

SRX SERIES SERVICES GATEWAYS' HIGH AVAILABILITY USING JUNOS AUTOMATION

Achieving High Availability on the SRX Series by Utilizing Junos Automation. An Example of the Robust Nature of Junos Automation.

Table of Contents

Introduction	1
Scope	1
Design Considerations	1
Description and Deployment Scenario	1
Configuring Junos Automation	2
Achieving HA with Track IP	2
Configuring Track IP Event Options	4
Validating the Track IP Configuration	4
Achieving HA with Interface Monitoring	4
Validating the Interface Monitoring Configuration	6
Global SRX Series HA Configuration Options	6
Summary	7
About Juniper Networks	7

Table of Figures

Figure 1: Track IP topology	1
Figure 2: Track IP configuration example	3
Figure 3: Event options configuration example	4
Figure 4: Interface monitoring configuration example	5
Figure 5: Interface monitoring configuration example	5
Figure 6: Chassis cluster configuration example	6

Introduction

The track IP feature was missing from the initial release of the Juniper Networks® SRX Series Services Gateways platform, and in response to this, a Junos automation script was created to implement this feature. Junos automation track IP implementation allows the user to utilize this critical feature on the SRX Series platforms. It allows for path and next-hop validation through the existing network infrastructure with the ICMP protocol. Upon the detection of a failure, the script executes a failover to the other node in an attempt to prevent downtime.

The second SRX Series high availability (HA) feature implemented is interface monitoring. Interface monitoring is an included feature on the platform but it may not meet all customer requirements. The script implements all of the features that are included on the SRX Series platform plus the ability to limit the failover of the control plane. In some scenarios, if the control plane is rapidly failed over, it can cause instability. This implementation of interface monitoring accounts for this possibility and limits the user to monitoring only the data plane.

Scope

This document is designed to teach the reader how to utilize the track IP and interface monitoring features enabled by Junos automation on the SRX Series. Upon completion of reading this document the user will be able to implement the feature on an SRX Series firewall.

Design Considerations

Junos automation is a powerful tool that allows the administrator to customize their Juniper Networks Junos® operating system-based device to meet their needs. Before implementing Junos automation, the administrator should become familiar with their usage. Improperly implementing Junos automation can cause network outages and/or unexpected downtime.

Hardware Requirements

- SRX Series platforms

Software Requirements

- Junos OS Release 9.3 or later

Description and Deployment Scenario

Service availability is one of the key responsibilities of the underlying network. Each customer's needs may vary. There are times that the administrator may want to customize the availability options outside the scope of the products built-in features. Track IP, as implemented in Junos automation, can be utilized as a supplementary feature to interface monitoring for high availability. The goal for track IP is to monitor an upstream gateway to ensure the path between it and the SRX Series is available. In the event of a failure, the script will detect the loss of availability and fail the chassis over to the next node in the cluster. Figure 1 shows an example of a track IP topology.

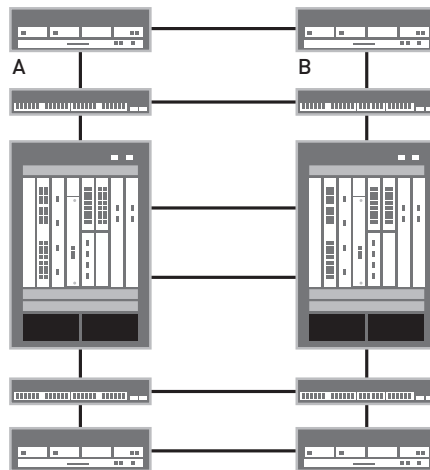


Figure 1: Track IP topology

Figure 1 depicts two routers using the track IP script. When the router labeled A is no longer accessible, the active SRX Series device is able to detect it. By using the track IP script, the device would failover to the second SRX Series device node. In doing so, the second node will now have access to the upstream device, allowing router A to now be accessible. This minimizes the interruptions and allows for traffic to continue passing over the network.

Interface monitoring is a feature that is built into Junos OS on the SRX Series. The built-in configuration can be utilized for both redundancy group 0 and group 1. However, it may be possible that during rapid failover of redundancy group 0 that instability can occur. For those who want redundancy group 0 to fail over in the event of an interface failure—and are unable to prevent rapid failures—the Junos automation script was created. The script operates just like the built-in feature. However, it prevents redundancy group 0 from being rapidly failed over.

