members and added to the cluster. The JBoss server will then synchronize its cache from the other nodes in the cluster. The typical restart time for JBoss is 2 to 5 minutes, but it can take more time depending on the number of apps installed, the number of devices being managed, the number of DMI schema versions installed, and so forth.

One JBoss server in the cluster will always act as the master of the cluster. The main purpose of the master designation is to host services that are deployed as cluster-wide singletons (HA singletons); for example, services that must be deployed on only one server in the cluster at any time. Junos Space uses a several services of this type, including the Job Poller service, which provides a single timer for scheduling jobs across the cluster, and the Distributed Resource Manager (DRM) service, which monitors and manages the nodes in the cluster. These services are deployed only on the JBoss server that is designated as the master.

NOTE: **This does not mean that the master does not host other services. Non-cluster singleton services are also hosted on the master server. Junos Space is configured such that the first JBoss server that comes up in the cluster becomes the master. If the master server goes down, other members in the JBoss cluster detect this and elect a new master.**

MySQL Cluster

The MySQL server runs on two nodes in the Junos Space cluster at any given time. These nodes form a logical active-standby cluster and both nodes listen on TCP port 3306 for database requests from JBoss servers. JBoss servers are configured to use the Virtual IP (VIP) address of the cluster to access database services. At any time, the VIP address is

owned by only one node in the cluster. Thus, the MySQL server on the node that owns the VIP address receives all database requests from JBoss and this server acts as the active database server, and the other server acts as the standby.

MySQL servers on each of the nodes are configured with unique server IDs. The master/slave relationship is also configured symmetrically on the nodes so that the server on Node-1 is configured with Node-2 as the master; and the server on Node-2 is configured with Node-1 as the master. Thus both nodes are capable of acting as a slave to the other, and the server running on the node which owns the VIP address acts as the master at any time, which ensures that the master-slave relationship switches dynamically as the VIP ownership switches from one node to the other. All transactions committed on the active (master) server are replicated to the standby (slave) server in near real time, by means of the asynchronous replication solution [2] provided by MySQL, which is based on the binary logging mechanism. The MySQL server operating as the master (the source of the database changes) writes updates and changes as “events” to the binary log. The information in the binary log is stored in different logging formats according to the database changes that are recorded. The slave server is configured to read the binary log from the master and to execute all the events in the binary log on the slave's local database.

If the MySQL server on a node goes down, the server is restarted automatically by the watchdog service on that node. Once restarted, the MySQL server should come up within 20 to 60 seconds. If this node owns the VIP address, JBoss might experience a brief database outage for this 20 to 60 second duration. Any requests which require database access will fail during this period. On the other hand, if the MySQL server goes down on the node that does not currently own the VIP address, there are no side effects noticed by JBoss. The watchdog service restarts the server and it comes back up in less than 1 minute. Once the server is back up, it will resynchronize with the master in the background and the resynchronization time will depend on the number of changes that occurred during the outage.

Related Documentation

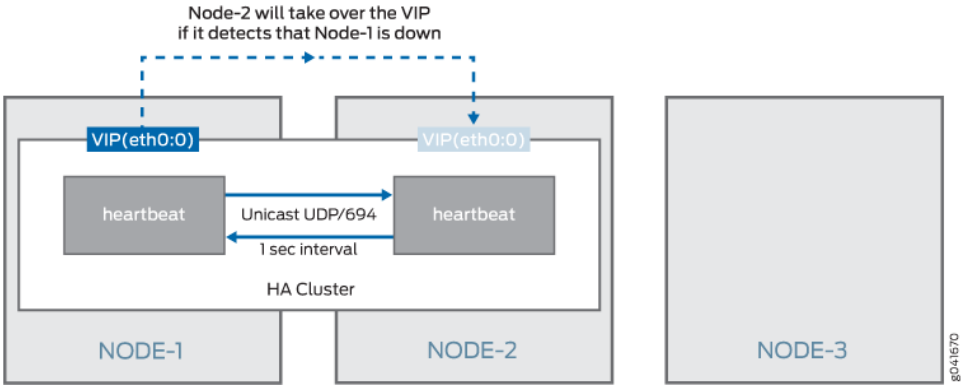
- Understanding Virtual IP Availability Within a Junos Space Cluster on page 19
- Understanding High Availability Nodes in a Cluster on page 20
- Configuring the Junos Space Cluster for High Availability Overview on page 24

Understanding Virtual IP Availability Within a Junos Space Cluster

Junos Space must ensure that the virtual IP (VIP) address is always available on one of the nodes in the cluster. This is essential for the HA solution because if the VIP address becomes unavailable, the entire cluster becomes unavailable to all user interface clients and NBI clients. In addition, none of the JBoss servers can access the database because they are all configured to the use the VIP address to connect to the database server. To protect against this scenario, Junos Space uses the heartbeat service (version 2.1.3 to version 3) provided by the Linux-HA project to ensure that the VIP address is always available on one of the nodes in the cluster. For information about the Linux-HA project, see the Linux HA User Guide.

Figure 5 on page 19 shows the heartbeat service that runs on two nodes in the cluster, which together form a Linux HA cluster.

Figure 5: Heartbeat Service on a Linux High Availability Cluster



The heartbeat service is configured symmetrically on both nodes to send a heartbeat message to the other node at a 1-second interval. Unicast messages to UDP port 694 are used to send the heartbeat messages. If a node misses 10 consecutive heartbeat messages from the other node, it will consider the other node as dead and initiate a failover to take ownership of the protected resource. The protected resource in this case is the VIP address of the cluster. When failover occurs, the virtual IP address is obtained using a method known as IP address takeover (for more information, see IP Address Take Over) whereby the newly activated node configures the VIP address on one of its interfaces (eth0:0 is used in Junos Space for this) and sends gratuitous ARP packets for the VIP address. All hosts on the network should receive these ARP packets and, from this point forward, send subsequent packets for the VIP address to this node. When the node that currently owns the VIP address crashes, an automatic failover of the VIP address to the other node in the cluster occurs in a little more than 10 seconds. When the crashed node comes back up (for example, in the case of a reboot), it joins the HA cluster and acts as the standby node. In other words, an automatic failback of the VIP address does not happen.

NOTE: The 10 seconds that it takes Junos Space to detect a failed node is applicable when the node crashes or becomes nonresponsive. However, in cases where the node is shut down or rebooted, or if the heartbeat service on the node is stopped by the Junos Space administrator, a message is sent to the heartbeat service on the other node and VIP failover occurs almost instantaneously.

Related Documentation

- Understanding the Logical Clusters Within a Junos Space Cluster on page 15
- Understanding High Availability Nodes in a Cluster on page 20
- Configuring the Junos Space Cluster for High Availability Overview on page 24

Understanding High Availability Nodes in a Cluster

A Junos Space cluster must include at least two nodes to achieve high availability (HA). If the cluster includes more than two nodes, the availability of the cluster does not increase, but the amount of load that the cluster can handle increases with each node added to the cluster. So at any given time, only two nodes in the cluster provide HA to the whole cluster. These two nodes alone (referred to as the HA nodes in the cluster) form the Linux HA cluster, the Apache Load Balancer cluster, and the MySQL cluster.

By default, the first two nodes added to the cluster function as the HA nodes. In the topic "Understanding the Logical Clusters Within a Junos Space Cluster" on page 15, the example shows that the first two nodes (Node-1 and Node-2) are HA nodes. If you were to delete Node-1 or Node-2 from the **Network Management Platform > Administration > Fabric** workspace, the system checks to see if other nodes in the cluster are available to replace the deleted HA node. The system then displays the list of capable nodes (only Node-3 in the example), which you can select. After you confirm the selected node, the Distributed Resource Manager (DRM) service adds the node to the HA cluster by sending requests to the Node Management Agent (NMA) running on the newly selected node. The following actions are initiated on the node added to the HA cluster:

- Apache HTTP server with the mod_proxy load balancer is started on the node and the node is configured with all JBoss nodes as members.
- After copying the database from the other MySQL server in the cluster, MySQL server is started on the node. This server is configured as a slave of the other MySQL server in the cluster and it resynchronizes with the master in the background. The existing MySQL server is also reconfigured to act as a slave of this new server to ensure a symmetric master/slave configuration on both.

Related Documentation

- Understanding the Logical Clusters Within a Junos Space Cluster on page 15
- Configuring the Junos Space Cluster for High Availability Overview on page 24

High Availability for Network Monitoring

The type of Junos Space cluster you create determines how high availability for the network monitoring service functions. A Junos Space fabric without Fault Monitoring and Performance Monitoring (FMPM) nodes uses the two high availability (HA) nodes in the cluster to protect the network monitoring service against node failures. However, when a Junos Space fabric includes one or more FMPM nodes, network monitoring functionality is disabled on the Junos Space nodes and enabled on the FMPM nodes.

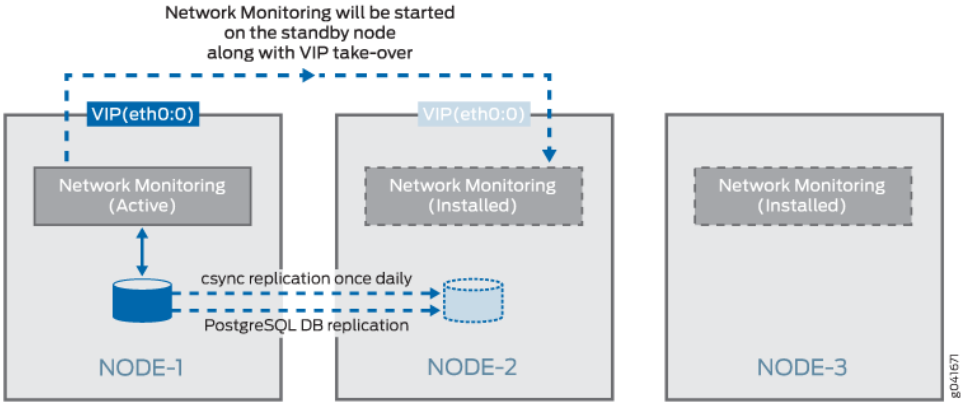
This topic includes the following sections:

High-Availability Fabric without FMPM Nodes

When a Junos Space fabric does not include FMPM nodes, the Junos Space cluster employs a hot-standby solution that uses the two high availability (HA) nodes in the cluster to protect the network monitoring service against node failures.

