Overview of High Availability

High Availability refers to the ability to establish redundancy of networking functionality and configuration data between two peer routers. This guide provides information about high availability, and how you can configure high availability on Cisco Catalyst 8000V Edge Software running on different cloud service providers.

The High Availability feature is supported for Cisco Catalyst 8000V Routers running on Microsoft Azure, Google Cloud Platform (GCP), and Amazon Web Services (AWS). A typical use case for the Cisco Catalyst 8000V is to interconnect two subnets within a virtual network. You can deploy Cisco Catalyst 8000V routers between the front-end (public) and the back-end (private) subnets. The Cisco Catalyst 8000V router represents a single point of failure for access to back-end resources. To mitigate this single point of failure, you must deploy two Cisco Catalyst 8000V routers between the two subnets.

The back-end subnet contains a routing table with entries pointing to the next hop router, which is one of the two Cisco Catalyst 8000V instances. The peer Cisco Catalyst 8000V routers communicate with one another over a tunnel using the Bi-directional Forwarding Detection (BFD) protocol. If the connection is lost between a router and a peer, BFD generates an event. This event causes the active router that is working to update the entries in the route table so that the routing table points to the default route.

The routing table controls the upstream traffic of the Cisco Catalyst 8000V router and the routing protocol configured on the router determines the path of the downstream traffic.

In cloud environments, it is common for virtual networks to implement a simplistic mechanism for routing, which is based on a centralized route table. However, you can also create multiple route tables, where each route table has a subnet assigned. This subnet acts as the source of route information, and the route table is populated automatically which includes one or more individual routes depending on the network topology. You can also configure the routes in the route table.

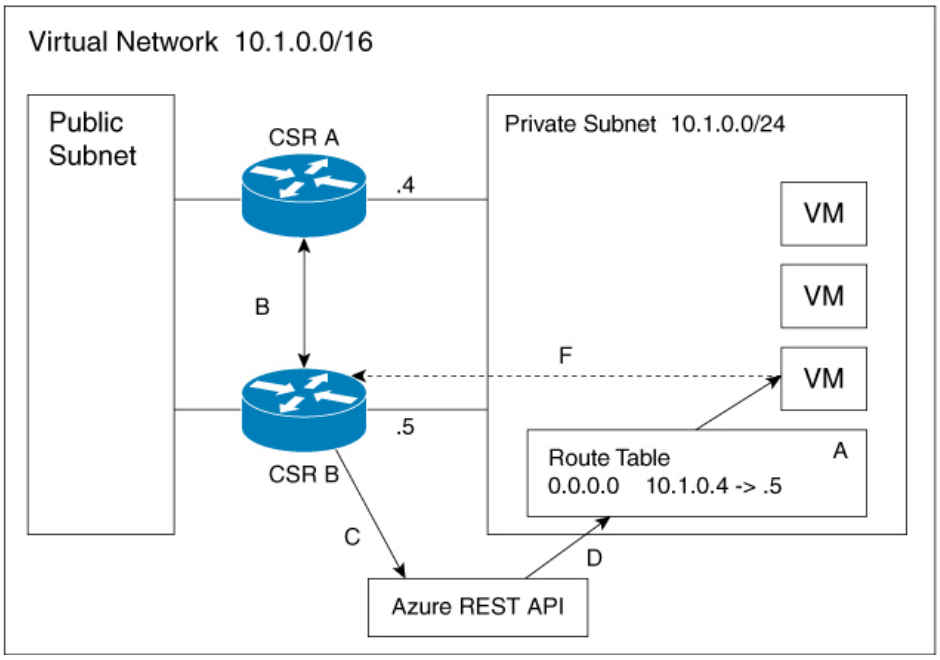
A subnet has a centralized route table, which allows two Cisco Catalyst 8000V routers to operate in a redundant mode. You can deploy two Cisco Catalyst 8000V routers in the same virtual network with their interfaces directly connected to subnets in the virtual network. You can add routes to the route table to point to one of the two redundant Cisco Catalyst 8000V routers. At any given time, one of the two Cisco Catalyst 8000V routers serves as the next-hop router for a subnet. This router is the active router for the subnet. The peer router is referred to as the passive router. The active router is the next hop for a given route destination.

