Bgp route reflector clusters

BGP route reflector clusters — BGP route reflectors used as an alternative method for full use of the mesh IBGP shall assist in scaling. The BGP route reflector cluster is used to provide redundancy in the BGP RR theme. BGP route reflectors and RR customers create a cluster. (Cluster = DAB RR + BGP RR clients) I'm explaining this topic in deep detail in my own, instructor led CCIE Enterprise Training, instructor led CCDE, Self Paced CCDE and also my specialized BGP Zero to Hero course. In IBGP topology, each BGP speaker must be logical in full mesa. So, each BGP router must be a direct IBGP peer. However, the route reflector is an exception. If you insert a BGP route reflector, the IBGP router will only create a BGP neighboring vessel with route reflectors. In this article, I will specifically mention the set of the route reflector and its design. For those who want to understand BGP Route Reflectors, I highly recommend my BGP Route Reflector in plain English post. What is bgp route reflector cluster ID ? The route reflector cluster ID is a four-byte BGP attribute and is taken from the BGP router ID of the route reflector by default. If two routers have the same BGP cluster ID, they belong to the same cluster. Before the route reflection, route reflectors add a cluster ID to the cluster list. If the route originates from the route reflector itself, the route reflector shall not draw up a cluster list. If the route is sent to an EBGP peer, RR removes the cluster list information. If the route is received from an EBGP peer, RR does not create a cluster list attribute. Why was the cluster list used ? The cluster list is used to prevent loops only route reflectors. The route reflector for customers does not use the cluster list attribute, so they don't know which cluster they belong to. If there are two route reflectors, are the route reflectors better the same or different in the IF cluster? If RR receives routes from an IBGP neighbor that has the same cluster ID, the routes are discarded. Let's start with the basic topology. The BGP route reflector cluster itself, CLuster ID Figure 1, uses the same cluster id in the diagram shown in Figure 1 above, R1 and R2 are route reflectors, and R3 and R4 are RR customers. Both route reflectors use the same cluster ID. Green lines represent physical connections. Red lines show IBGP connections. Assume that we use both route reflectors as cluster ID 1.1.1.1, which is R1 for router ID R1, and R2 to receive routes from R4. R1 and R2 receive routes from R3. Both R1 and R2 add 1.1.1.1 as route reflectors as cluster ID attributes that they send to each other. However, since they use the same set, they throw away routes from each other. Therefore, if RRs use the same cluster ID, RR clients must connect to both RRs. from R2). Of course, the IGP path passes through R1-R2-R4, because there is no physical path between R1-R4. If the physical link R1 and R2 go down, both IBGP sessions between R1-R4 and R2-R4 go down as well. Thus, the networks behind R4 can not be learned. Since routes can't be learned from R2 (same cluster ID), if the physical link is up and the IBGP session goes down between R1 and R4, the networks behind R4 won't be reachable, but if you have a BGP neighbor between feedback loops and physical topology is redundant, the possibility of IBGP sessions going down is very difficult. Note: Unnecessary physical links are common network design best practices. That's why below topology is a more realistic one. What if we add a physical link between R1-R4 and R2-R3? When a BGP route reflector shares the same cluster ID with an excessive redundancy image-2 route reflector uses the same cluster ID, the physical cross-connection is added between RR and RR clients in Figure 2 as physical cross-connections are added between R1 R4 and R2-R3. However, we use the same BGP cluster ID for route reflectors. Thus, if R2 forwards R4 routes to R1, R1 will abandon these routes. In addition, R1 will learn R4 routes through direct IBGP peering with R4. In this case, the IGP path will change to R1-R4 instead of R1-R2-R4. In a situation in which the R1 R4 physical link fails, the IBGP session does not go down if IGP converges the R1 R2 R4 path faster than the BGP session timeout (by default it). Thus, having the same cluster ID saves a lot of memory and CPU resource route reflectors even if link failures do not cause IBGP sessions to drop if there are enough redundancies on the network. If we could use other BGP Cluster ID in R1 and R2, R1 would accept reflected routes from R2 in addition to routes from direct peering to R4. Conclusion: Orhan Ergun recommends the same BGP cluster ID for route reflector redundancy if there is a resource problem for route reflectors. If there are no resource problems, having different Cluster IDs provides faster convergence in some cases (Dependent on topology) Otherwise, Route Reflectors will save an additional copy of each prefix that should not be advertised to Route Reflector customers anyway. To have a great understanding of SP Networks, you can check your new published Service Provider Networks Design and Architecture Perspective Book. This topic discusses route reflectors to simplify configuration and scaling support. Another way to reduce the workload of a route reflector other than the traffic transfer path is to use the non-install notification [edit protocols in the BGP family last name] hierarchy level. Starting with Junos OS Release 15.1, the non-install notification prevents interaction between the Routing Protocol daemon (rpd) and other components of the Junos system, such as kernel or distributed firewall daemon (dfwd). This interaction is prevented by preventing the publication of these components in all routing-related RPP route information bases (RIB), also known as routing tables. Note Release releases before In Junos OS release 15.1, you can reduce the workload of a route reflector that is not a traffic transfer route by using the expedition table export policy, which rejects routes made from BGP. Due to the full network requirement of BGP (IBGP), most networks use route reflectors to simplify configuration. The formula to calculate the number of sessions required for full eye is v * (v - 1)/2, where v is the number of BGP-enabled devices. The full mesh pattern doesn't scale well. Using a route