




Active Fabric Controller (AFC) User Guide



Notes, Cautions, and Warnings

-  **NOTE:** A NOTE indicates important information that helps you make better use of your computer.
-  **CAUTION:** A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.
-  **WARNING:** A WARNING indicates a potential for property damage, personal injury, or death.

© 2013 Dell Inc.

Trademarks used in this text: Dell™, the Dell logo, Dell Boomi™, Dell Precision™, OptiPlex™, Latitude™, PowerEdge™, PowerVault™, PowerConnect™, OpenManage™, EqualLogic™, Compellent™, KACE™, FlexAddress™, Force10™ and Vostro™ are trademarks of Dell Inc. Intel®, Pentium®, Xeon®, Core® and Celeron® are registered trademarks of Intel Corporation in the U.S. and other countries. AMD® is a registered trademark and AMD Opteron™, AMD Phenom™ and AMD Sempron™ are trademarks of Advanced Micro Devices, Inc. Microsoft®, Windows®, Windows Server®, Internet Explorer®, MS-DOS®, Windows Vista® and Active Directory® are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Red Hat® and Red Hat® Enterprise Linux® are registered trademarks of Red Hat, Inc. in the United States and/or other countries. Novell® and SUSE® are registered trademarks of Novell Inc. in the United States and other countries. Oracle® is a registered trademark of Oracle Corporation and/or its affiliates. Citrix®, Xen®, XenServer® and XenMotion® are either registered trademarks or trademarks of Citrix Systems, Inc. in the United States and/or other countries. VMware®, vMotion®, vCenter®, vCenter SRM™ and vSphere® are registered trademarks or trademarks of VMware, Inc. in the United States or other countries. IBM® is a registered trademark of International Business Machines Corporation.

2014-03

Rev. A00

Contents

1 Introduction	7
Supported Platforms.....	7
How the Active Fabric Works.....	7
Advantages of Active Fabric Deployment.....	8
Active Fabric Element Definitions and Requirements.....	8
AFC and SDN.....	10
2 Active Fabric Features.....	13
Active Fabric Resiliency.....	13
Link Resiliency.....	13
Switch Resiliency	13
Packet Forwarding.....	13
Policies.....	14
Important Points to Remember.....	14
Policy Configuration and Types.....	14
Policy Association.....	15
Network Policies.....	15
Endpoint Policies.....	15
Middlebox Policy.....	16
Filter Policy.....	16
Statistics.....	16
User Authentication	17
Remote Authentication Dial In User Service (RADIUS).....	18
Terminal Access Controller Access Control System (TACACS).....	18
3 Supported Technologies and Protocols.....	19
Bare Metal Provisioning (BMP).....	19
Destination Lookup Failure (DLF).....	19
DLF Port Roles.....	20
DLF Topology.....	20
High Availability (HA).....	21
HA and SDN.....	22
Deployment Model.....	22
Design	23
Role Determination.....	23
Checkpoints.....	24
Reconciliation Process.....	24
Link Aggregation Control Protocol (LACP) and NIC Teaming.....	25

Standard LACP Feature	26
LACP	27
NIC Teaming	27
Traffic Optimization over LACP and NIC Teaming.....	27
LACP Requirements and Limitations.....	27
NIC Teaming Requirements and Limitations.....	28
Mongo Database and Mongoose.....	28
OpenStack.....	28
Quality of Service (QoS).....	32
Types of QoS.....	32
Configuring a QoS Policy.....	32
Switched Port Analyzer (SPAN) and Remote Switched Port Analyzer (RSPAN).....	33
Virtual Link Trunking (VLT).....	33
Forwarding Database (FDB)Link Aggregation Control Protocol (LACP).....	33
VLT Link Failure.....	34
VLT Interconnect (VLTi) Failure.....	34
Head Node Failure.....	34
Leaf Node Failure.....	34
Multi-domain VLT (mVLT).....	35
Link Failure Scenarios.....	35
Node Failure.....	35
Multi-Stage mVLT Extension Support.....	35
4 REST APIs.....	37
REST Actions.....	37
REST Resources.....	37
Resources.....	38
REST Resource Attributes.....	40
Provider Objects Tenant Objects.....	40
Network Objects.....	41
Host Objects.....	41
Endpoint Objects.....	42
Policy Objects.....	42
Rule Objects.....	43
Network Policy Objects.....	44
Provider Tenant Policy Objects.....	44
Endpoint Policy Objects	45
Network Endpoint Policy Objects.....	46
Network Connection Objects.....	47
Network Endpoint Objects.....	47
Host Endpoint Objects.....	47
Switch Port Network Objects.....	47

FDB Entry Objects.....	48
Cluster Node Objects.....	48
Uplink Objects.....	48
Host Port Objects.....	49
WAN Port Objects.....	49
Port Monitoring Objects.....	50
Middlebox Port Objects.....	50
REST Information Retrieval.....	51
Viewing Information.....	51
Filtering Results.....	52
REST Request URIs.....	52
Requirements.....	52
System Resources and URIs.....	52
Provider Resources and URIs.....	55
Tenant Resources and URIs.....	56
Policy Resources and URIs.....	57
Switch Resources and URIs.....	59
Topology Resources and URIs.....	60
Uplink Resources and URIs.....	61
WAN Port Resources and URIs.....	62
Port Monitor Resources and URIs.....	62
Host Port Resources and URIs.....	63
Middlebox Port Resources and URIs.....	63
Flow Resources and URIs.....	63
Counter Resources and URIs.....	64
Edge Port Resources and URIs.....	65
Port Resources and URIs.....	65
StaticFlow Resources and URIs.....	66
Help Resources and URIs.....	66
Device Resources and URIs.....	66
QoS Resources and URIs.....	67
LACP Resources and URIs.....	67
Link Discovery Resources and URIs.....	67
REST Errors.....	67
5 Logging.....	69
Logging Sublevels.....	69
Component Logging.....	69
SDNC Database Logging.....	70
6 Upgrading RPM-Installed Software.....	71
Upgrading Single-Server Deployments.....	71

Upgrading Dual-Server Deployments.....	71
7 Active Fabric Controller User Interface.....	73
Supported Browsers.....	73
Components.....	73
HTTP and HTTPS Support.....	74
Features.....	74
GUI Icons.....	74
GUI Features.....	74
Help.....	75
Main Screen.....	75
Changing System Settings.....	76
Submitting Queries.....	76
Selecting Display Modes.....	77
Saving Queries.....	77
Submitting Saved Queries.....	78
Deleting Saved Queries.....	78
Saving Query Results.....	79
Displaying Saved Query Results.....	79
Deleting Saved Query Results	79
Applying Filters	79
Removing Filters.....	80
Refreshing Center Pane Content.....	80
Using Auto-Refresh.....	80
Pagination.....	80
Topology-Based Navigation and Information Retrieval.....	80
Configuring Resources.....	83
Query Builder.....	95
Summaries.....	96

Introduction

This document describes the Active Fabric Controller (AFC) and its role in the active fabric solution. The controller automatically discovers the physical topology of the fabric and uses a topology-specific forwarding scheme to provide multi-path support and forwarding between endpoints using the shortest path. AFC manages the switching nodes within the fabric to provide loop-free forwarding and high resiliency. You can use programmable application programming interfaces (APIs) provided by the controller to manage other components within the active fabric.

Supported Platforms

The AFC is supported on the following Dell Networking switches:

- S4810
- S4820T
- S6000
- MXL
- Input/Output Aggregator (IOA)

How the Active Fabric Works

Active fabric is a system of network components connecting virtual and physical server and storage devices. Each cluster of switches at the edge of the network is a domain. The fabric connects the controller software components using the edge ports of the switches. The switches connect to each other using internal ports. Elements connected to the fabric are considered end points; the ports connecting the end points to the controller are considered middle boxes.

The term *active fabric* is used collectively to refer to all networking components used by frames that are transferred to or from a virtual machine (VM), a physical device to or from another VM, or a physical device in the converged infrastructure. This includes:

- the chassis or rack interconnect switch, or top of rack (ToR)
- the switch in the chassis
- the converged network adapter (CNA)
- the virtual switch

The active fabric is treated as the logical equivalent of a network node. Active fabric systems are comprised of the following components:

- Orchestration engine (such as the Dell OpenStack m2 mechanism driver)
- Centralized control and management provided by the software-defined network (SDN)
- Switches (either inter- or intra-chassis)

In a multi-tenant environment, the active fabric provides the following logical components:

- Internal local area network (LAN) or virtual networks
- Internal storage networks
- External uplink networks and association with internal networks

The controller, which connects the end-point controller software components to other domains, is a key element of the active fabric that is responsible for workload management, network service provisioning, and fabric configuration using external applications. The controller's role can be a provider (manager of active fabric infrastructure) or a tenant (user of resources).

- The tenant creates and manages the workload profile, which must include the adaptor profile. You must select an associated endpoint port profile.
- The provider creates and manages the network profiles (LAN or SAN) and the port profiles (endpoint or uplink).

To simplify workload mobility and the addition or removal of infrastructure cards or other elements, an endpoint port profile is applied at startup when the workload is detected.

Advantages of Active Fabric Deployment

Active fabric is a solution designed for data centers that provides a programmable network of switches as a seamless entity. The advantages of implementing active fabric are:

- Bundling the network fabric in the server infrastructure
- Simplifying operations for the server or virtualization administrator
- Optimizing for cloud capability
- Simplifying control and management
 - Single point of control for management
 - No protocols – zero-touch deployment
 - Stateless autoconfiguration
- Combining virtual and physical networks
- Providing scalability and resiliency

Active Fabric Element Definitions and Requirements

The AFC provides multi-tenancy network connectivity for servers and VMs connected to the active fabric. Multi-tenancy enables tenants to create networks and VMs for running workloads. The active fabric isolates the traffic of the tenants.

Controller

- A software entity that manages a group of Dell SDN physical switches using OpenFlow to provide an abstract fabric view.

Head Node

- The access point to the network; also known as a gateway router or a spine node.

Host

- A server consisting of one or more endpoints that connect to different networks or the bare metal server running the application.

End Point

- A point of attachment to a software-managed network.
- Can be a VM running on a host or host network interface controller (NIC) cards.
- End points represented by a VM on a host are called VM end points.
- End points represented by host network card interfaces are called host end points.

External Providers

- A provider creates one or more tenants, networks, and policies.
- A provider creates networks using hosts with one or more endpoints and manages them as part of the provider infrastructure.
- A provider creates one or more policies and associates them with a specific tenant or tenant group. These policies enforce specific actions for all networks associated with the provider tenant or all endpoints used by hosts associated with a tenant of the specified provider.
- You can also associate a provider-created policy with a network managed by the provider or with endpoints on the hosts managed by the provider.
- A provider allocates one or more hosts, which contain one or more endpoints, to a tenant.

Leaf Node

- Devices connected to the head node; also known as edge devices.

Network

- A virtual network, logical switch, or Layer2 (L2) domain managed by the software.

Node

- Any device connected to a network or any device with an IP address.

Policy

- One or more rules that define the operations of a tenant, switch, link, network, host, or network connection.

Rule

- Consists of two components: match and action.
- The match component contains the flow attributes and the action components contains the expected switch behavior.

Switches and Switch Ports

- Any switch elements in the fabric.
- The controller can only gather data from the switch and cannot be used to configure the switch.

System

- The software creates an internal root provider. By default, any managed object that is not associated with an external provider is associated with the internal provider.
- All external providers must use a unique ID.

Tenants

- A tenant is any client on the active fabric network.

- A tenant is assigned an ID by the provider when it is created that identifies both the tenant and the provider. The ID must be unique for each tenant managed by the provider, but a provider can use a tenant ID also used by a different provider. The combination of the tenant ID and the provider ID creates a unique identification.
-
- The tenant creates networks and policies and connects endpoints to networks.
- The tenant creates one or more policies and associates them with a network managed by the tenant. The tenant can also create policies associated with host endpoints used by the tenant.

Up Link

- The edge port of the fabric connecting to the legacy network where traffic enters the ports.

AFC and SDN

In an SDN, an external controller-cluster manages the network and the resources on each switch. The software uses OpenFlow (OF) for communication between the controller and the switch.

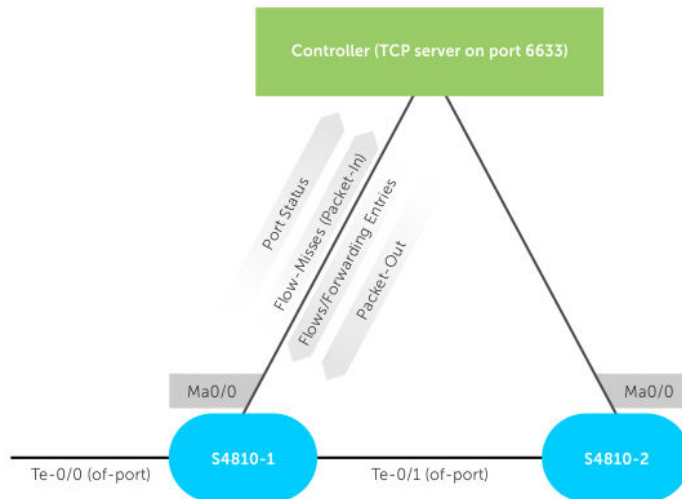


Figure 1. OpenFlow Topology — Overview

SDN offloads all switching and routing protocol state machines to the controller. A simplified and efficient software layer on the switch programs the forwarding tables.

Using OpenFlow, you can transmit the switch's ports and forwarding tables to the controller, allowing the controller to configure forwarding entries on the switch. OpenFlow also allows the controller to insert control packets through the switch, separating the control plane from the forwarding plane, and to redirect any missed flow packets from the switch to the controller.

The flows in OpenFlow allow the switch to match based on the following parameters:

- ingress port
- virtual local area network (VLAN) ID
- VLAN priority (vlan-pri)
- destination MAC address (DMAC)
- source MAC address (SMAC)
- EtherType
- session initiation protocol (SIP)
- dynamic IP (DIP)
- type of service (TOS)
- protocol
- transport source-port (`transport sport`)
- transport destination-port (`transport dport`)

The software forwards the match results out of one or more network ports, with the option to modify the packet headers.

The AFC supports OpenFlow version 1.3. For information about OpenFlow and SDN, refer to the *SDN Deployment Guide* on the Dell Networking documentation website.

Active Fabric Features

Active Fabric Resiliency

Active fabric provides link and node resiliency, as well as the ability to converge at comparable time with legacy networks. Active fabric provides more robust link and switch failure resiliency when compared with legacy networks.

Link Resiliency

If a link goes down in the active fabric physical network, the controller is notified of the port status change by the switches in the fabric. The controller receives the port status notification and handles the link down event. Link failures are handled differently depending on if the unavailable link is for an edge port between switches managed by the controller (internal link) or if the unavailable link is for an edge port connected to a server or legacy switch (external link). The software updates the network topology with the port status change. The type of technology used to manage the switch group determines how the link failure is managed.

The controller configures the following:

- Forwarding database (FDB) entries
- VLAN hardware table
- VLT blocking
- Ingress port blocking (for loop prevention)

Switch Resiliency

The controller configures the hardware tables for the FDB entries, VLAN, VLT blocking, and ingress port blocking for data traffic loop prevention. If a switch in the fabric is unavailable, the controller identifies the unavailable switch through the transmission control protocol (TCP) connection failure. The software uses OF messages to notify the rest of the active fabric that the switch is unavailable. The software disables all links from and to the switch and notifies the other fabric entities of the link state change. If the switch is at the edge of the switch group, the neighbor switch group's interface is also disabled. After the software notifies all fabric entities of the change, the software removes the unavailable switch from the fabric topology.

Packet Forwarding

The controller manages the switches within its domain using OF, which is enabled on each switch in the fabric connected to the controller. You must configure each switch using the controller information. Enable OF on all individual ports so that the OF agent running on the switch can inform the controller about the active ports available for OF negotiation.

After you enable OF on a port, all packets received on the port that do not match ingress forwarding criteria (unknown packets) are sent to the controller. The controller processes these packets using OF and programs the forwarding database (FDB) entries using the flow configuration messages. Each packet received is processed by the controller in stages, where each stage performs a different function. This is referred to as the “ingress pipeline.” For example, if the ingress packet is a link layer discovery protocol (LLDP) packet, the software uses it for topology mapping.

Policies

A policy is one or more rules that define the operation of a tenant, switch, link, network, host, or network connection. Each policy has three sections:

- the policy header that provides the policy ID, name, provider ID, and tenant ID
- the rule section that identifies the unique rule name and match criteria that specifies the resource
- the action section that defines the action (permit, block, or redirect) for traffic from the resource that meets the match criteria

The action and policy header information is defined when you create the policy. The rule section is defined when you associate a policy with a resource.

Important Points to Remember

- Use the GUI or REST APIs to create and configure policies.
- Policies are managed by a provider or a tenant.
- Associate policies with a network or an end point.
- To enable a policy, create the policy and apply a rule to it. To apply multiple rules for an endpoint or network, create multiple policies and associate them with the resource.
- You can apply a provider policy to provider resources and a tenant.
- If you apply a policy to a tenant, it applies to all of the tenant’s networks as well.
- When associating the policy, you can specify a list of resources, such as a list of endpoints or networks.
- Policy rules use the `flow-mod` OF message type. Each `flow-mod` message is associated with a priority that you use to program the content addressable memory (CAM) entries on the switch. The first rule with a host identity tag (HIT) has the highest priority within a logical group.
- A policy does not have an associated priority by default.
- Endpoint policies have a higher priority than network policies. For example, if you define a policy that drops any traffic from a specified endpoint and then define another policy that directs traffic from a network to a quality of service (QoS) profile, traffic received from the endpoint is dropped.
- Policies can only have one rule.
- You must specify the policy priority when associating a policy with a resource.
- The policy priority range is from 1 to 1023.
- If you create a network policy, the `flow-mod` messages for the rules in the policy are updated to include the network’s VLAN ID in the match criteria.

Policy Configuration and Types

You can configure the following policy types and behaviors:

- Network
- Endpoint

- Port mirroring (RSPAN/SPAN)
- Legacy using WAN ports
- Middlebox
- Filter
 - Permit
 - Deny
 - Redirect (to a middlebox, mirror, uplink, IP address, or VLAN)
- Service class (Platinum, Gold, or Silver)

You can associate these policy types with resources such as a network, endpoint, host port, or WAN port.

Policy Association

Associate policies with a resource, such as a tenant, network, or endpoint using the GUI or REST APIs. When you associate a policy with a resource, the controller generates `flow-mod` messages that include the associated resource information for the rules in the policy. The controller installs the policy rules on the applicable switches.

If you associate a policy with a network, the policy is applied on all edge ports for the specified network. Network policies apply to ingress traffic only.

If you associate a policy with a tenant, the controller associates that policy with all networks created by the specified tenant. You can flag this policy to be applied to any networks created in the future by the tenant.

Network Policies

Network policies define workload behavior. There are two components to a network policy:

- Policy definition — one or more rules that define priority and action for matching results
- Association points — policies that alter network or workload behavior

To configure network behavior, associate the policy with a specific network at the provider or tenant level. You can also configure policies to specify behavior on a per-workload basis or for a set of workloads. You must associate the policy to a specific network interface for the workload.

Endpoint Policies

You can associate an endpoint policy with either ingress or egress traffic. Policy rules applied to traffic from the endpoint are considered an ingress policy; policy rules applied to traffic to the endpoint are considered an egress policy. When you create an endpoint policy, the `flow-mod` messages for the policy rules are updated to include the endpoint address. To determine which action to take, match criteria use the endpoint address and the policy direction (ingress or egress) .

You can only configure an endpoint policy from the endpoint switch. Because the controller relies on a packet sent from the endpoint to determine the fabric edge, if you create an endpoint policy before the connectivity between the endpoint and the fabric edge is discovered, the controller does not install the `flow-mod` message update until connectivity is established. If endpoint connectivity changes, any associated policies are updated with the new endpoint information.

Middlebox Policy

A middlebox (also known as a network appliance) is a networking device that changes, inspects, filters, or otherwise processes traffic for purposes other than packet forwarding. Some examples of middleboxes include firewalls that filter unwanted or malicious traffic and network address translators that modify the source and destination addresses of packets. To create a middlebox policy, specify the IP address of the current destination and the IP address of the middlebox where traffic is redirected. The only action in a middlebox policy is redirect.

Filter Policy

A filter policy can be a network or endpoint policy. If you create a filter policy on a network or endpoint, define if traffic from the specified network or endpoint is permitted, blocked, or redirected. If you apply a filter policy to egress traffic, you must send the bidirectional traffic to the switch. If the switch receives only one-way traffic from the source to the destination and traffic to the destination is blocked, the policy is inoperable unless traffic is sent from the destination to the source as well. You must associate a destination endpoint that is attached to the fabric to apply the policy.

