CHAPTER 30

VrfApp Service

This chapter describes the DCNM web services' API methods for the VrfApp service.

Information About VrfApp Service

The Layer 3 functionality in a VDC can be further virtualized into multiple routing domains using virtual routing and forwarding (VRF) instances. A separate routing and forwarding table is maintained for each VRF. A Layer 3 interface (logical or physical) in a VDC can belong to exactly one VRF. A VRF is local to a VDC and each VDC can contain multiple VRFs. By default, each VDC will contain a VRF. By default, all Layer 3 interfaces belong to the default VRF. In addition, for each VDC, a management VRF is created automatically.

Cisco NX-OS supports a VRF-lite implementation. VRF-lite enables a service provider to support two or more VPNs, where the IP addresses can overlap among the VPNs. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as a VLAN SVI. A Layer 3 interface cannot belong to more than one VRF.

With VRF-lite, multiple customers can share one customer edge (CE), and only one physical link is used between the CE and the provider edge (PE). The shared CE maintains separate VRF tables for each customer and routes packets for each customer based on its own routing table. VRF-lite extends limited PE functionality to a CE device, giving it the ability to maintain separate VRF tables to extend the privacy and security of a VPN to the branch office.

addIpNetworkInterfaces

Add IpNetworkInterfaces to VRF.

Parameters

vrfInstanceNameId—InstanceNameId of VRF

ipNetworkInterfaceCol—List of IpNetworkInterface objects to be added.

Return Value

void

addIpNetworkInterfacesToDefaultVrf

Add IpNetworkInterfaces to Default VRF.

Parameters

ipNetworkInterfaceCol—List of IpNetworkInterface objects to be added.

Return Value

void

addIpv4StaticRoutes

Add a list of static routes to the specified network element.

Parameters

opContext—Operational context

InstanceNameId—Network element

vrfName—String specifying VRF name

Ipv4StaticRoute—List of static routes to add

Return Value

List of InstanceNameId

addIpv6StaticRoutes

Add a list of static routes to the specified network element.

Parameters

opContext—Operational context

InstanceNameId—Network element

vrfName—String specifying VRF name

Ipv6StaticRoute—List of static routes to add

Return Value

List of InstanceNameId

createVrfs

Create VRF on a VDC.

Parameters

neInstanceNameId—InstanceNameId of VDC

vrfCol—List of VRF objects used to create VRF instances.

Return Value

void

deleteVrfs

Removes VRFs from VDC.

Parameters

vrfInstanceNameIdCol—List of InstanceNameId of vrf to be deleted.

Return Value

void

getAllNetworkElements

Returns list VDCs in the network. This method will return neIds of VDCs that have the VRF created.

Return Value

List of VDC elements that have VRF enabled

getIpNetworkInterfaces

Returns list of IpNetworkInterfaces in a VRF.

Parameters

vrfInstanceNameId—Instance name ID of Vrf to be queried.

Return Value

List of IpNetworkInterface Objects.

getMulticastRoutingInstancesForVrf

Returns list of protocols instance Identifier that are enabled in a VRF.

Parameters

vrfInstanceNameId—Instance name ID of Vrf to be queried.

Return Value

List of RoutingInstances Objects for VRF.

getUnicastRoutingInstancesForVrf

Returns list of protocols instance Identifier that are enabled in a VRF.

Parameters

vrfInstanceNameId—Instance name ID of Vrf to be queried.

Return Value

List of RoutingInstances Objects for VRF.

getVrfsInNetworkElement

Returns list of VRFs in a VDC.

Parameters

neInstanceNameId—Instance name ID of network element for which Vrf are to be queried.

Return Value

List of Vrf.

removeStaticRoutes

Removes a list of static routes.

Parameters

opContext—Operational context

Ipv4StaticRoute—List of static routes to remove

Return Value

void