reflector, routers are grouped into clusters that are identified by numeric identifiers unique to the Autonomous System (AS). In the cluster, the BGP session must be configured from one router (route reflector) to each internal peer. This configuration fulfills the requirement for an IBGP full network. To use the as-to-route reflection, specify one or more routers as a route reflector, usually one for each presence point (POP). Route reflectors have a special BGP ability to readvertise routes made from the inner peer of other internal peers. Instead of requiring all internal peers to be fully mesaly with each other, the route reflection of the route requires only that the route reflector is fully uneeded by all internal peers. The route reflector and all its internal peers form the cluster as shown in Figure 1. NoteDemonny device network devices on each device that uses a route reflector must have an additional BGP feature license installed. For license information, see the software installation and upgrade guide.1. Figure 1: Single-route reflector topology (single cluster)Figure 1 shows that the router RR is configured as a reflector for the 127-route cluster. Other routers are selected by internal peers in the cluster. BGP routes are advertised on Router RR by one of their internal peers. RR then readvertises these routes to all other peers within the cluster. You can configure multiple clusters and link them by configuring the full screen of route reflectors (see Figure 2). Figure 2: The reflection of the route (multiple clusters) in Figure 2 shows the route reflectors RR 1, RR 2, RR 3 and RR 4 as fully eye-to-eye peers. When the router advertises a route to RR 1, RR 1 reads the route to other route reflectors, which in turn reads the route to the remaining routers. Route reflection allows you to reproduce the route throughout the AS without scaling problems caused by the full network requirement. Note A route reflector that supports multiple clusters does not accept a route with the same cluster ID from a router other than the client router. Therefore, you must configure a different cluster ID for redundant RR to reflect the route to other clusters. However, as the clusters become large, it is difficult to scale a full mesh with a route reflector, as is the full mesh net between the reflectors of the route. To help fix this problem, you can group your route reflector clusters into a hierarchical route reflection to group (see Figure 3). Figure 3: The reflection of the route (cluster clusters) in Figure 3 for clusters 2, RR 3 and RR 4 as route reflectors for clusters 127, 19 and 45 respectively. Instead of fully lining these route reflectors, the network administrator has configured them as part of another cluster (Set 6) for which the route reflector is RR 1. When the router advertises a route to RR 2, RR 2 readvertises the route to all routers in its cluster, and then readvertises the route from RR 1. RR 1 readvertises the route of the route to its cluster, and these routers spread the route down using their cluster. Example: Configuring a route reflectorThis example shows how to configure the route reflector. RequirementsOverviewConfigurationVerificationRequirements No special configuration is required before configuring this example after the device is initialized. The General Internal BGP (IBGP) enabled devices are fully eye-enabled because IBGP does not readvertise updates to other IBGP-enabled devices. A full eye is a logical eye achieved by configuring multiple neighboring notifications for each IBGP-enabled device. A full eye is not always physically full of mesh. Keeping a full eye (logical or physical) does not scale well in large placements. Figure 4 shows the IBGP network with a device A acting as a route reflector. Device B and device C are customers of the route reflector. Device D and device E are located outside the cluster, so they are not the adjacent class of the route reflector. Device A (route reflector) must have peer-to-peer relationships with all IBGP-enabled devices, including a neighbor's notification to customers (device B and device C) and nonclients (device D and device E). The cluster notification and cluster identifier must also be included. The cluster identifier can be any 32-bit value. This example uses the IP address of the route reflector LOOP interface. On device B and device C, route reflector customers, you only need one neighbor's notification that forms a peer-to-peer relationship with the route reflector, the A.On Device D and device E, nonclients, you need a neighbor's notification for each non-client device (No-E and E-to-D). You also need a neighboring notification route reflector (D-to-A and E-to-A). Device D and device E do not require neighboring notifications for client devices (device B and device C). TipDevices D and device E are considered neclients because they are clearly configured in peer-to-peer relationships between themselves. To make them RR client reflector clients, remove neighbor 192.168.5.5 notification from configuration device D and remove neighbor 192.168.0.1 notification from configuration device E. Figure 4: IBGP network, using route ReflectorConfigurationCLI Quick ConfigurationTo quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details needed to match the network configuration, and then copy and paste the commands to the CLI at the [edit] hierarchy level.set interfaces fe-0/0/0 unit 1 description to-Bset interfaces fe-0/0/0 unit 1 family inet address 10.10.10.1/30set interfaces fe-0/0/1 unit 3 description to-Dset interfaces fe-0/0/1 unit 3 family inet address 10.10.10.9/30set interfaces lo0 unit 1 family inet address 192.168.6.5/32set protocols bgp group internal-peers type internalset protocols bgp group internal-peers local-address 192.168.6.5set protocols bgp group internal-peers export send-ospfset protocols bgp group internal-peers neighbor 192.163.6.4set protocols bgp group internal-peers cluster 10.10.10.1set protocols bgp group internal-peers neighbor 192.168.40.4set protocols bgp group internal-peers neighbor 192.168.0.1set protocols ospf area 0.0.0.0 interface lo0.1 passiveset protocols ospf area 0.0.0.0 interface fe-0/0/0.1set protocols ospf area 0.0.0.0 interface fe-0/0/1.3set policy-options policy-statement send-ospf term 2 from protocol ospfset policy-options policy-statement send-ospf term 2 then