Configuring Junos Automation

There are two steps in configuring Junos automation. The first is placing the scripts on the devices that need to utilize the scripts and the second is to inform Junos OS about the scripts. All Junos automation scripts are located in the `/var/db/scripts` directory on the Junos OS-based device. There are four directories inside the scripts directory. A description of the directories is shown in Table 1.

Table 1: Junos Automation Script Directory Locations

Directory	Description
<code>/var/db/scripts/event</code>	This directory is where event scripts should be placed. When configuring scripts under the stanza <code>event options</code> , Junos OS utilizes the scripts named there from this location.
<code>/var/db/scripts/commit</code>	This directory is where commit scripts should be placed. When configuring scripts under the stanza <code>system/scripts/commit</code> , the scripts specified in that location are used from this location.
<code>/var/db/scripts/op</code>	This directory is where operational scripts should be placed. When configuring scripts under the stanza <code>system/scripts/op</code> , the scripts specified in that location are used from this location.
<code>/var/db/scripts/import</code>	This directory contains support scripts that are implemented by Juniper Networks and are a part of Junos OS. Scripts should not be removed or placed in this directory.

Scripts can be placed in the directories using secure copy (`scp`), or by initiating an FTP, TFTP, or HTTP get from the Junos OS-based device. Alternatively the script can be automatically pulled from a location based upon specifying a complete Uniform Resource Identifier (URI) in the Junos OS configuration. More details on automatic delivery of Junos automation script to devices are covered in the *Configuration and Diagnostic Automation Guide* in the Junos OS documentation. This documentation is considered the ultimate resource in implementing Junos automation and should be used as the main source of reference when utilizing Junos automation.

Achieving HA with Track IP

The track IP Junos automation implementation utilizes the event system. The event system allows for the triggering of events based upon receiving a specific system log or SNMP message. In this case, an event "TRACKIP" is triggered every 60 seconds, calling the track IP Junos automation script. The script reads the chassis cluster redundancy group information, searching for the track IP configuration elements. The elements are stored in a macro that is created in any redundancy group. Figure 2 is an example configuration for track IP.

```

apply-macro monitoring-options {
    clear-failover;
    full-failover;
}
reth-count 2;
control-ports {
    fpc 2 port 0;
    fpc 14 port 0;
}
redundancy-group 0 {
    node 0 priority 254;
    node 1 priority 1;
}
redundancy-group 1 {
    apply-macro track-gateway {
        server 1.1.1.222;
        weight 255;
        count 5; /* Optional */
        routing-instance Test; /* Optional */
        interval 2; /* Optional */
        wait 1; /* Optional */
    }
}
/* Optional */
node 0 priority 254;
node 1 priority 1;
}

```

Figure 2: Track IP configuration example

The macro requires that it begin with “track-” and contain a server variable and a weight. The server variable requires an IP address or hostname—this is the host that is tested three times. If the ping attempt is successful, then the script exits successfully. If the ping attempt fails, then it checks the weight. If the weight is equal to or greater than 255, then the firewall fails that redundancy group over to the other chassis. If there are multiple hosts listed and several of them fail, the weights are added and again checked to see if they equal or exceed 255. The track-ip script only can run within a 60-second window. Because of this, the total runtime is calculated with the following formula: $((60 / ((count * interval) + (wait - 1)))$. Any fractional number is remained so partial attempts are not run. It is important to calculate the total runtime so it does not exceed 60 seconds. All of the options that are available are specified in Table 2.

Table 2: Features and Benefits

Configuration Option	Description
Server	This specifies the IP address of the host that is tested. This option is required.
Interval	This specifies the time between ping attempts. The default is one-tenth of a second if not specified.
Weight	This declares the weight of the host. This option is required.
Count	This specifies the total attempts that are made to the server. The number of default attempts is three if it is not otherwise specified.
Routing Instance	This specifies the source routing instance for the ping attempt. This is optional and if it is not specified it uses the inet.0 routing table.
Wait	The wait option specifies the amount of seconds to wait for until exiting the ping. If not specified the default is zero seconds.