The Cisco Catalyst 8000V router uses the Bi-directional Failure Detection (BFD) protocol to detect whether a peer router is operating properly. An IP tunnel is created between the two peer routers and each router periodically sends a BFD protocol message to the other router. If one router fails to receive a BFD message from the peer for a specific period, the active router concludes that the peer router has failed.

If the active router fails, the route table for the subnet can be dynamically updated to change the next hop address for one or more routes so that they refer to the passive router. If the peer router detects the failure of the active router, the peer router uses the programmatic API to update the route table entries.

For a route table entry, configure which of the two Cisco Catalyst 8000V routers is the “primary” router. The other router is the passive router if it is configured as a “secondary” router. By default, all routes are configured as secondary.

Figure 1: High Availability - Topology



The subnet on the right has an address block of 12.1.0.0/24. The two Cisco Catalyst 8000V routers that are connected to this subnet provide a redundant path for traffic leaving this leaf subnet. The subnet is associated with a route table which provides the route information to the virtual machines attached to the subnet.

Consider this scenario: Initially the default route in the route table has the IP address of the next hop router - 12.1.0.4 (Cisco Catalyst 8000V A). All the traffic leaving the subnet goes through Cisco Catalyst 8000V A. Cisco Catalyst 8000V A is currently the active router for the default route. When Cisco Catalyst 8000V A fails, Cisco Catalyst 8000V B detects the failure as this router stops receiving BFD protocol messages from Cisco Catalyst 8000V A. Cisco Catalyst 8000V B writes to the route table via a RESTAPI to change the default route to the interface of Cisco Catalyst 8000V B on the 12.1.0.0/24 subnet, which is IP address 12.1.0.5. Cisco Catalyst 8000V B then becomes the active router for the route to the 15.0.0.0 network.

Step	Description
A	Cisco Catalyst 8000V A with address 12.1.0.4 is the active router for the 15.0.0.0 network.
B	Cisco Catalyst 8000V A fails. Cisco Catalyst 8000V B detects the failure using the BFD protocol.
C	Cisco Catalyst 8000V B uses an HTTP request to the Azure REST API.
D	Azure updates the 15.0.0.0 route in the user-defined route table to the IP address of Cisco Catalyst 8000V B.

Step	Description
E	Virtual machines see the route table update.
F	Packets from the virtual machines are now directed to Cisco Catalyst 8000V B.

High Availability Features

High Availability version supports several features. Here's an overview of high availability in Cisco Catalyst 8000V.

- **Cloud Agnostic**: This version of high availability is functional on Cisco Catalyst 8000V routers running on any cloud service provider. While there are some differences in the cloud terminology and parameters, the set of functions and scripts used to configure, control, and show the high availability features are common across the different cloud service providers. High Availability is supported in Cisco Catalyst 8000V routers running on AWS, Azure, and GCP. Check with Cisco for current support of high availability in the individual provider's clouds.
- **Active/active operation**: You can configure both Cisco Catalyst 8000V routers to be active simultaneously, which allows for load sharing. In this mode of operation, each route in a route table has one of the two routers serve as the primary router and the other router as the secondary router. To enable load sharing, take all the routes and split them between the two Cisco Catalyst 8000V routers.
- **Reversion to Primary Cisco Catalyst 8000V After Fault Recovery**: You can designate a Cisco Catalyst 8000V as the primary router for a given route. While this Cisco Catalyst 8000V is up and running, it is the next hop for the route. If this Cisco Catalyst 8000V fails, the peer Cisco Catalyst 8000V takes over as the next hop for the route, maintaining network connectivity. When the original router recovers from the failure, it reclaims ownership of the route and is the next hop router.
- **User-supplied Scripts**: The guestshell is a container in which you can deploy your own scripts. HA exposes a programming interface to user-supplied scripts. This implies that you can now write scripts that can trigger both failover and reversion events. You can also develop your own algorithms and triggers to control which Cisco Catalyst 8000V provides the forwarding services for a given route.
- **New Configuration and Deployment Mechanism**: The implementation of HA has been moved out of the Cisco IOS XE code. High availability code now runs in the guestshell container. For further information on guestshell, see the *Guest Shell* section in the Programmability Configuration Guide. The configuration of redundancy nodes is performed in the guestshell using a set of Python scripts.