Statistics

The software provides statistics for the switches and the controller using different types of counters. A counter provides statistical information for the entities in the infrastructure.

- The software's home page displays statistical information related to the system, provider, and tenants.
- The global system statistics provide the counter information for switches, providers, flows, and uplinks.

The provider and tenant statistics provide counter information for resources such as networks, endpoints, and policies. The system counters track the total number of flows. If the source or destination of a specific flow matches the endpoints in the provider or tenants, you can associate the flow with a specific tenant or provider. If a resource is associated with both the tenant and the provider, the counter information for the specified resource is stored in the tenant statistics.

Table 1. Statistic Types and Descriptions

Information Type	Description
Switch aggregate statistics	Displays counter information for each switch in the infrastructure or for all switches together. The information provided includes packet count, byte count, and flow count. You can clear the counters for all switches or for a switch specified by IP address.
Switch port statistics	Displays transmit and receive statistics for each port in the switch.
Database cluster counters	Displays statistics for the database server node cluster.
Database node counters	Displays statistics for the database server node.

Information Type	Description
SDNC node counters	Displays statistics for the SDNC node, which is specified by the SDNC node ID.
SDNC cluster counters	Displays the statistics for the SDNC node cluster.
Provider counters	Displays counter statistics, including information for resources such as hosts, networks, endpoints, policies, and flows, for each provider. Each provider is identified by a unique provider ID.
Provider network counters	Displays statistic information for each network associated with a specified provider or for a specific network.
Provider endpoint counters	Displays statistic information for each endpoint associated with a specified provider or for a specific endpoint.
Provider host counters	Displays statistic information for all hosts associated with a specified provider or for a specific host.
Provider policy counters	Displays policy statistics for all policies associated with a provider or for a specific policy.
Tenant counters	Displays counter statistics for all tenants associated with a provider or for a specified tenant, including resources such as hosts, networks, endpoints, policies, and flows.
Tenant network counters	Displays statistics for all networks associated with a specific tenant.
Tenant endpoint counters	Displays statistics for all endpoints associated with a specific tenant.
Tenant host counters	Displays statistics for all hosts associated with a specific tenant.
Tenant policy counters	Displays statistics for all policies associated with a specific tenant.
Flow counters	Displays all flow statistics, such as packet count and byte count. The information for all monitored flows originating from a specific endpoint, destined for a specific endpoint, or between two endpoints.
System counters	Displays counter information for the entire system, including statistics on switches, uplinks, providers, and flows.

User Authentication

To enable user authentication, the software uses remote authentication dial-in service (RADIUS) and terminal access controller access control system plus (TACACS+), along with other protocols. These

protocols secure Telnet or secure shell (SSH) access and are used in representational state transfer (REST) requests.

Remote Authentication Dial In User Service (RADIUS)

RADIUS is a networking protocol that provides centralized authentication, authorization, and accounting management for computers to connect to and use a network device. RADIUS is a client-server protocol that runs in the application layer, using the user datagram protocol (UDP) as transport. The RADIUS server is usually a daemon process running on a UNIX server.

In RADIUS, authentication and authorization are combined. If the username is found and the password is entered correctly, the RADIUS server returns an access-accept response, which includes parameters such as attribute-value pairs, that grants access to the user. The parameters, which include service type, protocol type, assigned IP address, access control lists (ACLs), and static routes to apply on the network attached storage (NAS), are configured in RADIUS.

Terminal Access Controller Access Control System (TACACS)

TACACS is a common authentication protocol for UNIX networks that allows a remote access server to forward a user's logon password to an authentication server to determine whether access is allowed to a specified system. TACACS is an encryption protocol and is therefore not as secure as TACACS+ and RADIUS protocols.

TACACS+ and RADIUS have replaced most of the earlier authentication protocols in recently built or updated networks. TACACS+ uses the transmission control protocol (TCP) port 49 and encrypts the entire packet except for the header. Dell Networking recommends TACACS as a more reliable protocol. While RADIUS combines authentication and authorization in a user profile, TACACS+ separates the two operations. TACACS+ extensions provide more types authentication requests and more types of response codes. You can implement the three separate protocols used by TACACS+ on different servers.

Supported Technologies and Protocols

AFC uses legacy switch technologies and protocols for installation and operation. To achieve multi-path L2 support and provide loop-free L2 networks, AFC uses different technologies and protocols in the switch hardware, including the following:

- Bare Metal Provisioning ([BMP](#))
- Destination Lookup Failure ([DLF](#))
- High Availability ([HA](#))
- Link Aggregation Control Protocol and Network Interface Card Teaming ([LACP and NIC Teaming](#))
- Switched Port Analyzer/Remote Switched Port Analyzer ([SPAN/RSPAN](#))
- Quality of Service ([QoS](#))
- Virtual Link Trunking ([VLT](#))


Bare Metal Provisioning (BMP)

The software uses BMP to ensure the OF switches have the correct software version and configuration. BMP uses DHCP to obtain the management IP addresses for the switches and TFTP to download the software version and configuration.

Before the active fabric (AF) controller starts up, the OF switches are upgraded to the appropriate firmware using BMP. The switches are also pre-set for optimal configuration, including the IP address of the AF controllers. All ports are configured as OF ports. The switches connect to the AF controller using their IP addresses. The AF controller uses the connected switches to build the active fabric. One AF controller is automatically assigned as the active controller and controls the switches.

To allow switches to communicate simultaneously with multiple controllers, you can configure up to two controller IP addresses on a switch. The primary controller will send and receive OF messages to or from the switches. The primary controller maintains an OF connection with the secondary controller.

When you enable BMP, AFC acts as the default DHCP server. The installation generates the `dhcpd.conf` for the DHCP server.


 **NOTE:** If BMP is enabled, ensure there is only one DHCP server in the management network.

For more information about BMP, refer to the relevant *Dell Networking Configuration Guide*.

Destination Lookup Failure (DLF)

When the switch checks the MAC destination address of the Ethernet frame to transmit it to the appropriate port, if the specified MAC address exists in the switch's L2 table, the frame is transmitted only to the port associated with that entry. If the MAC address is not included in the switch's L2 table, the frame is considered a destination lookup failure (DLF) and is transmitted to all forwarding ports on that VLAN. Unknown destination MAC packets are also considered DLF packets. A DLF tree is a variation of

spanning tree protocols (STPs) used by legacy networks. Unlike STP, there are no blocked ports in DLF protocols and all ports in the network are used for packet forwarding. The single shared tree is used only by the broadcast, unknown unicast, and multicast (BUM) packets. The known unicast traffic always uses the shortest path between two end points. DLF prevents data loops because all DLF traffic can only go through the nodes connected by the shared tree.

 **NOTE:** DLF provides node and link resiliency, but requires a longer convergence time than VLT. DLF does not support L2 multipath with paths that span switches.

When identifying a topology consisting of a subset of switches connected in a VLT-compatible method and using VLT, the controller uses the DLF tree protocol to connect the remaining switches and VLT-compatible switch groups.

DLF Port Roles

DLF roles are assigned to the ports of all switches in the network, except for the root switch. All other switches are analyzed and assigned DLF port roles. The defined port roles are:

- `DLF_ROOT_PORT` — A forwarding port on the switch that is closest to the root switch in terms of path cost.
- `DLF_DESIGNATED_PORT` — A forwarding port on the switch that designated to forward the traffic within the local area network.
- `DLF_ALTERNATE_PORT` — A forwarding port on the switch that is blocked using ingress port blocking to prevent network loops. This port forwards standard unicast traffic but not BUM traffic.

DLF Topology

You can configure a DLF topology for traffic blocking. The DLF traffic travels through the head nodes and returns to the leaf nodes if the endpoints are attached to leaf nodes. The DLF traffic uses the next hop and the unicast traffic uses the shortest path between endpoints. The DLF blocking supports DLF and unicast traffic requirements but does not affect unicast traffic.

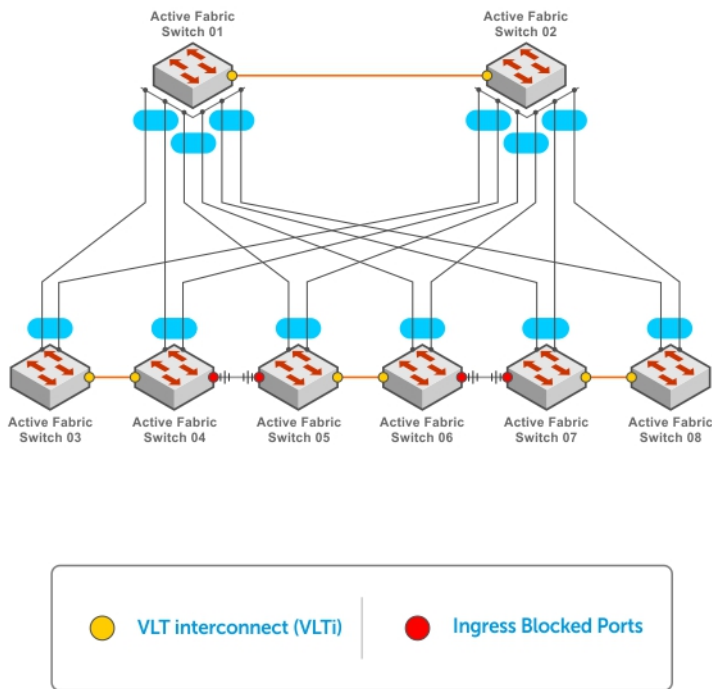


Figure 2. DLF Topology

High Availability (HA)

The active fabric controller (AFC) provides resiliency with minimal or no impact to data traffic using the high availability (HA) feature. With an HA redundancy model, two controllers (one active, one backup) work together to ensure data is maintained if a controller is unavailable. The AFC role determination logic follows the OF specification for switches using multiple controllers. The switch is configured with two controllers and establishes an OF connection with both controllers.

During installation, you select one controller as the active or primary controller and the other controller as the standby or secondary controller. However, controller roles are dynamic and are reassigned automatically, depending on controller availability. The only way to ensure one controller remains active and the other remains standby is to start the active controller first and wait until it is recognized as the primary controller before starting the second standby controller. As long as an active controller is not already present, the first controller recognized by AFC is assigned as the active controller. If an active controller is present, the other controller is automatically assigned as the standby controller.

An active controller configures forwarding entries in the OF switches. The standby controller does not receive asynchronous messages, except port status messages, from the OF switches in this mode. When the HA protocol detects an active controller failure, it reassigns the standby controller as the active controller without disrupting traffic.

HA and SDN

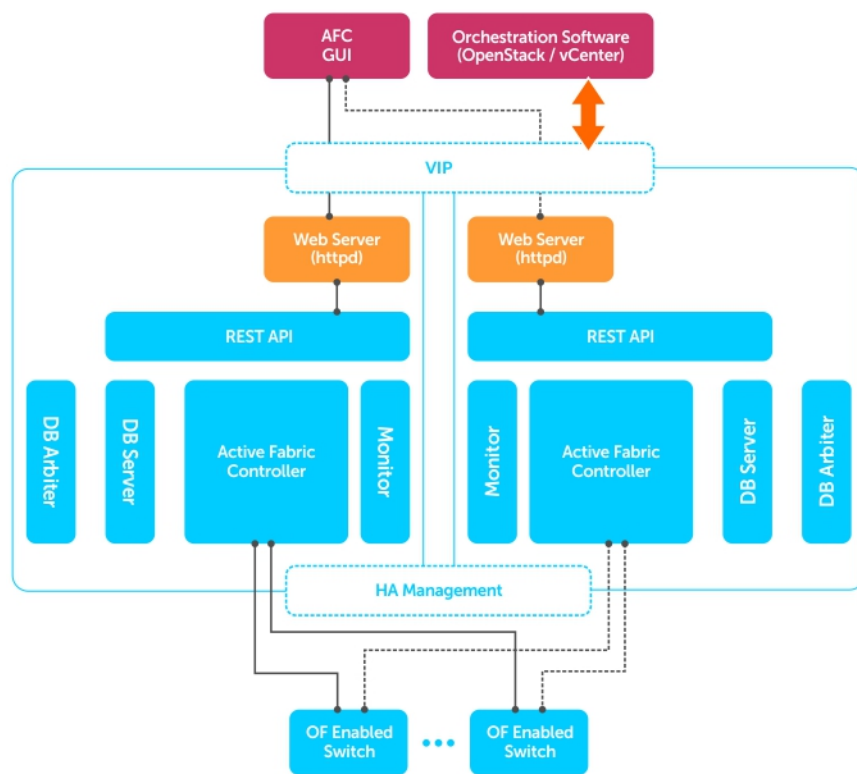
The HA protocol provides redundancy and hitless failover for the AFC during upgrades or failure. Dell Networking recommends using at least two controller servers configured for HA (one active, one standby) to enable this feature. To enable this feature on a single controller server, you can temporarily assign another server as a standby server, then initialize failover by disabling the current active server. When the upgrade completes, you can remove or reassign the server.

The active and standby controllers remain in sync using periodic checkpoints. High-level information, such as the last calculated topology and a list of end hosts, is included in the checkpoint data. The active controller hosts all network applications, such as the database server. The active controller can run as a VM, sharing the physical server with other components such as OpenStack. The standby controller can join or leave the group as needed without disrupting traffic. To ensure the standby controller has enough checkpoint data to become an active controller, it initializes a reconciliation process before becoming active. After this verification process completes, the controller's role changes to active without resetting any of the forwarding planes.

Deployment Model

The following example depicts a typical deployment with HA. The key components of this deployment are:

- AFC controllers are running in Active/Standby mode.
- The GUI uses AFC virtual IP (VIP) for REST communication and is unaware of the active and standby AFC controllers.
- The switches have two OF connections (one active, one standby).
- The switches send control and data packets over the OF connection to the active controller.



Design

HA involves the following four components:

- Role determination logic
- Messaging and transport infrastructure for checkpoints
- Initial snapshot and ongoing event synchronization for checkpoints
- Reconciliation during role transition from standby to active

When implementing this feature, be aware of the following design principles:

- The active controller determines the action to take and the standby server relies on the active controller.
- The reconciliation process minimizes the impact to data traffic.
- To be processed by the controller, the data must be valid .

Role Determination

The software uses a deterministic mechanism to decide which controller is the active controller. It also performs heartbeat checks and handles role transitions.

Checkpoints

Data originating from configurations or dynamic operations in the active controller is included in the checkpoint data.

- Configuration through REST APIs — The data resides on the active controller and in the database. The standby controller reads data from the database to build its own local data copy as part of the role transition process.
- Dynamic — The data is computed by the controller or learned through OF communication with switches.

Checkpoint data is dynamically communicated by the active controller to the standby controller. This process involves the following actions:

- The active controller provides a snapshot of the dynamically learned and computed data to the standby controller.
- The HA framework notifies the other controller software components to send snapshot data after a connection with the standby controller is established.
- The active controller performs event-based synchronization to update the standby controller.

When the standby controller starts up, the following process completes:

1. During initialization, the standby controller notifies the active controller that it is available.
2. The standby controller waits for a snapshot of the dynamic data from the primary controller.
3. The active controller prepares the snapshot and sends it to the standby controller.



NOTE: Do not modify the data while the snapshot is being prepared.

4. The standby controller processes the received snapshot.

When the controller is determined to be active, it restores the data from the database. After role assignment, the standby controller waits for snapshot data from the active controller. The active-standby synchronization process is initiated by the standby controller. After this initial synchronization, the rest of the process is event-driven and updates the standby controller based on updates from the active controller.

1. The active controller recognizes the standby controller when it receives the `Standby_Is_Ready` message from the standby controller. The cluster coordinator server establishes the relationship. After the standby controller is detected, the active controller creates a high-level snapshot, which includes topologies, SMAC learn values, and VLT configurations.
2. The HA framework sends a snapshot request to all network entities registered with the HA manager. After receiving this notification, each network element prepares its snapshot data and sends it to the standby controller.
3. The HA manager provides the infrastructure to transfer the data. Each network element opens its own connection to receive the snapshot request and send the snapshot data.
4. The HA framework uses the thrift mechanism to manage data serialization and deserialization.



NOTE: During the Snapshot in Progress state, the active controller does not process other events that would change the snapshot data. Instead, the events are queued for further processing after the snapshot process completes.

Reconciliation Process

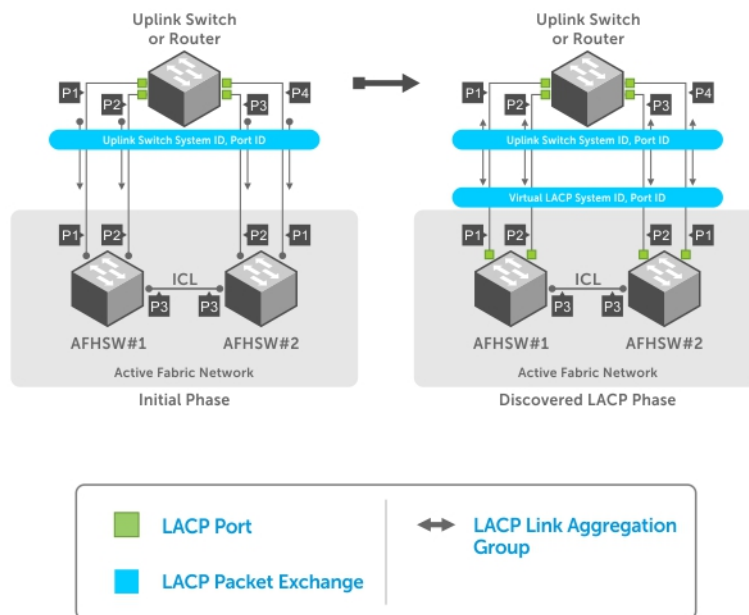
When an active controller is unavailable, the standby controller detects this and requests a role change to Active. The controller notifies the switches of the role change and begins the reconciliation process with

each switch. The reconciliation process ensures resiliency and minimal traffic disruption by verifying and synchronizing the controller's information with each switch in the active fabric. In case of traffic loss, the controller resolves the data discrepancies on the switches by adding or removing flows.

Link Aggregation Control Protocol (LACP) and NIC Teaming

To dynamically form an LACP logical group with extended connectivity from a single node to multiple nodes, the controller uses LACP to respond to messages from external ports on external legacy networks or servers connected using LACP. LACP enables the controller to dynamically aggregate multiple links as a logical group while eliminating loops and duplicate packet generation inside the group. Port parameter changes or link status changes are detected and synchronized with neighbors having updated members.

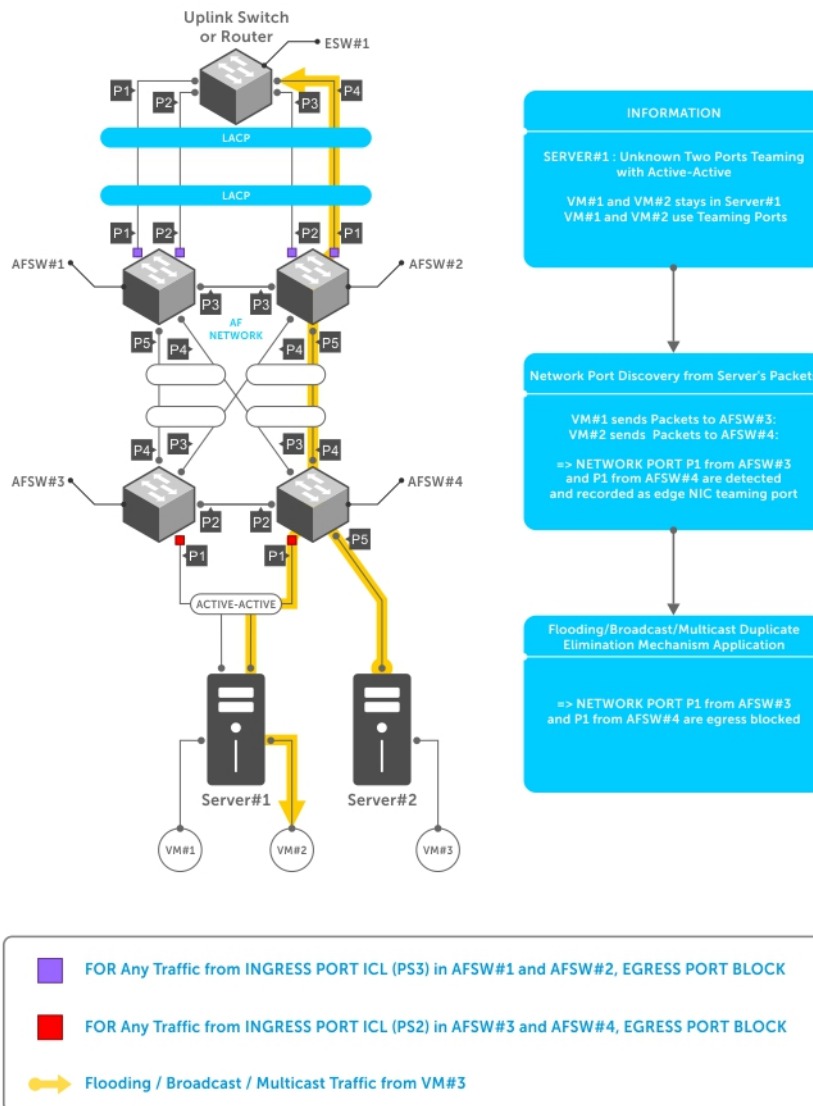
NOTE: All members of a LAG or port channel must be the same speed (1G, 10G, or 40G). Link speed combinations are not supported.