Configuring Track IP Event Options

When the script starts it tests to determine if the current routing engine (RE) is the primary for redundancy group 0—if it is not, then the script does not run. Only the primary RE is able to generate pings so it only executes the script. Figure 3 shows the event options configuration.

```
generate-event {
  TRACK-IP time-interval 60;
}
policy TRACK-IP {
  events TRACK-IP;
  then {
    event-script track-ip.xml;
  }
}
event-script {
  file track-ip.xml;
}
```

Figure 3: Event options configuration example

There are four important stanzas that are configured in Figure 3. The first is `generate-event`—this stanza generates a “TRACK-IP” event every 60 seconds. Then there is the policy “TRACK-IP.” This policy listens for the “TRACK-IP” event. When it detects the event it executes the `track-ip.xml` Junos automation script. Any output generated is sent to the destination `TRACKIPLOG`. The `event-script` stanza loads the `track-ip.xml` script and has Junos OS validate it so it is ready for execution.

Validating the Track IP Configuration

The configuration options that begin with “`apply-macro`” are all user-created options. Because of this Junos OS does not validate these by default. To create custom validation options a commit script must be used. To validate the track IP configuration the commit script “`srx-ha-validate.xml`” was created. The script must be placed in the `/var/db/scripts/commit` directory. Secondly, it must be added to the Junos OS configuration by using the command “`set system scripts commit file srx-ha-validate.xml`.” Upon the committing of this configuration the options for track IP are validated.

This prevents misspelling items such as “`routing-instance`” or giving numbers that are out of the range of the scripts’ capabilities. In the event that an option is not correctly configured, a warning is emitted. This does not notify the administrator that something is not right. It does not prevent the misconfiguration of track IP—it just creates a warning message. This was done to ensure interoperability with Juniper Networks Network and Security Manager. In the event that a warning message is received, simply review the message and resolve the error by correcting the configuration mistake.

Achieving HA with Interface Monitoring

To configure interface monitoring under any redundancy, an administrator would create an “`apply-macro monitor-interface`” stanza and optionally specify its weight as shown in Figure 3. If no weight is specified, it is assumed it is 255 and would trigger a failover in the event the interface fails. If the weight is not great enough the intermediate weight is noted in the configuration under the stanza “`apply-macro failover-interface-monitor`” in the chassis cluster section. When a failover occurs, a system log message is generated as type external and level critical.

```

reth-count 4;
control-ports {
    fpc 2 port 0;
    fpc 14 port 0;
}
apply-macro failover-interface-monitor {
    0 128;
}
redundancy-group 0 {
    apply-macro monitor-interfaces {
        xe-17/1/0;
        xe-5/1/0 128;
        xe-5/1/2 128;
    }
    node 0 priority 254;
    node 1 priority 1;
}
redundancy-group 1 {
    apply-macro track-host {
        interval 0.5;
        routing-instance BPSTest1;
        server 1.0.11.11;
        weight 255;
    }
    apply-macro monitor-interfaces {
        xe-17/1/0;
        xe-5/1/0;
        xe-5/2/0;
    }
    node 0 priority 254;
    node 1 priority 1;
}

```

Figure 4: Interface monitoring configuration example

To allow the interface monitoring to take effect, the event options stanza must be configured. This allows the interface monitoring to intercept the message "SNMP_TRAP_LINK_DOWN" and allows the monitor-interface script to act on the event. Using the configuration in Figure 5 activates the script to act on the interface down messages.

```

event-options {
    policy INTERFACE_MONITOR {
        events SNMP_TRAP_LINK_DOWN;
        then {
            event-script monitor-interface.xsl {
                arguments {
                    interface "${$.interface-name}";
                }
            }
        }
    }
    event-script {
        file monitor-interface.xsl;
    }
}

```

Figure 5: Interface monitoring configuration example

Validating the Interface Monitoring Configuration

The configuration options that begin with “apply-macro” are all user-created options. Because of this, Junos OS does not validate these by default. To create custom validation options a commit script must be used. To validate the monitor interface configuration the commit script “srx-ha-validate.xml” was created. The script must be placed in the /var/db/scripts/commit directory. Secondly, it must be added to the Junos OS configuration by using the command “set system scripts commit file srx-ha-validate.xml.” Upon the committing of this configuration, the options for monitor interface are validated.

