

# **NSX vSphere API Guide**

NSX 6.2 for vSphere

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition.

EN-001545-06



# Contents

About This Book	21
<b>1 Overview of NSX</b>	<b>23</b>
NSX Components	23
Data Plane	24
Control Plane	24
Management Plane	24
Consumption Platform	24
NSX Services	24
Logical Switches	25
Logical Routers	25
Logical Firewall	25
Logical Virtual Private Networks (VPNs)	25
Logical Load Balancer	25
Service Composer	25
NSX Extensibility	26
An Introduction to REST API for NSX Users	26
How REST Works	26
About the REST API	26
RESTful Workflow Patterns	27
For More Information About REST	27
Using the NSX REST API	27
Ports Required for NSX REST API	28
<b>2 Installing Components</b>	<b>29</b>
Installing Licenses	29
Evaluating NSX License Capacity Usage	29
Working with Network Virtualization Components	30
Install Network Virtualization Components	30
Delete Network Virtualization Components	31
Working with VXLAN for Logical Switches	31
Working with Controllers	32
Add Controller	32
Query Controllers	32
Query Controller Addition or Deletion Details	33
Query Controller Tech Support Logs	33
Delete Controller	33
Query Cluster Information	33
Modify Cluster Configuration	34
Add Controller Syslog Exporter	34
Query Controller Syslog Exporter	34
Delete Controller Syslog Exporter	35
Backup Controller Data	35
Working with Segment IDs	35
Add a new Segment ID Range	35
Query all Segment ID Ranges	35
Query a Specific Segment ID Range	36

Update a Segment ID Range	36
Delete a Segment ID Range	36
Configure VXLAN	37
Install VXLAN	37
Delete VXLAN	38
Delete VXLAN with vdsContext	38
Working with Network Scopes	38
Create a Network Scope	38
Edit a Network Scope	39
Update Attributes on a Network Scope	39
Query existing Network Scopes	39
Query a Specific Network Scope	40
Delete a Network Scope	41
Repair Logical Switches in a Network Scope	41
Reset Communication	41
Query Features on Cluster	42
Query Status of Specific Resources	42
Query Status of Child Resources	43
Query Status of Resources by Criterion	44
Installing NSX Edge	45
NSX Edge Services Gateway	45
Logical Router	46
Manage Tuning Configuration	46
Query Tuning Configuration	46
Modify Tuning Configuration	46
Installing NSX Edge Services Gateway	47
Installing a Logical Router	50
Working with Services	51
Install Security Fabric	52
Service Dependency	52
Deploying a Service with a Dependency	53
Identify Service Dependency	53
Uninstall Service Dependency	53
Query Installed Services	53
Query Details about a Service	54
Query Clusters	54
Query Agents on Host	55
Query Agent Information	55
Query Agents for Deployment	56
Working with Conflicting Agencies	57
Query Conflicts	57
Restore Conflicting Agencies	58
Delete Conflicting Agencies	58
Delete Deployment Units	58
Uninstalling Services	58

### 3 Managing the NSX Manager Appliance 61

Configuring NSX Manager with vCenter Server	61
Configure vCenter Server with NSX Manager	61
Query Configuration Details	62
Certificate Management	62
Generate CSR Certificate	62
Download CSR Certificate	63
Upload Certificate Chain	63

Query Certificates	63
Upload Keystore File	64
Resource Management	64
Query Global Appliance Manager Information	64
Query Summary Appliance Manager Information	64
Query Component Information	65
Reboot Appliance Manager	66
Query Appliance Manager CPU	66
Query Appliance Manager Uptime	66
Query Appliance Manager Memory	66
Query Appliance Manager Storage	67
Working with Network Settings	67
Query Network Information	67
Configure DNS Servers	67
Delete DNS Servers	68
Working with Time Settings	68
Configure Time Settings	68
Query Time Settings	68
Delete Time Settings	68
Working with Locale Settings	69
Configure Locale	69
Query Locale	69
Working with Syslog Servers	69
Configure Syslog Servers	69
Query Syslog Servers	70
Retrieves syslog servers.	70
Delete Syslog Servers	70
Deletes syslog servers.	70
Components Management	70
Query Components	70
Query Specific Component	71
Query Component Dependencies	71
Query Specific Component Dependents	72
Query Component Status	72
Toggle Specific Component Status	72
Restart Appliance Management Web Application	72
Working with Backup and Restore	73
Configure Backup Settings	73
Configure On-Demand Backup	74
Query Backup Settings	74
Delete Backup Configuration	75
Query Available Backups	75
Restore Data	75
Working with Tech Support Logs	75
Generate Tech Support Logs	75
Download Tech Support Logs	76
Querying NSX Manager Logs	76
Get NSX Manager System Events	76
Get NSX Manager Audit Logs	76
Working with Support Notifications	77
Query Notifications	77
Delete all Notifications	77
Acknowledge Notifications	77

- 4 Upgrading NSX Components 79**
  - Upgrading NSX Manager 79
    - Upload Upgrade Bundle 80
    - Query Upgrade Information 80
    - Begin Upgrade 80
    - Query Upgrade Status 81
  - Upgrading Controllers 81
    - Backup Controllers 81
    - Query Controller Upgrade Availability 82
    - Upgrade Controllers 82
    - Query Controller Upgrade Status 82
  - Upgrading Network Virtualization Components 82
  - Upgrading Distributed Firewall 83
  - Upgrading NSX Edge 83
  - Upgrading Services 84
  
- 5 User Management 85**
  - Configuring SSO on NSX Manager 85
    - Query SSO Details 86
    - Query SSO Configuration Status 86
    - Delete SSO Configuration 86
  - User Management 86
    - Get Information About a User 86
    - Enable or Disable a User Account 87
    - Change NSX Controller Password 87
    - Remove Role Assignment 88
  - Role Management 88
    - Get Role for a User 88
    - Get Role for a NSX Manager User 89
    - Add Role and Resources for a User 89
    - Change User Role 90
    - Get List of Possible Roles 90
    - Get List of Scoping Objects 91
    - Delete User Role 91
  
- 6 Grouping Objects 93**
  - Working with Security Groups 93
    - Create Security Group 93
    - Query Security Groups 95
    - Query Applicable Members for a Security Group 97
    - Query all Members of a Security Group 98
    - Query Security Group Objects 98
    - Query Security Groups that contain a Virtual Machine 99
    - Modify a Security Group 99
    - Delete a Security Group 99
    - Add Member to Security Group 99
    - Delete Member from Security Group 100
  - Working with IPsets 100
    - Create an IPset 100
    - Query IPsets 101
    - Query Details of an IPset 101
    - Modify an IPset 101
    - Delete an IPset 102
  - Working with MACsets 102
    - Create a MACset on a Scope 102

List MACsets Created on a Scope	102
Get Details of a MACset	103
Modify an Existing MACset	103
Delete a MACset	103
Working with Services	104
List Services on a Scope	104
Add Service	104
Get Details of a Service	105
Modify Service	105
Delete Service	106
Working with Service Groups	106
Add Service Group	106
Query Service Groups	106
Query Details of a Service Group	107
Modify Service Group Details	107
Delete Service Group from Scope	108
Working with the Members of a Service Group	108
Query Service Group Members	108
Add a Member to the Service Group	109
Delete a Member from the Service Group	109
Working with IP Pools	110
Add an IP Pool	110
Query IP Pool Details	110
Modify an IP Pool	111
Allocating a New IP Address	111
Allocating a Specific IP Address	112
Query all IP Pools on Scope	112
Query Allocated IP Addresses	113
Release an IP Address	113
Delete an IP Pool	114
Working with Tags	114
Create Security Tag	114
Apply Tag to Virtual Machine	114
Query Security Tags	114
Query Virtual Machines Assigned to Tag	115
Detach Tag from Virtual Machine	115
Delete Tag	115
<b>7 Working with Logical Switches</b>	<b>117</b>
Preparing for Logical Switches	118
Configuring Switches	118
Prepare Switch	118
Query Configured Switches	118
Query Configured Switches on Datacenter	119
Query Specific Switch	119
Delete Switch	120
Working with Segment IDs	120
Add a new Segment ID Range	120
Query all Segment ID Ranges	120
Query a Specific Segment ID Range	121
Update a Segment ID Range	121
Delete a Segment ID Range	121
Working with Multicast Address Ranges	121
Add a new Multicast Address Range	121
Query all Multicast Address Ranges	122

Get a Specific Multicast Address Range	122
Update a Multicast Address Range	122
Delete a Multicast Address Range	123
Working with Transport Zones	123
Create a Transport Zone	123
Edit a Transport Zone	123
Update Attributes on a Transport Zone	124
Query existing Transport Zones	124
Query a Specific Transport Zone	125
Delete a Transport Zone	126
Working with Logical Switches	126
Create a Logical Switch	126
Attach or Detach a Virtual Machine from a Logical Switch	126
Query all Logical Switches on a Transport Zone	127
Query all Logical Switches on all Transport Zones	127
Query a Specific Logical Switch	128
Modify Control Plane Mode	129
Delete a Logical Switch	129
Working with ARP Suppression and MAC Learning for Logical Switches	129
Managing the Logical Switch UDP Port	130
Get UDP Port	130
Update UDP Port	130
Querying Allocated Resources	130
Testing Multicast Group Connectivity	131
Test Multicast Group Connectivity in a Transport Zone	131
Test Multicast Group Connectivity in a Logical Switch	131
Performing Ping Test	132

## 8 NSX Edge Logical Router Management 133

Create a Logical Router	133
Query a Logical Router	135
Modify a Router	136
Deleting a Router	137
Working with Interfaces	138
Working with Management Interfaces	138
Configure Management Interfaces	138
Query Management Interfaces	138
Working with all Interfaces	139
Add Interfaces	139
Query Interfaces for a NSX Edge Router	139
Delete Interfaces	140
Delete all Interfaces	141
Manage an NSX Edge Router Interface	141
Retrieve Interface with Specific Index	141
Modify an Interface	141
Delete Interface Configuration	142
Configure Routes	142
Query Routes	145
Delete Routes	147
Manage Global Routing Configuration	147
Specify Global Configuration	148
Query Global Route	148
Manage Static Routing	148
Configure Static Routes	148

Query Static Routes	149
Delete Static Routes	150
Manage OSPF Routes for NSX Edge	150
Configure OSPF	150
Query OSPF	151
Delete OSPF	152
Manage BGP Routes for NSX Edge	152
Configure BGP	152
Query BGP	153
Delete BGP	154
Working with Bridging	155
Configure a Bridge	155
Query Bridge Configuration	155
Query BGP	155
Delete Bridge Configuration	156
<b>9 NSX Edge Services Gateway Management</b>	<b>157</b>
Query Installed Edges	158
Modifying NSX Edge Configuration	162
System Control Edge Configuration	166
Deleting NSX Edge	168
Configuring Edge Services in Async Mode	168
Query Async Job Status	168
Query all Jobs	168
Query active Jobs	169
Configuring Certificates	169
Working with Certificates	169
Create Certificate	169
Create Certificate or Certificate Chain for CSR	170
Query Certificates	170
Delete Certificate	170
Working with Certificate Signing Requests (CSRs)	170
Create CSR	171
Create Self Signed Certificate for CSR	171
Query CSRs	171
Working with Certificate Revocation List (CRL)	172
Create a CRL	172
Query CRL	172
Delete CRL	172
Working with NSX Edge Firewall	172
Configure Firewall	173
Query Firewall Configuration	174
Query Pre Rules	176
Append Firewall Rules	176
Add a Firewall Rule Above a Specific Rule	177
Query Specific Rule	178
Modify Firewall Rule	178
Delete a Firewall Rule	179
Delete Firewall Configuration	179
Manage Global Firewall Configuration	179
Query Global Firewall Configuration	179
Modify Global Configuration	180
Manage Default Firewall Policy	180
Query Default Firewall Policy	181

Modify Default Firewall Policy	181
Query Firewall Statistics	181
Query Firewall Statistics for Rule	182
Disable Firewall	182
Working with NAT	182
Configure NAT	182
Query NAT Rules for an Edge	183
Delete all NAT Rules	184
Add a NAT Rule above a Specific Rule	184
Append NAT Rules	185
Modify a NAT Rule	185
Delete a NAT Rule	185
Working with Routing	186
Configure Routes	186
Query Routes	190
Delete Routes	190
Manage Global Routing Configuration	190
Specify Global Configuration	190
Query Global Route	191
Manage Static Routing	191
Configure Static Routes	191
Query Static Routes	192
Delete Static Routes	192
Manage OSPF Routes for NSX Edge	193
Configure OSPF	193
Query OSPF	194
Delete OSPF	195
Manage ISIS Routes for NSX Edge	195
Configure ISIS	195
Query ISIS	196
Delete ISIS	197
Manage BGP Routes for NSX Edge	197
Configure BGP	198
Query BGP	199
Delete BGP	200
Working with Load Balancer	200
Configure Load Balancer	200
Query Load Balancer Configuration	207
Delete Load Balancer Configuration	207
Manage Application profiles	207
Append Application Profile	207
Modify Application Profile	208
Query Application Profile	208
Query all Application Profiles	208
Delete Application Profile	209
Delete all Application Profiles	209
Manage Application Rules	209
Append Application Rule	209
Modify Application Rule	210
Query Application Rule	210
Query all Application Rules	210
Delete Application Rule	210
Delete all Application Rules	211
Manage Load Balancer Monitors	211

Append Monitor	211
Modify Monitor	211
Query Monitor	212
Query all Monitors	212
Delete Monitor	213
Delete all Monitors	213
Manage Virtual Servers	213
Append Virtual Server	213
Query a Virtual Server	213
Query all Virtual Servers	214
Delete a Virtual Server	215
Delete all Virtual Server	215
Manage Backend Pools	215
Append Backend Pool	215
Modify a Backend Pool	216
Query Backend Pool Details	217
Query all Backend Pools	217
Delete a Backend Pool	219
Delete all Backend Pools	219
Query Statistics	219
Update Load Balancer Acceleration Mode	221
Update Load Balancer Member Condition	221
Configure DNS Servers	222
Configure DNS	222
Retrieve DNS Configuration	222
Delete DNS Configuration	223
Retrieve DNS Statistics	223
Working with DHCP Service	224
Configure DHCP	224
Query DHCP Configuration	226
Delete DHCP Configuration	226
Retrieve DHCP Lease Information	226
Append IP Pool to DHCP Configuration	227
Append Static Binding to DHCP Configuration	227
Delete DHCP Pool	228
Delete DHCP Static Binding	228
Working with DHCP Relay	228
Query DHCP Relay	229
Delete DHCP Relay Configuration	229
Working with High Availability (HA)	229
Retrieve High Availability Configuration	230
Delete High Availability Configuration	230
Force High Availability Failover	231
Working with Syslog	231
Configure Syslog	231
Query Syslog	231
Delete Syslog	231
Managing SSL VPN	232
Enable or Disable SSL VPN	232
Query SSL VPN Details	232
Manage Server Settings	232
Apply Server Settings	232
Query Server Settings	233
Configure Private Networks	233

Add Private Network	233
Modify Private Network	233
Query Specific Private Network	234
Delete Private Network	234
Delete all Private Networks	235
Apply All Private Networks	235
Configure Web Resource	235
Add Portal Web Resource	235
Modify Portal Web Resource	235
Query Portal Web Resource	236
Query all Web Resources	236
Delete Portal Web Resource	236
Deletes all Web Resources	237
Apply All Web Resources	237
Configure Users	237
Add User	237
Modify User	238
Query User Details	238
Delete User	238
Delete all Users	239
Apply all Users	239
Configure IP Pool	239
Add IP Pool	239
Modify IP Pool	240
Query IP Pool	240
Query all IP Pools	240
Delete IP Pool	241
Deletes all IP Pools	241
Apply all IP Pools	241
Configure Network Extension Client Parameters	242
Apply Client Configuration	242
Get Client Configuration	242
Configure Network Extension Client Installation Package	242
Add Client Installation Package	243
Modify Client Installation Package	243
Query Client Installation Package	244
Query all Client Installation Packages	244
Delete Client Installation Package	245
Delete all Client Installation Packages	245
Apply all Installation Packages	245
Configure Portal Layouts	246
Upload Portal Logo	246
Upload Phat Banner	246
Upload Client Connected Icon	246
Upload Client Disconnected Icon	247
Upload Client Desktop Icon	247
Upload Error Connected Icon	247
Apply Layout Configuration	247
Query Portal Layout	248
Configure Authentication Parameters	248
Upload RSA Config File	248
Apply Authentication Configuration	248
Query Authentication Configuration	250

Configure SSL VPN Advanced Configuration	250
Apply advanced configuration	250
Query Advanced Configuration	251
Working with Active Clients	251
Query Active Clients	251
Disconnect Active Client	252
Manage Logon and Logoff scripts	252
Upload Script	252
Configure Script Parameters	252
Modify Script Configuration	253
Query Script Configuration	253
Query All Script Configurations	253
Delete Script Configuration	254
Delete All Script Configuragtions	254
Apply All Script Configurations	254
Reconfigure SSL VPN	254
Query SSL VPN Configuration	258
Delete SSL VPN Configuration	261
Query SSL VPN Statistics	261
Enable or Disable SSLv3	262
Working with L2 VPN	262
Configure L2VPN	262
Query L2VPN	264
Query L2VPN Statistics	265
Enable L2VPN	265
Delete L2VPN	266
Working with IPSEC VPN	266
Retrieve IPSec Configuration	267
Retrieve IPSec Statistics	268
Query Tunnel Traffic Statistics	269
Delete IPSec Configuration	270
Managing an NSX Edge	270
Force Sync Edge	270
Redeploy Edge	270
Update DNS Settings	271
Modify AESNI Setting	271
Modify Edge Appliance Core Dump Setting	271
Modify Log Setting	271
Query Edge Summary	271
Query Edge Status	274
Query Edge Tech Support Logs	275
Manage CLI Credentials and Access	276
You can modify the CLI credentials and enable or disable SSH services for a Edge.	276
Modify CLI Credentials	276
Change CLI Remote Access	276
Manage Auto Configuration Settings	276
Modify Auto Configuration Settings	277
Query Auto Configuration Settings	277
Working with Appliances	277
Query Appliance Configuration	277
Modify Appliance Configuration	279
Change Appliance Size	279
Manage an Appliance	279
Working with Interfaces	281

Add Interfaces or Sub Interfaces	281
Retrieve Interfaces for a Edge	284
Retrieve Specified Interface	286
Modify Specified Interface	287
Delete Interfaces	289
Manage a Edge Interface	289
Retrieve Interface with Specific Index	289
Modify an Interface	290
Delete Interface Configuration	291
Query Interface Statistics	291
Query Statistics for all Interfaces	291
Query Statistics for Uplink Interfaces	292
Query Statistics for Internal Interfaces	292
Query Dashboard Statistics	293
<b>10 Firewall Management</b>	<b>295</b>
Configuring Firewall	297
Query Firewall Configuration	297
Filter Firewall Configuration	298
Modify Firewall Configuration	298
Delete Firewall Configuration	300
Working with Firewall Sections	301
Query Firewall Sections	301
Add Firewall Section	303
Modify Firewall Section	304
Delete Firewall Section	306
Working with Firewall Rules	306
Query Firewall Rule	307
Add Firewall Rule	307
Modify Firewall Rule	309
Delete Firewall Rule	310
Working with Layer3 Redirect Sections and Rules	310
Query Layer3 Redirect Rules (All)	310
Query Layer3 Redirect Section	310
Add Layer3 Redirect Section	311
Modify Layer3 Redirect Section	312
Delete Layer3 Redirect Section	314
Query Layer3 Redirect Rules	314
Add Layer3 Redirect Rule	315
Modify Layer3 Redirect Rule	316
Delete Layer3 Redirect Rule	317
Query Service Insertion Profiles	317
Query Status	322
Query Firewall Configuration Status	322
Query Layer3 Section Status	322
Query Layer2 Section Status	323
Working with Memory and CPU Thresholds	324
Configure Thresholds	324
Query Thresholds	325
Tuning Firewall Performance	325
Synchronizing and Enabling Firewall	326
Force Sync Host	326
Force Sync Cluster	326
Enable or Disable APIs for a Cluster	326
Importing and Exporting Firewall Configurations	327

