



DC – AI – Routing – Switching – Security – Wireless – Automation

From SRX Chassis Cluster to MNHA Configurations side by side

An HPE Juniper Networking Techpost

Laurentp

2026-07-13

HPE



Contents

1	From SRX Chassis Cluster to MNHA – Configurations side by side	2
1.1	Introduction	2
1.2	Cluster initialization	4
1.3	Cluster Groups	6
1.4	Interfaces and Zones	6
1.5	Cluster Communication Encryption	8
1.6	Security - Source NAT	10
1.7	Special Configuration – DHCP server	10
1.8	Common Configuration – Security Policies	10
1.9	Verification – Cluster Status	10
1.10	Verification – Cluster Interfaces	13
1.11	Verification – Cluster Statistics	13
1.12	Conclusion	14
1.13	Useful Links	14
1.14	Glossary	15

1 From SRX Chassis Cluster to MNHA – Configurations side by side

Laurent Paumelle - 07/13/2026

After the techpost “SRX clustering: from Chassis Cluster to MultiNode High Availability”, let’s deep dive into their respective configuration side by side. Below a simplified diagram showing the same architecture with an SRX pair segmenting 2 networks:



In this series, also mention the other techposts from various people:

- [Multi-Node High Availability Basics](#) (from Steven Jacques)
- [Hybrid MNHA with eBGP](#) (from James Rathburn)
- [SRX clustering: from Chassis Cluster to MultiNode High Availability](#) (from Laurent Paumelle)
- [MNHA, IPSec and Multiple Routing Instances](#)
- [DHCP on MNHA: Back to Basics](#) (from James Rathburn)
- [SRX MNHA with VRRP](#) (from Karel Hendrych)

As quick reminder, I’ll use terms “CC” for “legacy Chassis Cluster” and “MNHA” for the new “MultiNode High Availability”. See techposts above for more details.

1.1 Introduction

The topology to study is a simple network interconnection with an SRX pair sandwiched between the Trust/Left (in green) and Untrust/Right networks (in blue). The network mode is simple, Default Gateway for the left network, matching the most used “reth” (redundant interface) of a Chassis Cluster, and also the Virtual IP (VIP) of MNHA.

The left diagram below represents a CC and its right counterpart for MNHA. Some links between the SRX pair represent the synchronization link and packet-forwarding link, when appropriate: CTRL and FAB for CC and ICL, and ICD for MNHA (ICD not shown below since not used in this setup).

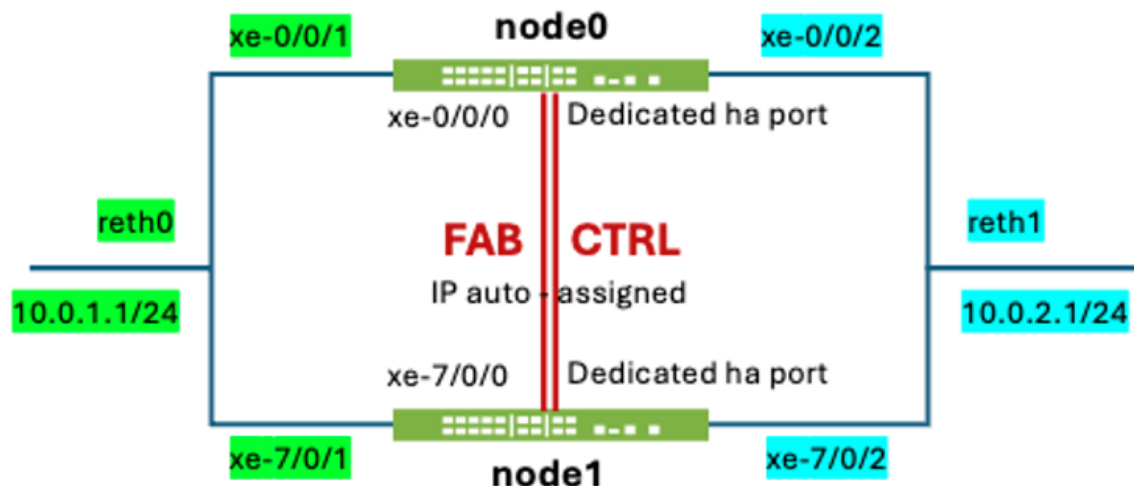
The networking is the same on both sides to make comparison simpler:

- 10.0.1.0/24 for Trust/Left, default gateway being reth0/VIP 10.0.1.1
- 10.0.2.0/24 for Untrust/Right, common IP being reth0/VIP 10.0.2.1
- Not shown on below diagrams are the management interface (fxp0) but it appears in some config statements. Its network is 192.168.100.0/24

Notes for CC:

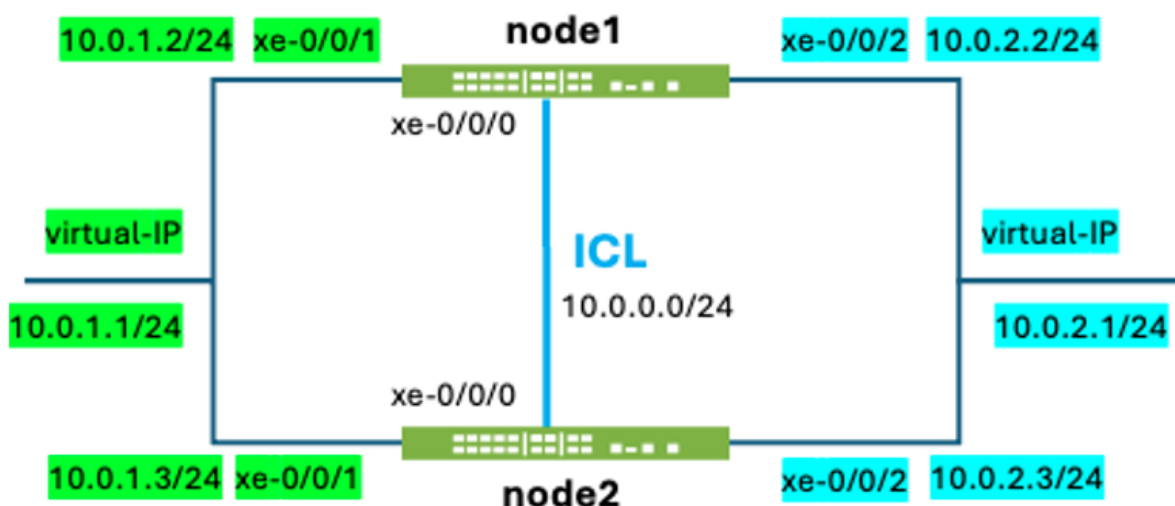
- Left/Right per device interfaces have no IP address (optional)

- CTRL and FAB links have requirements: dedicated L2 segment, jumbo frame support (mtu > 9000 bytes), disable igmp-snooping, preserve vlan-id 4094.
- CTRL link is used for state control between the 2 devices
- FAB link is used for session synchronization, and in the asymmetric case also forwards packets (Z-mode)
- Node numbering is node0 and node1



Notes for MNHA:

- Left/Right interfaces have an IP address as they can be used for monitoring and/or peering with BGP (not mandatory though)
- ICL is just a regular IP link and has no specific requirement and can be also routed, and IPsec encrypted.
- There is no dedicated ICL link so any interface (or sub-interface) can be used.
- If used, ICD helps handle asymmetric session establishments (and then have a slightly larger MTU with its UDP header)
- Node numbering is node1 and node2



The configurations below will show side by side (when not too large) for CC and MNHA. For CC, since it uses a single active Routing Engine (RE, aka management), there is no duplication of commands/configs across nodes (except at initial setup), as the chassis is “extended” between the 2 SRX. For MNHA, it’s often split into node1 and node2 commands and config as each has independent configurations. An optional mechanism exists in Junos to synchronize part of the configuration (commit sync).

Color used to make things clearer (hopefully):

- Green for trust/left side
- Blue for untrust/right side
- Yellow for cluster specific elements (but important to highlight)
- Pink for some other cluster IP elements

The content below will be split into 2 columns to compare configurations and outputs side by side:

CC - topic	MNHA - topic
Content1	Content2

1.2 Cluster initialization

To build a Chassis Cluster, we need to set initial commands and reboot. Same for MNHA, but in reverse order, it starts with the initial configuration, then reboot.

CC - Initial setup	MNHA - Initial setup
<pre># Before configuring, setting up the chassis cluster command and reboot. Operational commands (not in edit): # Node0 user@srx1> set chassis cluster cluster-id 1 node 0 reboot # Node1 user@srx2> set chassis cluster cluster-id 1 node 1 reboot</pre>	<pre># Initial configuration on both nodes then reboot: # Node1: ICL user@srx1> edit set system host-name mnha-srx1 set interfaces xe-0/0/0 unit 0 family inet address 10.0.0.1/24 set chassis high-availability local-id 1</pre>