Figure 6 on page 21 shows how network monitoring runs on two HA nodes in the cluster to protect the service in the event of node failure.

Figure 6: Linux High Availability Cluster



The network monitoring service is automatically installed on all nodes in the cluster. However, at any time, the network monitoring service runs only on the node that currently owns the virtual IP (VIP) address, and the service is responsible for all fault management and performance management functionality for the entire cluster. Network monitoring uses PostgreSQL 9.1 database for its storage needs. As Figure 6 on page 21 shows, real-time streaming replication with continuous archiving is set up between the two HA nodes (Node-1 and Node-2 in the cluster), which ensures that the network monitoring database on the standby node is continuously in sync with the network monitoring database on the active node. In addition, a cron job runs on the active node once a day at midnight to synchronize the network monitoring file system to the standby node, which ensures that all back-end configuration files that network monitoring uses are also synchronized between the two HA nodes.

When a VIP failover to the standby node occurs, network monitoring is automatically started on the node. The network monitoring service takes approximately 3 to 5 minutes to complete its initialization before it performs all fault monitoring and performance monitoring functionality for the cluster. Consequently, Junos Space users can expect a network monitoring outage to last approximately 3 to 5 minutes.

The watchdog service on the two HA nodes is responsible for ensuring that the network monitoring service is running on the HA node that owns the virtual IP address and is not running on the other (standby) HA node. As already noted, the watchdog service checks the status of all services on the node every second. If the watchdog service detects that the node owns the VIP address but does not run the network monitoring service, the watchdog service starts the network monitoring service and creates the cron job to synchronize fault management and performance management data to the other node. If the watchdog service detects that the node does not own the VIP address but is running the network monitoring service, the watchdog service shuts down the service and removes the cron job entry for data synchronization.

High-Availability Fabric with FMPM Nodes

If you manage a large or complex network, you might want to dedicate all your performance and network monitoring functionality to a special node called the Fault Monitoring and Performance Monitoring (FMPM) node. When you create a Junos Space fabric with one or more FMPM nodes, network monitoring functionality is disabled on all the Junos Space nodes and enabled on the FMPM nodes. When the first FMPM node is added to the fabric, network monitoring functionality is enabled on this node and the PostgreSQL 9.1 database runs on this node.

When you add a second FMPM node to the fabric, the first FMPM node functions as the primary node, and the second FMPM node functions as the standby node. The network monitoring service is automatically installed on both FMPM nodes in the FMPM team. However, at any time, the network monitoring service runs only on the FMPM node that currently owns the VIP address, and the service is responsible for all fault management (FM) and performance management (PM) functionality for the FMPM team. Network monitoring uses PostgreSQL 9.1 database for its storage needs.

Real-time streaming replication with continuous archiving is set up between the two FMPM nodes in the team, which ensures that the network monitoring database on the standby node is continuously in sync with the network monitoring database on the active node. In addition, a cron job runs on the active FMPM node once a day at midnight to synchronize the network monitoring file system to the standby FMPM node, which ensures that all back-end configuration files that network monitoring uses are also synchronized between the two FMPM nodes. When a VIP failover to the standby FMPM node occurs, network monitoring is automatically started on the second FMPM node. The network monitoring service takes approximately 3 to 5 minutes to complete its initialization before it performs all FM and PM functionality for the FMPM team. Consequently, Junos Space users can expect a network monitoring outage to last approximately 3 to 5 minutes.

The watchdog service on the two nodes is responsible for ensuring that the network monitoring service is running on the FMPM node which owns the virtual IP address and is not running on the other (standby) FMPM node. As already noted, the watchdog service checks the status of all services on the active FMPM node every second. If the watchdog

service detects that the FMPM node owns the VIP address but does not run the network monitoring service, the watchdog service starts the network monitoring service and creates the cron job to synchronize fault management and performance management data to the other node. If the watchdog service detects that the FMPM node does not own the VIP address but is running the network monitoring service, the watchdog service shuts down the service and removes the cron job entry for data synchronization.

Related Documentation

- Understanding How Devices Are Configured to Send SNMP Traps to Junos Space on page 23
- Understanding High Availability Nodes in a Cluster on page 20
- Configuring the Junos Space Cluster for High Availability Overview on page 24

Understanding How Devices Are Configured to Send SNMP Traps to Junos Space

Devices discovered in Junos Space are automatically configured to send SNMP traps to Junos Space.

The trap destination IP address that is configured on devices depends on whether a separate device management interface (eth3) is used on the node on which network monitoring is currently running. If the node uses the eth3 interface for device management, then the discovered devices are configured with the eth3 IP address. Otherwise, the discovered devices are configured with the virtual IP (VIP) address of the Junos Space cluster. If the VIP is configured as the trap destination, you do not need to reconfigure the trap destination on managed devices after a VIP failover because network monitoring will be started automatically on the node that currently owns the VIP and the node will start receiving the traps. However, if the eth3 IP address is configured as the trap destination, you must reconfigure all devices when a VIP failover occurs. This will be done automatically as part of the startup process of network monitoring on the second HA node. When network monitoring comes up on the new node, the trap destination on all managed devices will be automatically reconfigured to be the eth3 IP address for this node.

NOTE: **Automatic reconfiguration is not possible for devices whose connection with Junos Space is down at the time of the network monitoring failover. If there are any such devices, network monitoring stops receiving traps from these devices after the failover, and you will need to manually change the trap destination on these devices to the eth3 IP address of the node where network monitoring is currently running.**

Related Documentation

- High Availability for Network Monitoring on page 21
- Understanding High Availability Nodes in a Cluster on page 20

High Availability Characteristics of Junos Space Appliances

Junos Space Appliances (JA1500 and JA2500) incorporate the following fault tolerance features that prevent or minimize their downtime and contribute significantly to the availability of the overall cluster:

- Hot-swappable hard disk drives managed by a RAID controller
  - The JA1500 appliance has three hard drives in RAID 5 configuration and the JA2500 appliance has six hard drives in a RAID 10 configuration.
  - The hot swappable hard drives on Junos Space appliances are externally accessible in field replaceable trays, providing component high availability. You can remove and replace a hard disk without powering off the appliance or disrupting any functions performed by the appliance.
  - The RAID controller manages the hard disk drives and presents them as logical units.
- Option to install a redundant power supply module—Junos Space Appliances are shipped with a single AC power supply. However, you can install an additional power supply module that serves as a redundant power supply if one power supply module fails. If you install a second power supply module, ensure that you plug in each power supply module into a separate power circuit.

  When an appliance has an additional redundant, functioning power supply module that is plugged into a separate power circuit, the power supply modules are hot-swappable.
- Two cooling fans—Two externally accessible and hot-swappable cooling fans provide the required airflow and cooling for the appliance.

For detailed information about Junos Space Appliances, refer to the *Hardware Documentation* section of the *Junos Space Network Management Platform* page.

Related Documentation

- Junos Space High Availability Overview on page 3

Configuring the Junos Space Cluster for High Availability Overview

This topic provides an overview of the key steps required to configure a Junos Space cluster as a carrier-grade system with all high-availability capabilities enabled.

- Requirements on page 25
- Preparation on page 25
- Configuring the First Node in the Cluster on page 27
- Adding a Second Node to the Cluster on page 27
- Adding Additional Nodes to a Cluster on page 28
- Configuring FMPM Nodes on page 28
- Removing Nodes from a Cluster on page 28

Requirements

You can choose either Junos Space Appliances (JA1500 or JA2500) or Virtual Appliances for setting up a Junos Space cluster.

For a cluster of Virtual Appliances, the following recommendations apply for the underlying virtualization infrastructure on which the appliances are deployed:

- Use VMware ESX server 4.0 or later or VMware ESXi server 4.0, 5.0, 5.1, or 5.5 that can support a virtual machine.
- Deploy the two Junos Space Virtual Appliances (JSVA) on two separate servers.
- Each server must be able to dedicate 4 vCPUs or 2.66 GHz or more, 32 GB RAM, and sufficient hard disk for the Junos Space Virtual Appliance that it hosts.
- The servers should have similar fault tolerance features as the Junos Space appliance: dual redundant power supplies connected to two separate power circuits, RAID array of hard disks for storage, and hot-swappable fans.

NOTE: For more information on the requirements for the virtual appliance, refer to the *Deploying a Junos Space Virtual Appliance* topic in the *Virtual Appliance* documentation.

If you choose Junos Space appliances, you need to choose two instances of the corresponding SKUs for the appliance that you are using. In addition, order a second power supply module for each appliance in order to provide the redundant power supply module for each appliance.

Preparation

We recommend you use the following guidelines as you prepare a Junos Space cluster for high availability:

- The Junos Space cluster architecture allows you to dedicate one or two nodes solely for fault monitoring and performance monitoring functions. These are known as Fault Monitoring and Performance Monitoring (FMPM) nodes and are recommended when managing complex networks with a large number of devices and interfaces to be monitored. The advantage of this architecture is that fault and performance monitoring functions are localized within the FMPM nodes and the rest of the Junos Space nodes are freed up for other functions. One of the first decisions that you must make is whether to use FMPM nodes in your Junos Space cluster. If you choose to deploy FMPM nodes, we recommended that you have two of them so that the fault monitoring and performance monitoring services also have high availability. Currently, load balancing is not implemented across multiple FMPM nodes, so there is no need to have more than two FMPM nodes in a cluster.
- Junos Space appliance (hardware or virtual) utilizes two Ethernet interfaces: eth0 and eth3. The eth0 interface is used for all inter-node communication within the cluster and also for communication between GUI and NBI clients and the cluster. The

eth3 interface can be configured as the device management interface, in which case, all communication between the cluster and the managed devices occur over this interface. If the eth3 interface is not configured, all device communication also takes place over the eth0 interface. So, you must first decide whether or not to use eth3 as the device management interface. If you choose to use eth3, you should use eth3 for all appliances in the same cluster.