Save a Configuration	327
Query all Saved Configurations	327
Query a Saved Configuration	328
Modify a Saved Configuration	329
Delete a Saved Configuration	330
Export a Saved Configuration	330
Import a Saved Configuration	330
Working with SpoofGuard	331
Create SpoofGuard Policy	331
Modify SpoofGuard Policy	332
Query SpoofGuard Policy	332
Query all SpoofGuard Policies	333
Delete SpoofGuard Policy	334
SpoofGuard Operations	334
Get IP details	334
Approve IP Addresses	334
Publish Approved IP Addresses	335
Publish Approved IP Addresses for a Specific vNIC	335
Getting Flow Statistic Details	335
Get Flow Statistics	336
Get Flow Meta-Data	338
Flow Exclusion	339
Exclude Flows	339
Query Excluded Flows	340
Working with IPFix	341
Configure IPFix	341
Query IPFix Configuration	341
Delete IPFix Configuration	341
Excluding Virtual Machines from Firewall Protection	342
Add a Virtual Machine to the Exclusion List	342
Get Virtual Machine Exclusion List	342
Delete a Virtual Machine from Exclusion List	343
<b>11 Distributed Firewall Examples</b>	<b>345</b>
Introduction	345
NSX DFW REST API Functionalities	345
Distributed Firewall (DFW) REST API Call Examples	350
Distributed Firewall Configuration	350
Distributed Firewall Exclusions	368
CPU/Memory/CPS Configuration	369
Security Groups	370
Grouping Objects using IPSets	374
Grouping Objects using MACSets	376
<b>12 Service Composer Management</b>	<b>379</b>
Working with Security Policies	380
Creating a Security Policy	380
Description of Tags	382
Querying Security Policies	383
Edit a Security Policy	386
Delete a Security Policy	386
Export a Security Policy Configuration	387
Import a Security Policy Configuration	387
Query Security Actions for a Security Policy	388
Default Applied To Value for Firewall Rules	388

Query Default Applied To Value for Firewall Rules	388
Change Default Applied To Value for Firewall Rules	388
Working with Security Actions	389
Query Virtual Machines for a Security Action	389
Query Security Actions Applicable on a Security Group	389
Query Security Action Applicable on A Virtual Machine	393
Synchronizing Service Composer Rules with Distributed Firewall	394
Query Firewall Out-of-Sync Time Stamp	394
Synchronize Service Composer Firewall	394
Configuring Auto Save Draft for Service Composer	395
Query the Auto Save Draft Setting in Service Composer	395
Change the Auto Save Draft Setting in Service Composer	395
Query Security Policies Mapped to a Security Group	395
Query Service Provider Data	396
Query Security Group Effective Membership	396
Query Security Groups to which a VM Belongs	396
Status of Service Composer	397
System Alarms on Service Composer	397

### 13 Data Security Configuration 399

Data Security User Roles	399
Defining a Data Security Policy	400
Query Regulations	400
Enable a Regulation	400
Query Classification Value	401
Configure a Customized Regex as a Classification Value	401
View the List of Excludable Areas	401
Exclude Areas from Policy Inspection	402
Specify Security Groups to be Scanned	403
Query Security Groups Being Scanned	403
Configure File Filters	404
Saving and Publishing Policies	405
Query Saved Policy	405
Query Published Policy	406
Publish the Updated Policy	406
Data Security Scanning	407
Start, Pause, Resume, or Stop a Scan Operation	407
Query Status for a Scan Operation	407
Querying Scan Results	408
Get List of Virtual Machines Being Scanned	408
Get Number of Virtual Machines Being Scanned	408
Get Summary Information about the Last Five Scans	409
Get Information for Virtual Machines Scanned During Previous Scan	409
Retrieve Information About Previous Scan Results	409
Get XML Representation of Policy Used for Previous Scan	409
Querying Violation Details	411
Get List of Violation Counts	411
Get List of Violating Files	412
Get List of Violating Files in CSV Format	414
Get Violations in Entire Inventory	414
414	

### 14 Activity Monitoring 415

Data Collection	415
Enable Data Collection on a Single Virtual Machine	416

Disable Data Collection on a Single Virtual Machine	416
Override Data Collection	416
Turn On Kill Switch	416
Turn Off Kill Switch	417
Query Per Virtual Machine Data Collection	417
Query Resources	418
Prerequisites	418
View Outbound Activity	418
Parameter Values	418
View Inbound Activity	419
Parameter Values	419
View Interaction between Inventory Containers	420
Parameter Values	420
View Outbound AD Group Activity	420
Parameter Values	420
Query User Details	421
View Outbound Activity	421
Parameter Values	421
View Inbound Activity	422
Parameter Values	422
View Interaction between Inventory Containers	422
Parameter Values	422
View Outbound AD Group Activity	423
Parameter Values	423
View Virtual Machine Activity Report	423
Parameter Values	423
Query Discovered User Details	425
Working with Domains	426
Register a Domain with NSX Manager	426
Parameter Values for Register/Update Domain	426
Query Domains	427
Delete Domain	427
Working with LDAP Servers	428
Working with EventLog Servers	428
Working with Mapping Lists	429
Working with Activity Monitoring Syslog Support	429
<b>15 NSX Operations and Troubleshooting</b>	<b>431</b>
Communication Channel Health	431
Checking the Connection Status of a Single Host	431
Checking the Connection Status of a List of Hosts	431
Central CLI Methods	432
General Central CLI use in the API	432
Sample Central CLI command in the API	432
Traceflow	432
Creating Traceflows	433
Querying Traceflows	435
<b>16 Managing Hardware Gateways</b>	<b>439</b>
About the Hardware Gateway APIs	439
Managing Hardware Gateways	439
Install a Hardware Gateway	439
List all Hardware Gateways	440
Get a Hardware Gateway	441

- Update a Hardware Gateway 442
- Delete a Hardware Gateway Instance 442
- Managing Replication Clusters 442
  - Add or Delete Hosts on a Replication Cluster 442
  - Get a Replication Cluster 442
- Getting Hardware Gateway Inventory Information 444
  - Get Hardware Gateway Switches 444
  - Get Hardware Gateway Port Names for a Switch 444
- Managing Hardware Gateway Bindings 445
  - Get Hardware Gateway Bindings per Virtual Wire 445
  - Create a Hardware Gateway Binding 446
  - Get a List of Hardware Gateway Bindings 446
  - Get a Hardware Gateway Binding Object 446
  - Update a Hardware Gateway Binding Object 447
  - Delete a Hardware Gateway Binding 447
  - Manage Hardware Gateway Binding Objects 447
  - Get Statistic Information per Hardware Gateway Binding 448
- Connecting/Disconnecting a Hardware Gateway with a Virtual Wire 448
  - Attach a Hardware Gateway to a Virtual Wire 449
    - Option #1 449
    - Option #2 449
  - ?Detach a Hardware Gateway from a Virtual Wire 449
- Managing Bidirectional Forwarding Detection (BFD) 450
  - Set Global BFD Parameter Values 450
  - Get Global BFD Parameter Values 450
  - Get the Tunnel BFD Status 450
- 17 Managing NSX in a Cross-vCenter Environment 453**
  - Cross-vCenter Distributed Routing 453
    - Universal Distributed Logical Router 453
    - Cluster Level Locale ID 453
    - Host Level Locale ID 454
    - NSX Manager Roles 454
    - Universal Transport Zones 457
    - Universal Logical Switches 460
      - Universal Segment ID Pool (VNI Pool) 461
      - Universal Multicast Address Range 463
  - Distributed Firewall for Cross-vCenter NSX Environments 464
    - Universal Grouping Object Universal IP Sets (IP Address Groups) 474
      - 474
    - Universal MAC Sets 476
      - 476
    - Universal Services (Applications) 477
      - 477
    - Universal Service Groups (Application Groups) 478
      - If you are creating, modifying, or deleting universal service groups, you must run the API request on the primary NSX Manager. Universal service groups are read-only from secondary NSX Managers. 478
    - Universal Security Group 480
- 18 Task Framework Management 483**
  - About Task Framework 483
  - Query Job Instances for Job ID 484
  - Query Latest Job Instances for Job ID 485
  - Block REST Thread 485

Query Job Instances by Criterion	485
<b>19 vShield Endpoint Management</b>	<b>487</b>
Overview of Solution Registration	487
Registering a Solution with vShield Endpoint Service	488
Register a Vendor	488
Register a Solution	488
Altitude of a Solution	488
IP Address and Port for a Solution	488
Activate a Solution	489
Querying Registration Status of vShield Endpoint	489
Get Vendor Registration	489
Get Solution Registration	490
Get IP Address of a Solution	490
Get Activation Status of a Solution	490
Querying Activated Security Virtual Machines for a Solution	490
Query Activated Security Virtual Machines	490
Query Activation Information	491
Unregistering a Solution with vShield Endpoint	491
Unregister a Vendor	491
Unregister a Solution	491
Unset IP Address	492
Deactivate a Solution	492
Status Codes and Error Schema	492
Return Status Codes	492
Error Schema	493
<b>20 vCenter Object IDs</b>	<b>495</b>
Query Datacenter MOID	495
Query Datacenter ID	495
Query Host ID	495
Query Portgroup ID	496
Query VMID	496
<b>21 Deprecated APIs</b>	<b>497</b>
<b>Appendix A: Schemas</b>	<b>499</b>
Firewall Schemas	499
Firewall Configuration Schema	499
Firewall Section Schema	500
Firewall Sections Schema	501
Deprecated: vShield Manager Global Configuration Schema	501
Deprecated: ESX Host Preparation and Uninstallation Schema	506
Deprecated: vShield App Schemas	507
vShield App Configuration Schema	507
vShield App Firewall Schema	507
vShield App SpoofGuard Schema	510
vShield App Namespace Schema	512
Error Message Schema	513



# About This Book

---

This manual, the *NSX for vSphere API Guide*, describes how to install, configure, monitor, and maintain the VMware® NSX system by using REST API requests.

## Intended Audience

This manual is intended for anyone who wants to use REST API to programmatically control NSX in a VMware vSphere environment. The information in this manual is written for experienced developers who are familiar with virtual machine technology, virtualized datacenter operations, and REST APIs. This manual also assumes familiarity with vShield.

## VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation go to <http://www.vmware.com/support/pubs>.

## Document Feedback

VMware welcomes your suggestions for improving our documentation. If you have comments, send your feedback to [docfeedback@vmware.com](mailto:docfeedback@vmware.com).

## NSX Documentation

The following documents comprise the vShield documentation set:

- *NSX for vSphere Administration Guide*
- *NSX for vSphere Installation and Upgrade*
- *NSX API Programming Guide*, this guide

## Technical Support and Education Resources

The following sections describe the technical support resources available to you. To access the current version of this book and other books, go to <http://www.vmware.com/support/pubs>.

### Online and Telephone Support

To use online support to submit technical support requests, view your product and contract information, and register your products, go to <http://www.vmware.com/support>.

Customers with appropriate support contracts should use telephone support for the fastest response on priority 1 issues. Go to [http://www.vmware.com/support/phone\\_support](http://www.vmware.com/support/phone_support).

## Support Offerings

To find out how VMware support offerings can help meet your business needs, go to <http://www.vmware.com/support/services>.

## VMware Professional Services

VMware Education Services courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. Courses are available onsite, in the classroom, and live online. For onsite pilot programs and implementation best practices, VMware Consulting Services provides offerings to help you assess, plan, build, and manage your virtual environment. To access information about education classes, certification programs, and consulting services, go to <http://www.vmware.com/services>.

# Overview of NSX

---

IT organizations have gained significant benefits as a direct result of server virtualization. Server consolidation reduced physical complexity, increased operational efficiency and the ability to dynamically re-purpose underlying resources to quickly and optimally meet the needs of increasingly dynamic business applications.

VMware's Software Defined Data Center (SDDC) architecture is now extending virtualization technologies across the entire physical data center infrastructure. VMware NSX®, the network virtualization platform, is a key product in the SDDC architecture. With NSX, virtualization delivers for networking what it has already delivered for compute and storage. In much the same way that server virtualization programmatically creates, snapshots, deletes and restores software-based virtual machines (VMs), NSX network virtualization programmatically creates, snapshots, deletes, and restores software-based virtual networks. The result is a completely transformative approach to networking that not only enables data center managers to achieve orders of magnitude better agility and economics, but also allows for a vastly simplified operational model for the underlying physical network. With the ability to be deployed on any IP network, including both existing traditional networking models and next-generation fabric architectures from any vendor, NSX is a completely non-disruptive solution. In fact, with NSX, the physical network infrastructure you already have is all you need to deploy a software-defined data center.

With server virtualization, a software abstraction layer (server hypervisor) reproduces the familiar attributes of an x86 physical server (for example, CPU, RAM, Disk, NIC) in software, allowing them to be programmatically assembled in any arbitrary combination to produce a unique VM in a matter of seconds. With network virtualization, the functional equivalent of a network hypervisor reproduces the complete set of Layer 2 through Layer 7 networking services (for example, switching, routing, access control, firewalling, QoS, and load balancing) in software. As a result, these services can be programmatically assembled in any arbitrary combination, to produce unique, isolated virtual networks in a matter of seconds.

With network virtualization, benefits similar to server virtualization are derived. For example, just as VMs are independent of the underlying x86 platform and allow IT to treat physical hosts as a pool of compute capacity, virtual networks are independent of the underlying IP network hardware and allow IT to treat the physical network as a pool of transport capacity that can be consumed and re purposed on demand. Unlike legacy architectures, virtual networks can be provisioned, changed, stored, deleted, and restored programmatically without reconfiguring the underlying physical hardware or topology. By matching the capabilities and benefits derived from familiar server and storage virtualization solutions, this transformative approach to networking unleashes the full potential of the software-defined data center.

The chapter includes the following topics:

- [“NSX Components”](#) on page 23
- [“NSX Services”](#) on page 24
- [“An Introduction to REST API for NSX Users”](#) on page 26

## NSX Components

This section describes the components of the NSX solution.

## Data Plane

The NSX Data plane consists of the NSX vSwitch, which is based on the vSphere Distributed Switch (vDS) with additional components to enable services. Kernel modules (VIBs) run within the hypervisor kernel to provide services such as distributed routing and logical firewall and to enable VXLAN bridging capabilities.

The NSX vSwitch (vDS-based) abstracts the physical network and provides access-level switching in the hypervisor. It is central to network virtualization because it enables logical networks that are independent of physical constructs such as VLANs. Some of the benefits of the vSwitch are:

- Support for overlay networking with protocols (such as VXLAN) and centralized network configuration. Overlay networking enables the following capabilities:
  - Creation of a flexible logical Layer 2 (L2) overlay over existing IP networks on existing physical infrastructure without the need to re-architect any of the data center networks
  - Provision of communication (east.west and north.south), while maintaining isolation between tenants
  - Application workloads and virtual machines that are agnostic of the overlay network and operate as if they were connected to a physical L2 network
- Facilitates massive scale of hypervisors
- Multiple features such as Port Mirroring, NetFlow/IPFIX, Configuration Backup and Restore, Network Health Check, QoS, and LACP provide a comprehensive toolkit for traffic management, monitoring, and troubleshooting within a virtual network

Additionally, the data plane consists of gateway devices that can provide L2 bridging from the logical networking space (VXLAN) to the physical network (VLAN). The gateway device is typically an NSX Edge virtual appliance. NSX Edge offers L2, L3, perimeter firewall, load balancing, and other services such as SSL VPN and DHCP.

## Control Plane

The NSX control plane runs in the NSX controller. NSX controller is an advanced distributed state management system that controls virtual networks and overlay transport tunnels. It is the central control point for all logical switches within a network and maintains information about all virtual machines, hosts, logical switches, and VXLANs.

The controller does not have any dataplane traffic passing through it. Controller nodes are deployed in a cluster of odd-numbered members to enable high-availability and scale. Any failure of the controller nodes does not impact any data-plane traffic.

## Management Plane

The NSX management plane is built by the NSX Manager, the centralized network management component of NSX. It provides the single point of configuration and REST API entry-points.

The NSX Manager is installed as a virtual appliance on any ESX host in your vCenter Server environment.

## Consumption Platform

The consumption of NSX can be driven directly through the NSX Manager user interface. In a vSphere environment, this is available in the vSphere Web Client. Typically end users tie network virtualization to their cloud management platform for deploying applications. NSX provides rich integration into virtually any CMP through REST APIs. Out-of-the-box integration is also available through VMware vCloudAutomation Center, vCloud Director, and OpenStack with the Neutron plug-in for NSX.

## NSX Services

The NSX components work together to provide the following functional services.

## Logical Switches

A cloud deployment or a virtual data center has a variety of applications across multiple tenants. These applications and tenants require isolation from each other for security, fault isolation, and non-overlapping IP addresses. The NSX logical switch creates logical broadcast domains or segments to which an application or tenant virtual machine can be logically wired. This allows for flexibility and speed of deployment while still providing all the characteristics of a physical network's broadcast domains (VLANs) without physical layer 2 sprawl or spanning tree issues.

A logical switch is distributed and can span arbitrarily large compute clusters. This allows for virtual machine mobility (vMotion) within the data center without limitations of the physical Layer 2 (VLAN) boundary. The physical infrastructure is not constrained by MAC/FIB table limits, because the logical switch contains the broadcast domain in software.

## Logical Routers

Dynamic routing provides the necessary forwarding information between Layer 2 broadcast domains, thereby allowing you to decrease the size of Layer 2 broadcast domains and improve network efficiency and scale. NSX extends this intelligence to where the workloads reside for East-West routing. This allows more direct VM-to-VM communication without the costly or timely need to extend hops. At the same time, NSX logical routers provide North-South connectivity, thereby enabling tenants to access public networks.

## Logical Firewall

Logical Firewall provides security mechanisms for dynamic virtual data centers. The Distributed Firewall component of Logical Firewall allows you to segment virtual datacenter entities like virtual machines based on VM names and attributes, user identity, vCenter objects like datacenters, and hosts, as well as traditional networking attributes like IP addresses and VLANs. The Edge Firewall component helps you meet key perimeter security requirements, such as building DMZs based on IP/VLAN constructs, tenant-to-tenant isolation in multi-tenant virtual data centers, Network Address Translation (NAT), partner (extranet) VPNs, and user-based SSL VPNs.

The Flow Monitoring feature displays network activity between virtual machines at the application protocol level. You can use this information to audit network traffic, define and refine firewall policies, and identify threats to your network.

## Logical Virtual Private Networks (VPN)s

SSL VPN-Plus allows remote users to access private corporate applications. IPSec VPN offers site-to-site connectivity between an NSX Edge instance and remote sites. L2 VPN allows you to extend your datacenter by allowing virtual machines to retain network connectivity across geographical boundaries.

## Logical Load Balancer

The NSX Edge load balancer enables network traffic to follow multiple paths to a specific destination. It distributes incoming service requests evenly among multiple servers in such a way that the load distribution is transparent to users. Load balancing thus helps in achieving optimal resource utilization, maximizing throughput, minimizing response time, and avoiding overload. NSX Edge provides load balancing up to Layer 7.

## Service Composer

Service Composer helps you provision and assign network and security services to applications in a virtual infrastructure. You map these services to a security group, and the services are applied to the virtual machines in the security group.

Data Security provides visibility into sensitive data stored within your organization's virtualized and cloud environments. Based on the violations reported by NSX Data Security, you can ensure that sensitive data is adequately protected and assess compliance with regulations around the world.

## NSX Extensibility

VMware partners can integrate their solutions with the NSX platform, thus enabling customers to have an integrated experience across VMware products and partner solutions. Data center operators can provision complex, multi-tier virtual networks in seconds, independent of the underlying network topology or components.

## An Introduction to REST API for NSX Users

REST, an acronym for Representational State Transfer, is a term that has been widely employed to describe an architectural style characteristic of programs that rely on the inherent properties of hypermedia to create and modify the state of an object that is accessible at a URL.

### How REST Works

Once a URL of such an object is known to a client, the client can use an HTTP GET request to discover the properties of the object. These properties are typically communicated in a structured document with an HTTP Content-Type of XML that provides a representation of the state of the object. In a RESTful workflow, documents (representations of object state) are passed back and forth (transferred) between a client and a service with the explicit assumption that neither party need know anything about an entity other than what is presented in a single request or response. The URLs at which these documents are available are often “sticky,” in that they persist beyond the lifetime of the request or response that includes them. The other content of the documents is nominally valid until the expiration date noted in the HTTP Expires header.

---

**IMPORTANT** The REST APIs must be invoked by a user that is assigned the appropriate role for that API (Enterprise Administrator, vShield Administrator, Security Administrator, Auditor). Some examples of the functions each role can use:

The Enterprise Administrator role can configure and manage logical switches and perform all operations related to data security configuration.

The vShield administrator role can perform NSX data security operations such as installing virtual appliances and configuring port groups.

The Security Administrator role can create VXLAN networks, and can perform data security functions like creating and publishing policies and viewing violation reports (but not starting/stopping security scans).

The Auditor role can view configured policies and violations reports (read-only) related to data security.