	<pre> set chassis high-availability local-id local-ip 10.0.0.1 set chassis high-availability peer-id 2 peer-ip 10.0.0.2 set chassis high-availability peer-id 2 interface xe-0/0/0 set chassis high-availability peer-id 2 liveness-detection minimum-interval 300 set chassis high-availability peer-id 2 liveness-detection multiplier 3 # ICL zone: set security zones security-zone icl host-inbound-traffic system-services ping set security zones security-zone icl host-inbound-traffic system-services high-availability set security zones security-zone icl host-inbound-traffic system-services ike set security zones security-zone icl host-inbound-traffic protocols bfd set security zones security-zone icl host-inbound-traffic protocols bgp set security zones security-zone icl interfaces xe-0/0/0 commit and-quit user@ srx1> request system reboot </pre>
<pre> # After reboot configure chassis cluster on primary node only (single RE used): user@ srx1> edit set chassis cluster control-link-recovery set chassis cluster redundancy-mode active-backup # 2 reths (trust and untrust): set chassis cluster reth-count 2 # Dedicated interfaces already used for HA Control link # Set below interfaces for HA Fab link: set interfaces fab0 fabric-options member-interfaces xe- 0/0/0 set interfaces fab1 fabric-options member-interfaces xe- 7/0/0 # Node0 management: set groups node0 system host-name cluster-srx1 set groups node0 interfaces fxp0 unit 0 family inet address 192.168.100.33/24 set groups node0 system backup-router 192.168.100.1 destination 0.0.0.0/0 # Node1 management: set groups node1 system host-name cluster-srx2 set groups node1 interfaces fxp0 unit 0 family inet address 192.168.100.34/24 set groups node1 system backup-router 192.168.100.1 destination 0.0.0.0/0 # Apply to both nodes: set apply-groups "\${node}" commit </pre>	<pre> # Node2: ICL user@ srx2> edit set system host-name mnha-srx2 set interfaces xe-0/0/0 unit 0 family inet address 10.0.0.1/24 set chassis high-availability local-id 2 set chassis high-availability local-id local-ip 10.0.0.2 set chassis high-availability peer-id 1 peer-ip 10.0.0.1 set chassis high-availability peer-id 1 interface xe-0/0/0 set chassis high-availability peer-id 1 liveness-detection minimum-interval 300 set chassis high-availability peer-id 1 liveness-detection multiplier 3 # ICL zone: set security zones security-zone icl host-inbound-traffic system-services ping set security zones security-zone icl host-inbound-traffic system-services high-availability set security zones security-zone icl host-inbound-traffic system-services ike set security zones security-zone icl host-inbound-traffic protocols bfd set security zones security-zone icl host-inbound-traffic protocols bgp set security zones security-zone icl interfaces xe-0/0/0.0 commit and-quit user@ srx2> request system reboot </pre>
	<pre> # Same for node1 and node2: # Optional routing instance for ICL: set chassis high-availability peer-id 1 routing-instance MNHA-ICL set routing-instances MNHA-ICL instance-type virtual-router set routing-instances MNHA-ICL interface xe-0/0/0.0 </pre>

1.3 Cluster Groups

CC uses Redundancy Groups (RG) and MNHA Secure Redundancy Groups (SRG), but their meanings are (almost) the same: grouping functions together.

CC – Cluster Groups	MNHA - Cluster Groups
<pre># Primary group for Control Plane (RG0): set chassis cluster redundancy-group 0 node 0 priority 100 set chassis cluster redundancy-group 0 node 1 priority 90 # Second group for Data Plane (RG1): set chassis cluster redundancy-group 1 node 0 priority 100 set chassis cluster redundancy-group 1 node 1 priority 90</pre>	<pre># Node1: # SRG0: set chassis high-availability services-redundancy-group 0 peer-id 2 # SRG1: set chassis high-availability services-redundancy-group 1 deployment-type switching set chassis high-availability services-redundancy-group 1 mode active-backup set chassis high-availability services-redundancy-group 1 peer-id 2 set chassis high-availability services-redundancy-group 1 activeness-priority 100</pre>
	<pre># Node2: # SRG0: set chassis high-availability services-redundancy-group 0 peer-id 1 # SRG1: set chassis high-availability services-redundancy-group 1 deployment-type switching set chassis high-availability services-redundancy-group 1 mode active-backup set chassis high-availability services-redundancy-group 1 peer-id 2 set chassis high-availability services-redundancy-group 1 activeness-priority 90</pre>
<pre># Monitoring interfaces (global): set chassis cluster redundancy-group 1 interface-monitor xe-0/0/1 weight 128 set chassis cluster redundancy-group 1 interface-monitor xe-0/0/2 weight 128</pre>	<pre># Same on node1 and node2: # Monitoring interfaces (per node and SRG): set chassis high-availability services-redundancy-group 1 monitor interface xe-0/0/1 set chassis high-availability services-redundancy-group 1 monitor interface xe-0/0/2</pre>

1.4 Interfaces and Zones

By design, CC uses redundant interfaces between the 2 chassis. But the syntax is not “aggregate interface” (ae) but instead “reth”. The common IP are set on reth0 (trust) and reth1 (untrust).

We do not use the ae or reth equivalents in MNHA, since they do not need this syntax. We use a Virtual IP (VIP) instead, which fails over using a gratuitous ARP. Note that aggregate interfaces are possible, but per node (like xe-0/0/5 and xe-0/0/6 together in ae0 bundle), not across nodes as reth does.