The software automatically detects LACP-enabled external nodes. This extends connectivity from the standard one-to-one LACP connection to multi-to-one LACP connection by exposing the virtual system and port information when the external nodes configure LACP over multiple links as one group. The group is terminated into multiple internal switch nodes and the internal switch nodes are directly connected to each other.

The NIC teaming feature collects NIC teaming information from the orchestration tool and optimizes traffic flow originating from or destined for NIC teaming. NIC teaming extends coverage of connectivity to multiple internal switching nodes. The multi-node extension for LACP and NIC teaming allows traffic

optimization to the external LACP node or NIC teaming by syncing the MAC addresses learned over member ports in the same LAG to all member ports in the LAG.



Standard LACP Feature

To create LAGs with the active links and synchronize LAG port forwarding status over the active links, LACP allows two nodes to exchange handshake information for each link in the LACP-enabled links. The dynamic handshake mechanism of LACP prevents unidirectional link failure or port parameter mismatches from being used for active traffic and allows one node to notify an implementation limitation (such as the maximum number of ports in a LAG) to the other nodes. Dynamic control of ports into a LAG group from LACP also eliminates the possibility of a loop due to misconfiguration of the static LAG.

LACP

The current LACP standard supports dynamic LAG formation between two LACP-enabled nodes (one-to-one). The software's LACP allows multiple nodes in the active fabric network to form a single logical LACP group. To achieve this, the external LACP node configures the links to the active fabric network as a LAG with direct reachability among the multi-nodes in the active fabric that forms the logical LACP group.

The multi-node LACP uses egress port blocking when direct reachability between LACP nodes has been established. The direct reachability requirement is based on the split horizon rule. This requirement prevents duplication of flooding, broadcast, and multicast traffic to the member ports of the same LAG. The software uses the virtual system ID and port ID generated for each LACP group instead of using the actual MAC address and port ID to represent the multiple links from the multiple nodes as originating from one logical LACP node. If you use VLT or mVLT topologies, you can use a virtual link trunking interconnect (VLTi) port.

LACP neighbors are discovered by monitoring LACP packets from active fabric external ports. The software creates a single LACP packet for each active external port and waits for response packets from uplink and downlink external ports. The response LACP packet triggers LACP to collect detailed neighbor information about multi-homed connectivity to the active fabric network, the number of links from the external LACP node, and the configured priority information. The software builds the LACP LAG using the eligible discovered member ports and synchronizes the members into the LAG group during the handshake process. The software configures itself as a preferred system priority so that configuration information, such as ports excluded from LAGs and changing the status to standby, is recognized by neighbors. If the member LACP port does not receive any LACP packets, a pruning mechanism activates to change the LACP-enabled port to a normal port as it assumes the LACP configuration was removed.

NIC Teaming

After the network topology is determined and the NIC teaming information from the orchestration or hypervisor tool is established, the NIC teaming feature allows the controller to identify the NIC teaming ports and optimize the traffic flows. The software supports the following NIC teaming topologies:

- **Active-Standby** — Only ports with an active configuration are used for sending and receiving packets and standby ports remain in standby status until failover
- **LACP** — Port initiates dynamic handshake to form LAG and ports with agreed status are used for sending and receiving packets

Traffic Optimization over LACP and NIC Teaming

Both LACP and NIC teaming may require extending source MAC addresses learned on ports from one active fabric switch to the other switches in the active fabric. For configurations using both NIC teaming and active-standby, information is not transmitted to other nodes because only active links are used for sending or receiving traffic.

LACP Requirements and Limitations

- All members of a LAG or port channel must be the same speed (1G, 10G, or 40G). Combinations of different link speeds are not supported.
- Up to 16 ports are supported for a single node.
- You must directly connect each node in the multi-node LACP to another node. If you have not established full mesh networks over the nodes, LACP does not form a single LACP group.

- Egress port blocking for links with direct reachability is implemented to prevent duplicate packets and loops. In DLF topologies, you cannot configure the ingress blocking port over a direct link between LACP nodes. If DLF uses ingress blocking on the link between LACP nodes, the failure is handled as a link failure between the nodes forming the multi-node LACP.
- If the direct link between the active fabric nodes forming the LACP group is lost, the group is split into two groups. The LACP activates one LACP group and deactivates the other group by assigning different key values. The deactivated group disables the traffic forwarding links from the LACP neighbor to prevent loops and duplicate packets.
- The highest priority is assigned to the system priority and port priority in the LACP to enable link selection for the LAG.

NIC Teaming Requirements and Limitations

- Up to 16 ports are supported for a single node.
- You must directly connect each node in the multi-node team to another node. If you have not established full mesh networks over the nodes, all links from external nodes are blocked.
- Egress port blocking for links with direct reachability is implemented to prevent duplicate packets and loops. In DLF topologies, you cannot configure the ingress blocking port over a direct link between the NIC teaming nodes. If DLF uses ingress blocking on the link between NIC teaming nodes, the failure is handled as a link failure between the nodes.

Mongo Database and Mongoose

Mongo Database

The Dell Active Fabric Controller uses MongoDB for data persistence. For more information about MongoDB, Dell recommends the following link: <http://www.mongodb.org/about/introduction/>

Mongoose

Use Sleepy Mongoose to directly access MongoDB data. To access documents using Sleepy Mongoose, use the following URIs:

- `http://<ipAddress>:27080/<database-name>/<table-name>/_find`
- `http://localhost:27080/sdnc/controller_switch/_find`

OpenStack

You can use the AFC as an orchestration tool for OpenStack configurations. This open-source tool provides a tenant portal for creating networks, servers, and VMs.

The OpenStack configuration used with the Dell Neutron plug-in provides a reference model.

To separate management traffic from the tenant networks, the OpenStack management out-of-band network uses separate interfaces. The following illustrations depict two possible use cases.

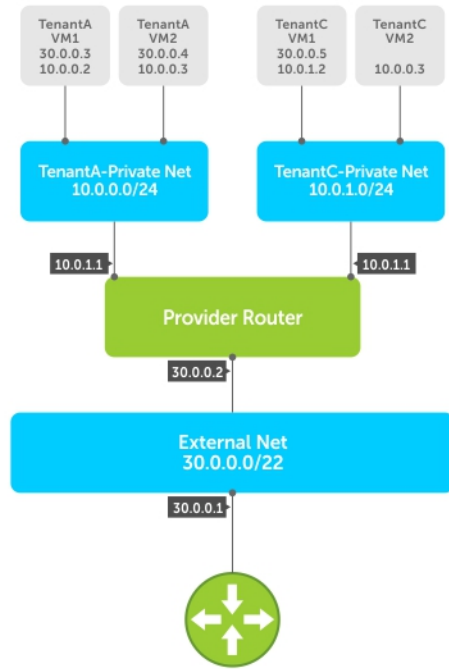


Figure 3. OpenStack Use Case: Provider Router

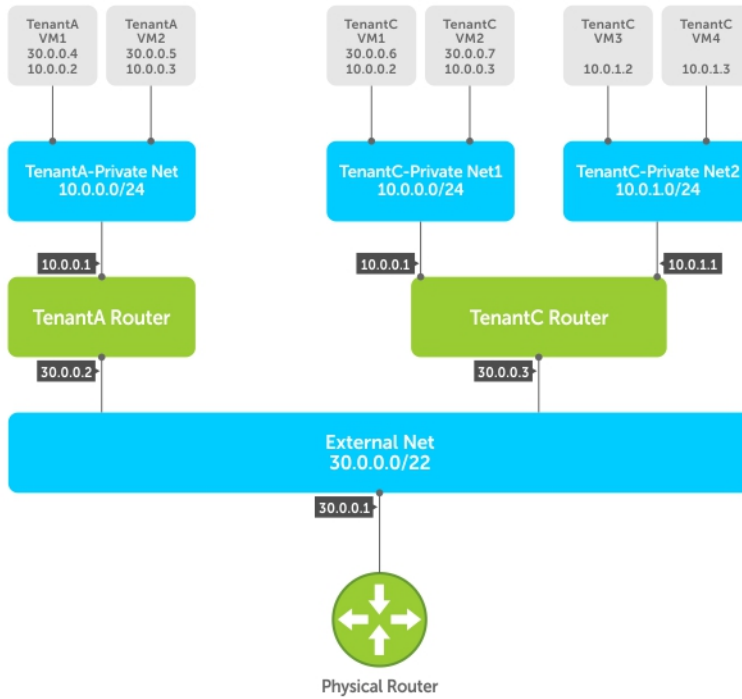


Figure 4. OpenStack Use Case: Per-Tenant Routers

The following illustration depicts a typical OpenStack configuration, where there is a management traffic interface for each node and GRE tunnels are used for traffic.

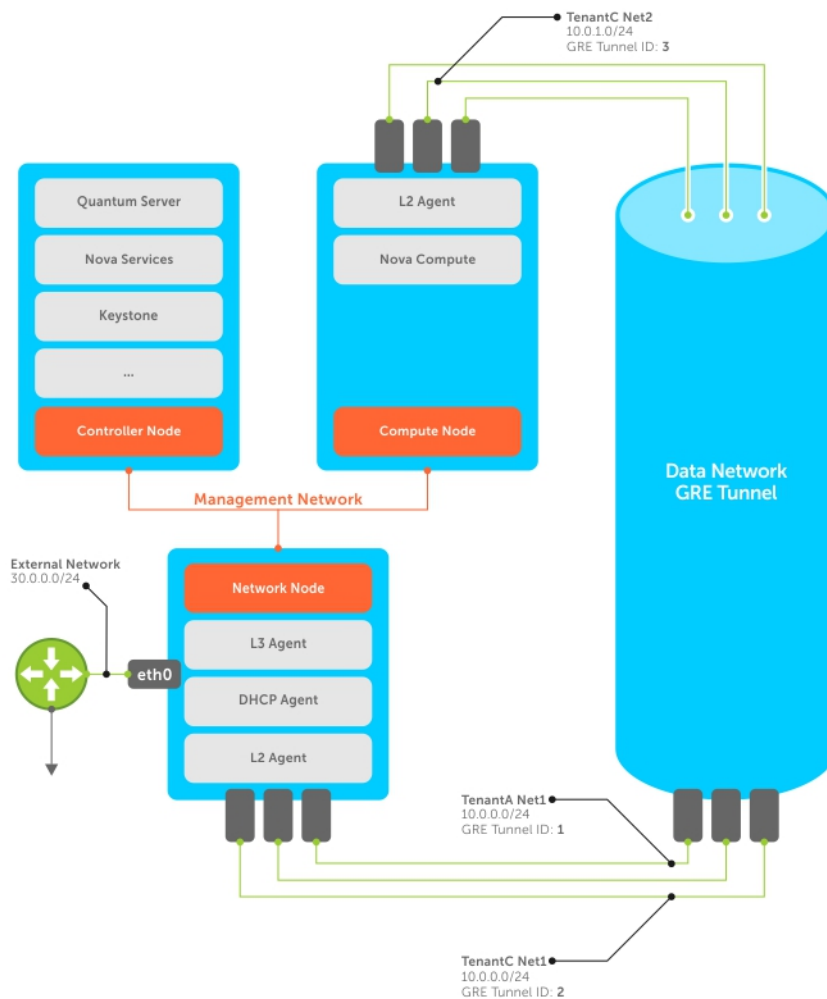


Figure 5. OpenStack Configuration


To separate traffic for each tenant, the tenants use unique VLAN ID tags. This example configuration includes OpenStack in a dual-server configuration and uses the Neutron plug-in for VLAN tagging and creation by OpenFlow-enabled Dell Networking switches.

The Neutron plug-in also allows you to create and manage virtual networks, with each virtual network containing one or more ports. You can attach a port on a virtual network to a network interface.

If you create a query, the Neutron plug-in displays a list of configured components and their status. The Neutron plug-in includes additional REST APIs to configure the physical switch and gathers information about hosts and virtual machines that are attached to a specific host. The Neutron plug-in supports the following APIs:

- `POST/networks/{networkID}`
- `GET/networks/{networkID}`
- `UPDATE/networks/{networkID}`
- `DELETE/networks/{networkID}`

- POST/networks/{networkID}/ports/{portID}
- GET/networks/{networkID}/ports/{portID}
- UPDATE/networks/{networkID}/ports/{portID}
- DELETE/networks/{networkID}/ports/{portID}
- POST/networks/{networkID}/ports/{portID}
- GET/networks/{networkID}/ports/{portID}
- UPDATE/networks/{networkID}/ports/{portID}
- DELETE/networks/{networkID}/ports/{portID}

 **NOTE:** You cannot use APIs to request information about routers and subnets.

The software is compatible with the OpenStack Horizon dashboard interface.

For more information about Dell OpenStack, refer to *Dell OpenStack-Powered Cloud Solution Reference Architecture*.

Quality of Service (QoS)

Three service levels are available: platinum, gold, and silver. The priority levels for the three service levels of QoS are predetermined. The following table provides the default queue weights for the available service levels:

Queue Number	Default Weight	Equivalent Percentage	Service Level
1	1	6.67%	Silver
2	2	13.33%	Gold
3	4	26.67%	DCB/DCBx
4	8	53.33%	Platinum

A QoS profile includes the service level, the queue priority, and the queue ID (for example: Platinum, 3, Q1). Packets are marked on the access switches so the QoS levels are valid throughout the active fabric.

Types of QoS

You can configure QoS for a flow (up to 12-tuple), an endpoint (VM or MAC), a network (VLAN) or a host (server). You can configure the following QoS type:

- **Network-based QoS** — Configure the requested service level on the switch ports specified in the policy. To configure the network-based QoS, the software uses a VLAN-based QoS OF extension.

Configuring a QoS Policy

You can configure a QoS policy on the GUI. First, choose the QoS service level. Specify the match criteria, then define the desired actions.

Switched Port Analyzer (SPAN) and Remote Switched Port Analyzer (RSPAN)

When you enable SPAN or RSPAN, port traffic is copied and sent to the specified uplink port that hosts the monitoring appliance. If you enable RSPAN, monitored traffic is also tagged with the specified VLAN ID. If you enable SPAN, no tags are added to the monitored traffic.

The controller advertises all external ports as potential edge ports. You can configure any potential edge port as a `mirrorTo` port or a monitor port. Use the GUI or REST APIs to define the policy for monitored traffic or specify the port for monitored traffic by associating a policy.

Virtual Link Trunking (VLT)

In a typical VLT topology, a connected pair of switches is known as a VLT pair and the link between them is an interconnect link. A VLT pair in a network topology acts as a single logical switch for all entities connected to both switches. The ports connected to the VLT pair are known as VLT ports. The switches within the VLT pair are also known as head nodes and the switches connected to the VLT pair are known as leaf nodes.

Forwarding Database (FDB)

A forwarding database (FDB) entry learned on a VLT port is configured on the equivalent VLT port on the VLT pair switch. In traditional networks, both switches in the VLT pair run a protocol to synchronize their FDB entries. Instead of using a protocol, AFC configures the FDB entries learned on one VLT port on the corresponding VLT port. If the leaf nodes are part of the fabric, the controller forms the port channel by including all leaf node ports connected to the VLT pair. If the leaf nodes are legacy switches connected to the fabric through uplink ports, the controller is notified that the legacy switch connected to the fabric switches using VLT. Alternatively, you can enable LACP on the legacy switches using uplinks to connect to the fabric .

Link Aggregation Control Protocol (LACP)

To identify VLT topologies in the fabric, you must enable LACP on the legacy switch for the controller . If you do not configure LACP on the legacy switches, or if you enabled NIC teaming on the server without LACP, the controller must be notified of the VLT connectivity from the external entities. By default, the controller listens for LACP packets on all fabric edge ports. If an LACP packet is received on the uplink port, the controller checks if a port channel can be configured. When it receives an LACP packet from the same entity on two different neighboring switches in the fabric, the controller responds with a virtual MAC address. Using this method, the controller can identify the VLT-compatible connectivity with a legacy switch. This also applies if the leaf nodes connected to the VLT pair are NIC teaming servers.

After the VLT pair is identified, the controller identifies the VLT ports and equivalent ports. After the port roles are identified, the controller configures VLT blocking on each of the VLT pairs to prevent data loops and duplicate packets. VLT blocking blocks all traffic from the interconnect link to the VLT port. If the leaf nodes are part of the fabric, the controller creates port channels on the leaf nodes. After pushing the VLT blocking configuration, the controller populates the FDB entries on the equivalent port when an FDB entry is learned on a VLT port.

If the VLT port is an internal port in the fabric, FDB entries are not learned on VLT ports, because a flow traveling through the VLT pair would have multiple paths between the VLT pair switches. The forwarding entries are programmed in both of the VLT pair switches based on the multiple paths between the endpoints.

VLT Link Failure

VLT link failures are handled based on the link type and the location of the VLT nodes (inside or outside the fabric). If a link between the head node and the leaf node fails, the controller performs the following actions:

1. Opens VLT blocking on the equivalent VLT port.
2. Redirects all flows destined for the impacted VLT port to the VLTi port.

VLT Interconnect (VLTi) Failure

VLTi link failures are handled by the controller based on whether the leaf nodes are part of the fabric or not. If the leaf nodes are not part of the fabric, the controller identifies one of the VLT pair nodes as the primary and the other node is assigned a backup role. The controller blocks all VLT ports on the switches assigned to the backup role. If the link between the VLT pair fails, the controller performs the following actions:

1. Identifies the primary VLT switch.
2. Administratively disables the VLT ports on the backup switch.
3. Flushes all forwarding entries destined for the blocked VLT ports.

This forces the legacy switch or server that is connected to the fabric to use the remaining link for packet forwarding. If the leaf nodes are part of the fabric, AFC analyzes the updated topology and reverts to DLF to manage the switch group. The controller performs the following actions to handle a VLTi failure:

1. Identifies the primary VLT switch.
2. Assigns the primary VLT switch as the root for the DLF tree.
3. Constructs the DLF tree for the DLF traffic.
4. Sets ingress port blocking for ports assigned as alternate ports.
5. Disables VLT blocking in both VLT head nodes.
6. Redirects any forwarding entries destined for the failed link by computing the next shortest path.


Head Node Failure

Head node failures are handled by the controller based on whether the leaf node is part of the fabric or not. If the head node fails and the leaf nodes are managed by the controller, the controller handles the failure as a VLTi failure.

Leaf Node Failure

If the leaf nodes are not part of the fabric, the controller disables VLT blocking on the remaining switch. Because the unavailable leaf node is an external entity, the controller does not make any configuration changes. The external entity receives the link down notification and the nodes handle the event based on their methodology. If the leaf node is unavailable, both of the VLT ports in the pair are unavailable as well. The controller disables VLT blocking on the ports connecting to the leaf node and removes the port channel created for these ports. The controller flushes the FDB entries configured on these ports.

Multi-domain VLT (mVLT)

 **NOTE:** Host NIC teaming from servers or LACP from external uplink switches in the same VLT pair is supported, but LACP or teaming connectivity across VLT pairs is not supported.

mVLT is a subset of VLT that you can use in a full square mesh network with higher resiliency for link and node failures and better link utilization. Unlike VLT, mVLT builds two VLT pairs, with each pair hosting two directly connected switches. The link between the two switches is known as a VLT interconnect (VLTi) link. Links between the two VLT pairs are considered VLT ports. The software configures the VLT ports from each switch node to the other VLT pair as a single LAG. LAGs help prevent loops in mVLT topologies.