---

### About the REST API

REST APIs use HTTP requests (often sent by script or high-level language) as a way of making idempotent remote procedure calls that create, modify, or delete objects defined by the API. A REST API is defined by a collection of XML documents that represent the objects on which the API operates. The HTTP operations themselves are generic to all HTTP clients. To write a RESTful client, you should understand HTTP protocol and the semantics of standard HTML markup. For NSX REST API, you must know three things:

- The set of objects that the API supports, and what they represent. For example, what are vDC and Org?
- How the API represents these objects. For instance, what is the XML schema for the NSX Edge firewall rule set? What do the individual elements and attributes represent?
- How the client refers to an object on which it wants to operate. For example, what is a managed object ID?

To answer these questions, you look at NSX API resource schemas. These schemas define a number of XML types, many of which are extended by other types. The XML elements defined in these schemas, along with their attributes and composition rules (minimum and maximum number of elements or attributes, or the prescribed hierarchy with which elements can be nested) represent the data structures of NSX objects. A client can “read” an object by making an HTTP GET request to the object’s resource URL. A client can “write” (create or modify) an object with an HTTP PUT or POST request that includes a new or changed XML body document for the object. Usually a client can delete an object with an HTTP DELETE request.

This document presents example requests and responses, and provides reference information on the XML schemas that define the request and response bodies.

## RESTful Workflow Patterns

All RESTful workflows fall into a pattern that includes only two fundamental operations, which you repeat in this order for as long as necessary.

- Make an HTTP request (GET, PUT, POST, or DELETE). The target of this request is either a well-known URL (such as NSX Manager) or a link obtained from the response to a previous request. For example, a GET request to an Org URL returns links to vDC objects contained by the Org.
- Examine the response, which can be an XML document or an HTTP response code. If the response is an XML document, it may contain links or other information about the state of an object. If the response is an HTTP response code, it indicates whether the request succeeded or failed, and may be accompanied by a URL that points to a location from which additional information can be retrieved.

## For More Information About REST

For a comprehensive discussion of REST from both client and server perspectives, see *RESTful Web Services* by Leonard Richardson and Sam Ruby, published 2007 by O'Reilly Media.

There are also many sources of information about REST on the Web, including:

- <http://www.infoq.com/articles/rest-introduction>
- <http://www.infoq.com/articles/subbu-allamaraju-rest>
- <http://www.stucharlton.com/blog/archives/000141.html>

## Using the NSX REST API

You have several choices for programming the NSX REST API: using Firefox, Chrome, or cURL. To make XML responses more legible, you can copy and paste them into an XML friendly editor such as `xmlcopyeditor` or `pspad`.

For PUT calls, you need to define the Content-type header (`content-type: application/xml`).

### To use the REST API in Firefox

- 1 Login to the vSphere Web Client.
- 2 If not already installed, locate the RESTClient Mozilla add-on, and add it to Firefox.
- 3 Click the REST-Client icon in the toolbar to open the REST client in a new tab.
- 4 Click **Authentication** and then **Basic Authentication**. This will result in the credentials being automatically added to the Request Header.
- 5 Select a method such as GET, POST, or PUT, and type the URL of a REST API. Click **Send**.  
Note: You might have been asked to accept or ignore the lack of SSL certificate upon logging in to the vSphere web-client.

Response Header, Response Body, and Rendered HTML appear in the bottom window.

### To use the REST API in Chrome

- 1 Search the Web to find the Simple REST Client, and add it to Chrome.
- 2 Click its globe-like icon to start it in a tab.
- 3 The Simple REST Client provides no certificate-checking interface, so use another Chrome tab to accept or ignore the lack of SSL certificate.
- 4 Type the URL of a REST API, and select a method such as GET, POST, or PUT.
- 5 In the Headers field, type the basic authorization line, as in the Important note above. Click **Send**.

Status, Headers, and Data appear in the Response window.

**To use the REST API in curl**

- 1 Install curl if not already installed.
- 2 In front of the REST URL, the `-k` option avoids certificate checking, and the `-u` option specifies credentials.

```
curl -k -u admin:default  
https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/user/admin
```

**Ports Required for NSX REST API**

The NSX Manager requires port 443/TCP for REST API requests.

# Installing Components

---

After the installation of NSX Manager, you can install other components as required.

This chapter includes the following topics:

- [“Installing Licenses”](#) on page 29
- [“Working with Network Virtualization Components”](#) on page 30
- [“Working with VXLAN for Logical Switches”](#) on page 31
- [“Installing NSX Edge”](#) on page 45
- [“Working with Services”](#) on page 51
- [“Working with Conflicting Agencies”](#) on page 57
- [“Uninstalling Services”](#) on page 58

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Installing Licenses

See Install and Assign NSX for vSphere License in the NSX Installation Guide for information.

### Evaluating NSX License Capacity Usage

The Licensing Capacity Usage API command reports usage of CPUs, VMs and Concurrent users for DFW and VXLAN.

---

#### Example 2-1. Evaluate licensing capacity usage

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/licensing/capacityusage>

Response Body:

```
<featureCapacityUsageList>
  <featureCapacityUsageInfo>
    <capacityUsageInfo>
      <capacityType>CPU_CAPACITY_TYPE</capacityType>
      <usageCount>16</usageCount>
    </capacityUsageInfo>
    <capacityUsageInfo>
      <capacityType>VM_CAPACITY_TYPE</capacityType>
      <usageCount>3</usageCount>
    </capacityUsageInfo>
    <capacityUsageInfo>
      <capacityType>CONCURRENT_USER_CAPACITY_TYPE</capacityType>
```

```

        <usageCount>3</usageCount>
    </capacityUsageInfo>
    <feature>dfw</feature>
</featureCapacityUsageInfo>
<featureCapacityUsageInfo>
    <capacityUsageInfo>
        <capacityType>CPU_CAPACITY_TYPE</capacityType>
        <usageCount>16</usageCount>
    </capacityUsageInfo>
    <capacityUsageInfo>
        <capacityType>VM_CAPACITY_TYPE</capacityType>
        <usageCount>3</usageCount>
    </capacityUsageInfo>
    <capacityUsageInfo>
        <capacityType>CONCURRENT_USER_CAPACITY_TYPE</capacityType>
        <usageCount>3</usageCount>
    </capacityUsageInfo>
    <feature>vxlan</feature>
</featureCapacityUsageInfo>
</featureCapacityUsageList>

```

---

## Working with Network Virtualization Components

As the demands on datacenters continue to grow and accelerate, requirements related to speed and access to the data itself continue to grow as well. In most infrastructures, virtual machine access and mobility usually depend on physical networking infrastructure and the physical networking environments they reside in. This can force virtual workloads into less than ideal environments due to potential layer 2 or layer 3 boundaries, such as being tied to specific VLANs.

Network virtualization allows you to place these virtual workloads on any available infrastructure in the datacenter regardless of the underlying physical network infrastructure. This not only allows increased flexibility and mobility, but increased availability and resilience.

Feature configuration is managed at a cluster level. Cluster preparation can be broken down into the following:

- Install VIB and non-VIB related action: Before any per-host config a VIB must be installed on the host. The feature can use this time to perform other bootstrapping tasks which do not depend on VIB-installation. e.g. VXLAN creates the vmknic-pg and sets up some opaque data.
- Post-VIB install: Prepare each host for the feature. In the case of VXLAN, create vmknics.

## Install Network Virtualization Components

You install the network infrastructure components in your virtual environment on a per-cluster level for each vCenter server, which deploys the required software on all hosts in the cluster. This software is also referred to as an NSX vSwitch. When a new host is added to this cluster, the required software is automatically installed on the newly added host. After the network infrastructure is installed on a cluster, Logical Firewall is enabled on that cluster.

### Example 2-2. Install network virtualization

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/nwfabric/configure>

Request Body:

```

<nwFabricFeatureConfig>
  <resourceConfig>
    <resourceId>CLUSTER MOID</resourceId>
  </resourceConfig>

```

```
</nwFabricFeatureConfig>
```

---

## Delete Network Virtualization Components

Removes previously installed VIBs, tears down NSX manager to ESX messaging, and remove any other network fabric dependent features like logical wires etc. If a feature like logical wire is being used in your environment, this call fails.

### Example 2-3. Delete network virtualization

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/nwfabric/configure
```

Request Body:

```
<nwFabricFeatureConfig>
  <resourceConfig>
    <resourceId>CLUSTER MOID</resourceId>
  </resourceConfig>
</nwFabricFeatureConfig>
```

---

## Working with VXLAN for Logical Switches

Configuring logical switches is a multi-step process. You must follow these steps in order to complete logical switch configuration. In lieu of multicast routing on the physical fabric, you can add NSX controllers in your environment. You can later associate one of these traffic forwarding mechanisms with a transport zone.

### Prerequisites

- You must have the Super Administrator or Enterprise Administrator role permissions to configure and manage logical switches.
- Install network virtualization components on the clusters that are to be part of the logical switch. See [“Install Network Virtualization Components”](#) on page 30.
- Ensure that you have the following software versions.
  - VMware vCenter Server 5.5 or later
  - VMware ESX 5.1 or later on each server
  - vSphere Distributed Switch 5.1 or later
- Physical infrastructure MTU must be at least 50 bytes more than the MTU of the virtual machine vNIC.
- Set Managed IP address for each vCenter server in the vCenter Server Runtime Settings. For more information, see vCenter Server and Host Management.
- If using DHCP for IP assignment for vmknics, verify that DHCP is available on VXLAN transport VLANs.
 

If using an IP pool for static IP assignment, selecting a gateway other than the default gateway of the ESX management network leverages a dedicated TCP stack (applies to VMware ESXi™ 5.5 or later).
- For Link Aggregation Control Protocol (LACP), it is recommended that you enable 5-tuple hash distribution.
- You must use a consistent distributed virtual switch type (vendor etc.) and version across a given network scope. Inconsistent switch types can lead to undefined behavior in your logical switch.

The control plane that manages logical networks and overlay transport can be set as one of the following:

- **Multicast:** Multicast IP addresses on physical network is used for the control plane. This mode is recommended only when you are upgrading from older VXLAN deployments. Requires PIM/IGMP on physical network.

- **Unicast:** The control plane is handled by an NSX controller. All traffic replication is handled locally by the hypervisor. No multicast IP addresses or special network configuration is required.
- **Hybrid:** The optimized unicast mode. Offloads local traffic replication to physical network. This requires IGMP snooping on the first-hop switch, but does not require PIM. First-hop switch handles traffic replication for the subnet.

## Working with Controllers

For the unicast or hybrid control plane mode, you must add an NSX controller to manage overlay transport and provide East-West routing. The controller optimizes virtual machine broadcast (ARP only) traffic, and the learning is stored on the host and the controller.

### Add Controller

Adds a new NSX controller on the specified given cluster. The `hostId` parameter is optional. The `resourcePoolId` can be either the `clusterId` or `resourcePoolId`.

The IP address of the controller node will be allocated from the specified IP pool. `deployType` determines the controller node memory size and can be small, medium, or large. However, different controller deployment types are not currently supported because the OVF overrides it and different OVF types require changes in the manager build scripts. Despite not being supported, an arbitrary `deployType` size must still be specified or an error will be returned.

#### Example 2-4. Add controller

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/controller
```

Request Body:

```
<controllerSpec>
  <name>nsx-controller-node1</name>
  <description>nsx-controller</description>
  <ipPoolId>ipPool-1</ipPoolId>
  <resourcePoolId>domain-c1</resourcePoolId>
  <hostId>host-1</hostId>
  <datastoreId>datastore-1</datastoreId>
  <deployType>medium</deployType>
  <networkId>dvportgroup-1</networkId>
  <password>MyTestPassword</password>
</controllerSpec>
```

---

### Query Controllers

Retrieves details and runtime status for controller. Runtime status can be one of the following:

- Deploying - controller is being deployed and the procedure has not completed yet.
- Removing - controller is being removed and the procedure has not completed yet.
- Running - controller has been deployed and can respond to API invocation.
- Unknown - controller has been deployed but fails to respond to API invocation.

#### Example 2-5. Query controllers

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/controller
```

Response Body:

```
<controllers>
```

```

<controller>
  <id>controller-...</id>
  <name>controllerA</name>
  <description>nvp-controller</description>
  <ipAddress>10.1.1.1</ipAddress>
  <status>RUNNING</status>
</controller>
...
</controllers>

```

---

### Query Controller Addition or Deletion Details

Retrieves status of controller creation or removal. The progress gives a percentage indication of current deploy / remove procedure.

#### Example 2-6. Query controller addition or deletion details

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/controller/progress/jobId
```

Response Body:

```

<controllerDeploymentInfo>
  <vmId>vm-1</vmId>
  <progress>90</progress>
  <status>PushingFile</status>
  <exceptionMessage></exceptionMessage>
</controllerDeploymentInfo>

```

---

### Query Controller Tech Support Logs

Retrieves controller logs. Response content type is application/octet-stream and response header is filename.

This streams a fairly large bundle back (possibly hundreds of MB).

#### Example 2-7. Query controller logs

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/controller/controllerId/techsupportlogs
```

---

### Delete Controller

Deletes NSX controller. When deleting the last controller from a cluster, the parameter forceRemovalForLast must be set to true.

#### Example 2-8. Delete controller

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/controller/controllerId?forceRemoval=true/false
```

---

### Query Cluster Information

Retrieves cluster wise configuration information for controller.

#### Example 2-9. Query cluster details

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/vdn/controller/cluster>

Response Body:

```
<controllerConfig>
  <sslEnabled>true</sslEnabled>
</controllerConfig>
```

---

### Modify Cluster Configuration

Modifies cluster wise configuration information for controller.

**Example 2-10.** Modify cluster configuration

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/vdn/controller/cluster>

Request Body:

```
<controllerConfig>
  <sslEnabled>true</sslEnabled>
</controllerConfig>
```

---

### Add Controller Syslog Exporter

Configures a syslog exporter on the specified controller node.

**Example 2-11.** Add controller syslog exporter

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/vdn/controller/controllerId/syslog>

Request Body:

```
<controllerSyslogServer>
  <syslogServer>10.135.14.236</syslogServer>
  <port>514</port>
  <protocol>UDP</protocol>
  <level>INFO</level>
</controllerSyslogServer>
```

---

### Query Controller Syslog Exporter

Retrieves details about the configured syslog exporter on the specified controller node.

**Example 2-12.** Query controller syslog exporter

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/vdn/controller/controllerId/syslog>

Response Body:

```
<controllerSyslogServer>
  <syslogServer>10.135.14.236</syslogServer>
  <port>514</port>
  <protocol>UDP</protocol>
  <level>INFO</level>
</controllerSyslogServer>
```

---

## Delete Controller Syslog Exporter

Deletes syslog exporter on the specified controller node.

### Example 2-13. Delete controller syslog exporter

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/controller/controllerId/syslog
```

---

## Backup Controller Data

Takes a snapshot of the control cluster from the specified controller node.

### Example 2-14. Backup controller data

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/controller/controllerId/snapshot
```

---

To retrieve the controller IDs, log in to the vSphere Web Client. Navigate to Networking & Security > Installation. The NSX Controller Nodes table lists the controller IDs (Name column) and IP addresses (Node column) of each controller.

The output of the GET call is an octet stream containing the controller snapshot. Example call to download the snapshot is as follows.

```
curl -u admin:default -H "Accept: application/octet-stream" -X GET -k
https://NSX-Manager-IP-Address/api/2.0/vdn/controller/controllerID/snapshot >
controller_backup.snapshot
```

## Working with Segment IDs

You must specify a segment ID pool for each NSX Manager to isolate your network traffic. If an NSX controller is not deployed in your environment, you must add a multicast address range to help in spreading traffic across your network and avoid overloading a single multicast address.

### Add a new Segment ID Range

You can add a segment ID range, from which an ID is automatically assigned to the logical switch.

### Example 2-15. Add a segment ID range

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments
```

Request Body:

```
<segmentRange>
  <name>name</name> <!-- Required -->
  <desc>description</desc> <!-- optional -->
  <begin>5000</begin> <!-- Required. Minimum value is 5000 -->
  <end>65535</end> <!-- Required. Maximum value is 16777216-->
</segmentRange>
```

---

The segment range is inclusive – the beginning and ending IDs are included.

### Query all Segment ID Ranges

You can retrieve all segment ID ranges.

**Example 2-16.** Get all Segment ID Ranges

---

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments`

Response Body:

```
<segmentRanges>
  <segmentRange>
    <id>1</id>
    <name>name</name>
    <desc>description</desc>
    <begin>5000</begin>
    <end>65535</end>
  </segmentRange>
</segmentRanges>
```

---

**Query a Specific Segment ID Range**

You can retrieve a segment ID range by specifying the segment ID.

**Example 2-17.** Get a specific Segment ID Range

---

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments/segmentId`

Response Body:

```
<segmentRange>
  <id>1</id>
  <name>name</name>
  <desc>description</desc>
  <begin>5000</begin>
  <end>65535</end>
</segmentRange>
```

---

**Update a Segment ID Range**

You can update the name, description, or end of a segment ID range.

**Example 2-18.** Update a Segment ID Range

---

Request:

PUT `https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments/segmentId`

Request Body:

```
<segmentRange>
  <end>10000</end>
  <name>name</name>
  <desc>description</desc>
</segmentRange>
```

---

**Delete a Segment ID Range**

You can delete a segment ID range.

**Example 2-19.** Delete a Segment ID Range

---

Request:

DELETE <https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments/segmentId>

---

## Configure VXLAN

### Example 2-20. Install VXLAN

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/nwfabric/configure>

Request Body:

```
<nwFabricFeatureConfig>
  <featureId>com.vmware.vshield.vsm.vxlan</featureId>
  <resourceConfig>
    <resourceId>CLUSTER MOID</resourceId>
    <configSpec class="clusterMappingSpec">
      <switch><objectId>DVS MOID</objectId></switch>
      <vlanId>0</vlanId>
      <vmknicCount>1</vmknicCount>
      <!-- ipPoolId is optional and if none is specified will assume DHCP for VTEP
      address assignment.-->
      <ipPoolId>IPADDRESSPOOL ID</ipPoolId>
    </configSpec>
  </resourceConfig>
  <resourceConfig>
    <resourceId>DVS MOID</resourceId>
    <configSpec class="vdsContext">
      <switch><objectId>DVS MOID</objectId></switch>
      <mtu>1600</mtu>
      <!-- teaming value can be one of
      FAILOVER_ORDER|ETHER_CHANNEL|LACP_ACTIVE|LACP_PASSIVE|LOADBALANCE_LOADBASED
      |LOADBALANCE_SRCID|LOADBALANCE_SRCMAC|LACP_V2 -->
      <teaming>ETHER_CHANNEL</teaming>
    </configSpec>
  </resourceConfig>
</nwFabricFeatureConfig>
```

---

## Install VXLAN

### Example 2-21. Install VXLAN with LACPv2

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/nwfabric/configure>

Request Body:

```
<nwFabricFeatureConfig>
  <featureId>com.vmware.vshield.nsxmgr.vxlan</featureId>
  <resourceConfig>
    <resourceId>CLUSTER MOID</resourceId>
    <configSpec class="clusterMappingSpec">
      <switch>
        <objectId>DVS MOID</objectId>
      </switch>
      <vlanId>0</vlanId>
      <vmknicCount>1</vmknicCount>
    </configSpec>
  </resourceConfig>
  <resourceConfig>
    <resourceId>DVS MOID</resourceId>
    <configSpec class="vdsContext">
      <switch>
        <objectId>DVS MOID</objectId>
      </switch>
    </configSpec>
  </resourceConfig>
```

```

    </switch>
    <mtu>1600</mtu>
    <teaming>LACP_V2</teaming> <!-- uplinkPortName should be as specified
        in vCenter. -->
    <uplinkPortName>LAG NAME</uplinkPortName>
  </configSpec>
</resourceConfig>
</nwFabricFeatureConfig>

```

---

## Delete VXLAN

Deletes VXLAN from the specified cluster. This does not delete the network virtualization components from the cluster.

### Example 2-22. Delete VXLAN

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/nwfabric/configure
```

Request Body:

```

<nwFabricFeatureConfig>
  <featureId>com.vmware.vshield.vsm.vxlan</featureId>
  <resourceConfig>
    <resourceId>CLUSTER MOID</resourceId>
  </resourceConfig>
</nwFabricFeatureConfig>

```

---

## Delete VXLAN with vdsContext

Deletes VXLAN from the specified cluster and also removes the vdsContext.