Also, MNHA can use an IP address on its physical interfaces, mostly used for monitoring or doing dynamic routing with peer routers. Those are set below.

CC – Redundant Interfaces (reth)	MNHA - Interfaces
<pre># AE/RETH for trust interface: set interfaces xe-0/0/1 gigether-options redundant-parent reth0 set interfaces xe-7/0/1 gigether-options redundant-parent reth0 # AE/RETH for untrust interface: set interfaces xe-0/0/2 gigether-options redundant-parent reth1 set interfaces xe-7/0/2 gigether-options redundant-parent reth1</pre>	
<pre># IPv4 address for trust interface: set interfaces reth0 unit 0 family inet address 10.0.1.1/24 # IPv4 address for untrust interface: set interfaces reth1 unit 0 family inet address 10.0.2.1/24</pre>	<pre># Node1: # IPv4 address for trust interface: set chassis high-availability services-redundancy-group 1 virtual-ip 1 ip 10.0.1.1/24 set chassis high-availability services-redundancy-group 1 virtual-ip 1 interface xe-0/0/1.0 set chassis high-availability services-redundancy-group 1 virtual-ip 1 use-virtual-mac # IPv4 address for untrust interface: set chassis high-availability services-redundancy-group 1 virtual-ip 2 ip 10.0.2.1/24 set chassis high-availability services-redundancy-group 1 virtual-ip 2 interface xe-0/0/2.0 set chassis high-availability services-redundancy-group 1 virtual-ip 2 use-virtual-mac</pre>
	<pre># Node2: # IPv4 address for trust interface: set chassis high-availability services-redundancy-group 1 virtual-ip 1 ip 10.0.1.1/24 set chassis high-availability services-redundancy-group 1 virtual-ip 1 interface xe-0/0/1.0 set chassis high-availability services-redundancy-group 1 virtual-ip 1 use-virtual-mac # IPv4 address for untrust interface: set chassis high-availability services-redundancy-group 1 virtual-ip 2 ip 10.0.2.1/24 set chassis high-availability services-redundancy-group 1 virtual-ip 2 interface xe-0/0/2.0 set chassis high-availability services-redundancy-group 1 virtual-ip 2 use-virtual-mac</pre>
<pre># Security Zones: # Trust zone: set security zones security-zone trust tcp-rst</pre>	<pre># Same on node1 and node2: # Trust zone: set security zones security-zone trust tcp-rst</pre>
<pre>set security zones security-zone trust host-inbound-traffic system-services ping set security zones security-zone trust interfaces reth0.0 # Untrust zone: set security zones security-zone untrust host-inbound-traffic system-services ping set security zones security-zone untrust interfaces reth1.0</pre>	<pre>set security zones security-zone trust host-inbound-traffic system-services ping set security zones security-zone trust interfaces xe-0/0/1.0 # Untrust zone: set security zones security-zone untrust host-inbound-traffic system-services ping set security zones security-zone untrust interfaces xe-0/0/2.0</pre>

Note: if using IPv4 and IPv6 addressing, this is just an additional family inet6 for CC. But for MNHA, IPv4 and IPv6 need to be on different sub-interfaces (VLAN tagged), except if using VRRP.

Note: this should be solved in Junos 26.2R1 with an IPv4 and IPv6 on same VIP.

CC – adding IPv6	MNHA - adding IPv6
<pre># IPv6 for trust interface: set interfaces reth0 unit 0 family inet6 address fd00:2026::10:0:1:1/64 # IPv6 for trust interface: set interfaces reth1 unit 0 family inet6 address fd00:2026::10:0:2:1/64</pre>	<pre># Node1: # Adding a vlan tag to both IPv4 and IPv6 interfaces: set interfaces xe-0/0/0 vlan-tagging set interfaces xe-0/0/0 unit 0 vlan-id 4 # IPv6 for trust interface: set interfaces xe-0/0/0 unit 1 vlan-id 6 set interfaces xe-0/0/0 unit 1 family inet6 address fd00:2026::10:0:1:2/64 # SRG1 IPv6 VIP on trust interface: set chassis high-availability services-redundancy-group 1 virtual-ip 3 ip fd00:2026::10:0:1:1/64 set chassis high-availability services-redundancy-group 1 virtual-ip 3 interface xe-0/0/1.1 set chassis high-availability services-redundancy-group 1 virtual-ip 3 use-virtual-mac # IPv6 for untrust interface: set interfaces xe-0/0/1 unit 1 vlan-id 6 set interfaces xe-0/0/1 unit 1 family inet6 address fd00:2026::10:0:2:2/64 # SRG1 IPv6 VIP on untrust interface: set chassis high-availability services-redundancy-group 1 virtual-ip 4 ip fd00:2026::10:0:2:1/64 set chassis high-availability services-redundancy-group 1 virtual-ip 4 interface xe-0/0/2.1 set chassis high-availability services-redundancy-group 1 virtual-ip 4 use-virtual-mac</pre>
	<pre># Node2: # Adding a vlan tag to both IPv4 and IPv6 interfaces: set interfaces xe-0/0/0 vlan-tagging set interfaces xe-0/0/0 unit 0 vlan-id 4 # IPv6 for trust interface: set interfaces xe-0/0/0 unit 1 vlan-id 6 set interfaces xe-0/0/0 unit 1 family inet6 address fd00:2026::10:0:1:3/64 set security zones security-zone trust interfaces xe-0/0/0.1</pre>