To prevent loops in the VLT pairs, the software uses the split horizon and one-copy rule methods. Any packet from one VLT pair to the other VLT pair is not sent back to the original VLT. Only one copy for any received flooding, broadcast, or multicast packet from one VLT pair is delivered to the other VLT pair. If a switch in the VLT pair drops any packets from the VLTi ports to the VLT ports, only the switch that originated the packets delivers the packets to the other VLT pair and the packets are not duplicated on the other switch. Use the split horizon method by applying VLT blocking rules to drop any packets from VLTi ports to VLT ports in all switches that form the mVLT topology. The learned MAC addresses on the LAG ports acting as VLT ports in one switch in the VLT pair are duplicated in the LAG ports acting as VLT ports in the other switch in the VLT pair. This replication for learned MAC addresses over VLT ports in the same VLT pair increases bandwidth efficiency by using multiple paths to the same destination.

Link Failure Scenarios

In an mVLT topology, link failure events are handled differently depending on where the failure occurs. If at least one VLT port from each switch in the VLT pairs is active, mVLT does not require intervention in the active switching network. If all VLT ports from one switch in the VLT pair experience a link failure event, the switch moves all MAC entries from the VLT ports to the VLTi port and reconfigures the other switch in the VLT pair to remove VLTi to VLT port blocking filters.

If a VLTi link failure occurs, AFC deletes the VLT pair from the switches. The controller configures both switches in the VLT pair as leaf nodes in another VLT pair. If you have configured NIC teaming or VLT on legacy switches on any connected servers, the controller disables all edge ports for the secondary VLT switch.

Node Failure

A single node failure triggers the deletion of the configured VLT pair. The node failure is handled the same way as a VLTi link fault. Based on the node failure event, the FDB entries are flushed and reprogrammed.

Multi-Stage mVLT Extension Support


As an extension model of the mVLT topology, you can connect multiple VLT pairs to the existing mVLT topology. You can attach the new VLT pair to only one existing VLT pair in the topology. The VLT pair can share multiple VLT pairs.

REST APIs

Representational state transfer (REST) functionality is supported using the Restlet framework for the Java platform. The RESTful web API is implemented using HTTP and REST principles and is a collection of resources, including the following requirements:

- The base URI for the web service (for example, `http://example.com/resources`)
- The Internet media type for the data (for AFC, this is JSON)
- The set of operations supported by the web service using HTTP methods (for example, GET, PUT, POST, or DELETE)
- The API is hypertext-driven.

AFC uses the RESTful web service as the programmatic access interface. To use REST APIs, enter the REST API on the AFC interface.

 **NOTE:** REST APIs are case-sensitive. Do not use spaces.

To view a list of supported REST APIs, enter `sdnc/v1/help` in the GUI or by click the Build Query link and switch to List view.

REST Actions

REST uses standard HTTP actions to update the resource state. The RESTful web service supports the following HTTP resource actions:

- GET
- POST
- PUT
- DELETE

REST Resources

The REST resources used by the software are grouped into the following categories:

- System
- Managed objects
 - Infrastructure objects
 - Virtual objects
- Graph
- Counter
- Log
- Alarm

Resources

The following section describes the resources used by the software and their categories.

Resource	Category	Description
Tenant	Virtual managed object	Any client of the software's services. A tenant is associated with a unique ID during configuration. A set of resources (such as networks, hosts, endpoints, network connections, or policies) is associated with each tenant.
Provider	Virtual managed object	A client that acts as an administrator for the network infrastructure and its tenants.
Network	Virtual managed object	Virtual network (also known as a logical switch or L2 domain) that is managed by the software.
Host	Virtual managed object	A server or VM consisting of one or more endpoints that connect to different networks.
Endpoint	Virtual managed object	Point of attachment (with a unique ID, such as a MAC address or IP address) to the network managed by the software.
Policy	Virtual managed object	One or more rules that define the operations of a tenant, switch, link, network, host, or network connection.
Rule	Virtual managed object	Consists of two components: match and action. The match component contains the flow attributes and the action component defines the switch's behavior.
System	System	Deployed software components and the running environment. The components in the system are the SDNC cluster, OFC cluster, and DB cluster.
SDNC cluster	System	A cluster of SDN controller nodes and any cluster-level attributes. Can be one or two nodes.

Resource	Category	Description
SDNC node	System	An SDN controller node and its attributes.
DB cluster	System	A cluster of database nodes and any cluster-level attributes. Can be one or two nodes.
DB node	System	A database server node and its attributes.
OFC cluster	System	A cluster of OpenFlow controller nodes and any cluster-level attributes. Can be one or two nodes.
OFC node	System	An OpenFlow controller node and its attributes.
Switch	Infrastructure managed object	All switches in the physical network. This resource identifies all attributes for switches connected to the SDN controller through OpenFlow.
Switch port	Infrastructure managed object	Switches running in Hybrid mode on an SDN network to receive legacy networking packets. This resource identifies the OpenFlow-enabled ports and provides their attributes.
Uplink	Infrastructure managed object	A connection between the fabric managed by the SDN controller and a device or network that is not managed by the SDN controller, such as a legacy network.
Fabric topology	Graph	Devices discovered by the SDN controller and the links between the devices.
Flow topology	Graph	A list of devices and their connections between two endpoints.
Network topology	Graph	The devices and active links between the devices in a logical network.
Provider topology	Graph	The devices and links between devices used by a specified provider.

Resource	Category	Description
Tenant topology	Graph	The devices and links between devices used by a specified tenant.
Link	Graph	The links between switches.
Counter	Counter	Statistics for resources managed by the software. Counters are always dynamic and return records in a specified format: label (string) or counter (integer).
Log	Log	A record of state information (such as configuration actions or events) for resources managed by the software that you can retrieve for later review.
Alarm	Alarm	A notification generated by the software. You must clear the alarm after responding to the situation that activated the alarm.

REST Resource Attributes

This section lists the attributes and their values that you must include in the requests for each REST resource as part of the JSON Object. The GET request for a resource has more attribute-value pairs, depending on the number of resources associated with the specified resource instance.

Provider Objects

Table 2. Provider Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
providerId	String	Yes	Provider ID
providerName	String	No	Name of provider
providerDescription	String	No	Description of provider

Tenant Objects

Table 3. Tenant Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
tenantId	String	Yes	Tenant ID assigned by the orchestration tool
providerId	String	Yes	Provider ID
tenantName	String	No	Name of tenant
tenantDescription	String	No	Description of tenant

Network Objects

Table 4. Network Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
networkId	String	Yes	Network ID assigned by administrator
tenantId	String	Yes (if the entry is associated with a tenant)	Tenant ID, if the tenant created the network (otherwise, not applicable)
providerId	String	Yes	Provider ID
networkName	String	No	Network name
networkVlanId	Number	Yes	VLAN ID for the network
networkIP	String	Yes	The IP address of the subnet
networkPrefix	Number	Yes	The IP prefix of the IP subnet

Host Objects

Table 5. Host Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
hostId	String	Yes	Unique ID that identifies the host
tenantId	String	Yes (if the entry is associated with a tenant)	Tenant ID, if the tenant created the host (otherwise, not applicable)
providerId	String	Yes	Provider ID

Attribute	Type	Mandatory for POST request?	Description
hostname	String	No	Name of the host
hostIPAddress	Array: String	Yes	IP address configured on the host
hostEndpoints	Array: Object	No	Endpoints associated with the host

Endpoint Objects

Table 6. Endpoint Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
endpointId	String	Yes	Unique ID that identifies the endpoint
tenantId	String	Yes (if the entry is associated with a tenant)	Tenant ID, if the tenant created the endpoint (otherwise, not applicable)
providerId	String	Yes	Provider ID
endpointName	String	No	Name of the endpoint
endpointIPAddress	Array: String	No	Configured IP address for the endpoint
endpointAddress	String	No	Address of the endpoint
endpointType	String	No	Type of the endpoint (for example, VM)

Policy Objects

Table 7. Policy Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
policyId	String	Yes	Policy ID
tenantId	String	Yes (if the entry is associated with a tenant)	Tenant ID, if the tenant created the policy (otherwise, not applicable)
providerId	String	Yes	Provider ID
policyName	String	No	Policy Name

Attribute	Type	Mandatory for POST request?	Description
policyType	String	Yes	Identifies the policy type (for example, mac-auth or instream-firewall)
policyRuleList	Array: Rule	Yes (must contain at least one rule)	Displays list of rules

Rule Objects

Table 8. Rule Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
ruleName	String	Yes	Name of the rule, which is used as the ID
rulePriority	Integer	No	Priority of rule
ruleMatch	JSON Object	Yes, a combination of attributes are required (refer to ruleMatch Sub-objects and Attributes)	Match attributes (refer to ruleMatch Sub-objects and Attributes)
ruleActions	JSON Array	Optional based on desired behavior (refer to ruleActions Sub-objects and Attributes)	Set of actions (refer to ruleActions Sub-objects and Attributes)

Table 9. ruleMatch Sub-objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
inputPort	Number	No	Ingress Port
dataLayerSource	String	No	MAC address in HEX format
dataLayerDestination	String	No	MAC address in HEX format
dataLayerVirtualLan	Number	No	VLAN ID
dataLayerType	Number	No	Ethernet type
networkSource	String	No	IPv4 address
networkDestination	String	No	IPv4 address
networkProtocol	Number	No	Protocol ID
transportSource	Number	No	L4 source port
transportDestination	Number	No	L4 destination port

Attribute	Type	Mandatory for POST request?	Description
networkTypeOfService	Number	No	TOS or DSCP values

Table 10. ruleActions Sub-objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
Type	Number	Yes (if specified)	Type of action
Length	Number	Yes	Length of action
Port	Number	No	Output port
VlanID	Number	No	VLAN TAG ID for adding VLAN tag

Network Policy Objects

Table 11. Network Policy Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
policyId	String	Yes	Policy ID
networkId	String	Yes	Network ID
tenantId	String	Yes (if associated with a tenant)	Tenant ID
providerId	String	Yes	Provider ID
policyDirection	String	Yes	Identifies if the policy is applied to ingress or egress traffic
priority	Number	No	Policy priority compared with other policies or applications configured on a network. The priority of a policy or application is relevant only to the associated resource. A single policy can be the highest priority for one network, but it may be the lowest priority for another network or host.

Provider Tenant Policy Objects

Table 12. Provider Tenant Policy Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
policyId	String	Yes	Policy ID
tenantId	String	Yes (if associated with a tenant)	Tenant ID
providerId	String	Yes	Provider ID
policyDirection	String	Yes	Identifies if the policy is applied to ingress or egress traffic
tenantPolicyType	String	Yes	Identifies if the policy is applied to the network, the endpoint, or the host
priority	Number	No	Policy priority compared to other policies or applications configured on a network. The priority of a policy or application is relevant only to the associated resource. A single policy can be the highest priority for one network, but it may be the lowest priority for another network or host.

Endpoint Policy Objects

Table 13. Endpoint Policy Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
policyId	String	Yes	Policy ID
endpointId	String	Yes	Endpoint ID
tenantId	String	Yes (if associated with a tenant)	Tenant ID
providerId	String	Yes	Provider ID
policyDirection	String	Yes	Identifies if the policy is applied to ingress or egress traffic
priority	Number	No	Policy priority compared to other policies or

Attribute	Type	Mandatory for POST request?	Description
			applications configured on a network. The priority of a policy or application is relevant only to the associated resource. A single policy can be the highest priority for one network, but it may be the lowest priority for another network or host.

Network Endpoint Policy Objects

Table 14. Network Endpoint Policy Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
policyId	String	Yes	Policy ID
networkId	String	Yes	Network ID
endpointId	String	Yes	Endpoint ID
tenantId	String	Yes (if associated with a tenant)	Tenant ID
providerId	String	Yes	Provider ID
policyDirection	String	Yes	Identifies if the policy is applied to ingress or egress traffic
priority	Number	No	Policy priority compared to other policies or applications configured on a network. The priority of a policy or application is relevant only to the associated resource. A single policy can be the highest priority for one network, but it may be the lowest priority for another network or host.

Network Connection Objects

Network connections objects are read-only.

Table 15. Network Connection Objects and Attributes

Attribute	Type	Description
networkId	String	Network ID
endpointId	String	Endpoint ID
tenantId	String	Tenant ID
providerId	String	Provider ID
switchId	String	Switch IP
portId	Number	Policy ID

Network Endpoint Objects

Table 16. Network Endpoint Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
networkId	String	Yes	Network ID
endpointId	String	Yes	Endpoint ID
tenantId	String	Yes (if associated with a tenant)	Tenant ID
providerId	String	Yes	Provider ID

Host Endpoint Objects

Host endpoint objects are read-only.

Table 17. Host Endpoint Objects and Attributes

Attribute	Type	Description
hostId	String	Host ID
endpointId	String	Endpoint ID
tenantId	String	Tenant ID
providerId	String	Provider ID

Switch Port Network Objects

Switch port network objects are read-only.

Table 18. Switch Port Network Objects and Attributes

Attribute	Type	Description
portId	Number	Port ID
switchId	String	Switch data path ID
networkId	String	Network ID
tenantId	String	Tenant ID
providerId	String	Provider ID

FDB Entry Objects

FDB entry objects are read-only.

Table 19. FDB Entry Objects and Attributes

Attribute	Type	Description
networkId	String	Network ID
endpointId	String	Endpoint ID
tenantId	String	Tenant ID
providerId	String	Provider ID
switchId	String	Switch ID
portId	Number	Port Number

Cluster Node Objects

Cluster node objects are read-only.

Table 20. Cluster Node Objects and Attributes

Attribute	Type	Description
clusterNodeId	String	IP address of the node

Uplink Objects


 **NOTE:** Include the provider ID in the string. The provider ID identifies the provider associated with the object.

Table 21. Uplink Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
uplinkId	String	Yes	Unique ID for the uplink
uplinkName	String	No	Name of the uplink
uplinkDescription	String	No	Description of the uplink

Attribute	Type	Mandatory for POST request?	Description
uplinkIp	String	No	IP address of the device that connects to the uplink
uplinkMac	String	No	MAC address of the device that connects to the uplink
providerId	String	Yes	Provider ID

Host Port Objects

Host port objects are read-only.


 **NOTE:** Include the provider ID in the string. The provider ID identifies the provider associated with the object.

Table 22. Host Port Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
HostPortId	String	Yes	IP address of the host port
HostPortName	String	No	Name of the host port
HostPortDescription	String	No	Description of the host port
HostPortIp	String	No	IP address of the device connected to the uplink
HostPortMac	String	No	MAC address of the device connected to the uplink

WAN Port Objects

WAN port objects are read-only.

Table 23. WAN Port Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
WanPortId	String	Yes	IP address of the WAN port
WanPortName	String	No	Name of the WAN port
WanPortDescription	String	No	Description of the WAN port

Attribute	Type	Mandatory for POST request?	Description
WanPortIp	String	No	IP address of the device connected to the WAN port
WanPortMac	String	No	MAC address of the device connected to the WAN port
providerId	String	Yes	Provider ID

Port Monitoring Objects

Port monitoring objects are read-only.


 **NOTE:** Include the provider ID in the string. The provider ID identifies the provider associated with the object.

Table 24. Port Monitoring Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
MonitorPortId	String	Yes	IP address of the port monitor
MonitorPortName	String	No	Name of the port monitor
MonitorPortDescription	String	No	Description of the port monitor
MonitorPortIp	String	No	IP address of the device connected to the port monitor
MonitorPortMac	String	No	MAC address of the device connected to port monitor

Middlebox Port Objects

Middlebox port objects are read-only.


 **NOTE:** Include the provider ID in the string. The provider ID identifies the provider associated with the object.

Table 25. Middlebox Port Objects and Attributes

Attribute	Type	Mandatory for POST request?	Description
MiddleBoxPortId	String	Yes	IP address of the middlebox port
MiddleBoxPortName	String	No	Name of the middlebox port
MiddleBoxPortDescription	String	No	Description of the middlebox port
MiddleBoxPortIp	String	No	IP address of the device connected to the middlebox port
MiddleBoxPortMac	String	No	MAC address of the device connected to the middlebox port

REST Information Retrieval

To retrieve the following information, use the REST APIs :

- All instances of a resource (for example, using GET to view all configuration information for all tenants)
- A single, specific instance of a resource (for example, configuration information for an identified tenant)
- Resource instances that match specified criteria (for example, configuration information for all tenants with names that include certain letters)

Viewing Information

All data is returned in JSON format. REST APIs use the following format for general information retrieval or view requests (as specified in RFC 3986):

```
<scheme>://<authority>[:<port>]/sdnc/v<version-number>/<dataset-type>/<resource-type>/[<resource-key> | ALL]?<query>
```

, where <resource-type> can optionally be multi-part (by separating items with a /) and the <resource-type>/[<resource-key> tuple can be repeated multiple times. The following list contains some examples of REST APIs:

- To view configuration information for all providers, enter `http://{ControllerIP}/afcui/sdnc/v1/providers/*`
- To view all configuration information for provider with {providerId} = "pro-1," enter `http://{ControllerIP}/afcui/sdnc/v1/providers/pro-1`
- To view all networks for provider with {providerId} = "pro-1," enter `http://{ControllerIP}/afcui/sdnc/v1/networks/*@pro-1`
- To view information for a specific network (net-10) of a provider (pro-1), enter `http://{ControllerIP}/afcui/sdnc/v1/networks/net-10@pro-1`
- To view aggregate statistics for all switches, enter `http://{ControllerIP}/afcui/sdnc/v1/counters/switches/*`

Filtering Results

REST APIs support result filtering. The software supports filters for regular expressions, offset/limit pagination, and selected fields. The REST API retrieves all available information, then applies the filter to the results. To filter results, use the URI format `<URI to Resource>/?*<filter>`. You can combine filters. In each query, you can only use each filter once.

- For a regular expression filter, use the query format `*?key=<fieldName>&value=<Regular Expression>`, where *fieldName* is the attribute in the response used as the filter criteria.
- To apply pagination, use the query format `*?offset=<num1>&limit=<num2>`, where *num1* is the starting item number and *num2* is the number of records to display. For example, `offset=1000&limit=10` displays items from 1000 to 1010.
- To limit fields, use the query format `*?fields=<field>,<field>,<field>` where *field* can be any field included in the results.

The following are examples of REST APIs that use filters:

- To view the names, IDs, and descriptions associated with all tenants, enter `http:// {ControllerIP}/afcui/sdnc/v1/providers/*? fields=providerName,providerId,providerDescription`
- To view for first 50 networks of provider pro-1, enter `http:// {ControllerIP}/afcui/sdnc/v1/networks/*@pro-1?offset=0&limit=50`
- To view all the networks of provider pro-1 where networkID starts with 09, enter `http:// {ControllerIP}/afcui/sdnc/v1/networks/*@pro-1?key=networkId&value=09.*`
- To view the first five networks of provider pro-1 where networkID starts with 09, enter `http:// {ControllerIP}/afcui/sdnc/v1/networks/*@pro-1? key=networkId&value=09.*&offset=0&limit=5`
- To view the first five networks of provider pro-1 where networkID starts with 09, but return only the field networkName, enter `http:// {ControllerIP}/afcui/sdnc/v1/networks/*@pro-1? key=networkId&value=09.* &offset=0&limit=5&fields=networkName`

REST Request URIs


This section lists the uniform resource identifiers (URIs) corresponding to the implemented REST resources. The URI is used for GET, POST, PUT, and DELETE actions.


Requirements

- All POST, GET, and PUT actions require input parameters in JSON object format. For more information, refer to [REST Resource Attributes](#).
- The PUT action is an incremental update: attributes are either created or replace the existing value.
- Provider resources use the following format: `{resourceID}@{providerID}` (where *{resourceID}* is the resource ID number and *{providerID}* is the provider ID number).
- Tenant resources use the following format: `{resourceID}@{tenantID}@{providerID}` (where *{resourceID}* is the resource ID number, *{tenantID}* is the tenant ID number, and *{providerID}* is the provider ID number).

System Resources and URIs

Table 26. System Resources and URIs

System Resource	URI	Description
SDNC Cluster	GET: <code>sdnc/v1/system/sdnccluster</code>	Displays summary for each node in the SDNC cluster.
DB Cluster	GET: <code>sdnc/v1/system/dbcluster</code>	Displays summary for each node in the database cluster.
Controller cluster	GET: <code>sdnc/v1/system/controllercluster</code>	Displays summary for each node in the controller cluster (applies to Primary/Backup mode only).
Cluster node	GET: <code>sdnc/v1/system/clusternodes/{clusterNodeID}</code>	Displays detailed resource usage information about the specified cluster node, where <i>{clusterNodeID}</i> is the node's IP address.
Log entries	GET: <code>sdnc/v1/system/logs/*</code>	Displays controller log entries from most recent to oldest. To specify start date and the number of results to display, use the OFFSET and LIMIT URL parameters. By default, the number of results returned is 1000.
		 NOTE: This action retrieves logs from the database. The table name is "logs." You can enable or disable this feature in the config.properties file. By default, logging is disabled.
View log status	GET: <code>sdnc/v1/system/logstatus</code>	Displays the current logging status (enabled or disabled) for all levels, sub-levels, and components.
Configure logging	PUT: <code>sdnc/v1/system/log/level/{DEBUG TRACE}/{ON OFF}</code>	Enables or disables the specified log level (debug or trace), including all sub-levels.