- You also must decide on the following networking parameters to be configured on the Junos Space appliances:
  - IP address and subnet mask for the interface “eth0”, the default gateway address, and the address of one or more name servers in the network.
  - IP address and subnet mask for the interface “eth3” if you choose to use a separate device management interface.
  - The virtual IP address to use for the cluster, which should be an address in the same subnet as the IP address assigned to the “eth0” interface.

    If you decide to use an FMPM cluster, you must choose a separate virtual IP address for the FMPM nodes. Please note that the FMPM virtual IP address need not be in the same subnet as the virtual IP address of the Junos Space nodes.
  - NTP server settings from which to synchronize the appliance’s time.
- The IP address that you assign to each Junos Space node in the cluster and the virtual IP address for the cluster must be in the same subnet. This is required for the IP address takeover mechanism to function correctly.

  It is possible to configure the FMPM nodes in a separate subnet.

  NOTE: **Strictly speaking, you can choose to deploy the non HA nodes in a different subnet. However, doing so will cause a problem if one of the HA nodes goes down and you want to promote one of the other nodes as an HA node. So, we recommend that you configure eth0 on all nodes in the same subnet.**
- Because JBoss servers on all the nodes communicate using UDP multicast to form and manage the JBoss cluster, you must ensure that UDP multicast is enabled in the network where you deploy the cluster nodes. You must also disable IGMP snooping on the switches interconnecting the cluster, or configure them explicitly to allow UDP multicast between the nodes.

  NOTE: **FMPM nodes do not participate in the JBoss cluster. Therefore, there is no need to enable UDP multicast between the FMPM nodes and the Junos Space nodes in the cluster.**

Configuring the First Node in the Cluster

After you power on the appliance and connect to its console, Junos Space displays a menu-driven command-line interface (CLI) that you use to specify the initial configuration of the appliance. To complete this initial configuration, you specify the following parameters:

- IP address and subnet mask for the interface “eth0”
- IP address of the default gateway
- IP address of the name server
- IP address and subnet mask for the interface “eth3”, if you choose to configure a cluster as described in the topic “Understanding the Logical Clusters Within a Junos Space Cluster” on page 15.
- Whether this appliance being added to an existing cluster. Choose “n” to indicate that this is the first node in the cluster.
- The virtual IP address that the cluster will use.
- NTP server settings from which to synchronize the appliance’s time.
- Maintenance mode user ID and password.

NOTE: **Make note of the user ID and password that you specify for maintenance mode, as you will need this ID and password to perform Network Management Platform software upgrades and database restoration.**

For detailed step-by-step instructions on configuring the appliance for initial deployment, refer to the Junos Space appliance documentation. After you have completed the initial configuration, all Junos Space services are started on the appliance and you can log in to the Network Management Platform User Interface from the virtual IP address assigned to it. At this stage, you have a single node cluster with no HA, which you can see by navigating to the **Network Management Platform > Administration> Fabric** workspace.

Adding a Second Node to the Cluster

In order to add a second node to the cluster, you must first configure the second appliance using its console. The process is identical to that of the first appliance except that you need to choose “y” when it you are prompted to specify whether this appliance will be added to an existing cluster. Make sure that the IP address you assign to this node is in the same subnet as the first node. You must also ensure its uniformity in using a separate device management interface (eth3). If you chose to use eth3 for the first node, choose the same for all additional nodes in the cluster.

After you configure the second appliance, you can log in to the Network Management Platform user interface of the first node at its virtual IP address to add the node to the cluster from the **Network Management Platform > Administration > Fabric > Add Fabric Node** workspace. To add the node to the cluster, specify the IP address assigned to the

eth0 interface of the new node, assign a name for the new node, and (optionally) schedule the date and time to add the node. The Distributed Resource Manager (DRM) service running on the first node contacts Node Management Agent (NMA) on the new node to make necessary configuration changes and add it to the cluster. The DRM service also ensures that required services are started on this node. After the new node joins the cluster, you can monitor its status from the **Network Management Platform > Administration > Fabric** workspace.

Adding Additional Nodes to a Cluster

The process for adding additional nodes is identical to the process for adding the second node. However, these additional nodes do not participate in any of the HA clusters in the fabric, unless explicitly promoted to that role if another HA node is removed.

Configuring FMPM Nodes

You can configure up to 2 FMPM nodes in a cluster. To configure FMPM nodes:

- For Junos Space appliances, refer to the following topics in the *Hardware Documentation* section of the *Junos Space Network Management Platform* documentation:
  - *Configuring a Junos Space Appliance as a Standalone or Primary FMPM Node*
  - *Configuring a Junos Space Appliance as a Backup or Secondary FMPM Node for High Availability*
- For Junos Space Virtual Appliances, refer to the following topics in the *Junos Space Virtual Appliance* documentation:
  - *Configuring a Junos Space Virtual Appliance as a Standalone or Primary FMPM Node*
  - *Configuring a Junos Space Virtual Appliance as a Backup or Secondary FMPM Node for High Availability*

Removing Nodes from a Cluster

If a node has failed and needs to be replaced, you can easily remove the node from the cluster. Navigate to the **Network Management Platform > Administration > Fabric** workspace, select the node you want to remove, and choose the **Delete Node** action. If the node being deleted is an HA node, the system will check if other nodes in the cluster can be elected as the replacement for the HA node being deleted. The system then shows the list of capable nodes (only Node-3 in this example) and allows you to choose from the available nodes. The process is described in "Understanding High Availability Nodes in a Cluster" on page 20.

Related Documentation

- Disaster Recovery Solution and Connectivity Requirements on page 44

Understanding Normal Operation of Master and Slave Clusters

During the normal operation of the master and slave clusters, you use the virtual IP (VIP) address of the master cluster to access its GUI and API for all network management services. On the master cluster, a cron job is run based on the configured schedule. To

view this job, you can use the **crontab -l** command, which invokes the **backupReal.sh** script at the scheduled time.

```
# crontab -l
0 2 * * *  /opt/jmp-geo/backup/script/backupReal.sh >>
/opt/jmp-geo/backup/backup.log 2>&1
```

This script first attempts to ping the DR slave VIP address. If the ping fails, an e-mail alert is sent to the configured e-mail address and the script then creates a backup of both the MySQL and PostgreSQL databases on the system, along with important files required for the correct functioning of all services. This backup is archived into a **tgz** file in the **/opt/jmp-geo/backup/data** directory. Only the most recent *N* files are kept in this directory as per the configured value of N (default is 3) and older files are purged. You can view a log of all activity tracked by the **backupReal.sh** script in the log file located at **/opt/jmp-geo/backup/backup.log.**

On the slave cluster, a cron job is created to pull backup files from the master cluster. You can view this job using the **crontab -l** command , which, as shown here, invokes the **poll.sh** script at the scheduled time.

```
# crontab -l
0 3 * * * /opt/jmp-geo/restore/script/poll.sh >> /opt/jmp-geo/restore/restore.log 2>&1
```

The **poll.sh** script transfers the most recent backup file from the master cluster using SCP. If the transfer fails, an email alert is sent to the configured email address. The backup files are stored in the **/opt/jmp-geo/restore/data** directory. The script ensures that only the most recent *N* files are kept in this directory and older files are purged. To view a log of all activity by the **poll.sh** script, see the log file **/opt/jmp-geo/restore/restore.log**.

As previously mentioned, TCP port 7804 is blocked on all slave cluster nodes. In addition, all device discovery jobs attempted on the slave cluster are forced to fail, which ensures that you cannot discover or manage any devices on the slave cluster during its normal operation.

Related Documentation

- Understanding High Availability Nodes in a Cluster on page 20
- Disaster Recovery Solution and Connectivity Requirements on page 44

CHAPTER 4

Disaster Recovery

Disaster Recovery Overview

Overview

Junos Space provides a means to recover from disaster, by enabling mirroring of the original Junos Space installation on a cluster of nodes at a geographically remote location. If the main Junos Space site failed due to a disaster such as an earthquake, the other site would take over.

The physical installation is a set of two geographically separate clusters: the DR Master cluster (the main site) and the backup or DR Slave cluster (the remote site). Backups contain:

- Junos Space Network Management Platform and other application databases
- Firewall rules
- SNMP configuration of Junos Space
- Device schema information
- Network monitoring database information
- Real-time performance monitoring information

The disaster recovery (DR) system is entirely driven by back-end scripts. Currently, these scripts must be configured manually.

You perform the following sequence of operations to set a disaster recovery system:

1. Back up the DR Master cluster to the DR Slave cluster. See "Creating the DR Master Cluster" on page 53.
2. If disaster overtakes the original DR Master, stop the DR Slave from pulling the backups from the DR Master. See "Creating the DR Slave Cluster" on page 56.
3. When your original DR Master comes back online, perform a reverse restore operation to convert the DR Master to a DR Slave. See "Performing a Reverse Restore Operation" on page 61

Prerequisites

The requirements for recovering your Junos Space installation from a disaster are as follows:

- The DR Master cluster at the primary site (which can be a single node or multiple nodes) and the DR Slave cluster at the remote site (a single node or multiple nodes) must be set up in exactly the same way, with all the same applications, device adapters, and so on.
- When a new node is added to the cluster, the backup and restore scripts must be rerun to update the configuration.
- Both clusters should be configured through the graphical user interface (GUI) with SMTP server information (see *Managing SMTP Servers*). This configuration enables both the DR Master and the DR Slave clusters to notify you by e-mail if the replications fail.

NOTE: We recommend that the e-mail server information is the same on both the DR Master and the DR Slave clusters to avoid the following situation:

If the DR Master is configured with e-mail server 1 and the DR Slave is configured with e-mail server 2, when restoring the database, e-mail server 2 is removed, and only e-mail server 1 remains.