### Example 2-23. Delete VXLAN

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/nwfabric/configure
```

Request Body:

```

<nwFabricFeatureConfig>
  <featureId>com.vmware.vshield.vsm.vxlan</featureId>
  <resourceConfig>
    <resourceId>CLUSTER MOID</resourceId>
    <configSpec class="map">
      <entry><keyclass="java.lang.String">vxlan</key><valueclass="java.lang
        .String">cascadeDeleteVdsContext</value></entry>
    </configSpec>
  </resourceConfig>
</nwFabricFeatureConfig>

```

---

## Working with Network Scopes

A network scope is the networking infrastructure within provider virtual datacenters.

### Create a Network Scope

You must specify the clusters that are to be part of the network scope. You must have the VLAN ID, UUID of the vCenter Server, and vDS ID.

**Example 2-24.** Create a network scope

Request:

POST `https://NSX-Manager-IP-Address/api/2.0/vdn/scopes`

Request Body:

```
<vdnScope>
  <name>tz-1</name>
  <description>Transport Zone 1</description>
  <clusters>
    <cluster>
      <objectId>domain-c59</objectId>
    </cluster>
  </clusters>
  <controlPlaneMode>UNICAST_MODE</controlPlaneMode>
</vdnScope>
```

**Edit a Network Scope**

You can add a cluster to or delete a cluster from a network scope.

**Example 2-25.** Create a network scope

Request:

POST `https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId?action=patch`

Request Body:

```
<vdnScope>
  <objectId>id</objectId>
  <clusters>
    <cluster>
      <objectId>domain-c59</objectId>
    </cluster>
  </clusters>
</vdnScope>
```

**Update Attributes on a Network Scope**

You can update the attributes of a network scope.

**Example 2-26.** Update attributes of a network scope

Request:

PUT `https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId/attributes`

Request Body:

```
<vdnScope>
  <objectId>vdsnScope-1</objectId>
  <name>name</name>
  <description>description</description>
</vdnScope>
```

**Query existing Network Scopes**

You can retrieve all existing network scopes.

**Example 2-27.** Get all network scopes

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/vdn/scopes>

Response Body:

```
<vdnScopes>
  <vdnScope>
    <objectId>vdnscope-2</objectId>
    <type>
      <typeName>VdnScope</typeName>
    </type>
    <name>name</name>
    <description>description</description>
    <revision>0</revision>
    <objectTypeName>VdnScope</objectTypeName>
    <extendedAttributes></extendedAttributes>
    <id>vdnscope-2</id>
    <clusters>
      <cluster>
        <objectId>domain-c124</objectId>
        <type>
          <typeName>ClusterComputeResource</typeName>
        </type>
        <name>vxlان-cluster</name>
        <scope>
          <id>datacenter-2</id>
          <objectTypeName>Datacenter</objectTypeName>
          <name>dc1</name>
        </scope>
        <extendedAttributes></extendedAttributes>
      </cluster>
      ...
    </clusters>
    <virtualWireCount>10</virtualWireCount>
  </vdnScope>
  ...
</vdnScopes>
```

---

## Query a Specific Network Scope

You can retrieve a specific network scope.

### Example 2-28. Get a network scope

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId>

Response Body:

```
<vdnScope>
  <objectId>vdnscope-2</objectId>
  <type>
    <typeName>VdnScope</typeName>
  </type>
  <name>name</name>
  <description>description</description>
  <revision>0</revision>
  <objectTypeName>VdnScope</objectTypeName>
  <extendedAttributes></extendedAttributes>
  <id>vdnscope-2</id>
  <clusters>
    <cluster>
      <objectId>domain-c124</objectId>
      <type>
        <typeName>ClusterComputeResource</typeName>
      </type>
```

```

    <name>vxlan-cluster</name>
    <scope>
      <id>datacenter-2</id>
      <objectTypeName>Datacenter</objectTypeName>
      <name>dc1</name>
    </scope>
    <extendedAttributes></extendedAttributes>
  </cluster>
  ...
</clusters>
<virtualWireCount>10</virtualWireCount>
</vdnScope>

```

---

### Delete a Network Scope

You can delete a network scope.

#### Example 2-29. Delete network scope

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId
```

---

### Repair Logical Switches in a Network Scope

You can use this method on a network scope to recreate missing distributed port groups for logical switches on a given network scope where *scopeId* is the scope. Success returns 200 and the job ID for the repair job. The location header also includes the URI of the job.

#### Example 2-30. Repair logical switches in a network scope

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId?action=repair
```

Response:

```
jobdata-20737
```

---

## Reset Communication

Resets communication between NSX Manager and a host or cluster. The *resourceId* is the ID of the cluster or host that is retrieved from the vCenter managed object browser. Examples include domain-7 or host-14

#### Example 2-31. Reset communication

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/nwfabric/configure?action=synchronize
```

Request Body:

```

<nwFabricFeatureConfig>
  <featureId>com.vmware.vshield.vsm.messagingInfra</featureId>
  <resourceConfig>
    <resourceId>resourceId</resourceId>
  </resourceConfig>
</nwFabricFeatureConfig>

```

Response:

```
jobdata-21662
```

---

## Query Features on Cluster

Retrieves all features available on the cluster.

### Example 2-32. Query features

---

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/nwfabric/features`

Response Body:

```
<featureInfos>  <!-- Can contain multiple featureInfo tags-->
  <featureInfo>
    <name>FEATURE_NAME</name>
    <featureId>FEATURE_ID</featureId>
    <version>FEATURE_VERSION</version>
  </featureInfo>
</featureInfos>
```

---

## Query Status of Specific Resources

### Example 2-33. Query status

---

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/nwfabric/status?resource=resourceId`

Response Body:

```
<resourceStatuses>
  <resourceStatus>
    <resource>
      <objectId>resource-id</objectId>
      <objectTypeName>ClusterComputerResource</objectTypeName>
      <nsxmgrUuid>jfldj</nsxmgrUuid>
      <revision>2</revision>
      <type>
        <typeName>ClusterComputerResource</typeName>
      </type>
      <name>c-1</name>
      <scope>
        <id>datacenter-2</id>
        <objectTypeName>Datacenter</objectTypeName>
        <name>dc-1</name>
      </scope>
      <clientHandle></clientHandle>
      <extendedAttributes></extendedAttributes>
    </resource>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.nwfabric.hostPrep</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>>false</updateAvailable>
      <status>RED</status>
      <message></message>
      <installed>>true</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.vxlan</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>>false</updateAvailable>
      <status>UNKNOWN</status>
      <installed>>false</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.messagingInfra</featureId>
      <featureVersion>5.5</featureVersion>
```

```

        <updateAvailable>false</updateAvailable>
        <status>UNKNOWN</status>
        <installed>false</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
        <featureId>com.vmware.vshield.firewall</featureId>
        <featureVersion>5.5</featureVersion>
        <updateAvailable>false</updateAvailable>
        <status>UNKNOWN</status>
        <installed>false</installed>
    </nwFabricFeatureStatus>
</resourceStatus>
</resourceStatuses>

```

---

## Query Status of Child Resources

### Example 2-34. Query status

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/nwfabric/status/child/parentresourceId
```

Response Body:

```

<resourceStatuses>
  <resourceStatus>
    <resource>
      <objectId>host-9</objectId>
      <objectTypeName>HostSystem</objectTypeName>
      <nsxmgrUuid>jfldj</nsxmgrUuid>
      <revision>4</revision>
      <type>
        <typeName>HostSystem</typeName>
      </type>
      <name>10.135.14.186</name>
      <scope>
        <id>domain-c34</id>
        <objectTypeName>ClusterComputerResource</objectTypeName>
        <name>c-1</name>
      </scope>
      <clientHandle></clientHandle>
      <extendedAttributes></extendedAttributes>
    </resource>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.nwfabric.hostPrep</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>false</updateAvailable>
      <status>RED</status>
      <message></message>
      <installed>true</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.vxlan</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>false</updateAvailable>
      <status>UNKNOWN</status>
      <installed>false</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.messagingInfra</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>false</updateAvailable>
      <status>UNKNOWN</status>
      <installed>false</installed>
    </nwFabricFeatureStatus>
  </nwFabricFeatureStatuses>

```

```

    <featureId>com.vmware.vshield.firewall</featureId>
    <featureVersion>5.5</featureVersion>
    <updateAvailable>>false</updateAvailable>
    <status>UNKNOWN</status>
    <installed>>false</installed>
  </nwFabricFeatureStatus>
</resourceStatus>
</resourceStatuses>

```

---

## Query Status of Resources by Criterion

### Example 2-35. Query status

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/nwfabric/status/alleligible/resource-type>

Response Body:

```

<resourceStatuses>
  <resourceStatus>
    <resource>
      <objectId>domain-c34</objectId>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <nsxmgrUuid>jfldj</nsxmgrUuid>
      <revision>2</revision>
      <type>
        <typeName>ClusterComputeResource</typeName>
      </type>
      <name>c-1</name>
      <scope>
        <id>datacenter-2</id>
        <objectTypeName>Datacenter</objectTypeName>
        <name>dc-1</name>
      </scope>
      <clientHandle></clientHandle>
      <extendedAttributes></extendedAttributes>
    </resource>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.nwfabric.hostPrep</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>>false</updateAvailable>
      <status>RED</status>
      <message></message>
      <installed>>true</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.vxlan</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>>false</updateAvailable>
      <status>UNKNOWN</status>
      <installed>>false</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.messagingInfra</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>>false</updateAvailable>
      <status>UNKNOWN</status>
      <installed>>false</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.firewall</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>>false</updateAvailable>
      <status>UNKNOWN</status>
      <installed>>false</installed>
    </nwFabricFeatureStatus>
  </resourceStatus>
</resourceStatuses>

```

```

    </nwFabricFeatureStatus>
  </resourceStatus>
  <resourceStatus>
    <resource>
      <objectId>domain-c32</objectId>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <nsxmgrUuid>jfldj</nsxmgrUuid>
      <revision>1</revision>
      <type>
        <typeName>ClusterComputeResource</typeName>
      </type>
      <name>c-2</name>
      <scope>
        <id>datacenter-12</id>
        <objectTypeName>Datacenter</objectTypeName>
        <name>dc-2</name>
      </scope>
      <clientHandle></clientHandle>
      <extendedAttributes></extendedAttributes>
    </resource>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.nwfabric.hostPrep</featureId>
      <updateAvailable>>false</updateAvailable>
      <status>UNKNOWN</status>
      <installed>>false</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.vxlan</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>>false</updateAvailable>
      <status>UNKNOWN</status>
      <installed>>false</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.messagingInfra</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>>false</updateAvailable>
      <status>UNKNOWN</status>
      <installed>>false</installed>
    </nwFabricFeatureStatus>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.firewall</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>>false</updateAvailable>
      <status>UNKNOWN</status>
      <installed>>false</installed>
    </nwFabricFeatureStatus>
  </resourceStatus>
</resourceStatuses>

```

---

## Installing NSX Edge

You can install NSX Edge as a services gateway or as a logical router.

### NSX Edge Services Gateway

The services gateway gives you access to all NSX Edge services such as firewall, NAT, DHCP, VPN, load balancing, and high availability. You can install multiple NSX Edge services gateway virtual appliances in a datacenter. Each NSX Edge virtual appliance can have a total of ten uplink and internal network interfaces. The internal interfaces connect to secured port groups and act as the gateway for all protected virtual machines in the port group. The subnet assigned to the internal interface can be a publicly routed IP space or a NATed/routed RFC 1918 private space. Firewall rules and other NSX Edge services are enforced on traffic between network interfaces.

Uplink interfaces of NSX Edge connect to uplink port groups that have access to a shared corporate network or a service that provides access layer networking. Multiple external IP addresses can be configured for load balancer, site-to-site VPN, and NAT services.

## Logical Router

The NSX Edge logical router provides East-West distributed routing with tenant IP address space and data path isolation. Virtual machines or workloads that reside on the same host on different subnets can communicate with one another without having to traverse a traditional routing interface.

A logical router can have eight uplink interfaces and up to a thousand internal interfaces.

## Manage Tuning Configuration

Starting in NSX 6.2.3, the tuning configuration API allows you to configure default values for NSX Edge configuration parameters, including publishing and health check timeouts, and CPU and memory reservation, which are applicable to all NSX Edges.

The values for the tuning configuration parameters have been set to sensible defaults and may not require any changes. However, based on datacenter capacity and requirements, you can change the default CPU and memory resource reservation percentages using this API.

This percentage is applied across all Edge VM Sizes {COMPACT, LARGE, QUADLARGE, XLARGE}.

The default values are:

- 100 % for CPU reservation
- 100 % for Memory reservation
- 1000 MHz per CPU

## Query Tuning Configuration

### Example 2-36. Get tuning configuration

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edgePublish/tuningConfiguration
```

Response Body:

```
<tuningConfiguration>
  <lockUpdatesOnEdge>>false</lockUpdatesOnEdge>
  <aggregatePublishing>>true</aggregatePublishing>
  <edgeVMHealthCheckIntervalInMin>0</edgeVMHealthCheckIntervalInMin>
  <healthCheckCommandTimeoutInMs>120000</healthCheckCommandTimeoutInMs>
  <maxParallelVixCallsForHealthCheck>25</maxParallelVixCallsForHealthCheck>
  <publishingTimeoutInMs>1200000</publishingTimeoutInMs>
  <edgeVCpuReservationPercentage>100</edgeVCpuReservationPercentage>
  <edgeMemoryReservationPercentage>100</edgeMemoryReservationPercentage>
  <megaHertzPerVCpu>1000</megaHertzPerVCpu>
</tuningConfiguration>
```

---

## Modify Tuning Configuration

You can retrieve the configuration by using the GET call in [Example 2-36](#), and replace the values as described below by using a PUT call.

- `lockUpdatesOnEdge` {true|false} - Default is false. Serialize specific Edge operations related to DHCP and vnic configuration to avoid concurrency errors when too many configuration change requests arrive at the same time.
- `aggregatePublishing` {true|false} - Default value is true (enabled). Speed up configuration change publishing to the NSX Edge by aggregating over the configuration versions.

- `edgeVMHealthCheckIntervalInMin` - Default value for time interval between NSX Edge VM's health check is 0, where NSX Manager manages the interval based on the number of NSX Edge VM's. A positive integer value overrides the default behavior.
- `healthCheckCommandTimeoutInMs` - Default timeout value for health check command is 120000.
- `maxParallelVixCallsForHealthCheck` - The maximum concurrent health check calls that can be made for NSX Edge VM's based on VIX communication channel is 25.
- `publishingTimeoutInMs` - The timeout value to publish a configuration change on NSX Edge appliance. Default is 1200000 (20 minutes).
- `edgeVCpuReservationPercentage` [0-100] - integer value, specifying the CPU reservation percentage which will be applied to the NSX Edge appliance. To disable this resource reservation, enter 0.
- `edgeMemoryReservationPercentage` [0-100] - integer value, specifying the memory reservation percentage which will be applied to the NSX Edge appliance. To disable this resource reservation, enter 0.
- `megaHertzPerVCpu` - integer value specifying the megahertz per each vCPU (1000, 1500, 2000)

---

**Example 2-37.** Modify tuning configuration

Request:

PUT `https://NSX-Manager-IP-Address/api/4.0/edgePublish/tuningConfiguration`

Request Body:

```
<tuningConfiguration>
  <lockUpdatesOnEdge>false</lockUpdatesOnEdge>
  <aggregatePublishing>true</aggregatePublishing>
  <edgeVMHealthCheckIntervalInMin>0</edgeVMHealthCheckIntervalInMin>
  <healthCheckCommandTimeoutInMs>120000</healthCheckCommandTimeoutInMs>
  <maxParallelVixCallsForHealthCheck>25</maxParallelVixCallsForHealthCheck>
  <publishingTimeoutInMs>1200000</publishingTimeoutInMs>
  <edgeVCpuReservationPercentage>100</edgeVCpuReservationPercentage>
  <edgeMemoryReservationPercentage>100</edgeMemoryReservationPercentage>
  <megaHertzPerVCpu>1000</megaHertzPerVCpu>
</tuningConfiguration>
```

---

## Installing NSX Edge Services Gateway

The NSX Edge installation API copies the NSX Edge OVF from the Edge Manager to the specified datastore and deploys an NSX Edge on the given datacenter. After the NSX Edge is installed, the virtual machine powers on and initializes according to the given network configuration. If an appliance is added, it is deployed with the specified configuration.

Installing an NSX Edge instance adds a virtual machine to the vCenter Server inventory, you must specify an IP address for the management interface, and you may name the NSX Edge instance.

The configuration you specify when you install an NSX Edge is stored in the database. If an appliance is added, the configuration is applied to it and it is deployed.

**NOTE** Do not use hidden/system resource pool IDs as they are not supported on the UI.

---

**Example 2-38.** Install Services Gateway

Request:

POST `https://NSX-Manager-IP-Address/api/4.0/edges/`

Request Body:

```
<edge>
  <datacenterMoid>datacenter-2</datacenterMoid>
  <name>org1-edge</name> <!-- optional. Default is vshield-<edgeId>. Used as a vm name
    on VC appended by "-<haIndex>" -->
```

```

<description>Description for the edge gateway</description> <!-- optional -->
<tenant>org1</tenant> <!-- optional. Will be used in syslog messages -->
<fqdn>org1edge1</fqdn> <!-- optional. Default is vShield-<edgeId>. Used to set
    hostname on the vm. Appended by "-<haIndex>" -->
<vseLogLevel>info</vseLogLevel> <!-- optional. Default is info. Other possible values
    are EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, DEBUG -->
<enableAesni>false</enableAesni> <!-- optional. Default is true -->
<enableFips>true</enableFips> <!-- optional. Default is false -->
<appliances> <!-- maximum 2 appliances can be configured. Until one appliance is
    configured, none of the configured features configured will serve the
    network -->
    <applianceSize>compact</applianceSize> <!-- optional. Default is compact. Other
        possible values are large|xlarge|quadlarge -->
    <enableCoreDump>true</enableCoreDump> <!-- optional. default is false. Enabling
        core-dump will deploy an extra disk for core-dump files, which will consume
        1GB for COMPACT, LARGE, and QUADLARGE, and 8G for XLARGE Edge.-->
    <appliance>
        <resourcePoolId>resgroup-53</resourcePoolId>
        <datastoreId>datastore-29</datastoreId>
        <hostId>host-28</hostId> <!-- optional -->
        <vmFolderId>group-v38</vmFolderId> <!-- optional -->
        <customField> <!-- optional -->
            <key>system.service.vmware.vsla.main01</key>
            <value>string</value>
        </customField>
        <cpuReservation> <!-- optional -->
            <limit>2399</limit>
            <reservation>500</reservation>
            <shares>500</shares>
        </cpuReservation>
        <memoryReservation> <!-- optional -->
            <limit>5000</limit>
            <reservation>500</reservation>
            <shares>20480</shares>
        </memoryReservation>
    </appliance>
</appliances>
<vnics> <!-- maximum 10 interfaces index:0-9 can be configured. Until one connected
    vnic is configured, none of the configured features will serve the network
    -->
    <vnic>
        <index>0</index>
        <name>internal0</name> <!-- optional. System has default Names. format vnic0 ...
            vnic7 -->
        <type>internal</type> <!-- optional. Default is internal. Other possible value
            is "uplink" -->
        <portgroupId>dvportgroup-114</portgroupId> <!-- Possible values here are
            portgroupIds or virtualwire-id. portgroupId needs to be defined if
            isConnected=true -->
        <addressGroups> <!-- Supports one or more addressGroup except on the Edge used
            for the distributed router which can only have a primary IP address. -->
            <addressGroup> <!-- vnic can be configured to have more than one
                addressGroup/subnets -->
                <primaryAddress>192.168.3.1</primaryAddress> <!-- This is mandatory for an
                    addressGroup -->
                <secondaryAddresses> <!-- Optional. Should be used to add/defined other
                    IPs used for NAT, LB, VPN, etc -->
                    <ipAddress>192.168.3.2</ipAddress>
                    <ipAddress>192.168.3.3</ipAddress> <!-- Optional. This way multiple IP
                        Addresses can be assigned to a vnic/interface -->
                </secondaryAddresses>
                <subnetMask>255.255.255.0</subnetMask> <!-- either subnetMask or
                    subnetPrefixLength should be provided. If both then subnetPrefixLength is
                    ignored -->
            </addressGroup>
            <addressGroup> <!-- vnic can be configured to have more than one
                addressGroup/subnets -->