System Resource	URI	Description
		 NOTE: By default, debug and trace is disabled for components at all sub-levels. Debug and trace log level status are set independently. Log status is applicable to all sublevels (for example, if you disable debug logging for sublevel 3, debug logging is also disabled for sublevels 1 and 2.
	PUT: <code>sdnc/v1/system/log/level/{DEBUG TRACE}/sublevel/{sublevel}{ON OFF}</code>	Enables or disables the specified log level (debug or trace), including all sub-levels. The sublevel range is from 1 to 5.
	PUT: <code>sdnc/v1/system/log/component/{componentName}/{componentStatus}</code>	Specify a component name and status flag.
	GET: <code>sdnc/v1/system/configfile/config.properties</code>	Displays the contents of the configuration file.
DB resource pipeline	GET: <code>sdnc/v1/system/dbresource/*</code>	Displays a server report for the database.
	GET: <code>sdnc/v1/system/dbresource/cluster</code>	Displays the cluster information for the database resource.
	GET: <code>sdnc/v1/system/dbresource/buildinfo</code>	Displays the build information for the database.
	GET: <code>sdnc/v1/system/dbresource/serverstatus</code>	Displays the database server status information.
Operational mode	GET: <code>sdnc/v1/system/operationalmode</code>	Displays the controller's operational mode (cluster, active, or standby).
Version	GET: <code>sdnc/v1/system/buildversion</code>	Displays the controller's software version.
API list	GET: <code>sdnc/v1/system/apis</code>	Displays a list available of REST request URIs.
Modules	GET: <code>sdnc/v1/system/modules/*</code>	Displays a status and description for all AFC software modules.
DB tables	GET: <code>sdnc/v1/system/db/tables/*</code>	Displays information on all tables in the database.

System Resource	URI	Description
DB Document	GET: <code>sdnc/v1/system/db/table/{tableName}</code>	Displays information about the specified table.

Provider Resources and URIs

Table 27. Provider Resources and URIs

Provider Resource	URI	Description
Providers	GET: <code>sdnc/v1/providers/*</code>	Displays information about each configured provider.
	GET, POST, DELETE: <code>sdnc/v1/providers/{providerID}</code>	Displays information about, creates, or deletes the specified provider.
Provider tenants	GET: <code>sdnc/v1/tenants/*@{providerID}</code>	Displays information about each tenant associated with the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/tenants/{tenantID}@{providerID}</code>	Displays information about, creates, or deletes the specified tenant associated with the specified provider.
Provider networks	GET: <code>sdnc/v1/networks/*@{providerID}</code>	Displays an array with an element for each network associated with the specified provider.
	GET, POST, PUT, DELETE: <code>sdnc/v1/networks/{networkID}@{providerID}</code>	Displays information about, creates, or deletes the specified network.
Provider hosts	GET: <code>sdnc/v1/hosts/*@{providerID}</code>	Displays each host associated with the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/networks/{hostID}@{providerID}</code>	Displays information about, creates, or deletes the specified host belonging to the specified provider.
Provider endpoints	GET: <code>sdnc/v1/endpoints/*@{providerID}</code>	Displays an array with an element for each endpoint associated with the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/endpoints/{endpointID}@{providerID}</code>	Displays information about, creates, or deletes the specified endpoint.
Provider network endpoints	GET, POST, DELETE: <code>sdnc/v1/networks/{networkID}/endpoints/{endpointID}@{providerID}</code>	Displays information about, creates, or deletes the specified endpoint on the specified network associated with the specified provider.

Provider Resource	URI	Description
Provider host endpoints	GET: <code>sdnc/v1/hosts/{hostID}/endpoints/*@{providerID}</code>	Displays information about each endpoint on the specified host associated with the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/hosts/{hostID}/endpoints/{endpointID}@{providerID}</code>	Displays information about, creates, or deletes the specified endpoint on the specified host associated with the specified provider.
Provider policies	GET: <code>sdnc/v1/policies/*@{providerID}</code>	Displays an array with an element for each policy associated with the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/policies/{policyID}@{providerID}</code>	Displays information about, creates, or deletes the specified policy.

Tenant Resources and URIs

Table 28. Tenant Resources and URIs

Tenant Resource	URI	Description
Tenant networks	GET: <code>sdnc/v1/networks/*@{tenantID}@{providerID}</code>	Displays an array with an element for each network associated with the specified tenant and provider.
	GET, POST, PUT, DELETE: <code>sdnc/v1/networks/{networkID}@{tenantID}@{providerID}</code>	Displays information about, creates, updates, or deletes the specified network.
Tenant hosts	GET: <code>sdnc/v1/hosts/*@{tenantID}@{providerID}</code>	Displays an array with an element for each host associated with the specified tenant of the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/hosts/{hostID}@{tenantID}@{providerID}</code>	Displays information about, creates, or deletes the specified host associated with the specified tenant of the specified provider.
Tenant policies	GET: <code>sdnc/v1/policies/*@{tenantID}@{providerID}</code>	Displays an array with an element for each policy associated with the specified tenant and provider.
	GET, POST, DELETE: <code>sdnc/v1/policies/{policyID}@{tenantID}@{providerID}</code>	Displays information about, creates, or deletes the specified policy.

Tenant Resource	URI	Description
Tenant endpoints	GET: <code>sdnc/v1/endpoints/*@{tenantID}@{providerID}</code>	Displays an array with an element for each endpoint associated with the specified tenant and provider.
	GET, POST, DELETE: <code>sdnc/v1/endpoints/{endpointID}@{tenantID}@{providerID}</code>	Displays information about, creates, or deletes the specified endpoint.
Tenant network policies	GET, POST, DELETE: <code>sdnc/v1/networks/{networkID}/policies/{policyID}@{tenantID}@{providerID}</code>	Displays information about, creates, or deletes the specified network policy for the specified network associated with the specified tenant of the specified provider.
Tenant network endpoints	GET, POST, DELETE: <code>sdnc/v1/networks/{networkID}/endpoints/{endpointID}@{tenantID}@{providerID}</code>	Displays information about, creates, or deletes the specified endpoint from the specified network associated with the specified tenant.
Tenant host endpoints	GET: <code>sdnc/v1/hosts/{hostID}/endpoints/*@{tenantID}@{providerID}</code>	Displays an array with an element for each endpoint on the specified host associated with the specified tenant of the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/hosts/{hostID}/endpoints/{endpointID}@{tenantID}@{providerID}</code>	Displays information about, creates, or deletes the specified endpoint on the specified host associated with the specified tenant.
Tenant endpoint policies	GET, POST, DELETE: <code>sdnc/v1/endpoints/{endpointID}/policies/{policyID}@{tenantID}@{providerID}</code>	Displays information about, creates, or deletes the specified endpoint policy.
Tenant endpoint on provider network	GET, POST, DELETE: <code>sdnc/v1/networks/{networkID}/endpoints/{endpointID}@{tenantID}@{providerID}</code>	Displays information about, creates, or deletes the specified tenant endpoint on the specified provider network associated with the specified tenant of the specified provider.

Policy Resources and URIs

Table 29. Policy Resources and URIs

Switch Resource	URI	Description
Policy rule	GET, POST, DELETE: sdnc/v1/rules/ {ruleID}@{policyID}@{providerID}	Displays information about, creates, or deletes the specified rule in the specified policy associated with the specified provider.
	GET, POST, DELETE: sdnc/v1/rules/ {ruleID}@{policyID}@{tenantID}@{providerID}	Displays information about, creates, or deletes the specified rule in the specified policy associated with the specified tenant on the specified provider.
Policies uplink	GET, POST, DELETE: sdnc/v1/uplinks/ {uplinkID}/policies/ {policyID}@{providerID}	Displays information about, creates, or deletes the specified policies on the specified uplink associated with the specified provider.
	GET, POST, DELETE: sdnc/v1/uplinks/ {uplinkID}/policies/ {policyID}@{tenantID}@{providerID}	Displays information about, creates, or deletes the specified policies on the specified uplink associated with the specified tenant on the specified provider.
Policy wide-area network (WAN) ports	GET, POST, DELETE: sdnc/v1/wanports/ {WANportID}/policies/ {policyID}@{providerID}	Displays information about, creates, or deletes the specified policies on the specified WAN port associated with the specified provider.
	GET, POST, DELETE: sdnc/v1/wanports/ {WANportID}/policies/ {policyID}@{tenantID}@{providerID}	Displays information about, creates, or deletes the specified policies on the specified WAN port associated with the specified tenant associated with the specified provider.
Policy host port	GET, POST, DELETE: sdnc/v1/hostport/ {hostportID}/policies/ {policyID}@{providerID}	Displays information about, creates, or deletes the specified policies on the specified host port on the specified provider.
	GET, POST, DELETE: sdnc/v1/hostport/ {hostportID}/policies/ {policyID}@{tenantID}@{providerID}	Displays information about, creates, or deletes the specified policies on the specified host port on the specified tenant associated with the specified provider.

Switch Resource	URI	Description
Policy associations	GET: <code>sdnc/v1/policies/associations/{policyID}/{providerID}</code>	Displays information about all associations for the specified policy associated with the specified provider.
	GET: <code>sdnc/v1/policies/associations/{policyID}@{tenantID}@{providerID}</code>	Displays information about all associations for the specified policy associated with the specified tenant associated with the specified provider.

Switch Resources and URIs

Table 30. Switch Resources and URIs

Switch Resource	URI	Description
Switches	GET: <code>sdnc/v1/switches/*</code>	Displays an array with an element for each switch in the infrastructure that is managed by the controller.
	GET: <code>sdnc/v1/switches/{switchID}</code>	Displays information about the specified switch.
Switch ports	GET: <code>sdnc/v1/switches/{switchID}/ports/*</code>	Displays an array with an element for each port on the specified switch.
	GET: <code>sdnc/v1/switches/{switchID}/ports/{portID}</code>	Displays information about the configuration of the specified port.
Switch links	GET: <code>sdnc/v1/switches/{switchID}/links/*</code>	Displays an array with an element for each link on the specified switch.
	GET: <code>sdnc/v1/switches/{switchID}/links/{linkID}</code>	Displays an array with an element for the specified link on the specified switch.
Switch endpoints	GET: <code>sdnc/v1/switches/{switchID}/endpoints/*</code>	Displays information on all endpoints on the specified switch.
	GET: <code>sdnc/v1/switches/{switchID}/endpoints/{endpointID}</code>	Displays information about the specified endpoint on the specified switch.
	GET: <code>sdnc/v1/switches/{switchID}/endpoints/*@{providerID}</code>	Displays information about each endpoint on the specified switch associated with the specified provider.

Switch Resource	URI	Description
	GET: <code>sdnc/v1/switches/{switchID}/endpoints/*@{tenantID}@{providerID}</code>	Displays information about each endpoint on the specified switch associated with the specified tenant associated with the specified provider.
	GET: <code>sdnc/v1/switches/{switchID}/endpoints/{endpointID}@{providerID}</code>	Displays information about the specified endpoint on the specified switch associated with the specified provider.
	GET: <code>sdnc/v1/switches/{switchID}/endpoints/{endpointID}@{tenantID}@{providerID}</code>	Displays information about the specified endpoint on the specified switch associated with the specified tenant associated with the specified provider.
Switch links	GET: <code>sdnc/v1/switches/{switchID}/links/*</code>	Displays information about each link on the specified switch.
	GET: <code>sdnc/v1/switches/{switchID}/links/{linkID}</code>	Displays information about the specified link on the specified switch.
Switch port endpoints	GET: <code>sdnc/v1/switchports/{portID}@{switchID}/endpoints/*</code>	Displays information on all endpoints on the specified port on the specified switch.
	GET: <code>sdnc/v1/switchports/{portID}@{switchID}/endpoints/{endpointID}</code>	Displays information about the specified endpoint on the specified port on the specified switch.
Switch local-area group (LAG) ports	GET: <code>sdnc/v1/switches/{switchID}/lports/*</code>	Displays information on all LAG ports on the specified switch.

Topology Resources and URIs

Table 31. Topology Resources and URIs

Topology Resource	URI	Description
Fabric topology	GET: <code>sdnc/v1/topology/fabric</code>	Displays physical topology, identifying all head nodes, leaf nodes, and adjacency information (links to other nodes) for each node.
	GET: <code>sdnc/v1/topology/all</code>	Displays physical topology, identifying all head nodes, leaf nodes, switches, middleboxes,

Topology Resource	URI	Description
	GET: <code>sdnc/v1/topology</code>	VMs, and endpoints, including adjacency information.
	GET: <code>sdnc/v1/topology/fabric/node/{nodeID}</code>	Displays all switch links in the topology.
	GET: <code>sdnc/v1/topology/logical</code>	Displays information about the specified node.
Provider topology	GET: <code>sdnc/v1/topology/provider/{providerID}</code>	Displays the logical topology, identifying relationships between switches, switch groups, and domains.
Tenant topology	GET: <code>sdnc/v1/topology/tenant/{tenantID}@{providerID}</code>	Displays physical topology, identifying all head nodes, leaf nodes, switches, middleboxes, VMs, and endpoints associated with the specified provider.
Network topology	GET: <code>sdnc/v1/topology/network/{networkID}@{tenantID}@{providerID}</code>	Displays physical topology, identifying all head nodes, leaf nodes, switches, middleboxes, VMs, and endpoints associated with the specified tenant.
Endpoint topology	GET: <code>sdnc/v1/topology/endpoints</code>	Displays physical topology, identifying all head nodes, leaf nodes, switches, middleboxes, VMs, and endpoints used by the specified network associated with the specified tenant and provider.
Flow topology	GET: <code>sdnc/v1/topology/flow/srcendpoint/{srcEndpointId}/dstendpoint/{dstEndpointId}</code>	Displays the physical topology, identifying endpoints and connectivity.
		Displays information about flows from the specified source endpoint to the specified destination endpoint.

Uplink Resources and URIs

Table 32. Uplink Resources and URIs

Uplink Resource	URI	Description
Uplinks	GET: <code>sdnc/v1/uplinks/*@{providerID}</code>	Displays an array with an element for each uplink associated with the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/uplinks/{uplinkID}@{providerID}</code>	Displays information about, creates, or deletes the specified uplink associated with the specified provider.
Binding uplinks	POST, DELETE: <code>sdnc/v1/binduplinks/{uplinkID}@{providerID}</code>	Create or delete the binding uplinks in the specified uplink associated with the specified provider.

WAN Port Resources and URIs

Table 33. WAN Port Resources and URIs

Resource	URI	Description
WAN port	GET: <code>sdnc/v1/wanports/*@{providerID}</code>	Displays an array with an element for each WAN port associated with the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/wanports/{WANportID}@{providerID}</code>	Displays information about, creates, or deletes the specified WAN port associated with the specified provider.

Port Monitor Resources and URIs

Table 34. Port Monitor Resources and URIs

Resource	URI	Description
Port monitoring	GET: <code>sdnc/v1/monitorports/*@{providerID}</code>	Displays an array with an element for each port monitor associated with the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/monitorports/{MonitorPortID}@{providerID}</code>	Displays information about, creates, or deletes the specified port monitor associated with the specified provider.

Host Port Resources and URIs

Table 35. Host Port Resources and URIs

Resource	URI	Description
Host ports	GET: <code>sdnc/v1/hostports/*@{providerID}</code>	Displays an array with an element for each host port associated with the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/hostports/{HostPortID}@{providerID}</code>	Displays information about, creates, or deletes the specified host port associated with the specified provider.

Middlebox Port Resources and URIs

Table 36. Middlebox Port Resources and URIs

Resource	URI	Description
Middlebox ports	GET: <code>sdnc/v1/middleboxports/*@{providerID}</code>	Displays an array with an element for each middlebox port associated with the specified provider.
	GET, POST, DELETE: <code>sdnc/v1/middleboxports/{MiddleboxPortID}@{providerID}</code>	Displays information about, creates, or deletes the specified middlebox port associated with the specified provider.

Flow Resources and URIs

Table 37. Flow Resources and URIs

Flow Resource	URI	Description
Flow resource	GET: <code>sdnc/v1/flows/srcendpoint/{srcEndpointId}</code>	Displays the flows for the specified source endpoint.
	GET: <code>sdnc/v1/flows/dstendpoint{dstEndpointId}</code>	Displays the flows for the specified destination endpoint.
	PUT: <code>sdnc/v1/flowmonitor/srcendpoint/{srcEndpointId}/state/{on off}</code>	Configures flow monitoring for all flows from the specified source endpoint to the specified state.
	PUT: <code>sdnc/v1/flowmonitor/dstendpoint/{dstEndpointId}/state/{on off}</code>	Configures flow monitoring for all flows from the specified destination endpoint to the specified state.

Counter Resources and URIs

Table 38. Counter Resources and URIs

Counter Resource	URI	Description
System counter statistics	GET: <code>sdnc/v1/counters/system</code>	Displays counter information for switches and providers.
Switch aggregate statistics	GET: <code>sdnc/v1/counters/switches/*</code>	Displays an array containing aggregate packet, byte, and flow statistics for each switch in the infrastructure.
	GET, DELETE: <code>sdnc/v1/counters/switches/{switchID}</code>	Displays or deletes the information for the specified switch, including aggregate packet, byte, and flow statistics.
Switch port statistics	GET, DELETE: <code>sdnc/v1/counters/switchports/*@{switchID}</code>	Displays or deletes information for transmit and receive statistics for each port in the specified switch.
	GET, DELETE: <code>sdnc/v1/counters/switchports/{portID}@{switchID}</code>	Displays or deletes information transmit and receive statistics for the specified port on the specified switch.
OFC Stats	GET: <code>sdnc/v1/counters/ofcstats/switches/*</code>	Displays an array containing statistics for OF messages for each switch.
	DELETE: <code>sdnc/v1/counters/ofcstats/switches/*</code>	Clears all OF-related counters for all switches.
	GET: <code>sdnc/v1/counters/ofcstats/switches/{switchID}</code>	Displays statistics for OF messages for the specified switch.
	DELETE: <code>sdnc/v1/counters/ofcstats/switches/{switchID}</code>	Clears all OF-related counters for the specified switch.
Provider Counters	GET: <code>sdnc/v1/counters/providers/*</code>	Displays provider statistics for each provider.
	GET: <code>sdnc/v1/counters/providers/{providerID}</code>	Displays provider statistics for the specified provider.
Tenant Counters	GET: <code>sdnc/v1/counters/tenants/*@{providerID}</code>	Displays tenant statistics for the specified provider.

Counter Resource	URI	Description
	GET: <code>sdnc/v1/counters/tenants/{tenantID}@{providerID}</code>	Displays tenant statistics for the specified tenant in the specified provider.
Flow counters	GET, DELETE: <code>sdnc/v1/counters/flows/srcendpoint/{srcEndpointId}</code>	Displays or deletes flow information for the specified source endpoint.
	GET, DELETE: <code>sdnc/v1/counters/flows/dstendpoint/{dstEndpointId}</code>	Displays or deletes flow information for the specified destination endpoint.
Switch queue statistics	GET, DELETE: <code>sdnc/v1/counters/queues/*@{switchID}</code>	Displays or deletes statistics for all port queues on the specified switch.
	GET, DELETE: <code>sdnc/v1/counters/queues/{portID}@{switchID}</code>	Displays or deletes statistics for the port queue of the specified port on the specified switch.

Edge Port Resources and URIs

Table 39. Edge Port Resources and URIs

Edge Port Resource	URI	Description
Edge port	GET: <code>sdnc/v1/edgeports/*</code>	Displays an array with an element for each edge port.

Port Resources and URIs

Table 40. Port Resources and URIs

Port Resource	URI	Description
WAN ports	GET: <code>sdnc/v1/wanports/*@{providerId}</code>	Displays information for all WAN ports.
	GET, POST, DELETE: <code>sdnc/v1/wanports/{WANportID}@{providerId}</code>	Displays information about, creates, or deletes information for the specified WAN port.
Host ports	GET: <code>sdnc/v1/hostports/*@{providerId}</code>	Displays information for all host ports.
	GET: <code>sdnc/v1/hostports/{hostportID}@{providerId}</code>	Displays information about the specified host port.
Middlebox ports	GET: <code>sdnc/v1/middleboxports/*@{providerId}</code>	Displays information for all middlebox ports.