- Both ICMP and SCP must be enabled between the DR Master and DR Slave clusters.
- Backup and restore operations cannot be performed on the same server.
- Backup configuration and restore configuration operations should be performed only on the VIP node of respective clusters. If a VIP switchover occurs, you need to rerun the backup or restore operation (depending on the role) on the new VIP node.

Related Documentation

- Creating the DR Master Cluster on page 53
- Creating the DR Slave Cluster on page 56
- Performing a Reverse Restore Operation on page 61

Understanding Failover Scenarios

Table 3 on page 33 identifies possible failure scenarios, describes how they are detected, provides recovery actions, and if applicable, explains any impact to the system as a result of the failure.

Table 3: Failover Scenarios

Scenario	Detection	Recovery	Impact

Table 3: Failover Scenarios *(continued)*

1) Active VIP node crashes.	The heartbeat service running on the standby VIP node detects the crash within 10 seconds as it will not receive any heartbeat messages from its peer. The JBoss clustering mechanism will enable JBoss servers on the other nodes to detect that the JBoss server on the failed node is unresponsive, in about 52 seconds.	The standby node immediately takes over the VIP address. Device connections served by the failed node will be migrated to the remaining nodes in the cluster. This process will start about 1 minute after the JBoss cluster members detect that the JBoss server on the failed node is down. The time it takes to complete will depend on the number of device connections to be migrated, the load on the remaining nodes, and so on. It is expected to complete within a couple of minutes. After the VIP address is taken over by the standby node, network monitoring service is started on the standby node. It takes around 3 to 5 minutes for network monitoring service to complete its initialization. It might take more time depending on the size of FM and PM data that is being maintained.	The VIP address will be unavailable for about 10 seconds until it is taken over by the standby node. GUI/API client access during this brief period will encounter transient errors. Also any SNMP traps sent by the devices to the VIP address during this interval will be lost. Device connectivity will be down for a couple of minutes for the devices whose connections were being served by the JBoss server on the failed node. Any jobs that were in progress on the failed node will be marked as failed with a reason stating that the node was down. Users will experience an outage of network monitoring functionality for about 3 to 5 minutes while the network monitoring service is being initialized on the standby node.
2) Standby VIP node crashes	JBoss clustering mechanism will enable JBoss servers on the other nodes to detect that the JBoss server on the failed node is non-responsive in about 52 seconds.	Device connections served by the failed node are migrated to the remaining nodes in the cluster. This process starts about 1 minute after the JBoss cluster members detect that the JBoss server on the failed node is down. The process completion time depends on the number of device connections to be migrated, the load on the remaining nodes, etc, but should complete within a few minutes.	Device connectivity will be down for a couple of minutes for the devices whose connections were being served by the JBoss server on the failed node. Any jobs that were in-progress on the failed node are marked as failed with the reason that the node was down.
3) eth0 on the active VIP node goes down	See the detection case described in scenario 1 above.	See the recovery case in scenario 1 above.	Refer to the impacts case in scenario 1 above.
4) eth0 on the standby VIP node goes down	See the detection case described in scenario 2 above.	See the recovery case in scenario 2 above.	Refer to the impacts case in scenario 2 above.

Table 3: Failover Scenarios *(continued)*

5) A non-VIP node crashes	The JBoss clustering mechanism enables JBoss servers on the other nodes to detect that the JBoss server on the failed node is non-responsive in about 52 seconds.	Device connections served by the failed node will be migrated to the remaining nodes in the cluster. This process will start about one minute after the JBoss cluster members detect that the JBoss server on the failed node is down. The time it takes to complete depends on the number of device connections to be migrated, the load on the remaining nodes, and so on. This should complete within a couple of minutes.	Device connectivity will be down for a couple of minutes for the devices whose connections were served by the JBoss server on the failed node. Any jobs that were in-progress on the failed node are marked as failed with a reason stating that the node was down.
6) eth0 on a non-VIP node goes down	See the detection case described in scenario 2 above.	See the recovery case in scenario 2 above.	Refer to the impacts case in scenario 2 above.
7) eth3 on a non-VIP node goes down	The device keepalive monitor detects that all device connections served by this node are down in 15 minutes, and marks the connection status of these devices as Down.	For Junos Space initiated connections, Junos Space will attempt to reconnect to these devices. Each attempt is made from the cluster node that is determined to be the least loaded in terms of the number of devices it manages. If other nodes in the cluster are significantly less loaded than this node according to this load-balancing check, reconnection attempts are made from those nodes and they will succeed. In this case, connectivity for these devices comes back up in a couple of minutes. If this node happens to be the least loaded, then all reconnection attempts are made from this node and these attempts will continue failing as long as eth3 remains down. In the case of device initiated connections, the device detects a connection failure in about 15 minutes, and then reconnects to another node in the cluster in the next few seconds.	Device connectivity will be down for the devices whose connections were being served by this node. Connectivity might be down for 15 minutes (best case) or until eth3 is brought back up (worst case). Also, the outage time might vary from device to device depending on which node is chosen to attempt a reconnection for that device. In the case of device initiated connections, the outage will last for a little more than 15 minutes.

Table 3: Failover Scenarios *(continued)*

8) eth3 on the active VIP node goes down.	See the detection case described in scenario 7 above.	Same as recovery case in scenario 7 above.	In addition to the device connectivity impacts described in scenario 7 above, the network monitoring service is also impacted because it runs only on the VIP node. The service will not receive any SNMP traps from any managed device because all devices are configured with the eth3 IP address of the VIP node as the trap destination. Also, all performance and fault monitoring of all managed devices will fail until eth3 is brought back up.
9) JBoss server on a node goes down	When the JBoss server on a node goes down, other nodes in the JBoss cluster will detect this immediately (around 2 seconds) as it results in their TCP connections to the crashed JBoss server getting closed by the OS.	Device connections served by the crashed JBoss server will be migrated to the other nodes in the cluster. This process will start about 1 minute after the JBoss cluster members detect that the JBoss server on the failed node is down. The time it takes to complete will depend on the number of device connections to be migrated, the load on the remaining nodes, and so on. It is expected to complete within a couple of minutes. The watchdog service (jmp watchdog) running on the node will detect that JBoss server is down and restarts it automatically. When the JBoss server comes back up, it will be automatically discovered by other cluster members and get added to the cluster. It will then synchronize its cache from the other nodes in the cluster. The typical restart time for JBoss is 2 to 5 minutes. However, it can take more time depending on the number of apps installed, the number of devices being managed, the number of DMI schema versions installed, and so on.	Device connectivity will be down for a couple of minutes for the devices whose connections were being served by the JBoss server that went down. Any jobs that were in progress on the crashed JBoss server will be marked as failed with a reason stating that the node was down.

Table 3: Failover Scenarios *(continued)*

10) MySQL server on the active VIP node goes down	If the MySQL server on a node goes down, the watchdog service detects the down MySQL server on that active node in about 1 to 2 seconds.	The watchdog service will immediately restart MySQL server on the node. Once restarted, the MySQL server comes up in around 20 to 60 seconds.	The MySQL server on the VIP node is the active database servicing all requests from all JBoss servers in the cluster. This effectively means that a brief database outage could be experienced by JBoss on all nodes for this duration (20 to 60 seconds). Any requests that require database access will fail during this period. This will result in failures encountered by GUI/API clients on their requests which internally require database access during this period. This will also result in failures of jobs that require database access during this period.
11) MySQL server on the standby VIP node goes down	If the MySQL server on a node goes down, the watchdog service detects the down MySQL server on that standby node in about 1 to 2 seconds.	The watchdog service will immediately restart MySQL server on the node. Once restarted, it takes around 20 to 60 seconds for the MySQL server to come up. Once it is back up, this server will resynchronize with the master in the background and the resynchronization time will depend on the number of changes that happened during the outage.	Since the MySQL server on the standby VIP node is not accessed by JBoss, its down time does not cause any adverse impacts that are noticed by the rest of the system or users of the system.
12) Apache HTTP server on the active VIP node goes down	If the Apache HTTP server on a node goes down, the watchdog service detects the down HTTP server on that node in about 1 to 2 seconds.	The watchdog service will immediately restart the Apache HTTP server on the node and it becomes ready for service in 1 second.	A brief service outage could be experienced by GUI and NBI clients until the Apache HTTP server is restarted. However, this outage is only for a few seconds (typically 2 seconds) and is hardly noticed by the clients.
13) Apache HTTP server on the standby VIP node goes down	If the Apache HTTP server on a node goes down, the watchdog service detects the down HTTP server on that node in about 1 to 2 seconds.	The watchdog service will immediately restart Apache HTTP Server on the node and it becomes ready for service in 1 second.	No impact.

Related Documentation

- How Different Operational Scenarios Impact Disaster Recovery Deployments on page 38
- Example: Setting Up a Disaster Recovery Solution on page 62

How Different Operational Scenarios Impact Disaster Recovery Deployments

The impact of a disaster recovery deployment can vary depending on the specific Junos Space operational scenario.

Software Upgrade

When you are ready upgrade the software in a disaster recovery (DR) deployment, we recommend that you first perform the upgrade on the slave cluster. By upgrading on the slave cluster first, you can verify that the upgrade process is working for the new version without impacting normal operations at the master site. By upgrading on the slave cluster first, you also ensure that new software and new database schema is first made available on the slave cluster to enable it to receive new backup files from the master cluster after its upgrade. After you successfully upgrade the slave cluster, you should install the upgrade on the master cluster. When the master cluster upgrade successfully finishes, we recommend that you immediately perform a backup on the master cluster and then transfer the backup to the slave cluster.

To perform a successful backup of the master cluster and transfer to the slave cluster:

1. Execute the following command on the master cluster to perform a complete backup of the master cluster and store the archive file in the **/opt/jmp-geo/backup/data** directory:

   ```
   # /opt/jmp--geo/backup/script/backupReal.sh
   ```

2. Execute the following command on the slave cluster to fetch the latest backup file from the master cluster:

   ```
   # /opt/jmp-geo/restore/script/poll.sh
   ```

When the Active VIP Node in the Master Cluster Fails

When the active VIP node in the master cluster fails, the master VIP address switches over to the standby VIP node in the master cluster. However, the master VIP continues to be available so that the slave cluster can use SCP to connect to the master's VIP. However, because the backup files are stored in the file system of the failed node, the slave will not see any backup files to be transferred. In addition, because the cron job was created only on the failed node, no more backups will be taken automatically.