acceptset routing-options router-id 192.168.6.5set routing-options autonomous-system 17set interfaces fe-0/0/0 unit 2 description to-Aset interfaces fe-0/0/0 unit 2 family inet address 10.10.10.2/30set interfaces fe-0/0/1 unit 5 description to-Cset interfaces fe-0/0/1 unit 5 family inet address 10.10.10.5/30set interfaces lo0 unit 2 family inet address 192.163.6.4/32set protocols bgp group internal-peers type internalset protocols bgp group internal-peers local-address 192.163.6.4set protocols bgp group internal-peers export send-ospfset protocols bgp group internal-peers neighbor 192.168.6.5set protocols ospf area 0.0.0.0 interface lo0.2 passiveset protocols ospf area 0.0.0.0 interface fe-0/0/0.2set protocols ospf area 0.0.0.0 interface fe-0/0/1.5set policy-options policy-statement send-ospf term 2 from protocol ospfset policy-options policy-statement send-ospf term 2 then acceptset routing-options router-id 192.163.6.4set routing-options autonomous-system 17set interfaces fe-0/0/0 unit 6 description to-Bset interfaces fe-0/0/0 unit 6 family inet address 10.10.10.6/30set interfaces lo0 unit 3 family inet address 192.168.40.4/32set protocols bgp group internal-peers type internalset protocols bgp group internal-peers local-address 192.168.40.4set protocols bgp group internal-peers export send-ospfset protocols bgp group internal-peers neighbor 192.168.6.5set protocols ospf area 0.0.0.0 interface lo0.3 passiveset protocols ospf area 0.0.0.0 interface fe-0/0/0.6set policy-options policy-statement send-ospf term 2 from protocol ospfset policy-options policy-statement send-ospf term 2 then acceptset maršrutēšanas opcijas router-id 192.168.40.4set maršrutēšanas opcijas autonomās sistēmas 17set saskarnes fe-0/0/0 vienība 4 apraksts to-Aset saskarnes fe-0/0/0 vienība 4 saimes inet adrese 10.10.10.10/30set saskarnes fe-0/0/1 vienība 7 apraksts to-Eset saskarnes fe-0/0/1 vienība 7 ģimenes inet adrese interfaces lo0 unit 4 family inet address 192.168.0.1/32set protocols bgp group internal-peers type internalset protocols bgp group internal-peers local-address 192.168.0.1set protocols bgp group internal-peers export send-ospfset protocols bgp group internal-peers neighbor 192.168.6.5set protocols bgp group internal-peers neighbor 192.168.5.5set protocols ospf area 0.0.0.0 interface lo0.4 passiveset protocols ospf area 0.0.0.0 interface fe-0/0/0.4set protocols ospf area 0.0.0.0 interface fe-0/0/1.7set policy-options policy-statement send-ospf term 2 from protocol ospfset policy-options policy-statement send-ospf term 2 then acceptset routing-options router-id 192.168.0.1set routing-options autonomous-system 17set interfaces fe-0/0/1 unit 8 description to-Dset interfaces fe-0/0/1 unit 8 family inet address 10.10.10.14/30set interfaces lo0 unit 5 family inet address 192.168.5.5/32set protocols bgp group internal-peers type internalset protocols bgp group internal-peers local-address 192.168.5.5set protocols bgp group internal-peers export send-ospfset protocols bgp group internal-peers neighbor 192.168.6.5set protocols bgp group internal-peers neighbor 192.168.0.1set protocols ospf area 0.0.0.0 interface lo0.5 passiveset protocols ospf area 0.0.0.0 interface fe-0/0/0.8set policy-options policy-statement send-ospf term 2 from protocol ospfset policy-options policy-statement send-ospf term 2 tad acceptset maršrutēšanas opcijas router-id 192.168.5.5set maršrutēšanas opcijas autonomās sistēmas 17Step-by-Step ProcedureConfiguring Route ReflectorStep-by-Step ProcedureŠajā piemērā nepieciešams, lai pārvietotos dažādos līmeņos konfigurācijas hierarhijā. Informāciju par navigāciju CLI skatiet sadaļā CLI redaktora izmantošana konfigurācijas režīmā CLI lietotāja Guide.To konfigurēt IBGP tīklā, izmantojot Juniper Networks Device A kā maršruta atstarotājs: Konfigurēt interfeises.user@A# set fe-0/0/0 vienība 1 apraksts uz-Buser@A # iestatīt fe-0/0/0 vienība 1 saimes inet adrese 10.10.10. 1/30user@A # kas fe-0/0/1 vienība 3 apraksts to-Duser@A # kas fe-0/0/1 vienība 3 ģimenes inet adrese 10.10.10.9/30user@A # noteikt lo0 vienība 1 ģimenes inet adrese 192.168.6.5/32Configure BGP, ieskaitot klastera identifikatoru un kaimiņu attiecības ar visām ibgp iespējotām ierīcēm autonomajā sistēmā (AS). Lietot arī politiku, kas pārdala OSPF maršrutus BGP. [rediģēt protokolus BGP grupas iekšējo vienaudžiem] user@A # noteikt tipu iekšējo user@A # iestatīt vietējā adrese 192.168.6.5user@A # iestatīt eksporta sūtīt-ospfuser@A # noteikt kaimiņš 192.163.6.4user@A # noteikt klasteru 10.10.10.1user@A # noteikt kaimiņš 192.168.40.4user@A # noteikt kaimiņš 192.168.0.1user@A # noteikt kaimiņš 192.168.5.5Configure statisko maršrutu vai interjera vārtejas protokolu (IGP). Šajā piemērā tiek izmantots OSPF. [rediģēt protokolus ospf platība 0.0.0.0] user@A # noteiktis interfeisis lo0.1 passiveuser@A # noteikt interfeisi fe-0/0/0.1user@A # noteikt interfeisi fe-0/0/1.3Konfigurēt politiku, kas OSPF maršrutus BGP. [edit policy-options policy-statement send-ospf term 2]user@A# set [rediģēt politikas opcijas politika-paziņojums sūtīt-ospf termins 2] & # set from protocol ospfConfigure router ID and autonomous system (AS) number.user@A # set router-id 192.168.6.5user@A # set autonomous system 17ResultsFrom configuration mode, confirm your configuration by entering display interfaces, display protocols, display policy options, and display routing options commands. If the output does not show the intended configuration, repeat the instructions in this example to correct the configuration.local-address 192.168.6.5 user@A# show policy-optionspolicy-statement send-ospf term 2 user@A# show routing-options. Note Repeat these steps for each cluster that you configure if other non-inconvenient devices are from Juniper Network. Otherwise, see the instructions in the documentation for your device. Configuring Client PeersStep-by-Step ProcedureConfiguring interfaces.user@B# set fe-0/0/0 unit 2 description to-Auser@B# set fe-0/0/0 unit 2 family inet address 10.10.10.2/30user@B # set fe-0/0/1 unit5 description to-Cuser@B # set fe-0/0/1 unit 5 family inet address 10.10.10.5/30user@B # set lo0 unit 2 family inet address 192.163.6.4/32Configure BGP neighboring relationship with RRroute reflector. Also use a policy that redeps OSPF routes in DAB. [edit protocols for BGP group internal peers] user@B# set type internaluser@B # set local address 192.163.6.4user@B# set export