1.5 Cluster Communication Encryption

MNHA natively supports IPsec encryption for its session synchronization link, also known as the ICL (Inter-Chassis Link). This needs the proper IKE daemon to be loaded (which is default in all latest Junos) or can be loaded manually.

CC – None	MNHA - IPsec Encryption
	<pre># Node1 and node2 for IPsec encrypted ICL: user@host> request system software add optional://junos-ike.tgz # Common to both nodes: set groups mnha-sync-group security ike proposal MNHA-prop description MNHA-ICL set groups mnha-sync-group security ike proposal MNHA-prop authentication-method pre-shared-keys set groups mnha-sync-group security ike proposal MNHA-prop dh-group group16 set groups mnha-sync-group security ike proposal MNHA-prop encryption-algorithm aes-256-gcm set groups mnha-sync-group security ike proposal MNHA-prop lifetime-seconds 300 set groups mnha-sync-group security ike policy MNHA-pol description iclpolicy set groups mnha-sync-group security ike policy MNHA-pol proposals MNHA-prop set groups mnha-sync-group security ike policy MNHA-pol pre-shared-key ascii-text "\$somePSK..." set groups mnha-sync-group security ike gateway MNHA-gw ike-policy MNHA-pol set groups mnha-sync-group security ike gateway MNHA-gw version v2-only</pre>
	<pre>set groups mnha-sync-group security ipsec proposal MNHA-prop description ICL-esp set groups mnha-sync-group security ipsec proposal MNHA-prop protocol esp set groups mnha-sync-group security ipsec proposal MNHA-prop encryption-algorithm aes-256-gcm set groups mnha-sync-group security ipsec proposal MNHA-prop lifetime-seconds 300 set groups mnha-sync-group security ipsec policy MNHA-pol description icl-ph2-pol set groups mnha-sync-group security ipsec policy MNHA-pol perfect-forward-secrecy keys group16 set groups mnha-sync-group security ipsec policy MNHA-pol proposals MNHA-prop set groups mnha-sync-group security ipsec vpn MNHA-vpn ha-link-encryption set groups mnha-sync-group security ipsec vpn MNHA-vpn ike gateway MNHA-gw set groups mnha-sync-group security ipsec vpn MNHA-vpn ike ipsec-policy MNHA-pol set apply-groups mnha-sync-group</pre>
	<pre># Node1: set groups mnha-vpn-peer1 chassis high-availability peer-id 2 vpn-profile MNHA-vpn set apply-groups mnha-vpn-peer1 # Node2: set groups mnha-vpn-peer2 chassis high-availability peer-id 1 vpn-profile MNHA-vpn set apply-groups mnha-vpn-peer2</pre>

Note: the IKE config above can also use certificates if any are present, but it is not required.

1.6 Security - Source NAT

Network Address Translation (NAT), uses the interface IP of the outbound interface on CC by default. But on MNHA, since each node has its own IP address, the VIP needs to be declared as a Source NAT Pool to serve the same purpose.

1.7 Special Configuration – DHCP server

When the SRX is used as a local DHCP Server for local machines (PCs, Wi-Fi, etc.), it simply uses reth0 on CC. In MNHA, the setting needs to be made on both nodes since the DHCP service is not synchronized (and not handled by a single process like CC). In case of a LAN DHCP Server or a DHCP Relay on SRX toward a DMZ-located DHCP Server (not on SRX), there are no special settings for MNHA, but the DHCP Relay still needs to be set up. Look at James's techpost DHCP on MNHA: Back to Basics, precisely on this subject, right à propos!

1.8 Common Configuration – Security Policies

We have seen the CC and MNHA-specific settings side by side, but the rest of the configuration remains the same. All security statements are usually identical between CC and MNHA for elements like: Zones, Policies, Addresses, UTM, IDP, Screens, ATP, Applications, etc.