Port Resource	URI	Description
	GET: <code>sdnc/v1/middleboxports/{middleboxportID}@{providerId}</code>	Displays information about the specified middlebox port.
	GET: <code>sdnc/v1/middleboxports/{middleboxportID}@{providerId}</code>	Displays information about the specified middlebox port.
	GET, POST, DELETE: <code>sdnc/v1/middleboxports/{middleboxportID}@{providerId}</code>	Displays information about, creates, or deletes the specified middlebox port.
Port monitoring	POST, DELETE: <code>sdnc/v1/monitorports/{mPortId}@{providerId}</code>	Creates or deletes information for the specified port monitor.

StaticFlow Resources and URIs

Table 41. StaticFlow Resources and URIs

StaticFlow Resource	URI	Description
StaticFlow	POST, DELETE: <code>static/wm/v1/staticflow</code>	Posts or deletes ACLs on switches for troubleshooting.

Help Resources and URIs

Table 42. Help Resources and URIs

Help Resource	URI	Description
Help	GET: <code>sdnc/v1/help</code>	Displays a list of all available APIs.

Device Resources and URIs

Table 43. Device Resources and URIs

Device Resource	URI	Description
Device	GET: <code>sdnc/v1/devices</code>	Displays information for all devices recognized by the controller.

QoS Resources and URIs

Table 44. QoS Resources and URIs

QoS Resource	URI	Description
Service class	POST, DELETE: <code>sdnc/v1/serviceclass/{serviceclass}/tenant/{tenantID}@{providerID}</code>	Create or delete QoS information for the specified tenant associated with the specified provider.

LACP Resources and URIs

Table 45. LACP Resources and URIs

Resource	URI	Description
LACP	GET: <code>sdnc/v1/lacp/lacpbond</code>	Displays information about NIC teaming bonds.
	GET, POST, PUT, DELETE: <code>sdnc/v1/lacp/staticbond/{bondID}</code>	Displays, creates, updates, or deletes the specified bond information.

Link Discovery Resources and URIs

Table 46. Link Discovery Resources and URIs

Resource	URI	Description
Provider link discovery	GET: <code>sdnc/v1/linkdiscovery/provider/{providerID}</code>	Displays physical topology, identifying head nodes, leaf nodes, switches, middleboxes, and endpoints associated with the specified provider.
Tenant link discovery	GET: <code>sdnc/v1/linkdiscovery/tenant/{tenantID}@{providerID}</code>	Displays physical topology, identifying head nodes, leaf nodes, switches, middleboxes, and endpoints associated with the specified tenant associated with the specified provider.
Network link discovery	GET: <code>sdnc/v1/linkdiscovery/network/{networkID}@{tenantID}@{providerID}</code>	Displays physical topology, identifying head nodes, leaf nodes, switches, middleboxes, and endpoints on the specified network associated with the specified tenant associated with the specified provider.

REST Errors

The following codes are returned for any errors encountered by REST.

Table 47. REST Error Codes and Responses

Condition	HTTP Code	Json Response (Single Object) in HTTP body	Json Response (Collection) in HTTP body
1	200 OK	{Object}	[Array]
2	404 Not found	{error message – not found}	{error message}
3	204 No content	No data	No data
4	400 Bad request	{error message}	{error message}
5	404 Invalid syntax	{error message – invalid syntax}	{error message}
6	500	{error message – internal error + contextual message}	{error message}
7	200 OK	No data	No data

Logging

The software uses different classes for logging, with each class providing different data depending on the message type and level. To configure the logging class, edit the `log-properties.xml` file. The log is saved as `SDNC_VERBOSE.log`, where each new version is appended with a number (for example, `SDNC_VERBOSE_1.log`).

- TRACE — Logs TRACE messages only. Provides more detailed information than the DEBUG level.
- DEBUG — Logs TRACE and DEBUG messages. Provides high-level information for debugging.
- INFO — Logs TRACE, DEBUG, and INFO messages. Provides information about the progress of the application at a low level, such as incoming and outgoing messages.
- WARN — Logs TRACE, DEBUG, INFO and WARN messages. Provides cautionary information, such as lost connections and threshold limits.
- ERROR — Logs TRACE, DEBUG, INFO, WARN, and ERROR messages. Provides information about critical situations.
- OFF — Disables logging.

Logging Sublevels

Logging sublevels provide five sublevels for the DEBUG and TRACE logging classes. You can classify messages from 1 to 5 based on priority, with 1 as the highest priority and 5 as the lowest.

- To enable debugging, enter `/sdnc/v1/system/log/level/debug/on`.
- To use the debugging sublevels, enter `sdnc/v1/system/log/level/debug/sublevel/{1-5}/on`.
- To disable debugging, enter `/sdnc/v1/system/log/level/debug/off`.
- To enable trace logging, enter `/sdnc/v1/system/log/level/trace/sublevel/2/on`.
- To use the trace logging sublevels, enter `/sdnc/v1/system/log/level/trace/sublevel/[1-5]/on`.
- To disable trace logging, enter `/sdnc/v1/system/log/level/trace/on`.

You can also configure batch-level updates. For example, if you enter `/sdnc/v1/system/log/level/debug/sublevel/3/on`, debugging is enabled on levels 1–3. To disable trace logging on level one and two, enter `sdnc/v1/system/log/level/trace/sublevel/2/off`.

Component Logging

Component logging provides additional tagging for DEBUG and TRACE sublevels. To enable component logging, enter `/sdnc/v1/system/log/component/*/on`. To disable component logging, enter `/sdnc/v1/system/log/component/*/off`. You can also specify components. For example, to enable component logging for the topology, enter `/sdnc/v1/system/log/component/topology/off`.

SDNC Database Logging

All INFO, WARN, and ERROR log messages are sent to the database. To allocate up to 3 MB for this information, edit the `logTableSizeMB=3` item in the `config.properties` file. Logs are time-stamped and organized in descending order, with the most recent information first. When the memory is full, the older information is removed and new information is automatically inserted. SDNC database logging is disabled by default. To enable SDNC database logging, configure `logs2db=true` in the `config.properties` file. To disable SDNC database logging, configure `logs2db=false` in the `config.properties` file.

Upgrading RPM-Installed Software

When you start the controller, select the software version to install. If there is an existing installation, the software upgrades the existing version of the AFC software by updating the binary files and libraries, reconfiguring the running environment, and upgrading the existing configuration variables.

Upgrading Single-Server Deployments


For a single-server deployment without HA, you must schedule downtime for an upgrade. Single-server deployments are not hitless.


1. Run the `install_afc.py` file.
2. Enter `1` to start a new HA configuration.
3. Enter `1` to select the server to upgrade.
4. Enter `Y` to confirm the upgrade.
5. Run the `setup_afc.sh` file.
6. Enter `false` to disable HA.
7. Update the AFC values or press `Enter` to continue the installation.
8. Update the multicast values or press `Enter` to continue the installation.
9. Enter `true` to enable BMP. To disable BMP, enter `false`.
10. Enter the netmask DHCP IP range.
11. Enter `Y` to enable Syslog. To disable Syslog, enter `N`.
12. If you enable Syslog, enter the configuration information for the Syslog server.
13. Enter `Y` to enable authentication. Enter `N` to disable authentication.
14. If you enable authentication, enter a number for the authentication mode:
 - Enter `1` for RADIUS.
 - Enter `2` for TACACS.
 - Enter `3` for no authentication.
15. Press `Enter` to update the listed IP addresses.

Upgrading Dual-Server Deployments

For a dual-server deployment with HA, the upgrade is hitless and no downtime is required.

1. Run the `install_afc.py` file.
2. Enter `2` to join an existing HA configuration.
3. Enter `1` to select the primary server.
4. Enter `Y` to confirm the transfer.

 **NOTE:** If the configuration values transfer successfully to the secondary node, you do not need to manually configure the secondary node.

5. Run the `setup_afc.sh` file.
6. Enter the IP address for the MongoDB server followed by a comma, then enter the IP address of the secondary server.
7. Enter `true` to enable HA.
8. Update the AFC values or press `Enter` to continue the installation.
9. Configure the AFC controller IP address and credentials in OpenStack.
10. Update the multicast values or press `Enter` to continue the installation.
 -  **NOTE:** The primary and secondary nodes must use the same multicast address, which cannot be used for any other HA configuration.
11. Enter `true` to enable BMP. To disable BMP, enter `false`.
12. Enter the netmask DHCP IP range.
13. Press `Enter` to update the listed IP addresses.

Active Fabric Controller User Interface

Use the GUI to access the AFC for monitoring and debugging or to control the active fabric infrastructure. You can also access the active fabric infrastructure using the OpenStack Horizon dashboard.

The software uses a web-based GUI to monitor the status of provisioned networks and to issue policies or other configuration data. The GUI uses Javascript, PHP, HTML, and CSS, and the web server uses Apache 2. The GUI accepts the user's request and redirects the action to the software using REST APIs to communicate with the software. When the software returns the result, the GUI displays it in the browser.

To use the GUI, type the URL in a browser (for example, `http://{ControllerIP}/afcui`, where `{ControllerIP}` is the IP address of the primary controller).

The GUI is query-based, meaning that the primary user input is a uniform resource identifier (URI). To view data, enter the appropriate URI in the "Enter a Query" text field. For example, to view a list of switches in the network, enter the following URI: `http://controller-ip-address:tcp-port-number/sdnc/v1/switches/*`, where `controller-ip-address` is the IP address of the controller and `tcp-port-number` is the TCP port number for the web service running the software. Supported URIs are provided in the [REST APIs](#) section. To view a list of APIs and build queries, click the **Build Query** link in the top banner of the GUI.

Supported Browsers

- Internet Explorer (9 or later)
- Mozilla Firefox (24.0 or later)
- Google Chrome (25.0 or later)
- Apple Safari (6.0.5 or later)

For optimal resolution, set your screen resolution to 1024 x 768 (or greater).


Components

The GUI uses existing Javascript packages, including the following:

- JQuery to render the HTML content
- Javascript Infovis Toolkit (JIT) to render the topology
- Backbone to provide the model-view-controller (MVC) framework for Javascript
- JSON2 to manage the HTTP call to PHP in JSON format (table format is used by default)
- Underscore to provide functional programming support

HTTP and HTTPS Support

The GUI supports both HTTP and HTTPS. The Apache server and PHP are used for HTTP and HTTPS communication between the GUI and the software.

 **NOTE:** Communication between the browser and the Apache server is independent of the communication between the Apache server and the software. Each communication type can use different protocols for support.











Features

The GUI supports the REST APIs used by the software and can perform GET, POST, and DELETE operations. You can also:

- View a graphical representation of the network topology
- Issue or configure a policy
- View basic system information for monitoring purposes
- Use the query tree view and builder to facilitate selection and assembly of REST API queries
- Filter results to provide regex, field selection, and pagination

GUI Icons


The following lists the icons used in the GUI and their associated functions. If you hover over an icon, the function displays.

Icon	Function
	UI help
	Options view
	Refresh
	Run query
	Clear query or text
	Add filter
	Delete filter
	Delete query or query results
	Save result
	Save query

GUI Features

The following features are available in the GUI.

Help

The help icon displays in several areas of the GUI. Click the  icon to read a brief description of the relevant function or form.

Main Screen

The GUI viewing area is divided into three sections:

- The left pane displays statistics for the connected system, such as the IP addresses, current status, and use of resources.
- The center pane displays data results for a user query or request.
- The right pane provides alternate methods to retrieve and display data.

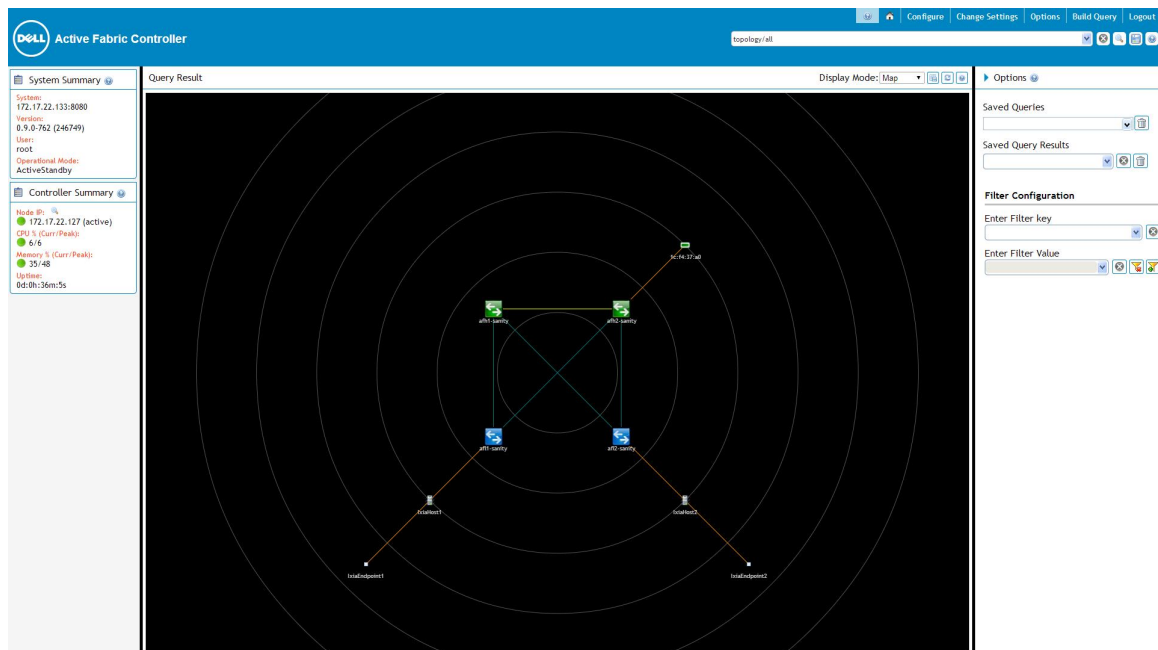




Figure 6. AFC GUI — Main Screen

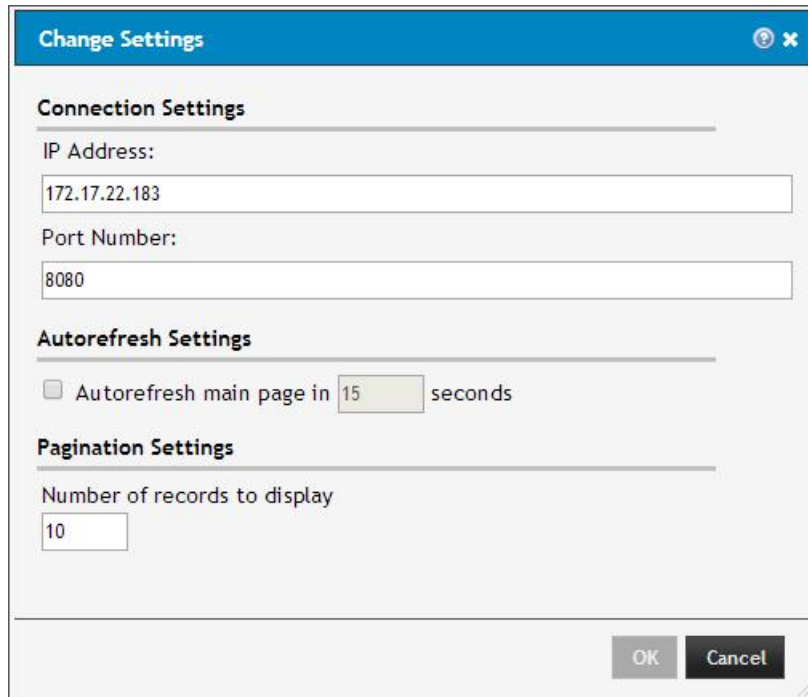
To create a larger viewing area for the center pane, click the  to close the right pane. Click the solid arrow at the top of the pane to close the pane or click the **Options** link above the banner. The center pane includes a header. The data displays in the left pane and icons or data function fields display in the right pane. The banner at the top of the GUI contains an input field to enter a query. There are icons and links for specific functions above the banner.

- Click the  icon to view a brief description of the GUI.
- Click the **Configure** link to add or delete resources.
- Click the **Change Settings** link to change the system settings.
- Click the **Options** link to open or close the right pane.
- Click the **Build Query** link to launch the interactive query builder.
- Click the **Logout** link to close the current browser session.

Changing System Settings

To configure settings for the following items, click the **Change Settings** link in the top banner. You can configure the following settings:

- AFC server IP address and port number
- Auto-refresh duration
- Number of query results to display



The screenshot shows a 'Change Settings' dialog box with the following fields and options:

- Connection Settings:**
 - IP Address: 172.17.22.183
 - Port Number: 8080
- Autorefresh Settings:**
 - Autorefresh main page in 15 seconds
- Pagination Settings:**
 - Number of records to display: 10

Buttons: OK, Cancel


Figure 7. AFC GUI — Change Settings Window

To direct all subsequent queries to the specified address, enter the controller's IP address and port number in the form that is displayed. The data is saved as a browser cookie and is automatically retrieved when you launch the GUI.

Submitting Queries

Enter the appropriate URI to submit a query in the text field in the banner. The URI can be complete (beginning with `http`) or incomplete. If you use an incomplete URI, the GUI inserts the preamble using the current system IP address and port number. For example, all of the following query formats generate a list of switches:

- `http://controller-ip-address:tcp-port-number/sdnc/v1/switches/*` (where *controller-ip-address* is the controller's IP address and *tcp-port-number* is the TCP port)
- `sdnc/v1/switches/*`
- `switches/*`

Press Enter to submit the query or click the  icon next to the text field. The GUI saves recently entered queries as browser cookies. You can view the list of recent queries by clicking the drop-down list for the entry field in the banner.

Auto-filtering is also supported in the entry field. When you type in the entry field, the drop-down list is automatically populated with all recent queries containing the entered text. Use the mouse to select a query or use the down arrow and Enter keys.

Selecting Display Modes

The GUI provides multiple viewing options for query results: Table, Tree, Map, Chart, and Tile. Table is the default viewing mode. You can apply Table and Tree viewing modes to any data results. Map, Chart, and Tile viewing modes require data in a particular format and you can only use them for specific query results. Use Map viewing mode for topology queries. To display statistical information, use Chart and Tile viewing modes. To select a viewing mode, click the appropriate viewing mode in the drop-down **Display Mode** list in the center pane header.

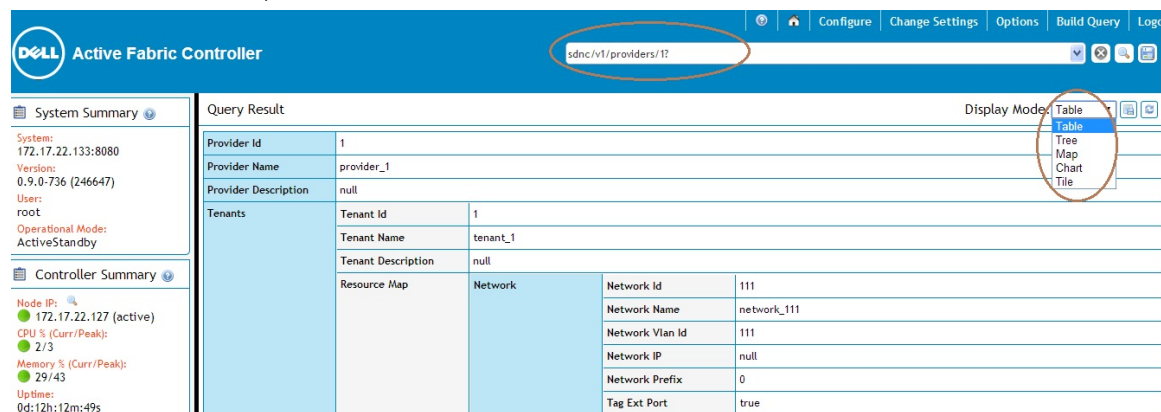



Figure 8. AFC GUI — Display Modes Menu

Saving Queries

To save a query, click the  icon next to the query entry field. Enter the query name and select the directory for the saved query in the Save Query form. The currently selected viewing mode is also saved with the query. To create a new folder for the saved query, click the **New Folder** button. Saved queries are stored on the server running the web service and are available for all users using that server to launch the GUI.