send-ospfuser@B # set neighbor to 192.168.6.5Configure OSPF. [edit protocols ospf area 0.0.0.0] user@B # set interface lo0.2 passiveuser@B # set interface fe-0/0/0.2user@B # set interface fe-0/0/1.5Configure a policy that redistributes OSPF routes in BGP. [edit policy options policy-statement send-ospf term 2] user@B # set from protocol ospfConfigure router ID and AS number.user@B # set router-id 192.163.6.4user@B # set autonomous system 17ResultsFrom configuration mode, confirm your configuration, display the interfaces, display protocols, display policy options, and display routing options commands. If the output does not show the intended configuration, repeat the instructions in this example to correct configuration.local-address 192.163.6.4;user@B# show policy-optionspolicy-statement send-ospf {user@B# show routing-options. NoteRepeat these steps for each client in the cluster in the BGP peer-to-peer framework of the cluster that you configure if the other client devices are from Juniper Network. Otherwise, see the instructions in the documentation for your device. Configuring Nonclient PeersStep-by-Step ProcedureConfigure interfaces.user@D# set fe-0/0/0 unit 4 description to-Auser@D# set fe-0/0/0 unit 4 family inet address 10.10.10 10/30user@D# set fe-0/0/1 item 7 description to -Euser@D # in fe-0/0/1 unit 7 family inet address 10.10.10.13/30user@D# fix unit 4 family inet address 192.168.0.1/32Configure BGP neighboring relationship with RRroute reflector and with other nonclient neighbor Also use a policy that redeps OSPF routes in DAB. [edit protocols for BGP group internal peers] user@D# set type internaluser@D # set local address 192.168.0.1user@D # set export send-ospfuser@D # set neighbor 192.168.6.5user@D # set neighbor 192.168.5.5Configure OSPF. [edit protocols ospf area 0.0.0.0] user@D # set interface lo0.4 passiveuser@D # set interface fe-0/0/0.4user@D # set interface fe-0/0/1.7Configure the policy that redistribute OSPF routes in BGP. [edit policy options policy statement send-ospf term 2] user@D # set from protocol ospfConfigure router ID and AS number. user@D # set router-id 192.168.0.1user@D # set autonomous system 17ResultsNo configuration mode, confirm your configuration, enter display interfaces, display protocols, display policy options, and display routing options commands. If the output does not show the intended configuration, repeat the instructions in this example to correct configuration.local-address 192.168.0.1;user@D# show routing-optionspolicy-statement send-ospf {user@D# show routing-options If you have finished configuring the device, enter execution from configuration mode. NoteRead these steps for each nonclient in the cluster that is not a valid BGP, a cluster that you configure if other non-inconvenient devices are from Juniper Networks. Otherwise, see the instructions in the documentation for your device. VerificationConfirm that the configuration is working correctly. When you test the BGP, NeighborsPurposeVerify BGP runs on configured interfaces, and a BGP session is created for each neighbor's address. ActionFrom darbības režīms, ievadiet parādīt BGP kaimiņš command.user@A> parādīt BGP neighborPeer: 192.163.6.4 +179 AS 17 Vietējie: 192.168.6.5 +62857 AS 17 Type: Izveidots (maršruta atstarotājs klients)Karogi: <Sync>Last State: OpenConfirm Last Event: RecvKeepAlive Last Error: None Export: [send-ospf] Options: <Preference localaddress= cluster= refresh=>Local Address: 192.168.6.5 Holdtime: 90 Preference: 170 Skaits atlūši: 0 Peer ID: 192.163.6.4 Vietējais ID: 192.168.6.5 Active Holdtime: 90 Keepalive Interval: 30 Peer index: 3 BFD: disabled, uz leju NLRI restart konfigurēts peer: inet-unicast NLRI advertised peer: inet-unicast NLRI for this session: inet-unicast Peer supports Refresh capability (2) Restart laiks konfigurēts peer: 120 Stale routes from peer are kept for: 300 Restart time requested by this peer: 120 NLRI, that peer supports restart : inet-unicast NLRI, that is restarting: inet-unicast NLRI received end-of-rib markers: inet-unicast NLRI end-of-rib markers sent: inet-unicast Peer supports 4 byte AS extension (peer-as 17) Peer does not support Addpath Table inet.0 Bit: 10 Rib State: BGP restart is complete Send state: sync Active prefixes: 0 Received prefixes: 6 Accepted prefixes: 6 Suppressed due to damping: 0 Advertised prefixes: 6 Last Traffic (seconds): Received 5 Sent 19 Checked 19 Input messages: Total 2961 Updates 7 Refreshes 0 Octet 56480 Output<Preference><Sync> Total 2945 updates 6 refreshes 0 octs 56235 output line[0]: 0 peer: 192.168.0.1 +179 AS 17 local: 192.168.6.5+60068 AS 17 Type: Internal position: Created (route reflected Client) Flags: <Sync>Last State: OpenConfirm Last Event: RecvKeepAlive Last Error: None Export: [send-ospf] Options: <Preference localaddress= cluster= refresh=>Local Address: 192.168.6.5 Holdtime: 90 Preference: 170 Unblock: 0 Peer ID: 192.168.0.1 Local ID: 192.168.6.5 Active Holdtime: 90 Keepalive Interval: 30 Peer index: 1 BFD: disabled, down NLRI restart configured peer: inet-unicast NLRI advertised peer: inet-unicast NLRI for this session: inet-unicast Peer supports Refresh capability (2) Restart time configured peer: 120 Stale routes from peer are held: 300 Restart time requested by this peer: 120 NLRI, which peer supports restart: inet-unicast NLRI, that resume is in talks about: inet-unicast NLRI received end rib markers: inet-unicast NLRI all end-of-rib markers sent: inet-unicast Peer supports 4 byte AS extension (peer-as 17) Peer does not support Addpath Table inet.0 Bit : 10000 RIB State: BGP restart is complete Send state: sync active prefixes: 0 Received prefixes: 6 Accepted prefixes: 6 Suppressed due to damping: 0 Advertised prefixes: 6 Last traffic (seconds) : Received 18 Sent 20 verified 12 Input messages: total 15 updates 4 refreshes 0 octets 32307 output queue[0]: 0 Peer: 192.168.5.5 +57458 AS 17 Local: 192.168.6.5+179 AS 17 Type: Internal country: Registered (route reflected Client)Flags: <Sync>Last country: OpenConfirm Last event : RecvKeepAlive Last error: No Export : [send-ospf] Options: <Preference localaddress= cluster= refresh=>Local Address: 192.168.6.5 Holdtime: 90 Preference: 170 Number flanges: 0 Peer ID: 192.168.5.5 Local ID: 192.168.6.5 Active Holdtime: 90 Keepalive Interval: 30 peer index: 2 BFD: disabled, down NLRI restart configured peer: inet-unicast NLRI advertised peer: inet-unicast NLRI for this session: inet-unicast Peer supports Refresh capability (2) Restart time configured peer: 120 Stale routes from peers are kept : 300 Restart time requires the following peer: 120 NLRI that peer supports restart: inet-unicast NLRI peer is restarting: inet-unicast NLRI received end rib marker: inet-unicast NLRI all end-of-rib markers sent: inet-unicast Peer supports 4 byte AS extension (peer-as 17) Peer does not support Addpath Table inet.0 Bit : 10000 RIB State: BGP restart is complete Send state: sync Active prep: 7 Accepted prefixes: 7 Suppressed due to damping: 0 Prefixes advertised: 6 Latest traffic (seconds): Received 17 Sent 3 Verified 9 Input messages: Total 2967 Updates 7 Refreshes 0 Octets 56629 Output messages: Total 2943 updates 6 Refreshes 0 oktēs 56197 izvades rinda[0]: 0 vienādranga: 192.168.40.4+53990 AS 17 local: 192.168.6.5+179 AS 17 Tips: Iekšējais stāvoklis: reģistrēts (maršruts<Preference><Sync>Last State: OpenConfirm Last Event: RecvKeepAlive Last Error: None Export: [send-ospf] Options: <Preference localaddress= cluster= refresh=>Local Address: 192.168.6.5 Holdtime: 90 Preference: 170 Number of flaps: 0 Peer ID: 192.168.40.4 Local ID: 192.168.6.5 Active Holdtime: 90 Keepalive Interval: 30 Peer index: 1 BFD: Disabled, Down NLRI restart configured peer: inet-unicast NLRI advertised peer: inet-unicast NLRI for this session: inet-unicast Peer supports Refresh capability (2) Restart time configured peer: 120 Stale routes from peer are kept for: 300 Restart time requested by this peer: 120 NLRI, which peer supports restart: inet-unicast NLRI peer is restarting: inet-unicast NLRI received end-of-rib marker: inet-unicast NLRI all end rib markers sent: inet-unicast Peer supports 4 byte AS extension (peer-as 17) Peer does not support Addpath Table inet.0 Bit : 10000 RIB State: BGP restart is complete Send state: sync active prefixes: 0 Received prefixes: 7 Accepted prefixes: 7 Suppressed damping due to: 0 Advertised prefixes: 6 Last traffic (seconds): Received 5 Sent 23 Checked 52 Input messages: Total 2960 updates 7 refreshes 0 octets 56112 Output messages: Total 2943 Updates 6 Refreshes 0 octs 56 7 output lines [0]: 0Verifying BGP groupsPurposeVerify BGP groups. ActionRuly mode, enter the Show bgp group command. Group type: Internal AS: 17 Name: 17 Internal Rank Index: 0 Flags: <Export Eval>Export: [send-ospf] Options: <Cluster>hold time: 0 total peers: 4 Created: 4 192.163.6.4+179 192.168.40.4+53990 192.168.0.1+179 192.168.5.5+57458 inet.: 0/0/26/16/0 Groups: 1 Peers: 4 External: 0 Internal: 4 Down Peers: 0 Covers: 0 Table Tot Roads Act Roads Suppressed History of a Wet State Waiting inet.0 26 0 0 0 0PurposeVerify, that the DAB configuration is correct. In ActionEngaged mode, enter the Show bgp summary command. Groups: 1 Peers: 4 Down Peers: 0 Table Tot Roads Act Roads Suppressed History of Wet State To Wait inet.0 26 0 0 0 0 0 Peer AS InPkt OutPkt OutQ Flanges Last Up/Dwn National #Active | 192.163.6.4 17 2981 2965 0 0 22:19:15 0/6/1/0 0/0/0/0 19 168.0.1 17 36 575 0 0 13:43 0/6/1/0 0/0/0/0 192.168.5.5 17 2988 2964 0 0 22:19:10 0/7/0/0 0/0/0/0 192.168.40.4 17 2980 2964 0 0 22:19:14 0/7/7/0 0/0/0/0 TargetRgp routes in the route table. ActionFrom operational mode, enter the show route command.inet.0: 12 destinations, 38 routes (12 active, 0 holddown, 10 hidden) + = Active Route, - = Last Active, * = Both *[Direct/0] 22:22:03 > via fe-0/0/0.1 [BGP/170] 22:20:55, MED 2, localpref 100, from 192.168.5.5 AS path: I > to 10.10.10.10 via fe-0/0/1.3 [BGP/170] 22:20:51, MED 3, localpref 100, from 192.168.40.4 AS path: I > to 10.10.10.2 via fe-0/0/0.1 *[Local/0] 22:22:03 Local via fe-0/0/0.1 10.10.10.4/30 *[OSPF/10] 22:20:51, metric 2 > to 10.10.10.2 via fe-0/0/0.1 [BGP/170] 22:20:55, MED 4, localpref 100, from 192.168.5.5 AS path: I > to 10.10.10.10 via fe-0/0/1.3 10.10.10.12/30 *[OSPF/10] 22:21:08, metric 2 > to 10.10.10.10 via fe-0/0/1.3 [BGP/170] 22:20:51, MED 4, localpref 100, from 192.168.40.4 AS path: I > to 10.10.10.2 via fe-0/0/0.1 [BGP/170] 22:20:55, MED 3, localpref 100, from 192.168.5.5 AS path: I > to 10.10.10.10 via fe-0/0/1.3 [BGP/170] 22:20:55, MED 3, localpref 100, from 192.168.40.4 AS path: I > to 10.10.10.2 via fe-0/0/0.1 192.168.5.5/32 *[OSPF/10] 22:21:08, metric 2 > to 10.10.10.10 via fe-0/0/1.3 [BGP/170] 00:15:24, MED 1, localpref 100, from 192.168.0.1 AS path: I > to 10.10.10.10 via fe-0/0/1.3 [BGP/170] 22:20:55, MED 4, localpref 100, from 192.168.40.4 AS path: I > to 10.10.10.2 via fe-0/0/0.1 192.168.6.5/32 *[Direct/0] 22:22:04 > via lo0.1 [BGP/170] 22:20:51 , MED 2, localpref 100, from 192.168.5.5 AS path : I > to 10.10.10.10 via fe-0/0/1.3 [BGP/170] 22:20:55, MED 2, localpref 100, from 192.168.40.4 AS path: I > to 10.10.10.2 via fe-0/0/0.1 192.168.40.4/32 *[OSPF/10] 22:21:13, metric 2 > to 10.10.10.2 via fe-0/0/0.1 [BGP/170] 22:20:55, MED 1, localpref 100, from 192.163.6.4 AS path: I > to 10.10.10.2 via fe-0/0/0.1 [BGP/170] 22:20:51, MED 4, localpref 100, from 192.168.5.5 AS path: I > to 10.10.10.10 via fe-0/0/1.3 224.0.0.5/32 *[OSPF/10] 22:22:07, metric 1 MultiRecvUnderstanding a Route Reflector That Belongs to Two Different ClustersThe purpose of route reflection is loop prevention when the internal BGP (IBGP) routing devices are not fully meshed. Lai to paveiktu, RRs pārkāpj vienu no normālas BGP darbības noteikumiem: viņi readvertise maršruti, kas gūti no iekšējā BGP peer citiem iekšējiem BGP vienaudžiem. Parasti viens RR ir tikai vienas kopas dalībnieks. Ņemiet vērā, piemēram, ka hierarhiskā RR dizainā otrā līmeņa RR var būt 1. līmeņa RR klienti, bet tie nevar būt savstarpēji klienti. Tomēr, ja divi RRs ir savstarpēji klienti un maršruti tiek atspoguļoti no viena klastera uz citu, klasteru sarakstā tiek iekļauts tikai viens no klastera snoņiem. Tas ir tāpēc, ka ir viens klastera ID klastera sarakstā ir piemērots cilpas novēršanai šajā gadījumā. 