This prevents misspelling items such as “interface” or giving numbers that are out of the range of the scripts’ capabilities. In the event that an option is not correctly configured, a warning is emitted. This does notify the administrator that something is not right. It does not prevent the misconfiguration of monitor interface—it just creates a warning message. This was done to ensure interoperability with the NSM platform. In the event that a warning message is received, simply review the message and resolve the error by correcting the configuration mistake.

Global SRX Series HA Configuration Options

Under the chassis cluster configuration the macro “monitoring-options” with the value of “clear-failover” can be applied. If this is configured, then after a failover of the redundancy group occurs the manual failover flag is cleared. This setting is optional. If it is not configured, then the manual failover needs to be cleared by the user. The second option that can be configured under the “monitoring-options” is the option “full-failover.” The full failover option triggers a full failover of all redundancy groups no matter which redundancy group failed its track IP checking. This option ensures that failed redundancy groups follow each other.

The design of the chassis cluster architecture is to allow the redundancy groups that pass data (redundancy group 1 and greater) to fail over between the cluster members as fast as possible to support the various changing conditions of the network. The control plane redundancy group 0 has some unique limitations that do not allow for this to occur. The design of the control plane redundancy group 0 is that upon boot it determines which chassis should be primary and sticks to that chassis member until a failure occurs. The two REs that are used, one per chassis, synchronize using the GRES mechanism. The graceful Routing Engine switchover design only allows the switching over of mastership between REs once per five minutes.

This is why RG0 is not meant to rapidly switch over between chassis and only in the event of a catastrophic failure. To prevent any graceful Routing Engine switchover synchronization issues, a timer has been implemented to stamp the last failover time for RG0. The timestamp shown in Figure 3 is in UNIX time (seconds since January 1, 1970). It is set upon the first failover of RG0.

```
chassis {
  cluster {
    apply-macro monitoring-options {
      clear-failover;
      full-failover;
    }
    apply-macro failover-monitoring {
      last-failover 1228859971;
    }
  }
}
```

Figure 6: Chassis cluster configuration example

While the chassis cluster technology is very robust it is not always aligned with the operational procedures of organizations. Because of this, the track IP Junos automation script accommodates these requests by also implementing the “follow the leader feature.” This feature that is enabled by default forces RG0 to go where RG1 is located. This would have occurred if RG0 were unable to fail over because it had been less than five minutes from the last failover. When the difference between the last failover and the current time is more than 300 seconds (five minutes), then RG0 automatically fails over to the node where RG1 is located.

Summary

To ensure network resiliency, Juniper Networks SRX Series Services Gateways maintain many checks to ensure their availability. The inclusion of track IP increases availability by allowing for monitoring of network hosts and upstream routers. This feature has been implemented in Junos automation because it was not currently available in Junos OS. The Interface monitoring feature on the SRX Series can correctly monitor and fail over redundancy groups in the event of an interface failure. However, because some customers require the monitoring of the control plane—and want to prevent any instability issues because of rapid failure of the control plane—the interface monitoring Junos automation script prevents the failover of the control plane in periods of less than five minutes.

About Juniper Networks

Juniper Networks, Inc. is the leader in high-performance networking. Juniper offers a high-performance network infrastructure that creates a responsive and trusted environment for accelerating the deployment of services and applications over a single network. This fuels high-performance businesses. Additional information can be found at www.juniper.net.

Corporate and Sales Headquarters

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or 408.745.2000
Fax: 408.745.2100
www.juniper.net

APAC Headquarters

Juniper Networks (Hong Kong)
26/F, Cityplaza One
1111 King's Road
Taikoo Shing, Hong Kong
Phone: 852.2332.3636
Fax: 852.2574.7803

EMEA Headquarters

Juniper Networks Ireland
Airside Business Park
Swords, County Dublin, Ireland
Phone: 35.31.8903.600
EMEA Sales: 00800.4586.4737
Fax: 35.31.8903.601

To purchase Juniper Networks solutions, please contact your Juniper Networks representative at 1-866-298-6428 or authorized reseller.

Copyright 2010 Juniper Networks, Inc. All rights reserved. Juniper Networks, the Juniper Networks logo, Junos, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners. Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.