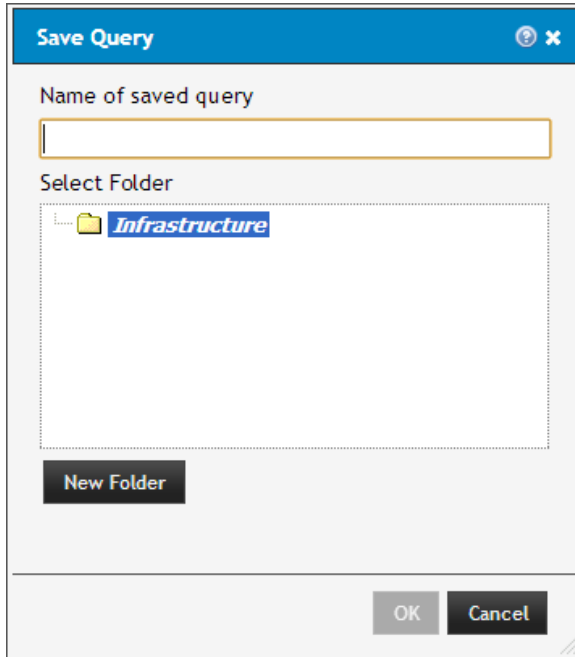


Figure 9. AFC GUI — Save Query

Submitting Saved Queries

To submit a saved query, open the drop-down **Saved Queries** list in the right pane. If the right pane is not visible, click the **Options** link above the banner to open the pane. Saved queries are listed in the directory that you select when you save the query. Locate the query you want to use and select it to submit the query. The data is displayed using the viewing mode that was used when the query was saved.

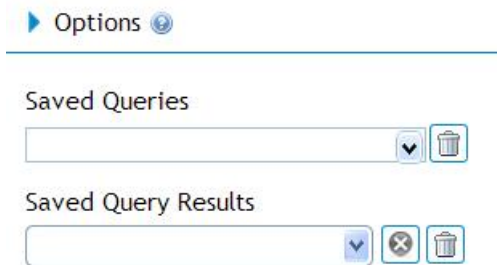




Figure 10. AFC GUI — Options

Deleting Saved Queries

To delete a saved query, locate and select the query you want to delete, then click the  icon. After you confirm the deletion, the query is deleted.

Saving Query Results

To save query results, click the  icon in the center pane header and enter a name for the saved results. Saved query results are stored on the server running the web service and are available for all users using that server to launch the GUI.

Displaying Saved Query Results

To display saved query results, select the results from the drop-down **Saved Query Results** list. The results are retrieved from the controller and displayed in the center pane.

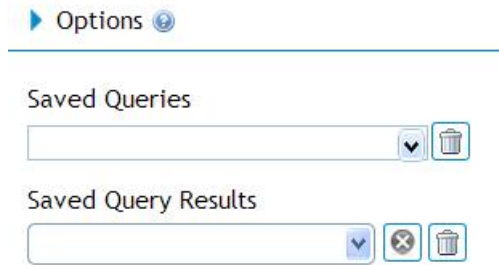




Figure 11. AFC GUI — Options

Deleting Saved Query Results

To delete saved query results, select the results from the drop-down **Saved Query Results** list, then click the  icon. After you confirm the deletion, the results are deleted.

Applying Filters

To filter data results, specify how the results should be filtered in the **Filter Configuration** section of the right pane. Enter the filter key in the **Enter Filter Key** entry field. The GUI automatically populates the filter keys that are applicable to the data results. To see the full list, open the drop-down list. Auto-filtering is supported for key names, so as you type in the entry field, the matching list of key names displays in the drop-down list. Select the appropriate key name. To clear any text in the entry field, click the  icon next to the entry field.

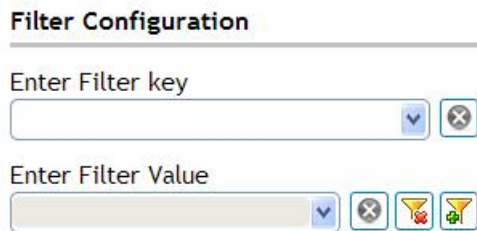



Figure 12. AFC GUI — Filter Configuration

After you select a filter key, enter a value in the **Enter Filter Value** entry field. The GUI saves previously entered filter values as browser cookies. To view a list of saved values, open the drop-down list, or begin


typing in the entry field to view a list of values containing the characters you entered. Select the appropriate listed value.

To apply the filter and refresh the data results, click the  icon. The GUI displays only the data that matches the selected filter criteria. This method is applicable only to query results containing multiple records. You can also retrieve data filtered by the server by providing filtering specifications as query parameters in the URI. For more information, refer to [REST APIs](#).

Removing Filters

The selected filter is applied until it is deleted. To remove a filter, click the  icon.

Refreshing Center Pane Content

To manually refresh the content that displays, click the  icon in the center pane header. The currently displayed query or saved results is updated with the latest data.

Using Auto-Refresh

To select auto-refresh settings, click the **Change Settings** link. You can only apply auto-refresh to query data. If the retrieved data is updated during the refresh or is different from the currently displayed data, the GUI updates the display.

Pagination

The GUI displays information and navigation links related to pagination in the center pane header. Pagination applies to the Table viewing mode only. By default, 10 entries display on each page. Click the **NEXT** link to view the next set of results or click the **PREV** link to view the previous set. To change the number of displayed results, click the **Change Settings** link.

Topology-Based Navigation and Information Retrieval

When you view the topology using the Map viewing mode, you can use topology-based navigation. For example, to retrieve and display the fabric topology, submit the following query: `topology/fabric` or `sdnc/v1/topology/fabric`. Select Map as the viewing mode from the drop-down **Display Mode** list. To view a brief description of the node in a tooltip, hover the cursor on a node in the topology display. When you move the cursor, the tooltip disappears.

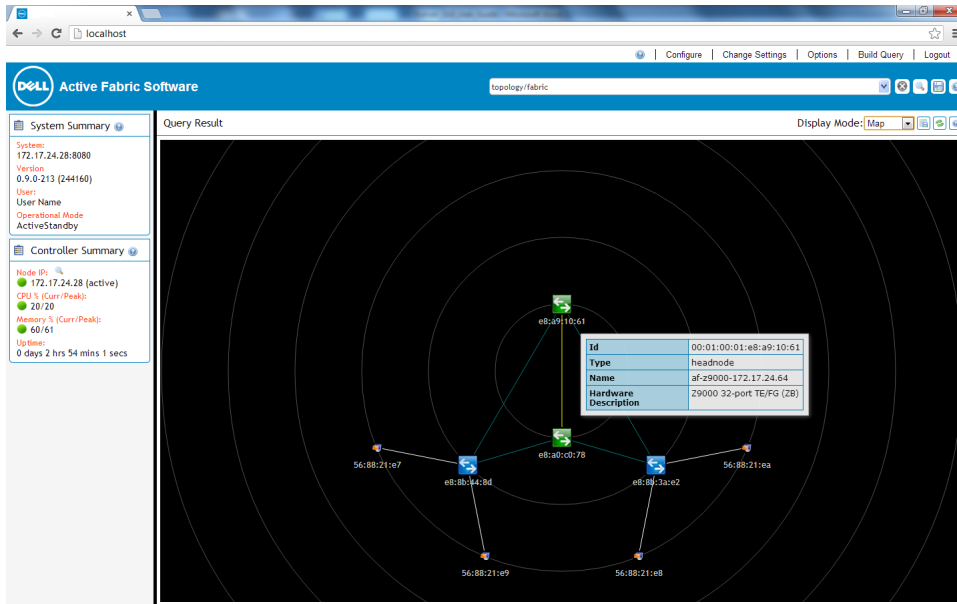


Figure 13. AFC GUI — Topology Tooltip

To drill down and view additional information, click a node or link in the topology display.

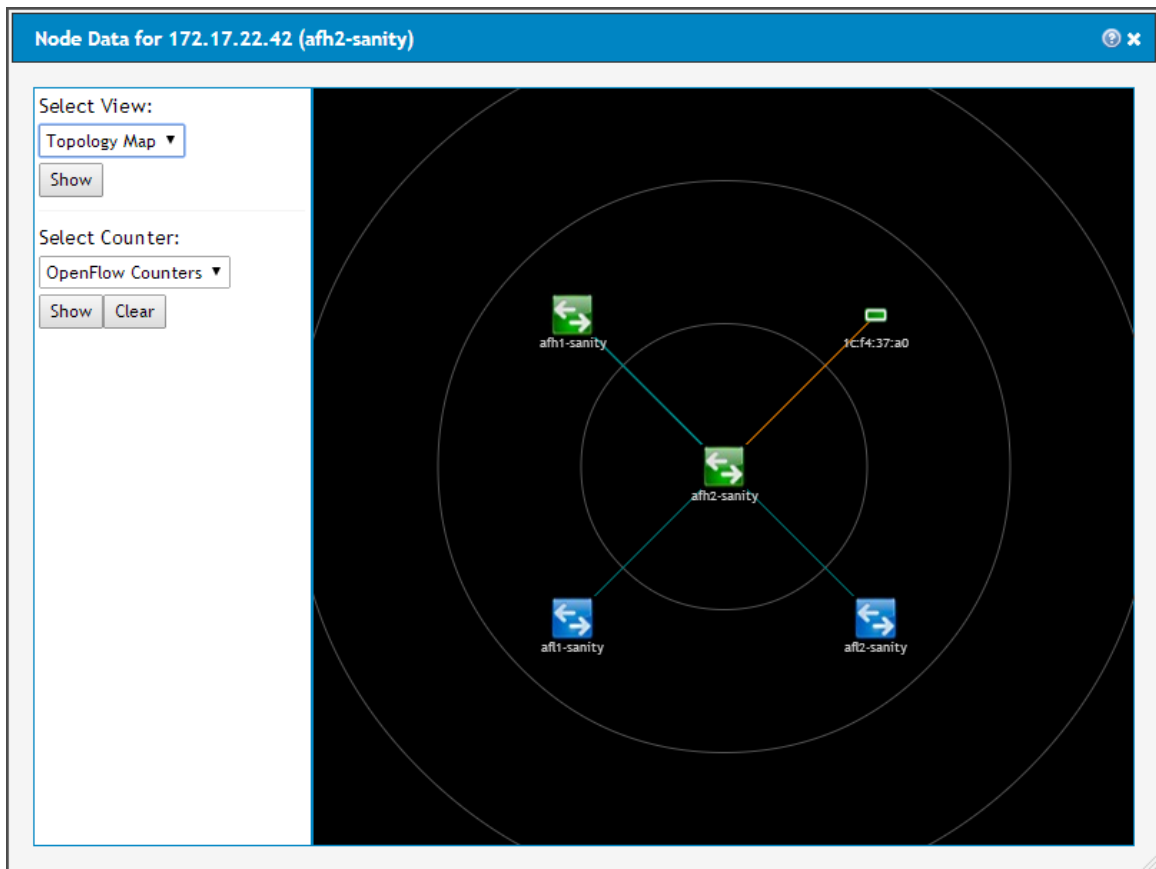


Figure 14. AFC GUI – Topology Node Data

To view more information about a node or link, select a viewing mode from the drop-down **Select View** list and click the **Go** button. The available viewing mode options vary depending on the type of node or link. The following table lists the viewing options by type and a description for each type.

Table 48. — Topology Navigation

Item of Interest	Available Viewing Options	Description
Switch	Topology Map	Displays the selected switch and all other devices (such as switches and endpoints) connected to the switch.
	Switch Details	Displays switch details such as type, hardware description, software description, and details for links connecting the switch to other devices.
Endpoint	Topology Map	Displays the selected endpoint and all switches or devices connected to the endpoint.
	Endpoint Details	Displays information about the endpoint and all links connecting the endpoint to other devices.
	Originating Flows	Displays details for all flows using the selected endpoint as a source.
	Terminating Flows	Displays details for all flows using the selected endpoint as a destination.
Switch-to-switch link	Link Details	Displays descriptions for devices at either end of the selected link.
Endpoint-to-switch link	Link Details	Displays descriptions for devices at either end of the selected link.

Counter Statistics

The software uses counters to produce statistics for a node or link. The list of available counters depends on the type of node or link you select. To view statistics based on counter data, select a counter from the **Select Counter** drop-down list and click the **Go** button. You can display the counter data in Chart or Tile viewing modes. To reset the selected counter, click the **Clear** button.

Table 49. — Counter Statistics

Item of Interest	Available Viewing Modes	Description
Switch	OpenFlow Counters	Displays OF statistics (such as the number of OF packets, active

Item of Interest	Available Viewing Modes	Description
		flows, and flow mods) for the selected switch in a Tile format.
	Switch Counters	Displays aggregate packet, byte, and error counts for the selected switch in a Tile format.
	Port Counters	Displays detailed packet and byte counts for each port on the selected switch as a Bar Chart categorized by counter type.
	Queue Counters	Displays detailed packet and byte counts for each priority queue for each port on the selected switch in a Bar Chart format.
Endpoint	Originating Flow Counters	Displays packet and byte statistics for all flows originating from the selected endpoint.
	Terminating Flow Counters	Displays packet and byte statistics for all flows terminating on the selected endpoint.

Flow Monitoring

To provide relevant statistics, the software can monitor specified flows. Flow monitoring is disabled by default. To select the category of flows to monitor, use the drop-down **Select Flow Monitoring** list. The options listed varies depending on the selected node or link. To begin flow monitoring for the selected flows, click the **Start** button. To disable flow monitoring for the selected flows, click the **Stop** button.

Table 50. — Flow Monitoring

Item of Interest	Available Viewing Modes	Description
Endpoint	Originating Flows	Enable or disable monitoring for all flows using the selected endpoint as the source.
	Terminating Flows	Enable or disable monitoring for all flows using the selected endpoint as the destination.

Configuring Resources

Using the Configure Menu

If you hover over the **Configure** link in the banner, a drop-down menu displays. Click to select an option and enter the required information in the form that displays. You can configure the following:

- Policies
- Policy Associations
- Fabric Edges

- Providers
- Tenants
- Networks
- Hosts
- Endpoints

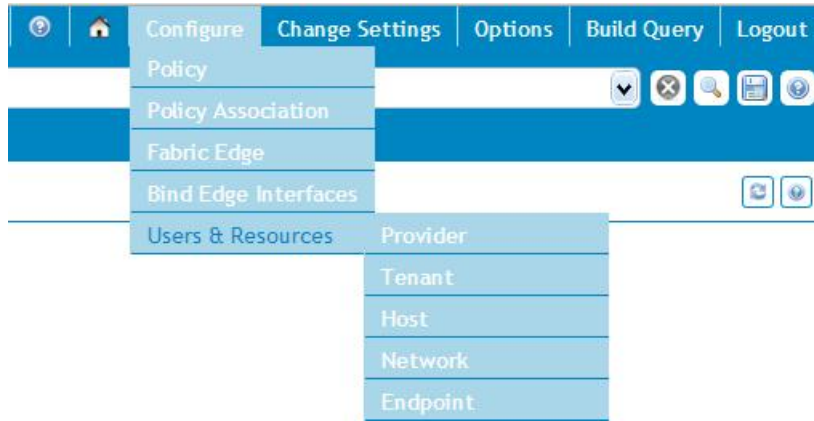



Figure 15. Configure Menu

 **NOTE:** ID names support only alphanumeric characters, underscores (_), hyphens (-), and periods (.).

Configuring a Policy

1. Click **Policy** in the **Configure** menu. The **Configure Policy** window displays.

Figure 16. Configure Policy Window

2. Select a provider from the drop-down **Provider ID:** list.
3. If the policy is tenant-specific, check the **Tenant Policy** checkbox. If you do not check this checkbox, the policy is provider-specific.
4. If applicable, select a tenant from the drop-down **Tenant ID:** list.
5. If you have already configured a policy, select it from the drop-down **Policy ID:** list and click **Add** to add the policy. If you have not configured a policy, continue to the next step.
6. Enter a unique ID number in the **Policy ID:** field.
7. Select the policy type from the drop-down **Policy Type** list.
8. Enter the match criteria for the policy in the appropriate fields in the **Select Match Criteria** section.
9. To define the policy's behavior, select an action from the drop-down **Action** list.
10. Select an option from the drop-down **Redirect To:** list and then select a destination from the drop-down **Target ID:** list.
11. When you have configured all policy attributes, click **Add**.

To delete a policy, select it from the drop-down **Policy ID:** list and click **Delete**.

Configuring a Policy Association

After you configure a policy, you can associate it with specific resources, such as a network or endpoint.

1. Click **Policy Association** in the **Configure** menu. The **Configure Policy Association** window displays.

Configure Policy Association

Select a Policy

Provider ID: *
1 : provider_1

Tenant Policy? Tenant ID: [Empty]

Policy ID: *
p1

Policy Details:

Policy Id	p1	
Policy Name	p1	
Policy Type	filter	
Policy Priority	0	
Policy Rule List	Rule Name	rule1
	Priority	0

Set Resource Allocation

Tenant Resource? Tenant ID: [Empty]

Resource Type: * [Empty] Resource ID: * [Empty]

Direction: [Empty] Priority: [Empty]

Add **Delete** **Close**

Figure 17. Configure Policy Association Window

2. Select a provider from the drop-down **Provider ID:** list.
3. If the policy is tenant-specific, check the **Tenant Policy** checkbox. If you do not check this checkbox, the policy is provider-specific.
4. If applicable, select a tenant from the drop-down **Tenant ID:** list.
5. Select a policy from the drop-down **Policy ID:** list.
6. Select the resource to associate with the policy:
 - (OPTIONAL) If the resource is tenant-specific, check the **Tenant Resource** checkbox.
 - (OPTIONAL) Select a tenant from the drop-down **Tenant ID** list.
 - Select the resource type from the drop-down **Resource Type** list.
 - Enter or select the resource from the drop-down **Resource ID** list.
 - Select the traffic direction from the drop-down **Direction** list.
 - (OPTIONAL) Enter the priority of the policy in the **Priority** field. The range is from 1 to 1023.
7. Click **Add** to confirm or click **Close** to close the window without making changes.

Configuring a Fabric Edge

You can configure fabric edges, including a middlebox, mirror, uplink, WAN port, or debug port with the interfaces used in the fabric.

1. Click **Fabric Edge** in the **Configure** menu. The **Fabric Edge** window displays.

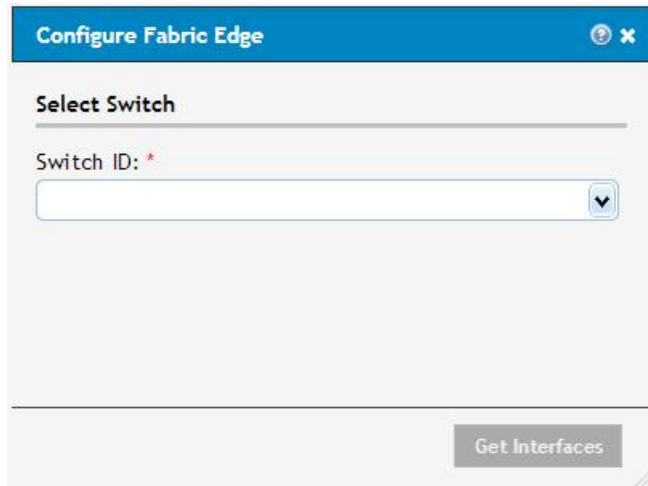


Figure 18. Configure Fabric Edge Window

2. Select a switch from the **Switch ID:** list and click **Get Interfaces** to display a list of ports available on the selected switch.
3. Select the radio button of the port you want to configure. The **Configure Fabric Edge Interface** window displays the selected port.

Configure Fabric Edge Interface

Port Number:
6

Switch Id:
afh1-sanity

Provider ID: *

Resource Type: *

Resource Id: *

Resource Name:

Resource Description:

SPAN RSPAN

RSPAN Vlan:


Add Delete Close

Figure 19. Configure Fabric Edge Interface Window

4. Select a provider from the drop-down **Provider ID** list.
5. Select the resource type (Middlebox, Mirror, Uplink, WAN port, or debug port) for the interface from the drop-down **Resource Type:** list.
6. Enter a unique ID number in the **Resource ID** field.
7. (OPTIONAL) Enter a name in the **Resource Name** field and a description in the **Resource Description** field for the interface.
8. If you enabled port mirroring, select the technology used for port mirroring (SPAN or RSPAN). If you select RSPAN, enter the corresponding VLAN ID.
9. Click **Add** to save the configuration or click **Close** to close the window without saving changes.

Configuring Binding Edge Interfaces

To create a single, logically-bound uplinks, you can bind uplinks within a provider.