1. tabulā ir apkopotas kārtulas, ko maršruta atstarotāji izmanto, aizpildot atspoguļotā maršruta klasteru sarakstu ar klastera table: Route Reflector RulesRouteRoute ScenarioConfigurationWhen reflecting the route from one client-client routerclient -> RR->--client-RR fills in the cluster ID that is associated with this client cluster list of the route. Reflecting the route from the client router client to the client routernon-client-> RR -> clientRr fills in the cluster ID associated with this client in the cluster list of the route. Reflecting the route from the client router to another client router that is in a different clusterclient1 -> RR -> client2 (another cluster)RR fills the cluster ID that is associated with the client1 cluster in the list before reflecting the cluster ID client2. The cluster ID associated with client 2 is not added. Reflecting the route from the client router - without the client router, which is located in another autonomous system. For example, if you have configured level 2 RR and 2 BGP groups, one for RR clients and one for Level 1 RR, and you reflect the route from one autonomous system to another autonomous system. Client -> RR ->-Client (different AS)RR cannot populated the cluster list with cluster ID before reflecting the route-client device because the cluster ID is specific to one autonomous system. Example: Configuring a route reflector that belongs to two different clustersThis example shows how to configure a route reflector (RRs) that belongs to two different clusters. This is not a common scenario, but it can be useful in some situations. RequirementsOverviewConfigurationVerificationRequirementsConfigure device interfaces and internal gateway protocol (IGP). For an example of setting RR, which includes the interface and IGP configuration, see example: Configuring a route reflector In this example, RR1 is a route reflector for both R2 and RR2. The route reflector RR1 has two different cluster S assigned, one is 5.5.5.5 rr1-R2 and 6.6.6.6 6 RR1 RR1 RR2. The RR2 device is a route reflector for the R4 device. Use Figure 5.5: Route reflector in two different combinationsThis example shows the BGP configuration on device RR1 and device RR2. ConfigurationCLI Quick ConfigurationTo quickly configure this example, copy the following commands, paste them into a text file, remove all line breaks, Change all required data to match the network configuration, and then copy and paste the cli commands at the [edit] hierarchy in the level.set protocols bgp group RR1_client type internalset protocols bgp group RR1_client local address 192.168.20.1set protocols bgp group RR1_client cluster 5.5.5.5set protocols bgp group RR1_client neighbor 192.168.1set protocols bgp group Non_client type internalset protocols bgp group Non_client local address 192.168.20.1set protocols BGP group Non_client neighbor 192.168.16.1set protocols BGP group RR1_to_RR2 type internalset protocols BGP group RR1_to_RR2 local address 192.168.20.1set protocols bgp group RR1_to_RR2 cluster 6.6.6.666 BGP group RR1_to_RR2 neighbor 192.168.40.1set protocols BGP group RR2_client type internalset protocols dab group RR2_client local address 192.168.40.1set protocols BGP group RR2_client cluster 7.7.7set protocols bgp group RR2_client neighbor 192. 168.32.1set protocols bgp group RR2_to_RR1 type internalset protocols BGP group RR2_to_RR1 local address 192.168.40.1set protocols bgp group RR2_to_RR1 neighbor 192.168.20.1Configurate device RR1Step-by-step ProcedureConfigure peer-to-peer relationship with device R2. [edit protocols for BGP Group RR1_client] user@RR1 # set internaluser@RR1 local address 192.168.20.1user@RR1 # set to cluster 5.5.5.5user@RR1# set neighbor 192168.48.1Configure peer-to-peer relationships with device R0. [edit protocols for BGP group Non_client] user@RR1 # set type internaluser@RR1 # set local addresses 192.168.20.1user@RR1 # set neighbor to 192.168.16.1Configure peer-to-peer relationship with device RR2. [edit protocols in the BGP Group RR1_to_RR2] user@RR1# set the type internaluser@RR1# set local addresses 192.168.20.1user@RR1# set to cluster 6.6.6.6user@RR1# set to neighbor 192.168.40.1ResultsNo configuration mode, confirm the configuration by entering the show protocols command. If the output does not display the intended configuration, repeat the instructions in this example to correct configuration.local-address 192.168.20.1;local address 192.168.20.1;local address 192.168.20.1; If you have finished configuring the device, enter the execution from configuration mode. Configure the RR2Step stepdging procedure for a deviceConfigure a peer-to-peer relationship with device R4. [edit protocols for BGP Group RR2_client] user@RR2 # set internaluser@RR2 local address 192.168.40.1user@RR2 # set to cluster 7.7.7.7user@RR2# set neighbor 192.168.32.1Configuring peer-to-peer relationships with device RR1. [edit protocols in BGP Group RR2_to_RR1] user@RR2 # set type internaluser@RR2 # set local address 192.168.40.1user@RR2 # set neighbor to 192.168.20.1ResultsNo configuration mode, confirm your configuration by entering the show protocol command. If the output does not display the intended configuration, repeat the instructions in this example to correct configuration.local-address 192.168.40.1;local address 192.168.40.1; If you have finished configuring the device, enter execution from configuration mode. VerificationConfirm that the configuration is working correctly. PurposeVerify that BGP is running a configured interface, and a BGP session is created for each neighboring address. In ActionRuly mode, enter the 5-star address command to display route ads in the BGP protocol. inet.0: 61 destinations, 61 routes (60 active, 0 holddown, 1 hidden) * 2.2.2.2/32 (1 entry, 1 announced) BGP Group RR1_to_RR2 Type Internal Nexthop: 192.168.48.1 Localpref: 100 AS Path: [100] I Cluster ID: 5.5.55.5 Originator ID: 192.168.48.1 MeaningThe 2.2.2.2/32 route originates from the device RR1 client peer, Device R2. When sending this route to an RR1 client, the device route is a 5.5.5.5 cluster ID ID that is cluster ID RR1-R2. GoalCheck a route ad from RR1 to RR2. In ActionFrom mode, enter the show route ad protocol BGP neighbor's address command.user@RR1 > show route ad protocol BGP 192.168.40.1 active path 1.1.1.1/32 extensiveinet.0: 61 destinations, 61 routes (60 active, hold, hold, 1 hidden) * 1.1.1.1/32 (1 entry, 1 announced) BGP group RR1_to_RR2 type Internal Nexthop: 192.168.16.1 Localpref: 100 AS path: [100] I Cluster ID: 6.6.6.6 Original ID: 192.168.16.1AnmetionThe 1.1.1.1/32 route originating from the device RR1-customer peer, device R0. When you send this route to the RR1 client, the RR2 device, the route is added to the 6.6.6.6 cluster ID that has cluster ID RR1 RR2. Device RR1 stores the incoming cluster ID from device RR2 by advertising to another client in another cluster (Device