To correct this problem, you must log in to the console of the node that now owns the master VIP and repeat the steps for configuring the master cluster and the steps for starting the backup process on the master cluster, as described in "Example: Setting Up

a Disaster Recovery Solution" on page 62. Completing these steps will ensure that the cron job is created on this node as well. No action is required on the slave cluster.

When the Active VIP Node in the Slave Cluster Fails

When the active VIP node in the slave cluster fails, the slave VIP address switches over to the standby VIP node in the slave cluster. However, the slave VIP remains available and the master cluster can still ping the slave VIP. However, because backup files are stored in the file system of the failed node, if the master site goes down, you cannot bring the slave cluster online because it has no backup files from which to perform a restore operation. In addition, because the cron job was created only on the failed node, no more backups are automatically fetched from the master cluster.

To address this problem, you must log in to the console of the node that now owns the master VIP and repeat the steps for configuring the slave cluster and the steps for starting the process of fetching backup files from the master, as described in "Example: Setting Up a Disaster Recovery Solution" on page 62. Performing these steps will ensure that the cron job gets created on this node as well. We also recommended that you perform an immediate backup transfer from the master by running the **poll.sh** script. No operations are required on the master cluster.

When a New Node Is Added to the Slave Cluster

If you do not use device initiated connections for any of your managed devices, adding a new node to the slave cluster has no impact on the DR setup. However, if you do use device-initiated connections, then we recommend that you add the device management IP of the new slave node to the outbound-ssh configuration of those devices to provide better device load distribution across slave nodes when the slave cluster is brought online. To add the device management IP of the new slave node to the outbound-ssh configuration of those devices, repeat the steps for configuring the master cluster and the steps for starting the backup process on the master cluster, as described in "Example: Setting Up a Disaster Recovery Solution" on page 62.

NOTE: **When you configure the master cluster, provide the device management IP of all slave nodes including the new node. Then starting the backup process on the master cluster will ensure that all these IP addresses are configured in the outbound-ssh stanza of all devices using device-initiated connections.**

When the Main Site Is Again Operational After a Disaster

After the main site becomes operational again after being down due to some disaster, you have several options. While the main site was down, you brought the slave cluster online as described in "Understanding How the Slave Cluster Is Brought Online when the Master Cluster Goes Down" on page 41, and it is now running in the initial or standalone (0) state.

You have the following two options:

- Configure the cluster at the main site (Site-1) as a slave to the cluster at Site-2.

We recommend this option because it is simpler to achieve and does not require any disruptions. This option requires that you perform the four steps described earlier in this topic, but with Site-2 as master and Site-1 as slave.

- Configure the cluster at the main site (Site-1) as master and the cluster at Site-2 as the slave.

  With this option, you want to get back the deployment you had before the disaster struck. In this case, you want to first ensure that the most recent state from Site-2 cluster is uploaded into the cluster at Site-1, which requires that you perform the following steps:

  1. Configure Site-2 cluster as the master:
     a. Run backup.sh config on the Site-2 VIP node.
     b. Provide Site-1 cluster details as details of the slave.
  2. Run **backupReal.sh** on the Site-2 VIP node to immediately perform a backup that creates a backup file in the **/opt/jmp-geo/backup/data** directory.
  3. Configure the Site-1 cluster as the slave:
     a. Run **restore.sh config** on the Site 1 VIP node.
     b. Provide Site 2 cluster details as details of the master cluster.
  4. Run **poll.sh** on the Site-1 VIP node to transfer the latest backup file from Site-2 to Site-1.
  5. Bring down the cluster at Site-2 or ensure that the Site-2 cluster VIP is not reachable from Site-1. Otherwise the **restore.sh** script will not allow a restore to be performed at Site-2.
  6. Restore the Site-1 cluster from the latest backup file:

     Run **restore.sh restore** on the Site-1 VIP node.

After performing the preceding steps, the Site-1 cluster is up and running in initial or stand alone state (0) with the latest data from the Site-2 cluster. You can now use the Site-1 cluster to manage your network. At this point, you can repeat the four steps described in "Example: Setting Up a Disaster Recovery Solution" on page 62 to configure the Site-1 cluster as master and the Site-2 cluster as slave.

Related Documentation

- Disaster Recovery Solution and Connectivity Requirements on page 44
- Understanding Failover Scenarios on page 33

Understanding How the Slave Cluster Is Brought Online when the Master Cluster Goes Down

When a disaster strikes the main site and the master cluster goes down, you can manually run a script on the slave cluster to bring it online. This script first pings the VIP address of the master cluster to verify that it is really down. If the ping succeeds, the script exits, and gives notification that the slave cluster cannot be brought online while the master cluster is running. Otherwise, the script uses the most recent backup file available to restore the complete state of the cluster from this file. As part of the restoration, the script also restarts all services.

You can also use the **restore.sh** script by specifying the string “restore” as the command-line argument to restore the complete state of the cluster:

```
# /opt/jmp-geo/restore/script/restore.sh restore
The DR Master is down, restore procedure continues.
The latest backup files is : /opt/jmp-geo/restore/data/870946000.tgz
Do you want to continue (yes/no):yes
Disaster Recover Procedure: The DR Master Cluster must
be down,
Stoping restore cron job...
Stopping crond: [ OK ]
Starting crond: [ OK ]
Demoting this cluster from the DR Slave Cluster Role
...
update cluster state successful
opening port 7804 on user1@host....
reloading firewall...
Starting jmp-firewall: [ OK ]
finish reloading
<response>
 <message>
 </message>
<status>SUCCESS</status>
</response>
Extracting backup files....
Set node into restore state
waiting for processes to stop
Restoring mysql database....
Keep the db backup metatdata files
Restoring postgresql database...
Manually stop opennms...
opennms is running..
Stopping OpenNMS...
Stopping OpenNMS: [ OK ]
DROP DATABASE
CREATE DATABASE
...
opennms is stopped..
Starting OpenNMS...
Starting OpenNMS: (not waiting for startup) [ OK ]
Syncing space system files over to other nodes...
Restoring user1@host....
```

```
Set node into restore mode...
No JBossas is currently running
Restore Junospace system config...
DR Master didn't start snmpd before.
Starting snmpd: [ OK ]
reloading firewall...
Starting jmp-firewall: [ OK ]
finish reloading
Set node to normal mode...
jmp-watchdog running
<response>
 <message>
 </message>
<status>SUCCESS</status>
</response>
#
```

The **/opt/jmp-geo/restore/script/restore.sh** script with the **restore** option performs the following operations:

- Verifies that the master VIP is not reachable.
- Asks you to confirm proceeding with restoration from the latest available backup file.
- Removes the cron job for fetching backup files from the other site.
- Sets the state of the cluster to initial or stand alone (0).
- Opens port 7804 on all nodes in the cluster.
- Stops all services.
- Restores MySQL and PostgreSQL databases from the dump contained in the backup file
- Copies system configuration files contained in the backup to appropriate locations.
- Configures all devices to send SNMP traps to the VIP address of the slave cluster. If eth3 is used for device management on slave nodes, the eth3 IP address of the active VIP node of the slave cluster will be configured as the trap destination, instead of the VIP address.
- Restarts all services.

After the restart is complete, you can access the GUI and API of this cluster from its VIP to perform all network management tasks. How long the complete restoration process takes depends on the amount of data that must be restored, which depends on the number of managed devices, the size of their configuration, the number of alarms and events in the database, and so forth. In most cases, the complete restoration process should take from 10 to 30 minutes.

NOTE: At the end of the restoration process, as described above, the cluster runs in the initial or stand-alone mode. When the main site becomes operational again, you can make the cluster at the main site a slave to this cluster—that is, the symmetric opposite of the configuration before the master cluster went down, or you can choose to revert back to the original state. If you choose to go back to the original state, you will need to disrupt normal operations on the cluster, as described in "When the Main Site Is Again Operational After a Disaster" in the topic "How Different Operational Scenarios Impact Disaster Recovery Deployments" on page 38.

Understanding the Impact of Switching Over from Master Cluster to Slave Cluster

To bring the slave cluster online requires the use of the latest backup available from the master cluster. In addition, it takes some time for the slave cluster to establish connectivity with managed devices and synchronize their configuration and inventory data. As a result, you should expect a certain amount of data loss during the process of switching from the master cluster to the slave cluster:

- Data that was created in the master cluster after the last backup will be lost. This includes items such as job history, audit log entries, service definitions, and services created by Services Activation Director, security policies and VPNs created by Security Director, and so on. However, device configuration and inventory data will be automatically collected and synchronized by the slave cluster once it establishes connection with the managed devices.
- Alarms and events that were collected by the master cluster after the last backup will be lost. The slave cluster will start receiving alarms and events once it establishes connection with the managed devices and configures its VIP address as the new trap destination on them.
- Performance data collected by the master cluster after the last backup will be lost. The slave cluster will start polling for PM data once it establishes connection with the managed devices.
- Any scheduled jobs that were created on the master cluster after the last backup but not executed before it went down will be lost.
- Recurrent job executions that were supposed to occur once or more between the time the master cluster going down and the slave cluster coming online will be lost. These jobs will start running again on the slave cluster from the next scheduled interval onwards.