```

```

    <primaryAddress>192.168.4.1</primaryAddress> <!-- This is mandatory for an
addressGroup -->
    <secondaryAddresses> <!-- Optional. Should be used to add/defined other
IPs used for NAT, LB, VPN, etc -->
        <ipAddress>192.168.4.2</ipAddress>
        <ipAddress>192.168.4.3</ipAddress> <!-- Optional. This way multiple IP
Addresses can be assigned to a vnic/interface -->
    </secondaryAddresses>
    <subnetPrefixLength>24</subnetPrefixLength>
</addressGroup>
<addressGroup> <!-- ipv6 addressGroup -->
    <primaryAddress>ffff::1</primaryAddress> <!-- This is mandatory for an
addressGroup -->
    <secondaryAddresses> <!-- Optional. Should be used to add/defined other
IPs used for NAT, LB, VPN, etc -->
        <ipAddress>ffff::2</ipAddress>
    </secondaryAddresses>
    <subnetPrefixLength>64</subnetPrefixLength> <!-- prefixLength valid values
1-128 -->
</addressGroup>
</addressGroups>
<macAddress> <!-- optional. When not specified, macAddresses will be managed by
VC -->
    <edgeVmHaIndex>0</edgeVmHaIndex>
    <value>00:50:56:01:03:23</value> <!-- optional. User must ensure that
macAddresses provided are unique withing the given layer 2 domain. -->
</macAddress>
<fenceParameter> <!-- optional -->
    <key>ethernet0.filter1.param1</key>
    <value>1</value>
</fenceParameter>
<mtu>1500</mtu> <!-- optional. Default is 1500 -->
<enableProxyArp>>false</enableProxyArp> <!-- optional. Default is false -->
<enableSendRedirects>>true</enableSendRedirects> <!-- optional. Default is true
-->
<isConnected>>true</isConnected> <!-- optional. Default is false -->
<inShapingPolicy> <!-- optional -->
    <averageBandwidth>200000000</averageBandwidth>
    <peakBandwidth>200000000</peakBandwidth>
    <burstSize>0</burstSize>
    <enabled>>true</enabled>
    <inherited>>false</inherited>
</inShapingPolicy>
<outShapingPolicy> <!-- optional -->
    <averageBandwidth>400000000</averageBandwidth>
    <peakBandwidth>400000000</peakBandwidth>
    <burstSize>0</burstSize>
    <enabled>>true</enabled>
    <inherited>>false</inherited>
</outShapingPolicy>
</vnic>
</vnics>
<cliSettings> <!-- optional. Default user/pass is admin/default, and remoteAccess is
false (i.e. disabled) -->
    <userName>test</userName> <!-- When you change the userName, you are overwriting
the current userName. -->
    <password>test123!</password> <!-- The password should be atleast 12 characters
long, must be a mix of alphabets, digits and special characters. Must
contain at-least 1 uppercase, 1 lowercase, 1 special character and 1 digit.
In addition, a character cannot be repeated 3 or more times
consectively.-->
    <remoteAccess>>false</remoteAccess> <!-- remote Access specifies whether cli console
access over ssh must be enabled. Relevant firewall rules to allow traffic
on port 22 must be opened by user/client. Please note: it is advisable to
restrict ssh access to Edge cli to only a limited ip addresses - so
firewall rules must be opened cautiously. -->
</cliSettings>
<autoConfiguration> <!-- optional -->

```

```

    <enabled>true</enabled> <!-- Optional. Default:true. If set to false, user should
        add the nat,firewall,routing config to control plane work for LB, VPN, etc
        -->
    <rulePriority>high</rulePriority> <!-- Optional. Default is high. Other possible
        value is low -->
</autoConfiguration>
<dnsClient> <!-- optional. if the primary/secondary are specified and the DNS service
    not, the primary/secondary will to used as the default of the DNS service.
    -->
    <primaryDns>10.117.0.1</primaryDns>
    <secondaryDns>10.117.0.2</secondaryDns>
    <domainName>vmware.com</domainName>
    <domainName>foo.com</domainName>
</dnsClient>
<queryDaemon> <!-- optional. defined for the sake of communication between SLB VM and
    edge vm for GSLB feature. -->
    <enabled>true</enabled> <!-- default to false-->
    <port>5666</port> <!-- default to 5666 -->
</queryDaemon>
</edge>

```

---

## Installing a Logical Router

Before installing a logical router, you must prepare the hosts on the appropriate clusters. For more information, see [“Working with Network Virtualization Components”](#) on page 30.

A logical router can have eight uplink interfaces and up to a thousand internal interfaces.

The user specified configuration is stored in the database and Edge identifier is returned to the user. This identifier must be used for future configurations on the given Edge.

If any appliance(s) are specified and at-least one connected interface/vnic is specified, then the appliance(s) are deployed and configuration is applied to them.

It is not possible to set the `<ecmp>true</ecmp>` property upon creation of a distributed logical router Edge and a subsequent API call is required to enable ECMP.

DHCP relay settings are not able to be used when creating a distributed logical router Edge and a subsequent API call is required to configure DHCP relay properties.

### Example 2-39. Install a logical router

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges
```

Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<edge>
  <datacenterMoid>datacenter-2</datacenterMoid>
  <type>distributedRouter</type>
  <!-- Mandatory to create "distributedRouter" edge. when absent, defaults to
    "gatewayServices" -->
  <appliances>
    <!-- Mandatory for "distributedRouter" edge. Atleast one appliance needs to be
      configured -->
    <appliance>
      <resourcePoolId>resgroup-20</resourcePoolId>
      <datastoreId>datastore-23</datastoreId>
    </appliance>
  </appliances>
  <mgmtInterface>
    <!-- Mandatory for "distributedRouter" edge -->
    <connectedToId>dvportgroup-38</connectedToId>
  </mgmtInterface>
</edge>

```

```

<addressGroups> <!-- Supports one or more addressGroup except on the Edge used for
    the distributed router which can only have a primary IP address. -->
    <addressGroup>
        <primaryAddress>10.112.196.165</primaryAddress>
        <subnetMask>255.255.252.0</subnetMask>
    </addressGroup>
</addressGroups>
</mgmtInterface>
<interfaces>
    <!-- Optional. Can be added later using modular APIs. Upto 999 interfaces
        supported. -->
    <interface>
        <type>uplink</type>
        <mtu>1500</mtu>
        <isConnected>true</isConnected>
        <addressGroups> <!-- Supports one or more addressGroup except on the Edge used
            for the distributed router which can only have a primary IP address. -->
            <addressGroup>
                <primaryAddress>192.168.10.1</primaryAddress>
                <!-- "distributedRouter" edge only supports IPv4 addresses -->
                <subnetMask>255.255.255.0</subnetMask>
            </addressGroup>
        </addressGroups>
        <connectedToId>dvportgroup-39</connectedToId>
        <!-- "distributedRouter" edge does not support legacy portGroups -->
    </interface>
    <interface>
        <type>internal</type>
        <mtu>1500</mtu>
        <isConnected>true</isConnected>
        <addressGroups> <!-- Supports one or more addressGroup except on the Edge used
            for the distributed router which can only have a primary IP address. -->
            <addressGroup>
                <primaryAddress>192.168.20.1</primaryAddress>
                <subnetMask>255.255.255.0</subnetMask>
            </addressGroup>
        </addressGroups>
        <connectedToId>dvportgroup-40</connectedToId>
    </interface>
</interfaces>
</edge>

```

---

The location header returns the edgeId of the installed router. You must use this ID to configure and manage this NSX Edge instance.

## Working with Services

The security fabric simplifies and automates deployment of security services and provide a platform for configuration of the elements that are required to provide security to workloads. These elements include:

- Internal components:
  - USVM
  - Endpoint Mux
  - Data Security
  - Logical Firewall
- External components
  - Partner OVF / VIBs
  - Partner vendor policy templates

For partner services, the overall workflow begins with registration of services by partner consoles, followed by deployment of the services by the administrator.

Subsequent workflow is as follows:

- 1 Select the clusters on which to deploy the security fabric (Mux, Traffic filter, USVM).
- 2 Specify an IP pool to be used with the SVMs (available only if the partner registration indicates requirement of static IPs)
- 3 Select portgroup (DVPG) to be used for each cluster (a default is pre-populated for the user).
- 4 Select datastore to be used for each cluster (a default is pre-populated for the user).
- 5 NSX Manager deploys the components on all hosts of the selected clusters.

Once you deploy the security fabric, an agency defines the configuration needed to deploy agents (host components and appliances). An agency is created per cluster per deployment spec associated with services. Agents are deployed on the selected clusters, and events / hooks for all the relevant actions are generated.

## Install Security Fabric

### Example 2-40. Install service

---

Request:

POST `https://NSX-Manager-IP-Address/api/2.0/si/deploy?startTime=time`

Request Body:

```
<clusterDeploymentConfigs>
  <clusterDeploymentConfig>
    <clusterId>cluster-id</clusterId>
    <datastore>ds-id</datastore> <!-- Used only in POST. Should be empty in PUT -->
    <services>
      <serviceDeploymentConfig>
        <serviceId>service-id</serviceId>
        <dvPortGroup>dvpg-id</dvPortGroup>
        <ipPool>ipPool</ipPool>
      </serviceDeploymentConfig>
    </services>
  </clusterDeploymentConfig>
</clusterDeploymentConfigs>
```

---

where:

- `dataStore` - Needs to be specified only in POST call. In PUT call, it should be left empty otherwise the call will fail.
- `dvPortGroup` - This is optional. If not specified, then user will set the Agent using vCenter Server.
- `ipPool` - This is optional. if not specified, IP address is assigned through DHCP.
- `startTime` - Time when the deployment task(s) are scheduled for. If this is not specified then deployment will happen immediately.

## Service Dependency

Services installed through the security fabric may be dependent on other services. When an internal service is registered, a `dependencyMap` is maintained with the `service-id` and implementation type of the internal service.

When partner registers a new service, the security fabric looks up its implementation type in the `dependencyMap` to identify the service it depends on, if any. Accordingly, a new field in Service object called `dependsOn-service-id` is populated.

## Deploying a Service with a Dependency

### Example 2-41. Deploy service

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/si/deploy
```

---

## Identify Service Dependency

Lists the service on which the specified service depends on.

### Example 2-42. Identify service dependency

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/si/deploy/service/serviceId/dependsOn
```

---

## Uninstall Service Dependency

Lists the service on which the specified service depends on.

### Example 2-43. Uninstall service dependency

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/si/deploy/cluster/clusterId
```

---

If you try to remove a service on which a service depends on and it is already installed, the uninstallation fails.

In order to uninstall services in any order, set parameter ignoreDependency true.

## Query Installed Services

Retrieves all services currently deployed on the cluster along with their status.

### Example 2-44. Query services

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/si/deploy/cluster/clusterId
```

Response Body:

```
<deployedServices>
  <deployedService>
    <deploymentUnitId>deploymentunit-1</deploymentUnitId>
    <serviceId>service-3</serviceId>
    <cluster>
      <objectId>domain-c41</objectId>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <nsxmgrUuid>42036483-6CF3-4F0F-B356-2EB1E6369C6F</nsxmgrUuid>
      <revision>2</revision>
      <type>
        <typeName>ClusterComputeResource</typeName>
      </type>
      <name>Cluster-1</name>
      <scope>
        <id>datacenter-21</id>
        <objectTypeName>Datacenter</objectTypeName>
        <name>nasingh-dc</name>
      </scope>
      <extendedAttributes></extendedAttributes>
    </cluster>
  </deployedService>
</deployedServices>
```

```

<serviceName>domain-c41_service-3</serviceName>
<datastore>
  <objectId>datastore-29</objectId>
  <objectTypeName>Datastore</objectTypeName>
  <nsxmgrUuid>42036483-6CF3-4F0F-B356-2EB1E6369C6F</nsxmgrUuid>
  <revision>1</revision>
  <type>
    <typeName>Datastore</typeName>
  </type>
  <name>datastore1</name>
  <extendedAttributes></extendedAttributes>
</datastore>
<dvPortGroup>
  <objectId>dvportgroup-45</objectId>
  <objectTypeName>DistributedVirtualPortgroup</objectTypeName>
  <nsxmgrUuid>42036483-6CF3-4F0F-B356-2EB1E6369C6F</nsxmgrUuid>
  <revision>2</revision>
  <type>
    <typeName>DistributedVirtualPortgroup</typeName>
  </type>
  <name>dvPortGroup</name>
  <scope>
    <id>datacenter-21</id>
    <objectTypeName>Datacenter</objectTypeName>
    <name>nasingh-dc</name>
  </scope>
  <extendedAttributes></extendedAttributes>
</dvPortGroup>
<serviceStatus>SUCCEEDED</serviceStatus>
</deployedService>
</deployedServices>

```

---

## Query Details about a Service

Retrieves detailed information about the service.

### Example 2-45. Query service

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/si/deploy/cluster/clusterId/service/serviceId
```

Response Body:

See [Example 2-44](#).

---

## Query Clusters

Retrieves all clusters on which the specified service is installed.

### Example 2-46. Query clusters

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/si/deploy/service/serviceId
```

Response Body:

See [Example 2-44](#).

---

## Query Agents on Host

Retrieves all agents on the specified host. The response body contains agent IDs for each agent, which you can use to retrieve details about that agent.

### Example 2-47. Query agents on host

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/si/host/hostId/agents
```

Response Body:

```
<fabricAgents>
  <agent>
    <agentId>nsxmragent-1</agentId>
    <agentName>agent name</agentName>
    <serviceId>service-6</serviceId>
    <serviceName>EndpointService</serviceName>
    <operationalStatus>ENABLED</operationalStatus>
    <progressStatus>IN_PROGRESS</progressStatus>
    <vmId>vm-92</vmId>
    <host>host-10</host>
    <allocatedIpAddress>
      <id>2</id>
      <ipAddress>10.112.5.182</ipAddress>
      <gateway>10.112.5.253</gateway>
      <prefixLength>23</prefixLength>
      <dnsServer1>10.112.0.1</dnsServer1>
      <dnsServer2>10.112.0.2</dnsServer2>
      <dnsSuffix />
      <subnetId>subnet-1</subnetId>
    </allocatedIpAddress>
    <serviceStatus>
      <status>WARNING</status>
      <errorId>partner_error</errorId>
      <errorDescription>partner_error</errorDescription>
    </serviceStatus>
    <hostInfo>
      <objectId>host-10</objectId>
      <objectTypeName>HostSystem</objectTypeName>
      <nsxmgrUuid>420369CD-2311-F1F7-D4AA-1158EA688E54</nsxmgrUuid>
      <revision>1</revision>
      <type>
        <typeName>HostSystem</typeName>
      </type>
      <name>10.112.5.173</name>
      <scope>
        <id>domain-c7</id>
        <objectTypeName>ClusterComputerResource</objectTypeName>
        <name>Kaustubh-CL</name>
      </scope>
      <clientHandle></clientHandle>
      <extendedAttributes></extendedAttributes>
    </hostInfo>
    <initialData>partner data if present</initialData>
  </agent>
</fabricAgents>
```

---

## Query Agent Information

Retrieves agent (agents (host components and appliances)) details.

### Example 2-48. Query agent details

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/si/agent/agentId>

Response Body:

```
<agent>
  <agentId>nsxmragent-1</agentId>
  <agentName>agent name</agentName>
  <serviceId>service-6</serviceId>
  <serviceName>EndpointService</serviceName>
  <operationalStatus>ENABLED</operationalStatus>
  <progressStatus>IN_PROGRESS</progressStatus>
  <vmId>vm-92</vmId>
  <host>host-10</host>
  <allocatedIpAddress>
    <id>2</id>
    <ipAddress>10.112.5.182</ipAddress>
    <gateway>10.112.5.253</gateway>
    <prefixLength>23</prefixLength>
    <dnsServer1>10.112.0.1</dnsServer1>
    <dnsServer2>10.112.0.2</dnsServer2>
    <dnsSuffix></dnsSuffix>
    <subnetId>subnet-1</subnetId>
  </allocatedIpAddress>
  <serviceStatus>
    <status>WARNING</status>
    <errorId>partner_error</errorId>
    <errorDescription>partner_error</errorDescription>
  </serviceStatus>
  <hostInfo>
    <objectId>host-10</objectId>
    <objectTypeName>HostSystem</objectTypeName>
    <nsxmgrUuid>420369CD-2311-F1F7-D4AA-1158EA688E54</nsxmgrUuid>
    <revision>1</revision>
    <type>
      <typeName>HostSystem</typeName>
    </type>
    <name>10.112.5.173</name>
    <scope>
      <id>domain-c7</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>Kaustubh-CL</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </hostInfo>
  <initialData>partner data if present</initialData>
</agent>
```

## Query Agents for Deployment

Retrieves all agents for the specified deployment.

**Example 2-49.** Query agents for deployment

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/si/deployment/deploymentunitId/agents>

Response Body:

```
<fabricAgents>
  <agent>
    <agentId>nsxmragent-1</agentId>
    <agentName>agent name</agentName>
    <serviceId>service-6</serviceId>
    <serviceName>EndpointService</serviceName>
    <operationalStatus>ENABLED</operationalStatus>
```

```

<progressStatus>IN_PROGRESS</progressStatus>
<vmId>vm-92</vmId>
<host>host-10</host>
<allocatedIpAddress>
  <id>2</id>
  <ipAddress>10.112.5.182</ipAddress>
  <gateway>10.112.5.253</gateway>
  <prefixLength>23</prefixLength>
  <dnsServer1>10.112.0.1</dnsServer1>
  <dnsServer2>10.112.0.2</dnsServer2>
  <dnsSuffix></dnsSuffix>
  <subnetId>subnet-1</subnetId>
</allocatedIpAddress>
<serviceStatus>
  <status>WARNING</status>
  <errorId>partner_error</errorId>
  <errorDescription>partner_error</errorDescription>
</serviceStatus>
<hostInfo>
  <objectId>host-10</objectId>
  <objectTypeName>HostSystem</objectTypeName>
  <nsxmgrUuid>420369CD-2311-F1F7-D4AA-1158EA688E54</nsxmgrUuid>
  <revision>1</revision>
  <type>
    <typeName>HostSystem</typeName>
  </type>
  <name>10.112.5.173</name>
  <scope>
    <id>domain-c7</id>
    <objectTypeName>ClusterComputerResource</objectTypeName>
    <name>kaustubh-CL</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
</hostInfo>
<initialData>partner data</initialData>
</agent>
</fabricAgents>

```

---

## Working with Conflicting Agencies

When the NSX Manager database backup is restored to an older point in time, it is possible that deployment units for some EAM Agencies are missing. These APIs help the administrator identify such EAM Agencies and take appropriate action.

### Query Conflicts

Retrieves conflicting Deployment Units and EAM Agencies, if any, and the allowed operations on them.

#### Example 2-50. Query conflicts

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/si//fabric/sync/conflicts
```

Response Body:

```

<fabricSyncConflictInfo>
  <fabricSyncConflictInfo>
    <conflictExist>true</conflictExist>
  <agencies>
    <agenciesInfo>
      <agencyConflictInfo>
        <agencyId>agency-150</agencyId>
        <agencyName>_VCNS_264_nasingh-cluster1_VMware Endpoint</agencyName>

```

```

    </agencyConflictInfo>
  </agenciesInfo>
  <allowedOperations>
    <conflictResolverOperation>DELETE</conflictResolverOperation>
    <conflictResolverOperation>RESTORE</conflictResolverOperation>
  </allowedOperations>
</agencies>
</fabricSyncConflictInfo>

```

---

## Restore Conflicting Agencies

Creates Deployment Units for conflicting EAM Agencies.

### Example 2-51. Query conflicts

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/si/fabric/sync/conflicts>

Request Body:

```

<conflictResolverInfo>
  <agencyAction>RESTORE</agencyAction>
</conflictResolverInfo>

```

---

## Delete Conflicting Agencies

Deletes conflicting EAM Agencies.

### Example 2-52. Delete conflicts

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/si/fabric/sync/conflicts>

Request Body:

```

<conflictResolverInfo>
  <agencyAction>DELETE</agencyAction>
</conflictResolverInfo>

```

---

## Delete Deployment Units

Deletes Deployment Units for conflicting EAM Agencies.

### Example 2-53. Query conflicts

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/si/fabric/sync/conflicts>

Request Body:

```

<conflictResolverInfo>
  <deploymentUnitAction>DELETE</deploymentUnitAction>
</conflictResolverInfo>

```

---

## Uninstalling Services

Uninstalls the specified services from the specified lusters.

**Example 2-54.** Uninstall services from a cluster

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/si/deploy/cluster/clusterId?services=  
service-id1,service-id2&startTime=time
```

---

where:

- *services* - list of service id's that needs to be uninstalled from the cluster. If this is not specified then all the services will be uninstalled.
- *startTime* - time when the uninstall will be scheduled for. If this is not specified then uninstall will happen immediately.

**Example 2-55.** Uninstall specified service from specified clusters

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/si/deploy/service/serviceId?clusters=  
cluster-id1,cluster-id2&startTime=time
```

---

where:

- *clusters* - list of cluster id's that service needs to be uninstalled from.
- *startTime* - time when the uninstall will be scheduled for. If this is not specified then uninstall will happen immediately.