R4). Understanding BGP Optimal Route ReflectionYou can configure BGP Optimal Route Reflections (BGP-ORR) with IS-IS and OSPF as an Interior Gateway Protocol (IGP) for the Route Reflector to advertise the best path to BGP-ORR customer groups. This is done using the shortest IGP metric from the customer's point of view, rather than the route reflector view. Groups of customers with the same or similar IGP topology can be grouped as one DAB peer group. You can configure the optimal route reflection to enable BGP ORR in this DAB peer-to-peer group. You can also configure one of the client nodes as the primary node (igp primary) in the BGP peer group so that IGP metrics from that primary node are used to select the best path and promote it to customers in the same BGP peer-to-peer group. Optionally, you can also select another client node as the backup node (igp-backup) that is used when the primary node (igp primary) goes down or is unreachable. To enable BGP-ORR, include the optimal route reflection at the notification hierarchy level [edit protocols bgp group group name]. NoteBGP-ORR only works when IGP is used to resolve the DAB route. BGP ORR does not work if MPLSLDP/RSVP is used for route resolution. NoteTo how BGP-ORR works, the BGP prefix must be resolved by using IGP. In normal Layer 3 VPN or Layer 2 VPN, VPLS, or MPN, or EEF cases, prefixes are resolved over inet.3. Inet.3, the main path protocol-next-hop prefix would be either RSVP or LDP (not the IGP protocol, such as IS-IS or OSPF). For BGP-ORR to work, you must configure the router to use the prefixes for inet.0 route resolution for Layer 3 VPN or Layer 2, or VPLS, or MPN, or EEF. You can do this by using the following commands:Layer 3 VPN prefix: user@host# set routing options resolution rib bgp.l3vpn.0 resolution-ribs inet.0 2 Layer VPN, or VPLS prefix:user@host# set routing options resolution rib bgp.l2vpn.0 resolution-ribs inet.0For EEF prefix: user@host# set routing options resolution rib bgp.evpn.0 resolution-ribs inet.0 MVPFor MVPN prefix: user@host # set options resolution options rib rib resolution ribs inet.0Other method is to leak IGP routes inet.3 and make them the primary route inet.3.Use the following CLI hierarchy to configure BGP-ORR: optimal route reflection (igp-primary ipv4-address; You can configure the BGP Optimal Route Reflection (BGP-ORR) with IS-IS and OSPF as an Internal Gateway Protocol (IGP) route reflector to promote the best path to the BGP ORR customer groups. This is done using the shortest IGP metric from the customer's point of view, rather than the route reflector view. To enable BGP-ORR, include the optimal route reflection at the notification hierarchy level [edit protocols bgp group group name]. Groups of customers with the same or similar IGP topology can be grouped as one DAB peer group. You can configure the optimal route reflection to enable BGP ORR in this DAB peer-to-peer group. Configure an optimal reflection of the route. [edit protocols in the BGP group group name] user@host# set the optimal route reflectionConfigure for one of the client nodes as the primary node (igp primary) in the BGP peer group, so that igp metrics from the primary node are used to select the best path and promote it to customers in the same BGP peer-to-peer group. [Edit protocols in bgp group group name optimal route reflection]user@host# igp primary ipv4-address;(optionally) configure another client node as backup node (igp-backup) that is used when the primary node (igp primary) goes out or is unreachable. [edit protocols bgp group group name optimal route reflection]user@host #igp-backup ipv4-address; Use the following CLI commands to monitor and prevent the DAB-ORR configuration:show the bgp group — see BGP-ORR primary and backup configurations— see is-IS BGP-ORR metrics (RIB).show ospf bgp-orr —showing is-IS BGP-ORR metric (RIB).show ospf bgp-orr — See OSPF BGP-ORR metric (RIB). Show ospf route records – View OSPF route records to show the active records in routing tables.Show route advertising protocol bgp peer - verify that routes are advertised according to BGP-ORR rules. BGP route server overview The DAB route server is the external DAB (EBGP) equivalent to an internal IBGP (IBGP) route reflector that simplifies the number of direct EBGP sessions required by the network. EBGP route servers shall be transparent for the distribution of the DAB attributes so that the route received from the route server transfers the BGP attribute set (NEXT_HOP, AS_PATH, MULTI_EXIT_DISC, AIGP and Community) if the route is from a directly connected EBGP peer. Like the IBGP route reflector, the EBGP route server is connected to a network outside the direct transfer path between EBGP peers using route server functionality. This connectivity can be via a direct physical link, or through Layer 2 networks, such as VPLS LAN or EEF, or via IP fabric point-to-point EBGP links that provide the reach of feedback peers (typical in data center networks). The BGP protocol has been improved to provide route server capability with the functions described in RFC 7947:attribute transparency NEXT_HOP, AS_PATH, MULTI_EXIT_DISC, AIGP, and Communities.Per-client policy control and multiple route server RIB to mitigate road hiding. BGP programmability improves route server functionality. The BGP JET bgp_route_service.proto API has been improved to support route server functionality as follows:EbGP route server application. Inject routes to a specific route server for rib selective advertising to client groups in client-specific RIBs.The BGP JET bgp_route_service.proto API includes a peer-to-peer object that identifies individual routes as EBGP or IBGP (default). Route server functionality is usually independent of the address family, although certain specific support for BGP attributes may be specific to the address family (for example, AIGP is only supported with labels andicast address families). Route server functionality is supported for all EBGP address families. The BGP attribute for the Transparency [RFC7947] attribute for the Transparency [RFC7947] route server is supported by modifying the normal DAB route distribution so that neither transient or transient attributes are removed or modified when routes are propagated. Changes to the normal EBGP activity are controlled by the route server client CLI configuration. The route server client