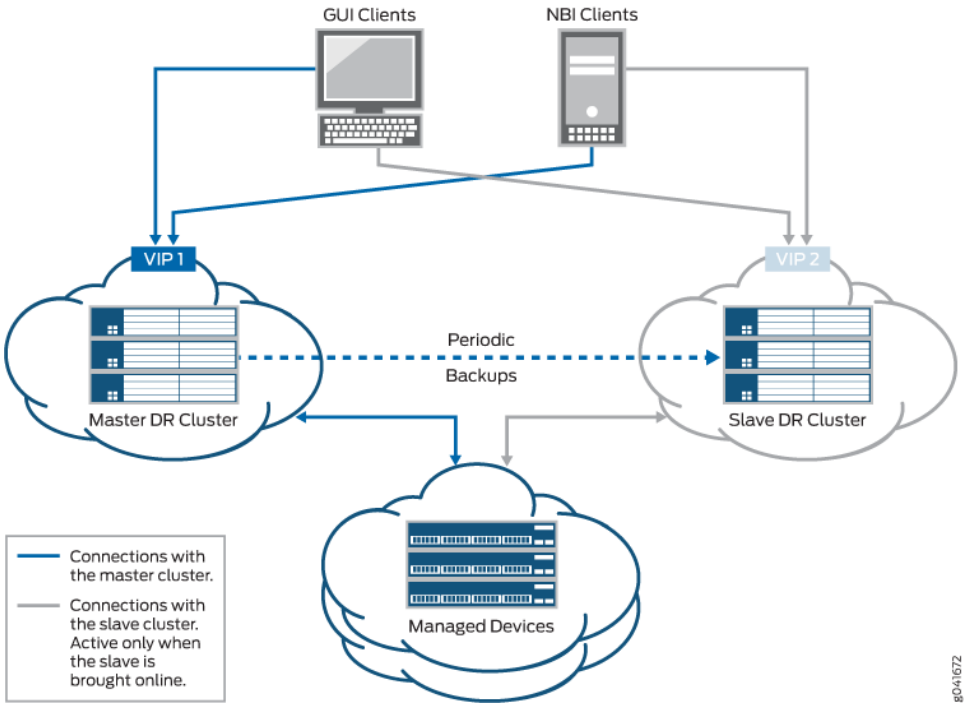
Related Documentation

- Understanding Failover Scenarios on page 33
- Disaster Recovery Solution and Connectivity Requirements on page 44
- Example: Setting Up a Disaster Recovery Solution on page 62

Disaster Recovery Solution and Connectivity Requirements

Junos Space cluster provides high availability and scalability to your network management solution. However, since all nodes in a cluster need to be within the same subnet, they are typically deployed in the same data center or within the same campus, which means that a Junos Space cluster by itself cannot survive disasters that can bring down both VIP nodes in the cluster. To avoid such a disaster, Junos Space supports a disaster recovery solution, as shown in Figure 7 on page 44, that allows a separate cluster at a different geographical location, which is known as the *DR slave cluster*, whereas the original cluster is known as the *DR master cluster*.

Figure 7: Disaster Recovery Solution



After the DR master cluster and DR slave cluster are deployed, you can log in to the console of the node that owns the VIP address of each cluster and run a few scripts on them to configure the master-slave relationship between the two clusters. After you have run the scripts on the clusters, the master cluster periodically takes a complete backup of its state, which is saved as a **.tgz** archive in its local file system. You can configure the script to specify the time and frequency of backups. This master cluster backup includes the entire database and all the files required for the correct functioning of all services on the cluster. The slave cluster will periodically copy this archive over to its local file system through SCP. Again, you configure the script to specify the time and desired frequency of the backup copy.

NOTE: The connectivity between the slave cluster and managed devices is disabled as long as the master cluster is functional. When the master cluster becomes unavailable due to a disaster, you can run a script on the node which owns the VIP address of the slave cluster to restore its state from the most recent backup that has been taken from the master cluster. At this time, all services on the slave cluster are restarted and the connectivity between the slave cluster and managed devices is established.

A disaster recovery solution must include the following connectivity requirements:

- Layer 3 connectivity is enabled between the two clusters, which means:
  - Any node in a cluster must be able to successfully ping (ICMP or ECHO) the VIP address of the other cluster.
  - Any node in a cluster must be able to use SCP to transfer files from the VIP address of the other cluster.
  - The bandwidth and latency of the connection between the two clusters must be such that the SCP transfer of a backup file from the master to the slave must complete before the next backup is available. This depends on the size of the backup file as well as the configured frequency with which backups are taken. You can use the following formula to compute the minimum required SCP transfer speed (*T*) between the master and the slave cluster:

    T = S x 8 / (N x 3600) Mbps

    where *S* is the size of the backup file expressed in MB, and *N* is the number of hours between two consecutive backups. Using the formula, you can see that if you can achieve a transfer speed of 1 Mbps between the two clusters, you can create daily backup files up to 10 GB in size.

    NOTE: The SCP protocol is not significantly impacted by high network latency; the bandwidth of the connection link between the two clusters is more important. Therefore, you must provide a connection link with sufficient bandwidth to achieve the minimum SCP transfer speed as calculated using the formula above.

- Independent Layer 3 connectivity between each cluster and managed devices is required.
- Independent Layer 3 connectivity between each cluster and GUI or NBI clients is required.

For an example disaster recovery configuration, see "Example: Setting Up a Disaster Recovery Solution" on page 62.

Related Documentation

- Understanding High Availability Nodes in a Cluster on page 20
- Configuring the Junos Space Cluster for High Availability Overview on page 24

CHAPTER 5

Frequently Asked Questions

What Steps Do I Need to Take When One of the High Availability Nodes in the Cluster Shows the Status “Down”?

The first step is to collect all the logs from the cluster as you will want these logs for troubleshooting later on. Navigate to the **Network Management Platform > Administration > Space Troubleshooting** workspace to download troubleshooting data and logs from the cluster. Then try to log in to the console of the appliance that is down. If you are able to log in, access the debug shell and check to see whether the jmp-watchdog and jboss services are up. If the services are not up, do a restart on them and wait 15 to 20 minutes to see if the node status changes to “Up”. If not, perform a reboot. Once these services are up, navigate to the **Network Management Platform > Administration > Fabric** workspace to add the appliance back to the cluster. If the node does not come up, perform a reboot and wait 15 to 20 minutes for the node status to change to “Up” in the **Network Management Platform > Administration > Fabric** workspace.

If you are unable to log in to the console, or a reboot does not solve the issue, navigate to the **Network Management Platform > Administration > Fabric** workspace and delete the node from the cluster. You can then promote another node (if available) as an HA node.

You can then try to re image the appliance from a USB stick that has the same version of Junos Space that is currently running on the other nodes in the cluster. If you are unable to re-image the appliance, you must get a replacement appliance and re image it.

NOTE: Whether you re-image the existing appliance or get a replacement appliance, you must reconfigure the appliance with exactly the same network settings that were configured previously.

Once the appliance is configured, navigate to the **Network Management Platform > Administration > Fabric** workspace to add the appliance back to the cluster.

How Can I Simulate a Virtual IP (VIP) Failover?

To simulate a VIP failover, you can shut down the heartbeat service on the node that currently owns the VIP.

1. Log in to the console of the node and enter “service heartbeat stop”.

   The heartbeat service on this node will shut down, at which point a message is automatically sent to its peer node. The peer node will take over the VIP in response to this message.

2. You can monitor the **/var/log/messages** file on both nodes to view the log entries related to the failover.

   As the messages will show, VIP failover is almost instantaneous in this case.

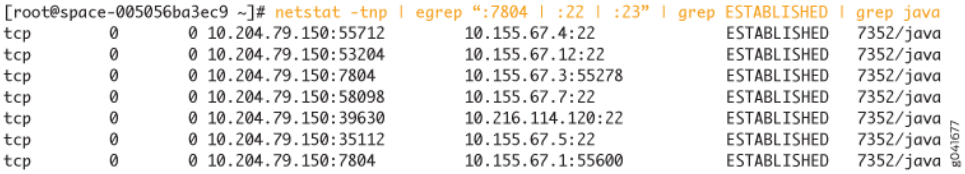
What Algorithm Does the Apache HTTP Load Balancer Use to Distribute Load Among Nodes in the Cluster?

Junos Space uses the mod_proxy load balancer module within the Apache HTTP server. The load balancer module is configured to use the default load-balancing method (lbmethod=byrequests), which provides a weighted round-robin algorithm. However, because all nodes in the cluster have the same weight, a round-robin strategy is used for load distribution. The load balancer is also configured to use sticky sessions (using the JSESSIONID cookie), which ensures that all HTTP requests that are part of the same login session are forwarded to the same node in the cluster.

How Do I Determine Which Device Connections Are Being Handled by a Node?

The easiest way is to SSH to the console of the node and use the **netstat** command, as shown in Figure 8 on page 48. The output will show one line for each device connected to this node. In this example, the node includes two device initiated connections and five space initiated connections. The IP addresses of the devices are displayed in the fifth column.

Figure 8: Using the netstat Command to SSH to the Console of a Node



How Do I Determine Which Node in the Cluster Is Handling My Network Management Platform User Interface Session?

Open you browser's dialog, which lists all the stored cookies, then find the site with the VIP address of the Junos Space cluster and expand it. The content of the cookie named JSESSIONID contains the hostname of the node which is serving your current session.

PART 2

Configuration

CHAPTER 6

Disaster Recovery

Creating the DR Master Cluster

To set up the main cluster, the DR Master cluster, run three scripts as described in the following sections:

Backup configuration and Restore configuration should be done only on the VIP node of the Master cluster. If a VIP switchover occurs, you must rerun the backup script on the new VIP node.

The role change from DR Slave to DR Master (backup to restore) and vice versa cannot be made directly. It can be made only after the initial role is stopped.

The scripts used are located at: **/opt/jmp-geo/backup/script/backup.sh – script**

NOTE:

- When a new node is added to the cluster, the backup and restore scripts must be rerun to update the configuration.
- After you run the restore script, the network monitoring node list might contain previous Junos Space servers as well.