- Reference the Chapter Map here, on page 4
- Topologies Supported, on page 4
- Redundancy Nodes, on page 4
- Event Types, on page 4

Reference the Chapter Map here

Topologies Supported

1-for-1 redundancy topology: If both the Cisco Catalyst 8000V routers have a direct connection to the same subnet, the routers provide a 1-for-1 redundancy. An example of 1-for-1 redundancy is shown in the preceding figure. All the traffic that is intended for a Cisco Catalyst 8000V only goes to one of the routers - the Cisco Catalyst 8000V that is currently active. The active Cisco Catalyst 8000V router is the next-hop router for a subnet. The other Cisco Catalyst 8000V router is the passive router for all the routes.

Load sharing topology: In this topology, both the Cisco Catalyst 8000V routers have direct connections to different subnets within the same virtual network. Traffic from subnet A goes to router A and traffic from subnet B goes to router B. Each of these subnets is bound to different route tables. If router A fails, the route table for subnet A is updated. Instead of router A being the next hop, the route entry is changed to router B as the next hop. If router B fails, the route table for subnet B is updated. Instead of router B being the next hop, the route entry is changed to router A as the next hop.

Redundancy Nodes

A redundancy node is a set of configuration parameters that specifies an entry in a route table. The next hop of a route is updated when an active router fails. To configure a redundancy node, you require the following information:

- Route Table – The identity of the route table in the cloud. Route table includes a region or group in which the table was created, an identifier for the creator or the owner of the table, and a name or identifier for the specific table. Optionally, you can specify an individual route within the table. If you do not specify an individual route, the redundancy node represents all the routes in the table.
- Credentials - Authentication of the identity of the Cisco Catalyst 8000V router. Each cloud provider handles the process of obtaining and specifying the credentials differently.
- Next Hop - The next hop address that is written to the route entry when a trigger event occurs. Next Hop is usually the interface of the Cisco Catalyst 8000V routers on the subnet that is protected.
- Peer Router - Identifies the redundant router that will forward traffic for this route after a failure occurs on this router.
- Router Role—Identifies whether the redundancy node serves in a primary or secondary role. This is an optional parameter. If you do not specify this value, the router role defaults to a secondary role.

Event Types

The high availability feature recognizes and responds to three types of events:

- **Peer Router Failure**: When the peer route fails, it is detected as a Peer Router Failure event. In response to this event, the event handler writes the route entry with the next hop address that is defined in the redundancy node. To enable this event to be generated, configure the BFD protocol to a peer router and associate the BFD peer under redundancy for cloud high availability.

- **Revert to Primary Router**: After a router recovers from a failure, the *Revert to Primary Router* event occurs. The purpose of this event is to ensure that the primary router for the route is re-established as the active router. This event is triggered by a timer and you need not configure this event. In the route table entry, the event handler changes the next hop address that is defined in the redundancy node only if it is different from the next hop address that is currently set for the route.

  This Revert to Primary Router event is generated periodically using a CRON job in the guestshell environment. The job is scheduled to run every 5 minutes and checks if each redundancy node that is configured in the primary mode has this router's next hop interface set in the route table. If the route table entry already points to this router's next hop interface, then an update is not required. If a redundancy node configuration of the mode parameter is secondary, then the *Revert to Primary Router* event is ignored.

- **Redundancy Node Verification**: The event handler detects a Redundancy Node Verification event and reads the route entry that is specified by the redundancy node. The event handler writes the same data back to the route entry. This event is not generated automatically or algorithmically. This event verifies the ability of the event handler to execute its functions. Execute a script, manually or programmatically, to trigger the Redundancy Node verification event. For further information about the verification event, see *User-Defined Triggers*, in the *Advanced Programming for High Availability on Microsoft Azure* section.