# Managing the NSX Manager Appliance

# 3

With the appliance management tool, you can manage:

- System configurations like network configuration, syslog, time settings, and certificate management etc.
- Components of appliance such as NSX Manager, Postgres, SSH component, Rabbitmq service etc.
- Overall support related features such as tech support logs, backup restore, status, and summary reports of appliance health.

The chapter includes the following topics:

- [“Configuring NSX Manager with vCenter Server”](#) on page 61
- [“Certificate Management”](#) on page 62
- [“Resource Management”](#) on page 64
- [“Components Management”](#) on page 70
- [“Working with Backup and Restore”](#) on page 73
- [“Working with Tech Support Logs”](#) on page 75
- [“Working with Support Notifications”](#) on page 77

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Configuring NSX Manager with vCenter Server

You can synchronize NSX Manager with a vCenter Server, which enables the Networking and Security tab in the vCenter Web Client to display your VMware Infrastructure inventory.

### Configure vCenter Server with NSX Manager

**Example 3-1.** Synchronize NSX Manager with vCenter server

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/services/vcconfig>

Request Body:

```
<vcInfo>
  <ipAddress>vc-1-01a.corp.local</ipAddress> <!-- Required. FQDN or IP address of
    vCenter server.-->
  <userName>administrator@vsphere.local</userName> <!-- Required. -->
  <password>VMware123</password> <!-- Required. -->
```

```

    <certificateThumbprint>D2:75:61:24:52:CA:B2:8D:D3:25:3F:78:11:2A:8F:94:5A:30:57:0D</c
      ertificateThumbprint> <!-- Required. Must be : delimited hexadecimal.
        -->
    <assignRoleToUser>true</assignRoleToUser> <!-- Optional. -->
    <pluginDownloadServer></pluginDownloadServer> <!-- Optional. -->
    <pluginDownloadPort></pluginDownloadPort> <!-- Optional. -->
  </vcInfo>

```

---

The <certificateThumbprint> property must be in : delimited hexadecimal. For example:

```
74:ED:6C:68:CF:92:C1:AE:C3:73:65:5F:EA:74:34:9D:84:36:2B:34
```

## Query Configuration Details

**Example 3-2.** Get vCenter Server configuration details on NSX Manager

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/vcconfig
```

Response Body:

```

<vcInfo>
  <ipAddress></ipAddress>
  <userName></userName>
  <certificateThumbprint>vCenter thumbprint</certificateThumbprint>
  <assignRoleToUser></assignRoleToUser>
  <vcInventoryLastUpdateTime></vcInventoryLastUpdateTime>
</vcInfo>

```

---

**Example 3-3.** Get default vCenter Server connection status

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/vcconfig/status
```

Response Body:

```

<vcConfigStatus>
  <connected></connected>
  <lastInventorySyncTime></lastInventorySyncTime>
</vcConfigStatus>

```

---

## Certificate Management

### Generate CSR Certificate

Generates CSR. Response header contains created file location.

**Example 3-4.** Generate CSR

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/1.0/appliance-management/certificatemanager
  /csr/nsx
```

Request Body:

```

<csr>
  <algorithm></algorithm>
  <keySize></keySize>
  <subjectDto>
    <commonName></commonName>
    <organizationUnit></organizationUnit>
  </subjectDto>
</csr>

```

```

    <organizationName></organizationName>
    <localityName></localityName>
    <stateName></stateName>
    <countryCode></countryCode>
  </subjectDn>
</csr>

```

---

## Download CSR Certificate

Downloads generated CSR from appliance.

### Example 3-5. Download CSR

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/certificatemanager
/csr/nsx
```

---

## Upload Certificate Chain

Input is certificate chain file which is a PEM encoded chain of certificates received from the CA after signing a CSR.

### Example 3-6. Upload certificate chain

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/1.0/appliance-management/certificatemanager
/uploadchain/nsx
```

---

## Query Certificates

Retrieves certificates.

### Example 3-7. Query certificates

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/certificatemanager
/certificates/nsx
```

Response Body:

```

<x509Certificates>
  <x509certificate>
    <subjectCn></subjectCn>
    <issuerCn></issuerCn>
    <version></version>
    <serialNumber></serialNumber>
    <signatureAlgo></signatureAlgo>
    <signature></signature>
    <notBefore></notBefore>
    <notAfter></notAfter>
    <issuer></issuer>
    <subject></subject>
    <publicKeyAlgo></publicKeyAlgo>
    <publicKeyLength></publicKeyLength>
    <rsaPublicKeyModulus></rsaPublicKeyModulus>
    <rsaPublicKeyExponent></rsaPublicKeyExponent>
    <sha1Hash></sha1Hash>
    <md5Hash></md5Hash>
    <isCa></isCa>
  </x509certificate>
</x509Certificates>

```

```

    <invalid></invalid>
  </x509certificate>
  . . .
</x509Certificates>

```

---

## Upload Keystore File

Input is PKCS#12 formatted NSX file along with password.

### Example 3-8. Upload file

---

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/appliance-management/certificatemanager
/pkcs12keystore/nsx?password="123"
```

---

## Resource Management

### Query Global Appliance Manager Information

Retrieves global information containing version information as well as current logged in user.

#### Example 3-9. Query global information

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/global/info
```

Response Body:

```

<globalInfo>
  <currentLoggedInUser>Joe</currentLoggedInUser>
  <versionInfo>
    <majorVersion>6</majorVersion>
    <minorVersion>0</minorVersion>
    <patchVersion>0</patchVersion>
    <buildNumber>1300000000</buildNumber>
  </versionInfo>
</globalInfo>

```

---

### Query Summary Appliance Manager Information

Retrieves system summary information such as address, dns name, version, CPU, memory, and storage.

#### Example 3-10. Query summary

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/summary/system
```

Response Body:

```

<systemSummary>
  <ipv4Address></ipv4Address>
  <dnsName></dnsName>
  <applianceName></applianceName>
  <versionInfo>
    <majorVersion></majorVersion>
    <minorVersion></minorVersion>
    <patchVersion></patchVersion>
    <buildNumber></buildNumber>
  </versionInfo>

```

```

<uptime></uptime>
<cpuInfoDto>
  <totalNoOfCPUs></totalNoOfCPUs>
  <capacity></capacity>
  <usedCapacity></usedCapacity>
  <freeCapacity></freeCapacity>
  <usedPercentage></usedPercentage>
</cpuInfoDto>
<memInfoDto>
  <totalMemory></totalMemory>
  <usedMemory></usedMemory>
  <freeMemory></freeMemory>
  <usedPercentage></usedPercentage>
</memInfoDto>
<storageInfoDto>
  <totalStorage></totalStorage>
  <usedStorage></usedStorage>
  <freeStorage></freeStorage>
  <usedPercentage></usedPercentage>
</storageInfoDto>
<currentSystemDate></currentSystemDate>
</systemSummary>

```

---

## Query Component Information

Retrieves summary of all available components available and their status information.

### Example 3-11. Query global information

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/summary/components
```

Response Body:

```

<componentsSummary>
  <componentsByGroup class="tree-map">
    <entry>
      <string></string>
      <components>
        <component>
          <componentId></componentId>
          <name></name>
          <description></description>
          <status></status>
          <enabled></enabled>
          <showTechSupportLogs></showTechSupportLogs>
          <usedBy>
            <string></string>
          </usedBy>
          <componentGroup></componentGroup>
        </component>
        <component>
          ...
        </component>
      </components>
    </entry>
    <entry>
      ...
    </entry>
  </componentsByGroup>
</componentsSummary>

```

---

## Reboot Appliance Manager

Reboots the appliance manager.

---

### Example 3-12. Reboot appliance

Request:

POST `https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/restart`

---

## Query Appliance Manager CPU

---

### Example 3-13. Query CPU

Request:

GET `https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/cpuinfo`

Response Body:

```
<cpuInfo>
  <totalNoofCPUs></totalNoofCPUs>
  <capacity></capacity>
  <usedCapacity></usedCapacity>
  <freeCapacity></freeCapacity>
  <usedPercentage></usedPercentage>
</cpuInfo>
```

---

## Query Appliance Manager Uptime

---

### Example 3-14. Query uptime

Request:

GET `https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/uptime`

Response Body:

<> days, <> hours, <> minutes

---

## Query Appliance Manager Memory

---

### Example 3-15. Query memory

Request:

GET `https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/meminfo`

Response Body:

```
<memInfo>
  <totalMemory>11996 MB</totalMemory>
  <usedMemory>6524 MB</usedMemory>
  <freeMemory>5471 MB</freeMemory>
  <usedPercentage>54</usedPercentage>
</memInfo>
```

---

## Query Appliance Manager Storage

### Example 3-16. Query storage

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/storageinfo
```

Response Body:

```
<storageInfo>
  <totalStorage></totalStorage>
  <usedStorage></usedStorage>
  <freeStorage></freeStorage>
  <usedPercentage></usedPercentage>
</storageInfo>
```

---

## Working with Network Settings

### Query Network Information

Retrieves network information such as configured host name, IP address, and DNS settings.

### Example 3-17. Query network details

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/network
```

Response Body:

```
<network>
  <hostName></hostName>
  <domainName></domainName>
  <networkIPv4AddressDto>
    <ipv4Address></ipv4Address>
    <ipv4NetMask></ipv4NetMask>
    <ipv4Gateway></ipv4Gateway>
  </networkIPv4AddressDto>
  <networkIPv6AddressDto>
    <ipv6Address></ipv6Address>
    <ipv6PrefixLength></ipv6PrefixLength>
    <ipv6Gateway></ipv6Gateway>
  </networkIPv6AddressDto>
  <dns>
    <ipv4Address></ipv4Address>
    <ipv6Address></ipv6Address>
    <domainList></domainList>
  </dns>
</network>
```

---

### Configure DNS Servers

Configures DNS servers.

### Example 3-18. Configure DNS

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/network/dns
```

Request Body:

```
<dns>
  <ipv4Address></ipv4Address>
```

```

        <ipv6Address></ipv6Address>
        <domainList></domainList>
</dns>

```

---

### Delete DNS Servers

Deletes DNS servers.

#### Example 3-19. Configure DNS

---

Request:

DELETE <https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/network/dns>

---

## Working with Time Settings

### Configure Time Settings

You can either configure time or specify the NTP server to be used for time synchronization.

#### Example 3-20. Configure time

---

Request:

PUT <https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/timesettings>

Request Body:

```

<timeSettings>
  <ntpServer>
    <string></string>
  </ntpServer>
  <datetime></datetime>
  <timezone></timezone>
</timeSettings>

```

---

### Query Time Settings

Retrieves time settings like timezone or current date and time with NTP server, if configured.

#### Example 3-21. Query time settings

---

Request:

GET <https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/timesettings>

Response Body:

```

<timeSettings>
  <ntpServer>
    <string></string>
    <string></string>
  </ntpServer>
  <datetime></datetime>
  <timezone></timezone>
</timeSettings>

```

---

### Delete Time Settings

Deletes NTP server.

**Example 3-22.** Delete NTP

Request:

```
DELETE https://NSX-Manager-IP-Address/api/1.0/appliance-management/system
        /timesettingsntp
```

**Working with Locale Settings****Configure Locale**

Configures locale.

**Example 3-23.** Configure locale

Request:

```
PUT https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/locale
```

Request Body:

```
<locale>
  <language>en</language>
  <country>US</country>
</locale>
```

**Query Locale**

Retrieves locale information.

**Example 3-24.** Query locale

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/locale
```

Response Body:

```
<locale>
  <language>en</language>
  <country>US</country>
</locale>
```

**Working with Syslog Servers**

If you specify a syslog server, NSX Manager sends all audit logs and system events from NSX Manager to the syslog server.

**Configure Syslog Servers**

Configures syslog servers.

**Example 3-25.** Configure syslog

Request:

```
PUT https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/syslogserver
```

Request Body:

```
<syslogserver>
  <syslogServer></syslogServer>
  <port></port>
```

```
<protocol></protocol>
</syslogserver>
```

---

### Query Syslog Servers

Retrieves syslog servers.

#### Example 3-26. Query syslog

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/syslogserver
```

Response Body:

```
<syslogserver>
  <syslogServer></syslogServer>
  <port></port>
  <protocol></protocol>
</syslogserver>
```

---

### Delete Syslog Servers

Deletes syslog servers.

#### Example 3-27. Delete syslog

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/1.0/appliance-management/system/syslogserver
```

---

## Components Management

### Query Components

Retrieves all Appliance Manager components.

#### Example 3-28. Query components

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/components
```

Response Body:

```
<components>
  <component>
    <componentId></componentId>
    <name></name>
    <description></description>
    <status></status>
    <enabled>true</enabled>
    <showTechSupportLogs></showTechSupportLogs>
    <usedBy>
      <string></string>
    </usedBy>
    <componentGroup></componentGroup>
  </component>
  ...
  <component>
    <componentId></componentId>
    <name></name>
```

```

    <description></description>
    <status></status>
    <enabled>true</enabled>
    <showTechSupportLogs></showTechSupportLogs>
    <usedBy>
      <string></string>
    </usedBy>
    <componentGroup>
    </componentGroup>
  </component>
</components>

```

---

## Query Specific Component

Retrieves details for the specified component ID.

### Example 3-29. Query component

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/components/component
/componentId
```

Response Body:

```

<component>
  <componentId></componentId>
  <name></name>
  <description> Manager</description>
  <status></status>
  <enabled></enabled>
  <showTechSupportLogs></showTechSupportLogs>
  <uses>
    <string></string>
    <string></string>
  </uses>
  <usedBy/>
  <componentGroup></componentGroup>
  <versionInfo>
    <majorVersion></majorVersion>
    <minorVersion></minorVersion>
    <patchVersion></patchVersion>
    <buildNumber></buildNumber>
  </versionInfo>
</component>

```

---

## Query Component Dependencies

Retrieves dependency details for the specified component ID.

### Example 3-30. Query component dependency details

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/components/componentId
/dependencies
```

Response Body:

```

<list>
  <string>VPOSTGRES</string>
  <string>RABBITMQ</string>
</list>

```

---

## Query Specific Component Dependents

Retrieves dependents for the specified component ID.

---

### Example 3-31. Query component dependents

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/components/componentId
/dependents
```

Response Body:

```
<list>
  <string></string>
  <string></string>
</list>
```

---

## Query Component Status

Retrieves current status for the specified component ID.

---

### Example 3-32. Query component status

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/components/componentId
/status
```

Response Body:

```
<result>
  <result class="status"></result>
  <operationStatus></operationStatus>
</result>
```

---

## Toggle Specific Component Status

Toggles component status.

---

### Example 3-33. Toggle status

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/appliance-management/components/componentId
/toggleStatus/command
```

---

## Restart Appliance Management Web Application

Restarts appliance management web application.

---

### Example 3-34. Restart web application

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/appliance-management/components/component
/APPMGMT/restart
```

## Working with Backup and Restore

You can back up and restore your NSX Manager data, which can include system configuration, events, and audit log tables. Configuration tables are included in every backup. Backups are saved to a remote location that must be accessible by the NSX Manager.

For information on backing up controller data, see [“Backup Controller Data”](#) on page 35.

### Configure Backup Settings

Configures backup on the Appliance Manager.

#### Example 3-35. Configure backup

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/1.0/appliance-management/backuprestore
/backupsettings
```

Request Body:

```
<backupRestoreSettings>
  <ftpSettings>
    <transferProtocol></transferProtocol>
    <hostNameIPAddress></hostNameIPAddress>
    <port></port>
    <userName></userName><password></password>
    <backupDirectory></backupDirectory>
    <filenamePrefix></filenamePrefix>
    <passiveMode></passiveMode>
    <useEPRT></useEPRT>
    <useEPSV></useEPSV>
  </ftpSettings>
  <backupFrequency>
    <frequency></frequency>
    <dayOfWeek></dayOfWeek>
    <hourOfDay></hourOfDay>
    <minuteOfHour></minuteOfHour>
  </backupFrequency>
  <excludeTables>
    <excludeTable></excludeTable>
    <excludeTable></excludeTable>
  </excludeTables>
</backupRestoreSettings>
```

---

where:

- transferProtocol: FTP, SFTP
- frequency: weekly, daily, hourly
- dayOfWeek: SUNDAY, MONDAY, ..., SATURDAY
- Hour of Day: [0 - 24 [
- Minute of hour: [0 - 60 [
- Exclude Tables: AUDIT\_LOG, SYSTEM\_EVENTS, FLOW\_RECORDS  
The tables specified in the excludeTables parameter are not backed up.

If you set up scheduled backups, the output is:

```
<scheduledBackupTaskDetails>
  <nextExecutionTime></nextExecutionTime>
</scheduledBackupTaskDetails>
```

---

You can use the following commands individually to configure a specific setting:

Configure FTP:

```
PUT https://NSX-Manager-IP-Address/1.0/appliance-management/backuprestore/backupsettings/ftpsettings
```

Specify tables that need not be backed up:

```
PUT https://NSX-Manager-IP-Address/1.0/appliance-management/backuprestore/backupsettings/excludedata
```

Set backup schedule:

```
PUT https://NSX-Manager-IP-Address/1.0/appliance-management/backuprestore/backupsettings/schedule
```

Delete backup schedule:

```
DELETE https://NSX-Manager-IP-Address/1.0/appliance-management/backuprestore/backupsettings/schedule
```

## Configure On-Demand Backup

You can take a backup NSX data at any given time.

### Example 3-36. On-demand backup

---

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/appliance-management/backuprestore/backup
```

---

## Query Backup Settings

Retrieves backup settings.

### Example 3-37. Query backup

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/backuprestore/backupsettings
```

Response Body:

```
<backupRestoreSettings>
  <ftpSettings>
    <transferProtocol></transferProtocol>
    <hostNameIPAddress></hostNameIPAddress>
    <port></port>
    <userName></userName><password></password>
    <backupDirectory></backupDirectory>
    <filenamePrefix></filenamePrefix>
    <passiveMode></passiveMode>
    <useEPRT></useEPRT>
    <useEPSV></useEPSV>
  </ftpSettings>
  <backupFrequency>
    <frequency></frequency>
    <dayOfWeek></dayOfWeek>
    <hourOfDay></hourOfDay>
    <minuteOfHour></minuteOfHour>
  </backupFrequency>
  <excludeTables>
    <excludeTable></excludeTable>
    <excludeTable></excludeTable>
  </excludeTables>
```

```
</backupRestoreSettings>
```

---

## Delete Backup Configuration

Deletes Appliance Manager backup configuration.

### Example 3-38. Delete backup settings

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/1.0/appliance-management/backuprestore
/backupsettings
```

---

## Query Available Backups

Retrieves list of all backups available at configured backup location.

### Example 3-39. Query backup

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/backuprestore/backups
```

Response Body:

```
<list>
  <backupFileProperties>
    <fileName></fileName>
    <fileSize></fileSize>
    <creationTime></creationTime>
  </backupFileProperties>
  ...
  <backupFileProperties>
    <fileName></fileName>
    <fileSize></fileSize>
    <creationTime></creationTime>
  </backupFileProperties>
</list>
```

---

## Restore Data

Restores backup from specified file.

### Example 3-40. Restore data

---

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/appliance-management/backuprestore
/restore?restoreFile=filename
```

---

## Working with Tech Support Logs

### Generate Tech Support Logs

Generates a tech support log. The location response header contains the location of the created tech support file. An example for the *componentId* option is NSX

**Example 3-41.** Generate tech support log

Request:

---

```
POST https://NSX-Manager-IP-Address/api/1.0/appliance-management/techsupportlogs
      /componentId
```

---

**Download Tech Support Logs**

Download tech support logs after it has been generated.

**Example 3-42.** Download tech support log

Request:

---

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/techsupportlogs/filename
```

---

**Querying NSX Manager Logs**

You can retrieve NSX Manager system event and audit logs.

**Get NSX Manager System Events**

You can retrieve NSX Manager system events.

**Example 3-43.** Get NSX Manager system events

Request:

---

```
GET https://NSX-Manager-IP-Address/api/2.0/systemevent?startIndex=0&pageSize=10
```

---

Where

- start index is an optional parameter which specifies the starting point for retrieving the logs. If this parameter is not specified, logs are retrieved from the beginning.
- page size is an optional parameter that limits the maximum number of entries returned by the API. The default value for this parameter is 256 and the valid range is 1-1024.

**Get NSX Manager Audit Logs**

You can get NSX Manager audit logs.

**Example 3-44.** Get NSX Manager audit logs

Request:

---

```
GET https://NSX-Manager-IP-Address/api/2.0/auditlog?startIndex=0&pageSize=10
```

---

Where

- start index is an optional parameter which specifies the starting point for retrieving the logs. If this parameter is not specified, logs are retrieved from the beginning.
- page size is an optional parameter that limits the maximum number of entries returned by the API. The default value for this parameter is 256 and the valid range is 1-1024.