To set up the main cluster, the DR Master cluster, run three scripts as described in the following sections:

Configuring the DR Master Cluster

Configuring the DR Master cluster enables you to input the following information, which is then stored in the **backup.properties** file:

- E-mail address for notifications
- DR Slave VIP IP address
- DR Slave device management IP addresses
- Number of backup files to be kept
- Time at which the backup should be run
- Number of days per week the backup should run

Run the script as follows. The output shown reflects the sample input.

```
[user1@host script]# ./backup.sh config
Please enter contact email address in case of Disaster Recovery Slave failure:

user1@example.com

Backup configurations...

Creating /etc/ssmtp/ssmtp.conf...

Creating /etc/ssmtp/revaliases...

Please enter DR Slave Cluster management ip(VIP) :

10.10.10.10

Please enter DR Slave Cluster device management ip(comma separated) :

10.10.10.63,10.10.10.65

checking ip: 10.10.10.63

checking ip: 10.10.10.65

Please enter max backup files to keep(default=3):

Notice: cron job takes format of digits joined by ',', For every instance enter '*'

Please enter hours of the day to run backup:

0

Please enter days of the week to run backup, Sun= 0, Sat=6:

6
```

NOTE: You should enter the hours of the day to run a backup operation in a 24-hour format.

Starting the Backup Operation for the DR Master Cluster

Starting the backup operation for the DR Master cluster causes a recurring job to be put in the cron. It can be viewed using **crontab –l**.

The backups are stored in the same server in **/opt/jmp-geo/backup/data** in TGZ . Verify the status of the backup process in **/opt/jmp-geo/backup/backup.log**. If the DR Slave is not available, you are notified by e-mail to the e-mail ID configured in the "Configuring the DR Master Cluster" on page 54 section.

If the device discovery mode is DIC, the script also adds the outbound-SSH of the DR Slave cluster's device management IP address to the Junos Space managed devices.

Run the script as follows. The output shown reflects the sample input.

```
[user1@host script]# ./backup.sh start
Demoting this cluster from the DR Master Cluster Role ...

update cluster state successful

Stoping backup cron job...

Stopping crond: [ OK ]

Starting crond: [ OK ]

Promoting this cluster to the DR Master Cluster Role ...

update cluster state successful

Adding DR Slave Cluster device management ip to devices ...

save cluster ip successful

save cluster ip successful

queue http://10.0.0.1:8080/api/hornet-q/queues/jms.queue.jmpgeoq4327 creation
successful

update-devices-with-ip 10.10.10.65 successful

delete http://10.0.0.1:8080/api/hornet-q/queues/jms.queue.jmpgeoq4327 successful

Starting backup cron job...

Stopping crond: [ OK ]

Starting crond:

The DR cron job is started on the DR master.
```

Stopping the Backup Operation

Do not transition from DR Master to DR Slave directly. Stop the initial role first. Choose one of the following methods of transitioning:

- Promote a normal cluster to DR Master.
- Demote a normal cluster to DR Slave.
- Disable a DR Master so that it becomes a normal cluster.
- Disable a DR Slave so that it becomes a normal cluster.

Stopping the backup operation removes the cron job and stops the backup operation from being performed.

Run the script as follows. The output shown reflects the sample input.

```
[user1@host script]# ./backup.sh stop
Demoting this cluster from the DR Master Cluster Role ...

update cluster state successful

Stoping backup cron job...

Stopping crond: [ OK ]

Starting crond: [ OK ]

[user1@host script]#
```

Related Documentation

- Disaster Recovery Overview on page 31
- Creating the DR Slave Cluster on page 56
- Performing a Reverse Restore Operation on page 61

Creating the DR Slave Cluster

The DR Slave cluster takes over when disaster has overtaken the DR Master cluster. The **/opt/jmp-geo/restore/script/restore.sh** script uses SCP to pull the backups from the DR Master cluster and when required, restore the DR Slave with the information from the DR Master.

The following four operations involved in setting up the DR Slave cluster:

Backup configuration and Restore configuration should be done only on the VIP node of the DR Master cluster or the DR Slave cluster. If a VIP switchover occurs, you must rerun the backup or restore script (depending on the role) on the new VIP node.

NOTE:

- When a new node is added to the cluster, the backup and restore scripts must be rerun to update the configuration.
- After you run the restore script, the network monitoring node list might contain previous Junos Space Servers as well.

The role change from Slave to Master (backup to restore) and vice versa cannot be made directly. It can be made only after the initial role is stopped.

The scripts used for this purpose are located at: **/opt/jmp-geo/restore/script/restore.sh – script** .

The following four operations are involved in setting up the DR Slave cluster:

- Configuring the DR Slave Cluster on page 57
- Starting to Pull the Backups from the DR Master on page 58
- Stopping Pulling the Backups from the DR Master on page 59
- Restoring on page 60

Configuring the DR Slave Cluster

Configuring the DR Slave cluster records the following information in the **restore.properties file:**

- E-mail address to receive notifications
- DR Master VIP address
- DR Master passwords, if there are multiple nodes
- SCP timeout
- Time at which the backups are to be pulled from the DR Master
- Number of days per week the backups are to be pulled from the DR Master

Run the script as follows. The output shown reflects the sample input.

[user1@host script]# ./restore.sh config
Please enter contact email address in case DR Master failure:

user1@example.com

Backup configurations...

Creating /etc/ssmtp/ssmtp.conf...

Creating /etc/ssmtp/revaliases...

Please enter DR Master Cluster management ip(VIP) :

10.10.10.10

Please enter DR Master Cluster VIP node admin passwords(comma separated):

abc123

Please enter scp timeout in seconds:

120

Notice: cron job takes format of digits joined by ',', For every instance enter '*' Please enter hours of the day to pull backup files:

0

Please enter days of the week to pull backup files, Sun= 0, Sat=6:

0

Testing SCP from DR Master to DR Slave...

Starting to Pull the Backups from the DR Master

The script shown in this section starts pulling the backups from the DR Master cluster.

It creates a cron job entry, which can be viewed by using **crontab –l**.

If the DR Master is not available, you receive a e-mail notification.

The copied files are located in the **/opt/jmp-geo/restore/data** folder. The restore polling status is located in **/opt/jmp-geo/restore/restore.log**.

At this point, the script blocks all connections to devices because this is a slave cluster (that is, no devices can be discovered).

Run the script as follows. The output shown reflects the sample input.

[user1@host script]# ./restore.sh startPoll
Enabling this cluster to the DR Slave Cluster Role ...

update cluster state successful

blocking port 7804 on host....

reloading firewall...

Starting jmp-firewall: [OK]

finish reloading

<response>

<message>

</message>

<status>SUCCESS</status>

```
</response>

Starting restore cron job...

Stopping crond: [ OK ]

Starting crond: [ OK ]
```

Stopping Pulling the Backups from the DR Master

The script in this section stops pulling the backups from the DR Master, and thereby demotes the cluster from the DR Slave cluster role and removes the cron job entry.

Do not transition from DR Master to DR Slave directly. Stop the initial role first. Choose one of the following methods of transitioning:

- Promote a normal cluster to DR Master.
- Demote a normal cluster to DR Slave.
- Disable a DR Master so that it becomes a normal cluster.
- Disable a DR Slave so that it becomes a normal cluster.

Stopping the backup removes the cron job and stops the backup operation being performed.

Run the script as follows. The output shown reflects the sample input.

```
[user1@host script]# ./restore.sh stopPoll
Stoping restore cron job...

Stopping crond: [ OK ]

Starting crond: [ OK ]

Demoting this cluster from the DR Slave Cluster Role ...

update cluster state successful

opening port 7804 on host....

jmp-firewall is stopped. Skip reloading

<response>

<message

</message>

<status>SUCCESS</status>

</response>
```

Restoring

Running the restore script enables the DR Slave to take over the management role when disaster overtakes the DR Master. The script carries out the following four operations:

1. Stops JBoss and the network monitoring service, inflates the files from the latest backup that was pulled, and brings the whole system back up.
2. Enables all connections to the devices.

NOTE: You cannot run the restore script when the DR Master is present and online. This procedure is for disaster recovery scenarios only.

3. If the devices were originally discovered using DIC mode, reconfigures Junos Space-managed devices to point to the DR Slave cluster so that devices connect back to the DR Slave cluster
4. Reconfigures all the devices to point the SNMP trap group to the DR Slave cluster, so that traps and alarms are received by the DR Slave cluster.

Run the script as follows. The output shown reflects the sample input.

```
[user1@host script]# ./restore.sh restore
The DR Master is down, restore procedure continues.

The latest backup files is : /opt/jmp-geo/restore/data/825763000.tgz

Do you want to continue (yes/no):

yes

Disaster Recover Procedure: The DR Master Cluster must be down,

turning this DR Slave Cluster to be in service ...

update cluster state successful

opening port 7804 on user1@host....

reloading firewall...

Starting jmp-firewall: [ OK ]

finish reloading

<response>

<message>

</message>

<status>SUCCESS</status>
```

```
</response>

Extracting backup files....

Set node into restore state
```

Related Documentation

- Disaster Recovery Overview on page 31
- Creating the DR Master Cluster on page 53
- Performing a Reverse Restore Operation on page 61

Performing a Reverse Restore Operation

You perform a reverse restore to reestablish a disaster recovery system by creating a new DR Slave at a site geographically separate from the site where your new DR Master is located. For example, if your original DR Master was in Chicago, and your DR Slave was in London, if the London site is overtaken by a further disaster, you would get your original site, Chicago, back online, and then create a DR Slave in Chicago because London would be the new DR Master.

This topic provides instructions for performing a reverse restore.

1. Configure your new DR Master (in the example above, the London site) for backup. See "Creating the DR Master Cluster" on page 53.
2. At the new DR Slave site, reinstall the same version of Junos Space Network Management Platform with the same IP addresses, applications and adapters used originally (in the example above, Chicago). See the Prerequisites section of "Disaster Recovery Overview" on page 31.
3. Configure the new DR Slave site for the restore operation. See "Creating the DR Slave Cluster" on page 56.

NOTE: After you run the restore script, the network monitoring node list might contain previous Junos Space servers as well.

Related Documentation

- Disaster Recovery Overview on page 31
- Creating the DR Master Cluster on page 53
- Creating the DR Slave Cluster on page 56

Example: Setting Up a Disaster Recovery Solution

This example shows how to set up a disaster recovery solution for the Junos Space Network Management Platform.