CLI configuration [edit protocols bgp group group name] at the hierarchy level implements the ggp attribute transparency action of the route server. The route server provides the transparency of attributes for both single-hop EBGP and multi-hop configurations. Therefore, the route server-client configuration implicitly includes functionality without nexthop-change for single hop and multiple hop sessions. Multi-hop BGP sessions must be configured with a multifunction keyword. The route server-client can configure [edit protocol bgp] hierarchy protocol, group, or neighboring levels. Route server-client configuration is applicable only if the group type is external. When you configure the route server-client internal group, a configuration error is issued when you try to execute. There are no parameters for route server-to-client configuration. The normal BGP privilege applies to the route-to-client configuration. NoteAttributes are processed independently of the transparency of the attributes. Even if the route server cannot send future hops unmodified, other attributes are sent transparently as applicable to these attributes. This is a sample route server-client configuration: Next-HopThe next-hop attribute should not be modified by imposing next-hop itself or otherwise modifying next-hop unless clearly configured by policy. The route server is to distribute the BGP next hop under third-party next-hop rules RFC 4271.Next-hop operation is specified for the following route server scenarios:In the case of LAN and WAN interconnections, the route server is connected to client peers through a shared LAN and WAN subnet, third-party future hops route server advertises without changes as defined in RFC 4271. In the case of a multiconnection of a data center, the route server is connected to client peers through a multi-part interconnection, the EBGP multihop must be configured, and without the nexthop change, the CLI configuration operation is indirectly determined by the route server-client configuration. The next hops received are advertised by the route server without changes, As one additional third-party behavior defined in RFC 4271.In in other cases, such as point-to-point connections between route servers and client peers, the usual next hop advertising procedures are used to prevent ad side hops that BGP peers might reject (for example, most DAB implementations reject by default the next hop addresses that are not covered by subnet address in non-multipoint sessions. AS PathAS-Path may not be modified by using the local AS number of the route server in advance. Configure the route server client CLI in peer-to-peer play in local AS number prepending ads. Additionally, the show route ad protocol for the BGP peer CLI team is changed to peers who are route server clients, such as local AS not showing advertised metrics. Other AttributesMULTI_EXIT_DISC attribute (optional, transient) must be multiplied as received. All community attributes, including non-advertised, non-exportable and non-transient extended communities, must be multiplied as received. The accumulated IGP (AIGP) attribute (optional, transient) must be multiplied as received. NoteJunos OS supports AIGP only for BGP-LU address families (IPv4 and IPv6). The BGP Route Server client RIBA route server client-specific RIB has a different view of BGP Loc-RIB, which can include different routes for the same destination than other views. Route server clients can associate with one single client-specific view or shared rib by using their peer-to-peer groups. In order to enable different routes to be advertised to different customers for the same destination, it is conceptually necessary to allow multiple BGP path selections to take place for the same destination, but in different customer/group contexts. In order to enforce a high level of flexible policy control requirement using client/group path selection, the DAB route server uses the approach using non-forwarding routings (NFIs) for many DAB pipeline selection, including BGP path selection, Loc-RIB, and policy. The route server is configured to group the GGP groups for route server clients into separate cases of odd-forward routing. This approach attracts the fact that the BGP running routing instance has no road choice and has ribs that are separate from BGP running in other routing cases. Policy requirements and considerationsTo distribute routes between route server clients, routes are leaked between RIBs cases based on configured policies. Route server configuration policy control includes the following considerations:Route server clients must be configured on the same main instance or routing instance to receive the same Loc-RIB view. Route server clients must be configured in their route instance to receive completely unique Loc-RIB views. Route server clients must configure different DAB peer groups in the same routing instance to have different export policies in the same loc-rib view. For route server client-specific RIB views, to receive all routes from other tables by default, the instance's import policy for full-eye is configured with an instance. Configuring instance-import with a policy that contains an example of any:instance-any can be used: policy statement ... Term... nopolitik statement ... nopolitik statement ... Term... topolicy-notice... there are no parameters. The use of an instance in policies other than import of an instance has no effect. Configuring many different routing instances and peer-to-peer groups affects scale and performance. The BGP transfer context CLI configuration [edit protocols for bgp group neighbor] at the hierarchy level divides the routing instance into a BGP neighbor configuration instance and the transfer instance. The BGP transfer context CLI configuration also supports an odd forwarding case with DAB peers configured as route server-client, where the specified instance can transfer a main or routing instance of a non-suction type. Starting with Junos OS Release 15.1, the non-install notification prevents interaction between the Routing Protocol daemon (rpd) and other components of the Junos system, such as kernel or distributed firewall daemon (dfwd). In pre-Junos OS Release 15.1, you can reduce the workload of route reflectors other than traffic transfer routes by using a transfer table export policy that rejects routes from BGP. Bgp.