## Working with Support Notifications

### Query Notifications

Retrieves all system generated notifications.

---

**Example 3-45.** Query notifications

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/notifications
```

Response Body:

```
<notifications>
  <notification>
    <id></id>
    <notification></notification>
    <notificationStatus></notificationStatus>
  </notification>
</notifications>
```

---

### Delete all Notifications

Deletes all system generated notifications regardless of whether they have been acknowledged.

---

**Example 3-46.** Delete notifications

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/1.0/appliance-management/notifications
```

---

### Acknowledge Notifications

Acknowledges a notification. The notification is then deleted from the system.

---

**Example 3-47.** Acknowledge notification

---

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/appliance-management/notifications
      /NotificationId/acknowledge
```

---



# Upgrading NSX Components

---

To upgrade to NSX 6.2, you must first upgrade the NSX Manager, and then upgrade the other components in the order in which they are documented.

NSX components must be upgraded in the following order:

- 1 NSX Manager
- 2 Controller
- 3 Network virtualization components
- 4 Distributed Firewall
- 5 NSX Edge
- 6 Services

This chapter includes the following topics:

- [“Upgrading NSX Manager”](#) on page 79
- [“Upgrading Controllers”](#) on page 81
- [“Upgrading Network Virtualization Components”](#) on page 82
- [“Upgrading Distributed Firewall”](#) on page 83
- [“Upgrading NSX Edge”](#) on page 83
- [“Upgrading Services”](#) on page 84

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Upgrading NSX Manager

If you are upgrading from vCloud Networking and Security to NSX, all grouping objects from vShield Manager 5.5 are carried over to NSX. Though new objects in NSX can be created only at a global scope, the scope of upgraded objects is maintained, and these objects can be edited. For example, you can nest the following security groups within an upgraded security group at the datacenter scope:

- security groups created at same datacenter scope (via REST only)
- security groups created at portgroup scope, which fall under the datacenter (via REST only)

Security groups created at the global scope cannot be nested under a security group created at a lower scope such as a datacenter.

All users and associated roles are carried over to NSX as well.

In all API calls for upgrading the NSX Manager, the *componentId* parameter can be `NSX` or `NSXAPIGMT`.

## Upload Upgrade Bundle

### Example 4-1. Upload upgrade bundle

---

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/appliance-management/upgrade/uploadbundle
      /componentId
```

---

## Query Upgrade Information

After the upgrade bundle is uploaded, you can query upgrade details such as pre-upgrade validation warning or error messages along with pre-upgrade questions.

### Example 4-2. Query upgrade information

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/upgrade/information
     /componentId
```

Response Body:

```
<upgradeInformation>
  <fromVersion></fromVersion>
  <toVersion></toVersion>
  <upgradeBundleDescription></upgradeBundleDescription>
  <preUpgradeQuestionsAnswers>
    <preUpgradeQuestionAnswer>
      <questionId></questionId>
      <question></question>
      <questionAnserType></questionAnserType>
    </preUpgradeQuestionAnswer>
    ...
    <preUpgradeQuestionAnswer>
      <questionId></questionId>
      <question></question>
      <questionAnserType></questionAnserType>
    </preUpgradeQuestionAnswer>
  </preUpgradeQuestionsAnswers>
  <upgradeStepsDto>
    <step>
      <stepId></stepId>
      <stepLabel></stepLabel>
      <description></description>
    </step>
    ...
    <step>
      <stepId></stepId>
      <stepLabel></stepLabel>
      <description></description>
    </step>
  </upgradeStepsDto>
  <warningMessages></warningMessages>
</upgradeInformation>
```

---

## Begin Upgrade

Starts upgrade process.

### Example 4-3. Start upgrade

---

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/appliance-management/upgrade/start
      /componentId
```

Request Body:

```
<preUpgradeQuestionsAnswers>
  <preUpgradeQuestionAnswer>
    <questionId>preUpgradeChecks1:Q1</questionId>
    <question>Do you want to enable SSH?</question>
    <questionAnserType>YESNO</questionAnserType>
    <answer>YES</answer>
  </preUpgradeQuestionAnswer>
  ...
</preUpgradeQuestionsAnswers>
```

---

## Query Upgrade Status

Retrieves upgrade status.

**Example 4-4.** Query upgrade status

---

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/upgrade/status
     /componentId
```

Response Body:

```
<upgradeStatus>
  <stepStatus>
    <upgradeStep>
      <stepId></stepId>
      <stepLabel></stepLabel>
      <description></description>
    </upgradeStep>
    <status></status>
  </stepStatus>
  <status></status>
  <existingBundleFileName></existingBundleFileName>
</upgradeStatus>
```

---

## Upgrading Controllers

The controllers in your environment are upgraded at the cluster level. It is recommended that you upgrade the controllers during a maintenance window.

Prerequisites

- Ensure that all of the controllers are in the normal state. Upgrading is not possible when the controllers are in the disconnected state.
- Ensure that the cluster has formed a majority (quorum). The upgrade cannot be launched without a majority. A majority is best achieved with an odd number, such as three or five nodes.
- During the upgrade, when there is a temporary non-majority state, existing virtual machines do not lose networking. However, newly created virtual machines might drop traffic. We recommend that you not provision new VMs, move VMs, or allow DRS to move VMs during the upgrade. New network creation is automatically blocked during the upgrade.

## Backup Controllers

You should take a snapshot of your controllers before beginning the upgrade.

**Example 4-5.** Backup controller

Request:

---

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/controller/controllerId/snapshot
```

---

The output of the GET call is an octet stream containing the controller snapshot. To download the snapshot, use the following REST API call:

```
curl -u admin:default -H "Accept: application/octet-stream" -X GET
      -khttps://NSXManagerIPAddress/api/2.0/vdn/controller/controllerID/snapshot
      >controller_backup.snapshot
```

## Query Controller Upgrade Availability

You can run the query command to check if a controller upgrade is available.

**Example 4-6.** Query controller upgrade availability

Request:

---

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/controller/upgrade-available
```

---

## Upgrade Controllers

The upgrade call to start an upgrade on the controller cluster. The upgrade is sequentially done on all the controller nodes. The status is updated on the node and cluster level.

**Example 4-7.** Upgrade controllers

Request:

---

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/controller/cluster/upgrade
```

---

## Query Controller Upgrade Status

You can query the status of a controller upgrade on the cluster level.

**Example 4-8.** Query Controller Upgrade Status

Request:

---

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/controller/cluster/upgrade
```

---

## Upgrading Network Virtualization Components

After NSX Manager is upgraded, previously prepared clusters must have the 6.x network virtualization components installed.

**Example 4-9.** Upgrade network virtualization components

Request:

---

```
PUT https://NSX-Manager-IP-Address/api/2.0/nwfabric/configure
```

Request Body:

```
<nwFabricFeatureConfig>
  <resourceConfig>
    <resourceId>CLUSTER MOID</resourceId>
  </resourceConfig>
```

```
</nwFabricFeatureConfig>
```

---

## Upgrading Distributed Firewall

After upgrading NSX Manager, controllers, and network virtualization components, you must enable Distributed Firewall.

**Example 4-10.** Get current state of firewall functioning after upgrade to NSX

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/state
```

Response Body:

```
<firewallSwitchState>
  <contextId>globalroot-0</contextId>
  <userId>System</userId>
  <timestamp>1416013165873</timestamp>
  <state>forward</state>
</firewallSwitchState>
```

---

Possible values for the state parameter are:

- backwardCompatible: This is the default state after an upgrade from vCloud Networking and Security to NSX, which means that vShield App is being used for protection instead of Distributed Firewall.
- backwardCompatibleReadyForSwitch: Once the clusters are prepared with NSX binaries, this state is enabled. You can switch to Distributed Firewall only after firewall is in this state
- switchingToForward: This is an intermediate state when you change firewall to Distributed Firewall.
- forward: This is the default state for green field deployments or after you have switched from vShield App to Distributed Firewall.
- switchFailed: This state is unlikely, but may be present if NSX Manager failed to switch to Distributed Firewall.

After your environment is ready for NSX, use the following call to enable Distributed Firewall.

**Example 4-11.** Enable Distributed Firewall

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/state
```

---

## Upgrading NSX Edge

Upgrades NSX Edge Services Gateway or Logical Router. The appliances are upgraded and feature configurations are retained and upgraded

**Example 4-12.** Upgrade Edge

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId?action=upgrade
```

---

**IMPORTANT** The location header returns the edgeld of the upgraded NSX Edge. You must use this ID to configure and manage this Edge instance.

If Edge in the previous release was installed using hidden/system resource pool IDs, the UI may show unusual behavior.

## Upgrading Services

Upgrades service to recent version.

### Example 4-13. Query clusters

---

Request:

PUT `https://NSX-Manager-IP-Address/api/2.0/si/deploy/?startTime=time`

Request Body:

```
<clusterDeploymentConfigs>
  <clusterDeploymentConfig>
    <clusterId>clusterId</clusterId>
    <datastore>datastoreId</datastore>
    <services>
      <serviceDeploymentConfig>
        <serviceId>serviceId</serviceId>
        <serviceInstanceId>serviceInstanceId</serviceInstanceId>
        <dvPortGroup>dvpgId</dvPortGroup>
        <ipPool>ipPoolId</ipPool>
      </serviceDeploymentConfig>
    </services>
  </clusterDeploymentConfig>
</clusterDeploymentConfigs>
```

---

The `datastore`, `dvPortGroup`, and `ipPool` variables should either not be specified or have same value as provided at time of deployment.

# User Management

---

In many organizations, networking and security operations are handled by different teams or members. Such organizations may require a way to limit certain operations to specific users. This topic describes the options provided by NSX to configure such access control. NSX also supports Single Sign On (SSO), which enables NSX to authenticate users from other identity services such as Active Directory, NIS, and LDAP.

User management in the vSphere Web Client is separate from user management in the CLI of any NSX component.

The chapter includes the following topics:

- [“Configuring SSO on NSX Manager”](#) on page 85
- [“User Management”](#) on page 86
- [“Role Management”](#) on page 88

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Configuring SSO on NSX Manager

Integrating the single sign on (SSO) service with NSX improves the security of user authentication for vCenter users and enables NSX to authenticate users from other identity services such as AD, NIS, and LDAP.

With SSO, NSX supports Security Assertion Markup Language (SAML) tokens from a trusted source to authenticate REST API calls. NSX Manager can also acquire authentication SAML tokens for use with other VMware solutions.

### Example 5-1. Configure SSO

---

Request:

POST `https://NSX-Manager-IP-Address/api/2.0/services/ssoconfig`

Request Body:

```
<ssoConfig>
  <ssoLookupServiceUrl>https://vc-1-01a.corp.local:443/lookservice/sdk</ssoLookupServiceUrl> <!-- Required. FQDN or IP address of SSO server.-->
  <ssoAdminUsername>administrator@vsphere.local</ssoAdminUsername> <!-- Required. -->
  <ssoAdminUserpassword>vmware</ssoAdminUserpassword> <!-- Required. -->
  <certificateThumbprint>D2:75:61:24:52:CA:B2:8D:D3:25:3F:78:11:2A:8F:94:5A:30:57:0D</certificateThumbprint> <!-- Required. -->
</ssoConfig>
```

## Query SSO Details

### Example 5-2. Get SSO details

---

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/services/ssoconfig`

Response Body:

```
<ssoconfig>
  <ssoLookupServiceUrl>https://vc-1-01a.corp.local:443/lookservice/sdk</ssoLookupServiceUrl>
  <ssoAdminUsername>administrator@vsphere.local</ssoAdminUsername>
</ssoConfig>
```

---

## Query SSO Configuration Status

### Example 5-3. Get SSO configuration status

---

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/services/ssoconfig/status`

Response Body:

```
<boolean></boolean>
```

---

## Delete SSO Configuration

### Example 5-4. Delete SSO configuration

---

Request:

DELETE `https://NSX-Manager-IP-Address/api/2.0/services/ssoconfig/`

---

## User Management

The authentication and authorization APIs include methods to manage users and roles.

### Get Information About a User

You can retrieve information about a user.

#### Example 5-5. Get information about a user

---

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/user/userId`

Request Body:

```
<userInfo>
  <objectId />
  <type>
    <typeName />
  </type>
  <name />
  <revision />
  <objectTypeName />
  <userId />
  <fullname />
```

```

<email />
<isLocal />
<isEnabled />
<isGroup />
<hasGlobalObjectAccess />
<accessControlEntry>
  <role />
  <resource>
    <objectId />
    <type>
      <typeName />
    </type>
    <name />
    <revision />
    <objectTypeName />
    <scope>
      <id />
      <objectTypeName />
      <name />
    </scope>
  </resource>
  ...
</accessControlEntry>
</userInfo>

```

---

User information includes user name, full name, email address, whether local or not, whether enabled, resource objects, roles, and scope.

## Enable or Disable a User Account

You can disable or enable a user account, either local user or vCenter user. When a user account is created, the account is enabled by default.

### Example 5-6. Enable or disable a user account

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/user/userId/enablestate/value
```

---

The *value* can be 0 (zero) to disable the account, or 1 (one) to enable the account.

This API returns “204 No Content” if successful.

## Change NSX Controller Password

You can change the NSX controller password by using the following API call:

### Example 5-7. Change NSX controller password

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/vdn/controller/credential
```

Sample:

```
PUT 203.0.113.23/api/2.0/vdn/controller/credential
<controllerCredential>
  <apiPassword>VMware-12345!</apiPassword>
</controllerCredential>
```

---

If the password change succeeds, the call returns a response code of “200 OK”.

## Remove Role Assignment

The first API removes the NSX role assignment for a vCenter user, without affecting the vCenter account. The second API removes a vCenter user's roles.

---

### Example 5-8. Remove role assignment

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/user/userId
```

---

### Example 5-9. Delete a user role

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/role/userId
```

---

Both APIs return "204 No Content" if successful.

## Role Management

When assigning or retrieving the role for a user, you cannot use a backslash (\) in the user name (`userId` parameter). Instead of specifying `Domain\user1` as the user name, say `user1@Domain`.

### Get Role for a User

You can retrieve information about the role assigned to this user.

---

#### Example 5-10. Get user role

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/role/userId
```

Request Body:

```
<accessControlEntry>
  <role></role>
  <resource>
    <objectId></objectId>
    <type>
      <typeName></typeName>
    </type>
    <name></name>
    <revision></revision>
    <objectTypeName></objectTypeName>
    <scope>
      <id></id>
      <objectTypeName></objectTypeName>
      <name></name>
    </scope>
  </resource>
  <resource>...</resource>
  ...
  ...
</accessControlEntry>
```

---

Possible roles are `super_user`, `vshield_admin`, `enterprise_admin`, `security_admin`, and `auditor`.

## Get Role for a NSX Manager User

You can retrieve information about users who have been assigned a NSX Manager role (local users as well as vCenter users with the NSX Manager role).

### Example 5-11. Get user role

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/users/vsm
```

Response Body:

```
<users>
  <userInfo>
    <objectId />
    <type>
      <typeName />
    </type>
    <name />
    <revision />
    <objectTypeName />
    <userId />
    <fullName />
    <email />
    <isLocal />
    <isEnabled />
    <isGroup>false</isGroup>
    <hasGlobalObjectAccess />
    <accessControlEntry>
      <role />
      <resource>
        <objectId />
        <type>
          <typeName />
        </type>
        <name />
        <revision />
        <objectTypeName />
        <scope>
          <id>group-d1</id>
          <objectTypeName />
          <name />
        </scope>
      </resource>
    </accessControlEntry>
  </userInfo>
  <userInfo>...</userInfo>
</users>
```

---

Possible roles are super\_user, vshield\_admin, enterprise\_admin, security\_admin, and auditor.

## Add Role and Resources for a User

You can add role and accessible resources for the specified user. It affects only vCenter users, not local users.

You cannot use a backslash (\) in the user name (userId parameter). Instead of specifying Domain\user1 as the user name, say user1@Domain.

Set isGroup=true to assign a role to a group and isGroup=false to assign a role to a user.

### Example 5-12. Update user role

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/role/userId??isGroup
      =true/false
```

Request Body:

```
<accessControlEntry>
  <role>new-role</role>
  <resource>
    <resourceId>resource-num</resourceId>
    ...
  </resource>
</accessControlEntry>
```

---

This API returns “204 No Content” if successful.

#### **Example 5-13.** Assign NSX Role to CLI user

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/role/userId?iscli=true>

Request Body:

```
<accessControlEntry>
  <role>new-role</role>
  <resource>
    <resourceId>resource-num</resourceId>
    </resource>
</accessControlEntry>
```

---

Possible roles for local CLI users are security\_admin and auditor.

## Change User Role

You can update the role assignment for a given user. The API returns an output representation specifying a new `<accessControlEntry>` for the user.

#### **Example 5-14.** Change user role

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/role/userId>

Request Body:

```
<accessControlEntry>
  <role>new_role</role>
  <resource>
    <resourceId>resource-num</resourceId>
    ...
  </resource>
</accessControlEntry>
```

---

Possible roles are super\_user, vshield\_admin, enterprise\_admin, security\_admin, and auditor.

## Get List of Possible Roles

You can retrieve the possible roles in NSX Manager.

#### **Example 5-15.** Get possible roles

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/roles>

Response Body:

```
<list>
```

```

    <string></string>
    <string></string>
    ...
</list>

```

---

## Get List of Scoping Objects

You can retrieve a list of objects that can be used to define a user's access scope.

### Example 5-16. Get scoping objects

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/usermgmt/scopingobjects
```

Response Body:

```

<scopingObjects>
  <object>
    <objectId />
    <type>
      <typeName />
    </type>
    <name />
    <revision />
    <objectTypeName />
    <scope>
      <id />
      <objectTypeName />
      <name />
    </scope>
  </object>
  <object>
    <objectId />
    <type>
      <typeName />
    </type>
    <name />
    <revision />
    <objectTypeName />
    <scope>
      <id />
      <objectTypeName />
      <name />
    </scope>
  </object>
  ...
</scopingObjects>

```

---

The scoping objects are usually managed object references or vCenter Server names of datacenters and folders.

## Delete User Role

You can delete the role assignment for the specified vCenter user. Once this role is deleted, the user is removed from NSX Manager.

You cannot delete the role for a local user.

### Example 5-17. Delete role

---

Request:

DELETE <https://NSX-Manager-IP-Address/api/2.0/usermgmt/role/userId>

---

# Grouping Objects

---

The Grouping feature enables you to create custom containers to which you can assign resources.

The chapter includes the following topics:

- [“Working with Security Groups”](#) on page 93
- [“Working with IPsets”](#) on page 100
- [“Working with MACsets”](#) on page 102
- [“Working with Services”](#) on page 104
- [“Working with Service Groups”](#) on page 106
- [“Working with IP Pools”](#) on page 110
- [“Working with Tags”](#) on page 114

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Working with Security Groups

A security group is a collection of assets or grouping objects from your vSphere inventory.

### Create Security Group

You can create a new security group on a global scope. Inheritance is not allowed.

The response of the call has 'Location' header populated with the URI using which the created object can be fetched.