CC and MNHA – Security Policies
<pre># Security policies: set security policies from-zone trust to-zone untrust policy trust-to-untrust-permit match source-address any set security policies from-zone trust to-zone untrust policy trust-to-untrust-permit match destination-address any set security policies from-zone trust to-zone untrust policy trust-to-untrust-permit match application any set security policies from-zone trust to-zone untrust policy trust-to-untrust-permit then permit set security policies from-zone trust to-zone untrust policy trust-to-untrust-permit then log session-close set security policies global policy global-last-deny match source-address any set security policies global policy global-last-deny match destination-address any set security policies global policy global-last-deny match application any set security policies global policy global-last-deny then deny set security policies global policy global-last-deny then log session-init set security policies default-policy deny-all set security policies pre-id-default-policy then log session-close</pre>

1.9 Verification – Cluster Status

After this setup, reboots, and finalizing the cluster elements, the verification can show the cluster status in each mode.

CC – Status	MNHA - Status																																								
# Primary node (single shared RE): user@srx1> show chassis cluster status Monitor Failure codes: <table><tr><td>CS Cold Sync monitoring</td><td>FL Fabric Connection monitoring</td></tr><tr><td>GR GRES monitoring</td><td>HW Hardware monitoring</td></tr><tr><td>IF Interface monitoring</td><td>IP IP monitoring</td></tr><tr><td>LB Loopback monitoring</td><td>MB Mbuf monitoring</td></tr><tr><td>NH Nexthop monitoring</td><td>NP NPC monitoring</td></tr><tr><td>SP SPU monitoring</td><td>SM Schedule monitoring</td></tr><tr><td>CF Config Sync monitoring</td><td>RE Relinquish monitoring</td></tr><tr><td>IS IRQ storm</td><td></td></tr></table> Cluster ID: 1 <table><tr><th>Node</th><th>Priority</th><th>Status</th><th>Preempt</th><th>Manual</th><th>Monitor-failures</th></tr><tr><td>node0</td><td>100</td><td>primary</td><td>no</td><td>no</td><td>None</td></tr><tr><td>node1</td><td>90</td><td>secondary</td><td>no</td><td>no</td><td>None</td></tr></table> Redundancy group: 0 , Failover count: 0 node0 100 primary no no None node1 90 secondary no no None Redundancy group: 1 , Failover count: 0	CS Cold Sync monitoring	FL Fabric Connection monitoring	GR GRES monitoring	HW Hardware monitoring	IF Interface monitoring	IP IP monitoring	LB Loopback monitoring	MB Mbuf monitoring	NH Nexthop monitoring	NP NPC monitoring	SP SPU monitoring	SM Schedule monitoring	CF Config Sync monitoring	RE Relinquish monitoring	IS IRQ storm		Node	Priority	Status	Preempt	Manual	Monitor-failures	node0	100	primary	no	no	None	node1	90	secondary	no	no	None	# Node1: user@srx1> show chassis high-availability information Node failure codes: <table><tr><td>HW Hardware monitoring</td><td>LB Loopback monitoring</td></tr><tr><td>MB Mbuf monitoring</td><td>SP SPU monitoring</td></tr><tr><td>CS Cold Sync monitoring</td><td>SU Software Upgrade</td></tr></table> Node Status: ONLINE Local-Id: 1 Local-IP: 10.0.0.1 HA Peer Information: Description: mnha-vsrx2 Peer Id: 2 IP address: 10.0.0.2 Interface: xe-0/0/0.0 Routing Instance: default Encrypted: YES Conn State: UP Configured BFD Detection Time: 3 * 300ms Cold Sync Status: COMPLETE Services Redundancy Group: 0 Current State: ONLINE	HW Hardware monitoring	LB Loopback monitoring	MB Mbuf monitoring	SP SPU monitoring	CS Cold Sync monitoring	SU Software Upgrade
CS Cold Sync monitoring	FL Fabric Connection monitoring																																								
GR GRES monitoring	HW Hardware monitoring																																								
IF Interface monitoring	IP IP monitoring																																								
LB Loopback monitoring	MB Mbuf monitoring																																								
NH Nexthop monitoring	NP NPC monitoring																																								
SP SPU monitoring	SM Schedule monitoring																																								
CF Config Sync monitoring	RE Relinquish monitoring																																								
IS IRQ storm																																									
Node	Priority	Status	Preempt	Manual	Monitor-failures																																				
node0	100	primary	no	no	None																																				
node1	90	secondary	no	no	None																																				
HW Hardware monitoring	LB Loopback monitoring																																								
MB Mbuf monitoring	SP SPU monitoring																																								
CS Cold Sync monitoring	SU Software Upgrade																																								