Requirements

The following prerequisites apply for configuring a disaster recovery solution:

- Two separate clusters must be deployed at two geographical locations.
- We recommend that both clusters have the same number of nodes so that, even in the case of a disaster, the DR site can operate with the same capacity as the primary site. However, it is also possible to have the DR slave cluster configured with fewer nodes.
- Both clusters are installed with the same versions for both the Junos Space Network Management Platform as well as all high-level applications and the device adapters installed on them.
- Connectivity requirements, as described in "Disaster Recovery Solution and Connectivity Requirements" on page 44.
- You must configure the same SMTP server on both clusters to receive e-mail alerts related to the DR solution from both clusters. You can perform this task from the **Network Managment Platform > Administration > SMTP Servers** workspace.

Overview

The Junos Space Network Management Platform is a software product packaged as an appliance. You can connect multiple appliances (both physical and virtual) together to form a single cluster. All services within the cluster are provided from a single virtual IP (VIP) address. This architecture provides protections against any single point of failure in the cluster. If any node in the cluster fails, all services will continue to be available. To prevent against the possible loss of data in the event of a geographic disaster, you can configure two separate clusters (master and slave clusters) to provide further protections in the event all nodes in a single cluster are compromised. This example outlines the basic steps for setting up a deployment of master and slave clusters.

Topology

Figure 9 on page 63 shows an example deployment of master and slave clusters with three nodes in each cluster to illustrate each step.

Figure 9: Example Deployment of Master and Slave Clusters

Sites:	Site-1 (Master)			Site-2 (Slave)		
Node:	Node-1	Node-2	Node-3	Node-1	Node-2	Node-3
eth0 IP:	10.1.1.1	10.1.1.2	10.1.1.3	10.2.1.1	10.2.1.2	10.2.1.3
eth3 IP:	20.1.1.1	20.1.2.1	20.1.3.1	20.2.1.1	20.2.2.1	20.2.3.1
VIP:	10.1.1.4			10.2.1.4		

g041673

Configuration

To configure a disaster recovery solution, perform these tasks in order:

- Configuring the Master Cluster on page 64
- Starting the Backup Process on the Master Cluster on page 66
- Configure the Slave Cluster on page 67
- Starting the Process of Fetching Backup Files from the Master on page 69

Configuring the Master Cluster

Step-by-Step Procedure

In this task, you configure the master cluster with details of the slave cluster and the required backup schedule. You can use the **backup.sh** script to perform this task by passing the string config as the command-line argument, which is illustrated in the following example.

```
 # /opt/jmp-geo/backup/script/backup.sh config
Please enter contact email address in case of Disaster
Recovery Slave failure:
user2@example.com
Backup configurations...
Creating /etc/ssmtp/ssmtp.conf...
Creating /etc/ssmtp/revaliases...
Please enter DR Slave Cluster management ip(VIP) :
10.2.1.4
Please enter DR Slave Cluster device management
ip(comma separated) :
20.2.1.1,20.2.2.1,20.2.3.1
checking ip: 20.2.1.1
checking ip: 20.2.2.1
checking ip: 20.2.3.1
Please enter max backup files to keep(default=3):
Notice: cron job takes format of digits joined by ',',
For every instance enter '*'
Please enter hours of the day to run backup:
2
0 Please enter days of the week to run backup, Sun= 0,
Sat=6:
*
#
# cat /opt/jmp-geo/backup/script/backup.properties
STANDBY_VIP=10.2.1.4
STANDBY_DEVICE_MGT_IP=20.2.1.1,20.2.2.1,20.2.3.1
CONTACT_EMAIL=user2@example.com
MAX_FILES=3
CRON_HOURS=2
CRON_WDAYS=*
#
```

The **backup.sh** script prompts you for the following details:

- e-mail address to be notified when the master detects that the slave cluster is down.
- VIP address of the slave cluster.
- Comma-separated list of the device management IP addresses of all the nodes in the slave cluster. If you configured eth3 as the device management interface on the slave nodes, you must ether their eth3 addresses here. Otherwise you must provide their eth0 addresses.
- The maximum number of backup files to keep at any time. The default is 3.
- The cron schedule for the backup process, to be entered as hours of day and days of the week following cron conventions. The recommendation is to configure for a daily backup to happen at an hour when the system is used least. For example, see the transcript above which configures a daily backup at 2 AM.

The script saves the entered values into a text file named **backup.properties** at the same location as the script. This file is read by the script during subsequent tasks.

Starting the Backup Process on the Master Cluster

Step-by-Step Procedure

This task is performed using the same **backup.sh** script, which passes the string start as the command-line argument. The script does not require any inputs and receives its required values from the **backup.properties** file created in the previous task "Configuring the Master Cluster" on page 64.

```
# /opt/jmp-geo/backup/script/backup.sh start
Demoting this cluster from the DR Master Cluster Role...
update cluster state successful
Stopping backup cron job...
Stopping crond: [ OK ]
Starting crond: [ OK ]
Promoting this cluster to the DR Master Cluster Role...
update cluster state successful
Adding DR Slave Cluster device management ip to devices  ...
save cluster ip successful
queue http://127.0.0.1:8080/api/hornetq/
queues/jms.queue.jmpgeoq7886 creation successful
update-devices-with-ip 20.2.3.1 successful
delete http://127.0.0.1:8080/api/hornetq/
queues/jms.queue.jmpgeoq7886 successful
Starting backup cron job...
Stopping crond: [ OK ]
Starting crond: [ OK ]
#
```

The **backup.sh** script does the following:

- Sets the state of the cluster as Master (1) in the database.
- Updates the outbound-ssh configuration stanza of devices connected with device initiated connections to contain the device management IP addresses of all slave cluster nodes.

NOTE: In a disaster recovery configuration, the device management IP addresses of the nodes in the slave cluster should be added to the device configuration, as shown in this task, so that when a disaster strikes, the devices are already configured with the required information to connect to the slave cluster. Also, after you complete the task "Starting the Process of Fetching Backup Files from the Master" on page 69, port 7804 is blocked on the slave cluster during normal operation, thus preventing devices from accidentally getting connected to the slave cluster. When the slave cluster is brought online in response to a disaster striking the master cluster, port 7804 is opened, which allows the devices to connect to the slave cluster.

- Creates a cron job for creating a backup of the cluster as specified in the configured schedule.

Configure the Slave Cluster

Step-by-Step Procedure

In this task, you configure the slave cluster with details of the master cluster and the required schedule for fetching backup files from the master. You can use the **restore.sh** script to perform this task by passing the string **config** as the command-line argument as shown below.

```
# /opt/jmp-geo/restore/script/restore.sh config
Please enter contact email address in case DR Master
failure:
user1@example.com
Backup configurations...
Creating /etc/ssmtp/ssmtp.conf...
Creating /etc/ssmtp/revaliases...
Please enter DR Master Cluster management ip(VIP) :
10.1.1.4
Please enter DR Master Cluster VIP node admin
passwords(comma separated):
abc123
Please enter scp timeout in seconds:
120
Notice: cron job takes format of digits joined by ',',
For every instance enter '*' Please enter hours of the
day to pull backup files:
3
Please enter days of the week to pull backup files,
Sun= 0, Sat=6:
*
Testing SCP from DR Master to DR Slave...
#
#
# cat /opt/jmp-geo/restore/script/restore.properties
ACTIVE_VIP=10.1.1.4
ADMIN_PASS=abc123
CONTACT_EMAIL=user1@example.com
SCP_TIMEOUT=120
CRON_HOURS=3
CRON_WDAYS=*
#
```

The **restore.sh** script prompts you for the following details:

- Email address to be notified when the slave fails to transfer a backup file from the master cluster.
- Virtual IP address of the master cluster.
- Comma-separated list of the admin passwords used on the nodes in the master cluster. If you specified the same admin password on all nodes (recommended), this is that password.
- SCP timeout (in seconds) to be used to detect SCP failures.
- The cron schedule for fetching the latest backup file, to be entered as hours of day and days of the week following cron conventions. Recommendation is to configure for a daily transfer to happen one hour after the master has taken the backup. For example, see the transcript on the right which configures a daily transfer at 3AM.

The script saves the entered values into a text file named **restore.properties** at the same location as the script. This file is read by the script during subsequent steps. The script also attempts to fetch a dummy file from the master cluster to verify that it can perform an SCP transfer from the master cluster by using the supplied credentials.

Starting the Process of Fetching Backup Files from the Master

Step-by-Step Procedure

In this task, you use the **restore.sh** script and enter "startPoll" as the command line argument. The script does not ask for any inputs but gets required values from the **restore.properties** file that you created in the previous task "Configure the Slave Cluster" on page 67.

```
# /opt/jmp-geo/restore/script/restore.sh startPoll
Enabling this cluster to the DR Slave Cluster Role ...
update cluster state successful
blocking port 7804 on user1@host....
reloading firewall...
Starting jmp-firewall: [ OK ]
finish reloading
<response>
 <message>
 </message>
<status>SUCCESS</status>
</response>
Starting restore cron job...
Stopping crond: [ OK ]
Starting crond: [ OK ]
#
```

The **restore.sh** script does the following:

- Sets the state of the cluster as Slave (2) in the database
- Blocks the port 7804 on all nodes in the cluster.

NOTE: In a disaster recovery configuration, the device management IP addresses of the nodes in the slave cluster are added to the device configuration, as shown in the task "Starting the Backup Process on the Master Cluster" on page 66, so that the devices are configured with the required information to connect to the slave cluster, thus protecting your system in the event disaster strikes. After completing this task, port 7804 is blocked on the slave cluster during normal operation, thus preventing devices from accidentally getting connected to the slave cluster. However, when the slave cluster is brought online in response to a disaster striking the master cluster, port 7804 is opened, which allows the devices to connect to the slave cluster.

- Creates a cron job to fetch the latest backup file from the master cluster as specified in the configured schedule.

Related Documentation

- Understanding How the Slave Cluster Is Brought Online when the Master Cluster Goes Down on page 41

PART 3

Index

Index