 **NOTE:** You cannot bind an uplink more than once.

1. Click **Bind Edge Interfaces** in the **Configure** menu. The **Bind Edge Interfaces** window displays.

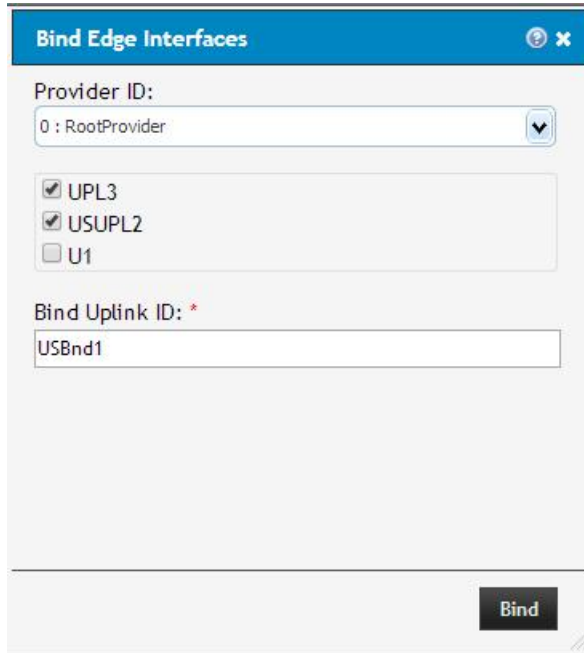


Figure 20. AFC GUI – Bind Edge Interfaces Window

2. Select a provider from the drop-down **Provider ID:** list.
3. Select two or more uplinks to bind from the list of uplinks that displays.
4. Enter a unique ID for the bound links in the **Bind Uplink ID:** field.
5. Click **Bind** to bind the selected uplinks.

Configuring Users and Resources

To create and configure users or resources including providers, tenants, networks, hosts, or endpoints, click **Users & Resources** in the **Configure** menu. The **Users & Resources** menu contains the following submenu items:

- Provider
- Tenant
- Network
- Host
- Endpoint

Configuring Providers

1. Click the **Configure** menu, click **Users & Resources**, then click **Provider**. The **Configure Provider** window displays.

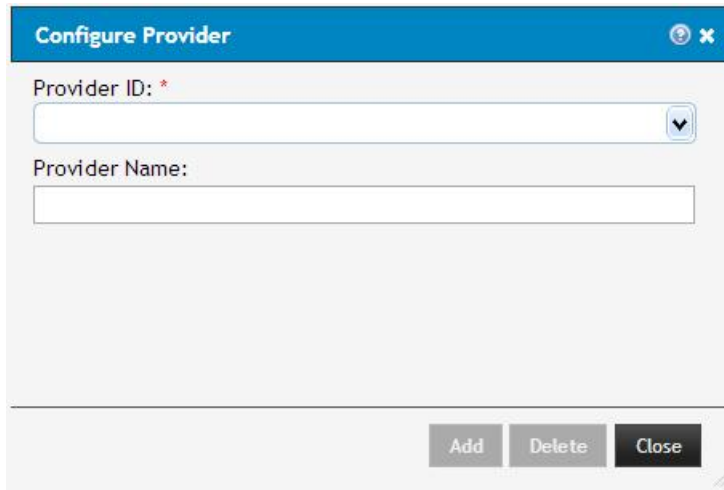



Figure 21. AFC GUI – Configure Provider Window

2. Enter a unique ID in the **Provider ID:** field.
3. (OPTIONAL) Enter a name to identify the provider in the **Provider Name:** field.
4. Click **Add** to save the configured provider or click **Close** to close the window without saving changes.

 **NOTE:** To delete a configured provider, select it from the drop-down **Provider ID:** list and click **Delete**.

Configuring Tenants

1. Click the **Configure** menu, click **Users & Resources**, then click **Tenant**. The **Configure Tenant** window displays.

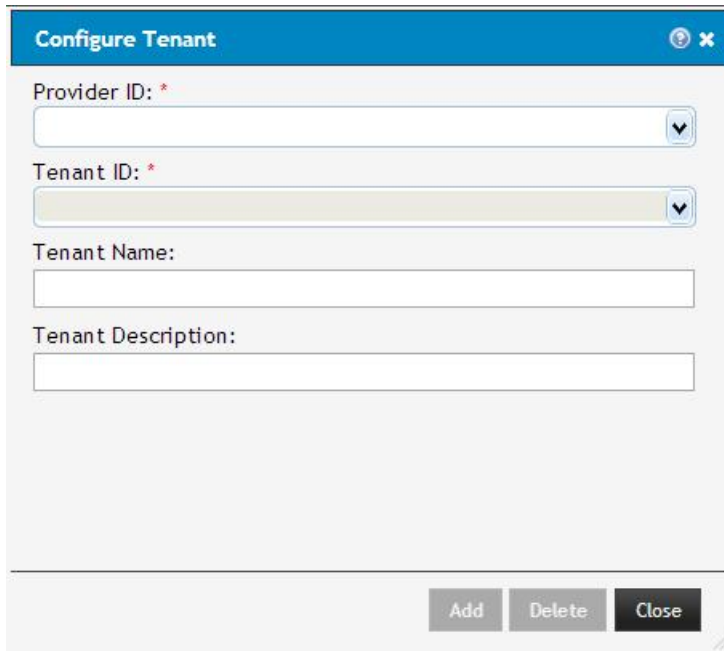



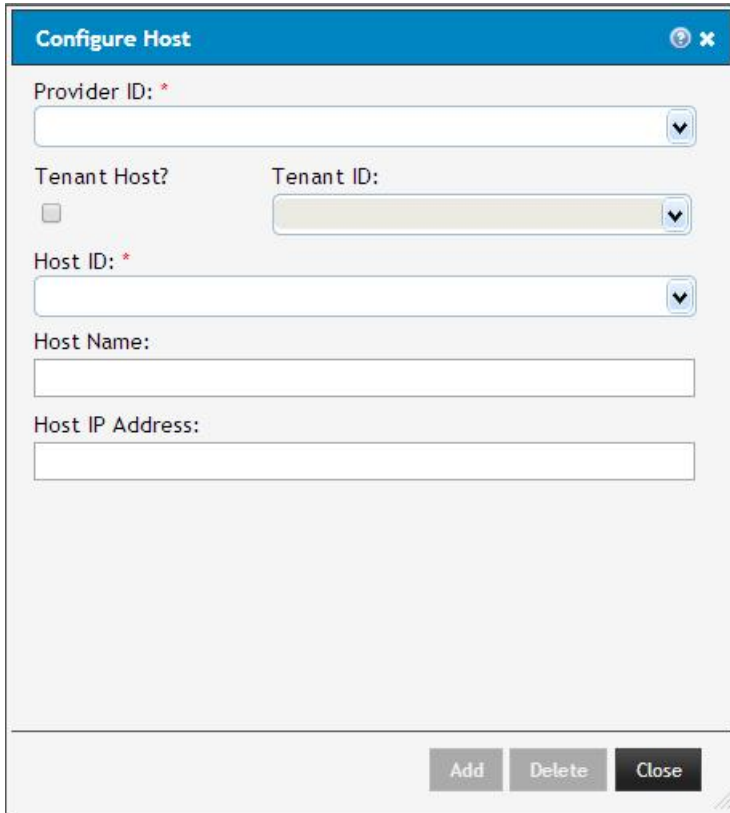
Figure 22. AFC GUI – Configure Tenant Window

2. Select the provider to associate with the tenant from the drop-down **Provider ID:** list.
3. Enter a unique ID in the **Tenant ID:** field.
4. (OPTIONAL) Enter a name to identify the tenant in the **Tenant Name** field and a description of the tenant in the **Tenant Description** field.
5. Click **Add** to save the configured tenant or click **Close** to close the window without saving changes.

 **NOTE:** To delete a provider, select the provider from the drop-down **Provider ID:** list and the tenant from the drop-down **Tenant ID:** list, then click **Delete**.

Configuring Hosts


1. Click the **Configure** menu, click **Users & Resources**, then click **Host**. The **Configure Host** window displays.



The screenshot shows a 'Configure Host' dialog box with the following fields and controls:

- Provider ID: ***: A drop-down menu.
- Tenant Host?**: A checkbox.
- Tenant ID:**: A drop-down menu.
- Host ID: ***: A drop-down menu.
- Host Name:**: A text input field.
- Host IP Address:**: A text input field.
- Buttons:** 'Add', 'Delete', and 'Close' at the bottom right.

Figure 23. AFC GUI — Configure Hosts Window

2. Select the provider from the drop-down **Provider ID:** list.
 3. To associate the host with a tenant, check the **Tenant Host** checkbox and select a tenant from the drop-down **Tenant ID:** list.
 4. Enter a unique ID for the host in the **Host ID:** field.
 5. (OPTIONAL) Enter a name to identify the host in the **Host Name** field and the IP address of the host in the **Host IP Address** field.
 6. Click **Add** to save the configured host or click **Close** to close the window without saving changes.
-  **NOTE:** To delete a host, select the provider and tenant (if applicable) from the drop-down lists. Select the host to delete from the drop-down **Host ID:** list and click **Delete**.

Configuring Networks

1. Click the **Configure** menu, click **Users & Resources**, then click **Network**. The **Configure Network** window displays.

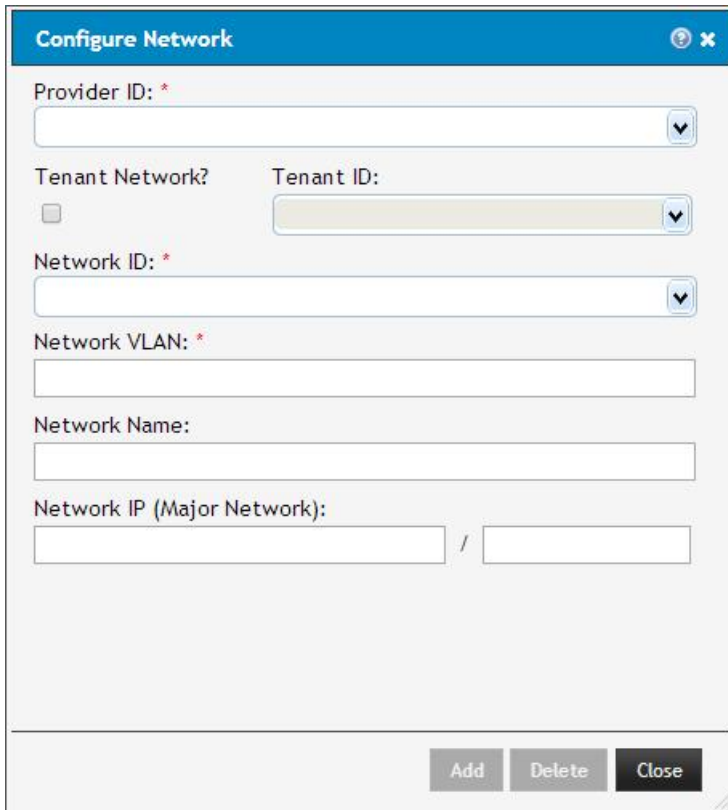


Figure 24. AFC GUI – Configure Network Window

2. Select the provider from the drop-down **Provider ID:** list.
3. To associate the network with a tenant, check the **Tenant Network** checkbox and select a tenant from the drop-down **Tenant ID:** list.
4. Enter a unique ID for the network in the **Network ID:** field.
5. Enter the VLAN ID for the network in the **Network VLAN** field. The range is from 0 to 4094.
6. (OPTIONAL) Enter a name to identify the network in the **Network Name** field and the IP address, with subnet, of the network in the **Network IP** field.
7. Click **Add** to save the configured network or click **Close** to close the window without saving changes.



NOTE: To delete a network, select the provider and tenant, if applicable, from the drop-down lists. Select the network to delete from the drop-down **Network ID:** list and click **Delete**.

Configuring Endpoints

1. Click the **Configure** menu, click **Users & Resources**, then click **Endpoint**. The **Configure Endpoint** window displays.

The screenshot shows the 'Configure Endpoint' dialog box. It contains the following fields and controls:


- Provider ID: ***: A dropdown menu.
- Tenant Endpoint?**: A checkbox.
- Tenant ID:**: A dropdown menu.
- Endpoint ID: ***: A dropdown menu.
- Endpoint Type: ***: A dropdown menu.
- Endpoint Name:**: A text input field.
- Endpoint MAC Address: ***: A text input field.
- Endpoint IP:**: A text input field.
- Host ID:** and **Network ID:**: Two dropdown menus.
- Buttons:** 'Add', 'Delete', and 'Close' at the bottom right.

2. Select the provider from the **Provider ID:** drop-down list.
3. To associate the endpoint with a tenant, check the **Tenant Endpoint** checkbox and select the tenant from the drop-down **Tenant ID:** field.
4. Enter a unique ID for the endpoint in the **Endpoint ID:** field.
5. Select the type of endpoint to configure from the drop-down **Endpoint Type** list. The available choices are host or virtual machine (VM).
6. (OPTIONAL) Enter a name to identify the endpoint in the **Endpoint Name** field and the MAC address for the endpoint in the **Endpoint MAC Address** field.
7. (OPTIONAL) Enter the IP address of the endpoint.
8. To link the endpoint with another host or network, select a host or network from the drop-down lists.



NOTE: The **Host ID** and **Network ID** lists are populated based on the selected provider and tenant.

- Click **Add** to save the configured endpoint or click **Close** to close the window without saving changes.

 **NOTE:** To delete an endpoint, select the provider and tenant, if applicable, from the drop-down lists. Select the endpoint to delete from the **Endpoint ID** list and click **Delete**.

Query Builder

To build a query, click the **Build Query** link in the banner. An API tree displays in the center pane.

Help

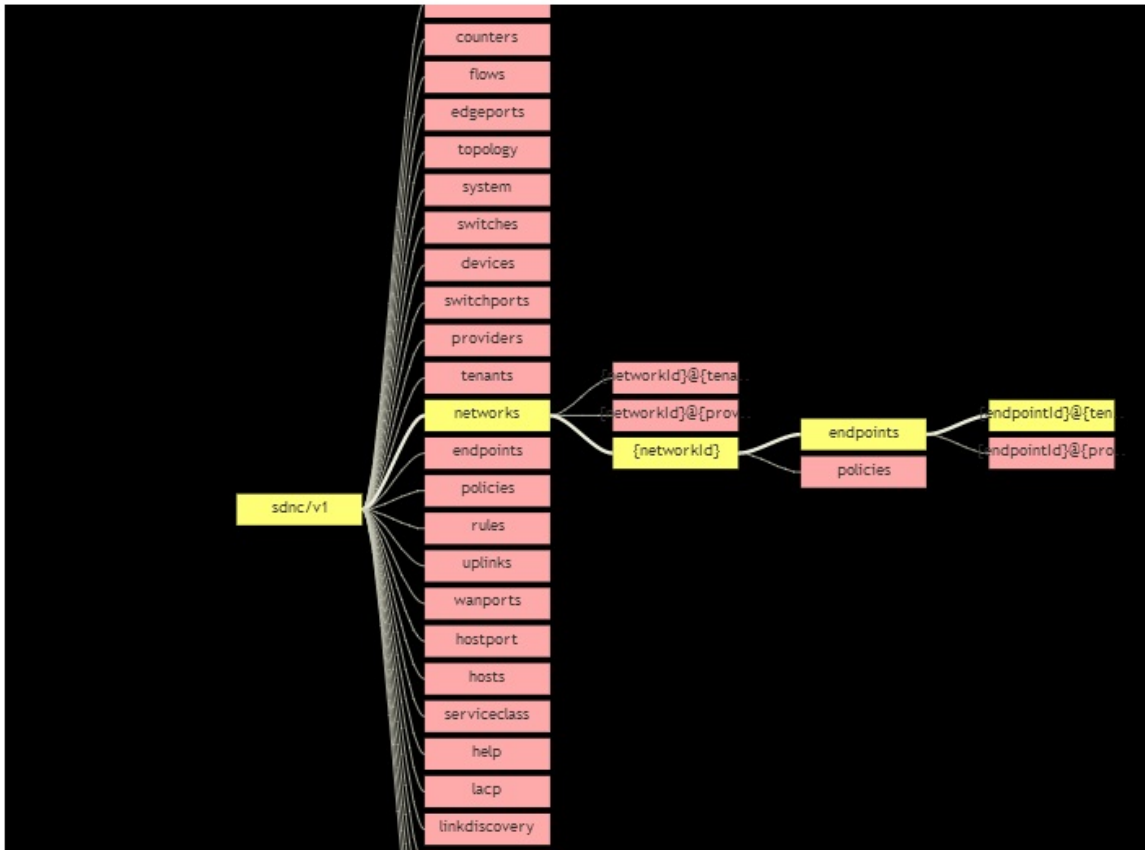


Figure 25. AFC GUI — Query Builder

To view the APIs as a list, click the *Switch to List View* icon. To restore the tree view, click the *Switch to Tree View* icon. To navigate the tree, click on any node. The node expands to reveal other related nodes. If you click the last node in a branch, the completed query displays in the text field in the banner. If more input is required to complete a selected API, a form displays where you can enter the requested data. When you click **OK**, the completed query displays in the text field in the banner. To submit the query, press the **Enter** key in the text field or click the *Run Query* icon.

Figure 26. AFC GUI — Build Query Window

Filtering Results

The query input form allows you to enter specific criteria for filtering results. Enter a Key-Value pair to retrieve only records containing matches for the specified key or attribute. The value can be a regular expression. To retrieve data results starting with a specific number, enter the number in the **Offset** entry field. To specify the number of returned results, enter a number in the **Limit** entry field. To retrieve results that include specific values, enter a comma-separated list of attribute names in the **Fields** entry field. All other results do not display. You can specify filters in the URI (for example, `sdnc/v1/switches/*?key=swId&value=.23.&offset=0&limit=10&fields=swId,hardwareDescription`). For more information about filtering results, refer to [REST Information Retrieval](#).

Summaries

The GUI displays information about the control software in the left pane. This information is categorized into two groups: System Summary and Controller Summary.

System Summary

The system summary includes the following information:

- **System IP** — If you install the software in a Primary/Backup configuration, the IP address of the primary server displays. If you install the software in a single controller configuration, the controller IP address displays.
- **Version** — Displays the software version, including build number.
- **User Name** — Displays the name of the current user.
- **Operational Mode** — The software can operate in one of two modes: Primary/Backup and Cluster.
 - In Primary/Backup mode, the software uses two nodes, with each node containing all required software components. The active node is responsible for all control functions. The standby node takes over if the active node is unavailable.
 - In Cluster mode, the software uses one or more nodes, with each node containing one or more components of the control software. The components form clusters and the responsibility for control functions is shared by all components in the cluster.

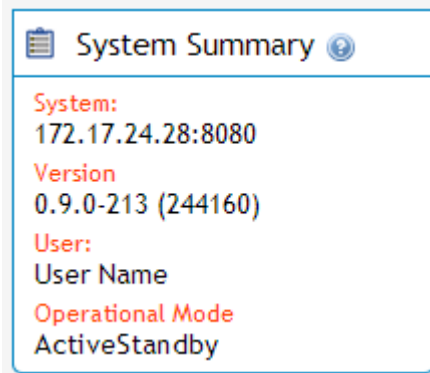


Figure 27. AFC GUI — System Summary

Controller Summary

A controller summary displays information about each configured controller. The controller summary includes the following information:

- **Node IP** — Displays the IP address of the node running the software and the controller’s role (active or standby). The colored dots indicate the role as well — green for active or standalone and yellow for standby. To view more information about the node’s operating environment and current operating conditions, click the Get Details icon adjacent to the node IP.
- **CPU %** — Displays current and peak values. The peak value represents the highest recorded CPU consumption since startup. The colored dots indicate conditions where CPU consumption exceeded pre-set thresholds. You cannot configure the thresholds.
 - *Red* — Indicates current consumption is greater than 90%
 - *Yellow* — Indicates current consumption is between 61 and 89%
 - *Green* — Indicates current consumption is less than 60%
- **Memory %** — Displays current and peak values. The peak value represents the highest recorded memory consumption since startup. The colored dots indicate conditions where memory consumption exceeded pre-set thresholds. You cannot configure the thresholds.
 - *Red* — Indicates current consumption is greater than 90%
 - *Yellow* — Indicates current consumption is between 61 and 89%
 - *Green* — Indicates current consumption is less than 60%
- **Uptime** — Displays the days, hours, minutes, and seconds the node has been operational since last startup.

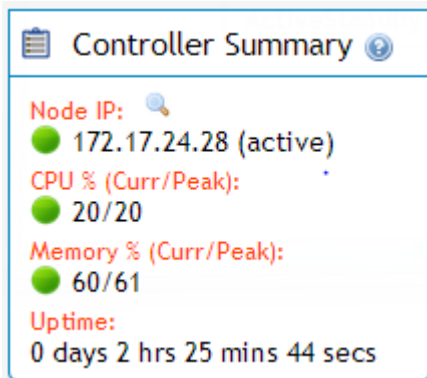


Figure 28. AFC GUI — Controller Summary