<pre>node0 100 primary yes no None node1 90 secondary yes no None (primary:node0)</pre>	Peer Information: Peer Id: 2 SRG failure event codes: - BF BFD monitoring - IP IP monitoring - IF Interface monitoring - CP Control Plane monitoring Services Redundancy Group: 1 - Deployment Type: SWITCHING Status: ACTIVE - Activeness Priority: 200 - Preemption: DISABLED - Process Packet In Backup State: NO - Control Plane State: READY - System Integrity Check: N/A - Failure Events: NONE Peer Information: - Peer Id: 2 Status: BACKUP - Health Status: HEALTHY - Fallover Readiness: READY
# No need on second node since a single RE runs at a time.	# Node2: user@srx2> show chassis high-availability information Node failure codes: - HW Hardware monitoring LB Loopback monitoring - MB Mbuf monitoring SP SPU monitoring - CS Cold Sync monitoring SU Software Upgrade Node Status: ONLINE Local-Id: 2 Local-IP: 10.0.0.2 HA Peer Information: Description: mnha-vsrx1 Peer Id: 1 IP address: 10.0.0.1 Interface: xe-0/0/0.0 Routing Instance: default Encrypted: YES Conn State: UP Configured BFD Detection Time: 3 * 300ms Cold Sync Status: COMPLETE Peer Forwarding IP: 10.0.0.1 Interface: xe-0/0/3.0 Peer ICD Conn State: UP Services Redundancy Group: 0 - Current State: ONLINE - Peer Information: Peer Id: 1 SRG failure event codes: - BF BFD monitoring - IP IP monitoring - IF Interface monitoring - CP Control Plane monitoring Services Redundancy Group: 1 - Deployment Type: SWITCHING Status: BACKUP - Activeness Priority: 100 - Preemption: DISABLED - Process Packet In Backup State: YES - Control Plane State: READY - System Integrity Check: COMPLETE - Failure Events: NONE Peer Information: - Peer Id: 1 Status: ACTIVE

	Health Status: HEALTHY Failover Readiness: N/A
--	---

1.10 Verification – Cluster Interfaces

CC – Cluster Interfaces	MNHA - Cluster Interfaces
# Primary node (single shared RE): <pre>{primary:node0} user@srx1> show chassis cluster interfaces Control link status: Up Control Interfaces: Index Interface Monitored-Status Internal-SA Security 0 em0 Up Disabled Disabled Fabric link status: Up Fabric Interfaces: Name Child-Interface Status Security (Physical/Monitored) fab0 ge-0/0/0 Up / Up Disabled fab0 fab1 ge-7/0/0 Up / Up Disabled fab1 Redundant-ethernet Information: Name Status Redundancy-group reth0 Up 1 reth1 Up 1 Redundant-pseudo-interface Information: Name Status Redundancy-group lo0 Up 0 Interface Monitoring: Interface Weight Status Redundancy-group (Physical/Monitored) xe-0/0/1 255 Up / Up 1 xe-0/0/2 255 Up / Up 1</pre>	# Same on node1 and node2: <pre>user@srx1> show chassis high-availability peer-info HA Peer Information: Description: mnha-vsrx2 Peer-ID: 2 IP address: 10.0.0.2 Interface: xe-0/0/0.0 Routing Instance: default Encrypted: YES Conn State: UP Cold Sync Status: COMPLETE Internal Interface: st0.16000 Internal Local-IP: 180.100.1.1 Internal Peer-IP: 180.100.1.2 Internal Routing-Instance: __juniper_private1__ Packet Statistics: Receive Error : 0 Send Error : 0 Packet-type Sent Received SRG Status Msg 10 9 SRG Status Ack 9 6 Attribute Msg 8 4 Attribute Ack 4 4</pre>

1.11 Verification – Cluster Statistics

CC – Statistics	MNHA - Statistics
# Primary node (single shared RE): <pre>user@srx1> show chassis cluster data-plane statistics Services Synchronized: Service name RT0s sent RT0s received Translation context 0 0 Incoming NAT 0 0 Resource manager 0 0 DS-LITE create 0 0 Session create 2466 0 IPv6 session create 0 0 IPv4/6 session RTO ACK 0 0 Session close 2443 0 IPv6 session close 0 0 Session change 383 0 IPv6 session change 0 0 ALG Support Library 0 0 Gate create 0 0 Session ageout refresh requests 0 67 IPv6 session ageout refresh requests 0 0 Session ageout refresh replies 57 0 IPv6 session ageout refresh replies 0 0 IPSec VPN 0 0 Firewall user authentication 0 0</pre>	# Node1 and same on Node2: <pre>user@srx1> show chassis high-availability data-plane statistics Services Synchronized: Service name RT0s sent RT0s received Translation context 0 0 Incoming NAT 0 0 Resource manager 0 0 DS-LITE create 0 0 Session create 3 39 IPv6 session create 0 0 IPv4/6 session RTO ACK 0 0 Session close 2 39 IPv6 session close 0 0 Session change 0 5 IPv6 session change 0 0 ALG Support Library 0 0 Gate create 0 0 Session ageout refresh requests 0 0 IPv6 session ageout refresh requests 0 0 Session ageout refresh replies 0 0 IPv6 session ageout refresh replies 0 0 IPSec VPN 0 0 Firewall user authentication 0 0</pre>

[_CUT_]			[_CUT_]		
JSF SLB	0	0	JSF SLB	0	0
APPID	3214	0	APPID	0	677
JSF MGCP MAP	0	0	JSF MGCP MAP	0	0
JSF H323 ALG	0	0	JSF H323 ALG	0	0
JSF RAS ALG	0	0	JSF RAS ALG	0	0
JSF SCCP MAP	0	0	JSF SCCP MAP	0	0
JSF SIP MAP	0	0	JSF SIP MAP	0	0
PST_NAT_CREATE	0	0	PST_NAT_CREATE	0	0
PST_NAT_CLOSE	0	0	PST_NAT_CLOSE	0	0
PST_NAT_UPDATE	0	0	PST_NAT_UPDATE	0	0
JSF TCP STACK	0	0	JSF TCP STACK	0	0
JSF IKE ALG	0	0	JSF IKE ALG	0	0
ALG RTO FRAG	0	0	ALG RTO FRAG	0	0
Packet stats	Pkts sent	Pkts	Packet stats	Pkts sent	Pkts
received			received		
ICD Data	0	0	ICD Data	0	0

1.12 Conclusion

Chassis Cluster and MNHA have been using different methods and settings from the beginning. However the concepts are quite close (RG vs SRG). Hoping this techpost will help compare and migrate from Chassis Cluster to MNHA; it was not meant to explore all possible options, but it should give the user some options. Another techpost would cover the other possible configurations in the 3 main network modes of MNHA: Default Gateway (L2), Hybrid (L2L3) and Routing (L3).

Also, this has been shown at the angle of console/cli access and configuration (easier to compare side by side than screenshots), but Security Director Cloud can also help with those settings; it would be another techpost to develop.

See the other tech posts below for other explanations and architectures.

1.13 Useful Links

Other techposts on MNHA:

- [Multi-Node High Availability Basics](#) (from Steven Jacques)
- [Hybrid MNHA with eBGP](#) (from James Rathburn)
- [SRX clustering: from Chassis Cluster to MultiNode High Availability](#) (from Laurent Paumelle)
- [MNHA, IPSec and Multiple Routing Instances](#)
- [DHCP on MNHA: Back to Basics](#) (from James Rathburn)
- [SRX MNHA with VRRP](#) (from Karel Hendrych)

High Availability = Chassis Cluster

- [Chassis Cluster Overview](#)
- [SRX Series Chassis Cluster Configuration Overview](#)
- [SRX Chassis Cluster Slot Numbering](#) (all SRX models)
- [SRX Getting Started - Configure Chassis Cluster \(High Availability\) - KB15650](#)
- [SRX HA Configuration Generator](#) (SRX Branch, vSRX, SRX1500, SRX4100, SRX4200)

Multi-Node High Availability

- [Multinode High Availability](#)
- [Example: Configure Multinode High Availability in a Default Gateway Deployment](#)

- [Example: Configure Multinode High Availability in a Hybrid Deployment](#)
- [Example: Configure Multinode High Availability in a Layer 3 Network](#)
- [Example: Configure IPSec VPN in Active-Active Multinode High Availability in a Layer 3 Network](#)
- [Asymmetric Traffic Flow Support in Multinode High Availability](#)

1.14 Glossary

Terminology used for Cluster:

- CC = “Chassis Cluster”
- L2HA = Layer 2 High Availability = Chassis Cluster
- CTRL = Control Link (dedicated interface)
- FAB = Fabric Link (dedicated interface)
- RG = Redundancy Group
- RTO = Real Time Objects (states to synchronize: sessions, NAT, ALG, IPsec...)

Terminology used for MNHA:

- MNHA = MultiNode High Availability
- ICL = Inter Chassis Link (similar to CTRL link)
- ICD = Inter Chassis Datapath (similar to FAB link)
- IDL = Inter Domain Link (more than 2 nodes scenario)
- SRG = Service Redundancy Group (similar to RG)
- RTO = Real Time Objects

Junos Terminology:

- RE = Routing Engine on Junos
- NSR = Non Stop Routing (allowing protocols such as BGP to hold its route states when RE restarts)
- ISSU = In Service Software Update (updating the 2 nodes of a cluster, one at a time, without disturbing traffics)

Network Layers:

- L2 = simple Layer 2 network, i.e. same IP broadcast domain
- L3 = Layer 3, refers to routed networks
- L2L3 = mix of Layer 2 and Layer 3, also named Hybrid
- L3-L7 = L3 + Layer 4 (TCP/UDP/ICMP/...) to Layer 7 (applications such as HTTP/S, DNS, Google, AWS, etc...)