#### Example 6-1. Create new security group

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/bulk/scopeId
```

where *scopeId* is globalroot-0

Request Body:

```
<securitygroup>
  <objectId></objectId>
  <objectTypeName></objectTypeName>
  <vsmUuid></vsmUuid>
  <revision></revision>
  <type>
    <typeName></typeName>
  </type>
  <name></name>
```

```

<scope>
  <id></id>
  <objectTypeName></objectTypeName>
  <name></name>
</scope>
<clientHandle></clientHandle>
<extendedAttributes></extendedAttributes>
<member>
  <objectId></objectId>
  <objectTypeName></objectTypeName>
  <vsmUuid></vsmUuid>
  <revision></revision>
  <type>
    <typeName></typeName>
  </type>
  <name></name>
  <scope>
    <id></id>
    <objectTypeName></objectTypeName>
    <name></name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
</member>
<member>
  ...
</member>
<member>
  ...
</member>
<excludeMember>
  <objectId></objectId>
  <objectTypeName></objectTypeName>
  <vsmUuid></vsmUuid>
  <revision></revision>
  <type>
    <typeName></typeName>
  </type>
  <name></name>
  <scope>
    <id></id>
    <objectTypeName></objectTypeName>
    <name></name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
</excludeMember>
<excludeMember>
  ...
</excludeMember>
<excludeMember>
  ...
</excludeMember>
<dynamicMemberDefinition>
  <dynamicSet>
    <operator></operator>
    <dynamicCriteria>
      <operator></operator>
      <key></key>
      <criteria></criteria>
      <value></value>
    </dynamicCriteria>
    <dynamicCriteria>
      ...
    </dynamicCriteria>
  </dynamicSet>
</dynamicSet>
  ...

```

```

    </dynamicSet>
  </dynamicMemberDefinition>
</securitygroup>

```

---

where `dynamicMemberDefinition` includes the following:

- `dynamicSet` represents a rule set as represented on the UI. There can be multiple dynamic sets inside dynamic member definition.
- `operator`: specifies how to combine the results of two dynamic sets. The operator present in this dynamic set is used to combine the result of the dynamic set(s) evaluated previously with the result of this dynamic set.  
The combining takes place serially. Consider three dynamic sets DS1, DS2 and DS3  
The possible values for this field are "AND" and "OR".
- `dynamicCriteria` defines the actual criteria for the membership. There can be multiple `dynamicCriteria` inside a `dynamicSet`.  
All the `dynamicCriteria` in a `dynamicSet` must have the same operator.
- `key` specifies the object and the attribute on which the condition has to be applied. Eg: "VM.name". The key can be any object attribute that is supported by the `DynamicMember` API.
- `criteria` specifies the condition that has to be applied to the key with respect to the value. Different conditions are defined for different datatypes. For string datatype, the condition can be "=", "!=", "contains", "does not contain", etc. For numerical datatypes, condition can be "=", "!=", "<", etc.
- `value` is a string to which key has to be compared using the criteria.

## Query Security Groups

You can retrieve all the security groups that have been created on a specific scope.

Due to the dynamic nature of security groups, changes to the virtual machine listing of security groups or changes to the services associated with a virtual machine are likely to get reflected a few seconds after the security group change. Hence, there should be a delay of a few seconds between a security group modification and running a GET call on it.

### Example 6-2. Query all security groups on NSX Manager

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/scope/scopeId
```

where *scopeId* is `globalroot-0` or `datacenterId` or `portgroupId` in upgrade use cases

Response Body

```

<list>
  <securitygroup>
    <objectId></objectId>
    <objectTypeName></objectTypeName>
    <nsxmgrUuid></nsxmgrUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name></name>
    <scope>
      <id></id>
      <objectTypeName></objectTypeName>
      <name></name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <member>
      <objectId></objectId>

```

```

    <objectTypeName></objectTypeName>
    <nsxmgrUuid></nsxmgrUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name></name>
    <scope>
      <id></id>
      <objectTypeName></objectTypeName>
      <name></name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </member>

  <member>
    ...
  </member>

  <member>
    ...
  </member>

  <excludeMember>
    <objectId></objectId>
    <objectTypeName></objectTypeName>
    <nsxmgrUuid></nsxmgrUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name></name>
    <scope>
      <id></id>
      <objectTypeName></objectTypeName>
      <name></name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </excludeMember>
  <excludeMember>
    ...
  </excludeMember>
  <excludeMember>
    ...
  </excludeMember>
  <dynamicMemberDefinition>
    <dynamicSet>
      <operator></operator>
      <dynamicCriteria>
        <operator></operator>
        <key></key>
        <criteria></criteria>
        <value></value>
      </dynamicCriteria>
      <dynamicCriteria>
        ....
      </dynamicCriteria>
    </dynamicSet>
    <dynamicSet>
      ....
    </dynamicSet>
  </dynamicMemberDefinition>
</securitygroup>
<securitygroup>
  ....
</securitygroup>

```

```

<securitygroup>
  ....
</securitygroup>
</list>

```

---

The following command retrieves details for the specified security group:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId
```

The following command retrieves all internal security groups on the NSX Manager. Internal security groups are used internally by the system and are not created or managed by end users. You should not modify these.

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/internal/scope/scopeId
```

## Query Applicable Members for a Security Group

You can retrieve a list of valid elements that can be added to a security group. Because security group allows only specific type of container elements to be added, this list helps you determine all possible valid elements that can be added.

**Example 6-3.** Get applicable members for a security group

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/scope/scopeId
/memberTypes
```

where *scopeId* is globalroot-0 or datacenterId or portgroupId in upgrade use cases

Response Body:

```

<list>
  <basicinfo>
    <objectId></objectId>
    <objectTypeName></objectTypeName>
    <nsxmgrUuid></nsxmgrUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name></name>
    <scope>
      <id></id>
      <objectTypeName></objectTypeName>
      <name></name>
    </scope>
    <clienthandle></clienthandle>
    <extendedAttributes></extendedAttributes>
  </basicinfo>
  <basicinfo>
    ...
  </basicinfo>
  <basicinfo>
    ...
  </basicinfo>
</list>

```

---

Note that this API command requires a slash (/) at the end.

Use the following command to retrieve members of a specific type under a scope:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/scope/scopeId/members
/memberType
```

## Query all Members of a Security Group

You can retrieve a list of all members that belong to a security group.

### Example 6-4. Get members of security group

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId
```

Response Body:

```
<securitygroup>
  <objectId>securitygroup-22</objectId>
  <objectTypeName>SecurityGroup</objectTypeName>
  <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
  <revision>2</revision>
  <type>
    <typeName>SecurityGroup</typeName>
  </type>
  <name>SG-WEB2</name>
  <description></description>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>false</inheritanceAllowed>
  <member>
    <objectId>vm-77</objectId>
    <objectTypeName>VirtualMachine</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>42</revision>
    <type>
      <typeName>VirtualMachine</typeName>
    </type>
    <name>web-vm-02</name>
    <scope>
      <id>domain-c26</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>VXLAN-COMPUTE-1</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </member>
</securitygroup>
```

---

## Query Security Group Objects

Retrieves list of entities (IpNodes, MacNodes, VmNodes, or VnicNodes) that belong to a specific security group.

### Example 6-5. Query security group members

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId/translation/virtualmachines
```

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId/translation/ipaddresses
```

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId/translation/macaddresses
```

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId/translation/vnics
```

---

## Query Security Groups that contain a Virtual Machine

Retrieves list of security groups to which the specified virtual machine belongs to.

### Example 6-6. Query Security Groups that contain a Virtual Machine

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/lookup/virtualmachine/virtualMachineId
```

---

## Modify a Security Group

To modify a security group, you must query it first and then modify the output. The modified output can then be specified as the request body.

### Example 6-7. Modify a security group

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/bulk/objectId
```

Request Body:

See [Example 6-1](#).

---

## Delete a Security Group

You can delete an existing security group.

### Example 6-8. Delete a security group

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId?force=true/false
```

---

## Add Member to Security Group

You can add a new member to a security group.

### Example 6-9. Add a member to a security group

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId/members/member-moref?failIfExists=true/false
```

---

No input representation is needed. On success, this request returns 200 HTTP OK.

Default value for failIfExists is 'true'.

If failIfExists=false:

If the member is not already present in the SecurityGroup, the API adds the member to the SecurityGroup.

If the member is already present in the SecurityGroup, the API will be a no-op and will return silently.

If `failIfExists=true`:

If the member is not already present in the SecurityGroup, the API adds the member to the SecurityGroup.

If the member is already present in the SecurityGroup, the API call fails with the below error:

```
ERROR:
<error>
  <details>The object vm-1000 is already present in the system.</details>
  <errorCode>203</errorCode>
  <moduleName>core-services</moduleName>
</error>
```

## Delete Member from Security Group

This API removes a member from a security group.

**Example 6-10.** Delete member from a security group

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/securitygroupId
/members/member-moref?failIfAbsent=true/false
```

---

No input representation is needed. On success, this request returns 200 HTTP OK.

Default value for `failIfExists` is `'true'`

If `failIfAbsent=false`:

If the member is present in the SecurityGroup, the API removes the member from the SecurityGroup.

If the member is not present in the SecurityGroup, the API will be a no-op and will return silently.

If `failIfAbsent=true`:

If the member is present in the SecurityGroup, the API removes the member from the SecurityGroup.

If the member is not present in the SecurityGroup, the API will fail with the below error:

```
ERROR:
<error>
  <details>The requested object : vm-1000 could not be found. Object identifiers are
  case sensitive.</details>
  <errorCode>202</errorCode>
  <moduleName>core-services</moduleName>
</error>
```

## Working with IPsets

You can group a set of IP addresses into an IPSet.

### Create an IPset

All IPsets are created on the global scope.

**Example 6-11.** Create IPset

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/ipset/scopeId
```

where `scopeId` is `globalroot-0` or `datacenterId` in upgrade use cases

Request Body:

```
<ipset>
  <type>
    <typeName>IPSet</typeName>
  </type>
  <description>New IP Set</description>
  <name>IP-Set-1</name>
  <value>10.112.201.8-10.112.201.14</value>
  <inheritanceAllowed>false</inheritanceAllowed>
</ipset>
```

---

In the request body example, a range of IP addresses on the 10.112 net is specified (201.8 to 201.14).

## Query IPsets

You can retrieve all the IPsets.

**Example 6-12.** List IPsets on a scope

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/ipset/scope/scopeId
```

where *scopeId* is globalroot-0 or datacenterId in upgrade use cases

---

## Query Details of an IPset

You can retrieve details about an IPset.

**Example 6-13.** Get details of an IPset

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/ipset/ipsetId
```

---

The *ipsetId* is as returned by listing the IPset on a scope.

## Modify an IPset

You can modify an existing IPset and retrieve details about the modified IPset.

**Example 6-14.** Modify an IPset

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/ipset/ipsetId
```

Request Body:

```
<ipset>
  <objectId>ipset-2</objectId>
  <objectTypeName>IPSet</objectTypeName>
  <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
  <revision>6</revision>  <!-- value incremented by 1-->
  <type>
    <typeName>IPSet</typeName>
  </type>
  <name>dmz_app1_web</name>
  <description></description>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
```

```

    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>false</inheritanceAllowed>
  <value>192.168.201.100/24,192.168.202.100/24,192.168.203.100/24</value> <!-- update
    IP information here separated with comma.-->
</ipset>

```

---

In the above example, the revision parameter is incremented and the value parameter lists the updated IP addresses.

## Delete an IPset

You can delete an IPset. The trailing boolean flag indicates forced or unforced delete. With forced delete, the object is deleted even if used in other places such as firewall rules, causing invalid referrals. For unforced delete, the object is deleted only if it is not used by other configuration; otherwise the delete fails.

### Example 6-15. Delete an IPset

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/ipset/ipsetId?force=true\false
```

---

## Working with MACsets

### Create a MACset on a Scope

You can create a MACset on the specified scope. On success, the API returns a string identifier for the new MACset.

### Example 6-16. Create MACset on a scope

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/macset/scopeId
```

where *scopeId* is globalroot-0 or datacenterId in upgrade use cases

Request Body:

```

<macset>
  <objectId></objectId>
  <type>
    <typeName />
  </type>
  <description>Some description</description>
  <name>TestMACSet1</name>
  <revision>0</revision>
  <objectTypeName></objectTypeName>
  <value>22:33:44:55:66:77,00:11:22:33:44:55,aa:bb:cc:dd:ee:ff</value>
</macset>

```

---

where the value parameter can include a single MAC identifier or a comma separated set of MAC identifiers.

### List MACsets Created on a Scope

You can retrieve all the MACsets that were created on the specified scope.

**Example 6-17.** List MACsets on a scope

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/services/macset/scopeId`where *scopeId* is globalroot-0 or datacenterId in upgrade use cases**Get Details of a MACset**

You can retrieve details about a MACset.

**Example 6-18.** Get details of a MACset

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/services/macset/macsetId`The *macsetId* is as returned by listing the MACset on a scope.**Modify an Existing MACset**

You can modify an existing MACset and retrieve details about the modified MACset.

**Example 6-19.** Modify details of a MACsets

Request:

PUT `https://NSX-Manager-IP-Address/api/2.0/services/macset/macsetId`

Request Body:

```
<macset>
  <objectId></objectId>
  <type>
    <typeName />
  </type>
  <description>Some description</description>
  <name>TestMACSet1</name>
  <revision>2</revision>  <!-- value incremented by 1-->
  <objectTypeName></objectTypeName>
  <value>22:33:44:55:66:77,00:11:22:33:44:55</value>  <!-- update IP information here
    separated with comma.-->
</macset>
```

where:

- the *macsetId* is as returned by listing the MACset on a scope
- the revision parameter is incremented and the value parameter lists the updated IP addresses

**Delete a MACset**

You can delete a MACset. The trailing boolean flag indicates forced or unforced delete. With forced delete, the object is deleted even if used in other places such as firewall rules, causing invalid referrals. For unforced delete, the object is deleted only if it is not used by other configuration; otherwise the delete fails.

**Example 6-20.** Delete a MACset

Request:

---

DELETE <https://NSX-Manager-IP-Address/api/2.0/services/macset/macsetId?force=true>false>

---

## Working with Services

### List Services on a Scope

You can retrieve a list of services that have been created on the scope specified by managed object reference *moref*.

**Example 6-21.** List services on a given scope

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/application/scope/scopeId>

where *scopeId* is globalroot-0 or datacenterId in upgrade use cases

---

A non-existent scope results in a 400 Bad Request error.

### Add Service

You can create a new service on the specified scope.

**Example 6-22.** Add a service

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/services/application/scopeId>

where *scopeId* is globalroot-0 or datacenterId in upgrade use cases

Request Body:

```
<application>
  <objectId></objectId>
  <type>
    <typeName/>
  </type>
  <description>Some description</description>
  <name>TestApplication1</name>
  <revision>0</revision>
  <objectTypeName></objectTypeName>
  <element>
    <applicationProtocol>UDP</applicationProtocol>
    <value>9,22-31,44</value>
  </element>
</application>
```

---

For applicationProtocol, possible values are:

- TCP
- UDP
- ORACLE\_TNS
- FTP
- SUN\_RPC\_TCP
- SUN\_RPC\_UDP
- MS\_RPC\_TCP
- MS\_RPC\_UDP
- NBNS\_BROADCAST

- NBDG\_BROADCAST

Only TCP and UDP support comma separated port numbers and dash separated port ranges. Other protocols support a single port number only.

On success, this call returns a string identifier for the newly created application, for instance Application-1. The location header in the reply contains the relative path of the created Application and can be used for further GET, PUT, and DELETE calls.

## Get Details of a Service

You can retrieve details about the service specified by *applicationgroupId* as returned by the call shown in [Example 6-22](#).

### Example 6-23. Retrieve details about a service

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/application/applicationId
```

Response Body:

```
<application>
  <objectId>application-45</objectId>
  <type>
    <typeName>Application</typeName>
  </type>
  <name>TestApplication1</name>
  <revision>1</revision>
  <objectTypeName>Application</objectTypeName>
  <scope>
    <id>datacenter-2</id>
    <objectTypeName>Datacenter</objectTypeName>
    <name>Amo1DC</name>
  </scope>
  <inheritanceAllowed>false</inheritanceAllowed>
  <element>
    <applicationProtocol>UDP</applicationProtocol>
    <value>9,22-31,44</value>
  </element>
</application>
```

---

A non-existent application ID results in a 404 Not Found error.

## Modify Service

You can modify the name, description, applicationProtocol, or port value of a service.

### Example 6-24. Modify service

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/application/applicationId
```

Request Body:

```
<application>
  <objectId>Application-1</objectId>
  <type>
    <typeName>Application</typeName>
  </type>
  <description>Some description</description>
  <name>TestApplication</name>
  <revision>3</revision>    <!-- value incremented by 1-->
  <objectTypeName>Application</objectTypeName>
  <element>
```

```

    <applicationProtocol>TCP</applicationProtocol>
    <value>10,29-30,45</value>
  </element>
</application>

```

---

The call returns XML describing the modified service.

### Delete Service

You can delete a service by specifying its <applicationgroup-id>. The force= flag indicates if the delete should be forced or unforced. For forced deletes, the object is deleted irrespective of its use in other places such as firewall rules, which invalidates other configurations referring to the deleted object. For unforced deletes, the object is deleted only if it is not being used by any other configuration. The default is unforced (false).

#### Example 6-25. Delete service

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/application/applicationId
```

---

## Working with Service Groups

### Add Service Group

You can create a new service group on the specified scope.

#### Example 6-26. Add a service group

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/scopeId
```

where *scopeId* is globalroot-0 or datacenterId in upgrade use cases

Request Body:

```

<applicationGroup>
  <description>Some description</description>
  <name>TestApplication1</name>
  <revision>0</revision>
  <inheritanceAllowed>false</inheritanceAllowed>
</applicationGroup>

```

---

### Query Service Groups

You can retrieve a list of service groups that have been created on the scope specified by managed object reference *moref*.

#### Example 6-27. List service groups on a given scope

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/scope/scopeId
```

where *scopeId* is globalroot-0 or datacenterId in upgrade use cases

Response Body:

```

<list>
  <applicationGroup>
    <objectId>applicationgroup-1</objectId>
    <type>

```

```

        <typeName>ApplicationGroup</typeName>
      </type>
    <name>testglobalAG</name>
    <description></description>
    <revision>2</revision>
    <objectTypeName>ApplicationGroup</objectTypeName>
    <scope>
      <id>globalroot-0</id>
      <objectTypeName>GlobalRoot</objectTypeName>
      <name>Global</name>
    </scope>
    <extendedAttributes></extendedAttributes>
    <inheritanceAllowed>false</inheritanceAllowed>
    <member>
      <objectId>application-37</objectId>
      <type>
        <typeName>Application</typeName>
      </type>
      <name>SMTP</name>
      <revision>3</revision>
      <objectTypeName>Application</objectTypeName>
      <scope>
        <id>globalroot-0</id>
        <objectTypeName>GlobalRoot</objectTypeName>
        <name>Global</name>
      </scope>
      <extendedAttributes></extendedAttributes>
    </member>
  </applicationGroup>
</list>

```

---

A non-existent scope results in a 400 Bad Request error.

## Query Details of a Service Group

You can retrieve details about the service group specified by *applicationgroupId* as returned by the call shown in [Example 6-22](#).

**Example 6-28.** Retrieve details about a service group

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/applicationgroupId
```

---

A non-existent application ID results in a 404 Not Found error.

## Modify Service Group Details

You can modify the name, description, applicationProtocol, or port value of a service group.

**Example 6-29.** Modify service group

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/applicationgroupId
```

Request Body:

```

<applicationGroup>
  <objectId>applicationgroup-1</objectId>
  <type>
    <typeName>ApplicationGroup</typeName>
  </type>
  <name>testglobalAG-updated</name>
  <description>updated with description</description>

```

```

<revision>3</revision> <!-- value incremented by 1-->
<objectTypeName>ApplicationGroup</objectTypeName>
<scope>
  <id>globalroot-0</id>
  <objectTypeName>GlobalRoot</objectTypeName>
  <name>Global</name>
</scope>
<extendedAttributes></extendedAttributes>
<inheritanceAllowed>false</inheritanceAllowed>
<member>
  <objectId>application-37</objectId>
  <type>
    <typeName>Application</typeName>
  </type>
  <name>SMTP</name>
  <revision>3</revision>
  <objectTypeName>Application</objectTypeName>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <extendedAttributes></extendedAttributes>
</member>
</applicationGroup>

```

---

The call returns XML describing the modified service.

## Delete Service Group from Scope

You can delete a service group by specifying its `<applicationgroup-id>`. The `force=` flag indicates if the delete should be forced or unforced. For forced deletes, the object is deleted irrespective of its use in other places such as firewall rules, which invalidates other configurations referring to the deleted object. For unforced deletes, the object is deleted only if it is not being used by any other configuration. The default is unforced (false).

### Example 6-30. Delete service group

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup
/applicationgroupId
```

---

## Working with the Members of a Service Group

### Query Service Group Members

You can get a list of member elements that can be added to the service groups created on a particular scope. Since service group allows only either services or other service groups as members to be added, this helps you get a list of all possible valid elements that can be added to the service.

### Example 6-31. Retrieve member elements

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/scope/scopeId/members
```

where `scopeId` is `globalroot-0` or `datacenterId` in upgrade use cases

Response Body:

```

<list>
  <basicinfo>
    <objectId>applicationgroup-3</objectId>

```

```

<type>
  <typeName>ApplicationGroup</typeName>
</type>
<name>AGDC-1</name>
<description>AG created in DC</description>
<revision>1</revision>
<objectTypeName>ApplicationGroup</objectTypeName>
<scope>
  <id>datacenter-2</id>
  <objectTypeName>Datacenter</objectTypeName>
  <name>Datacenter</name>
</scope>
  <extendedAttributes></extendedAttributes>
</basicinfo>
<basicinfo>
  <objectId>application-36</objectId>
  <type>
    <typeName>Application</typeName>
  </type>
  <name>ORACLE_TNS</name>
  <revision>2</revision>
  <objectTypeName>Application</objectTypeName>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <extendedAttributes></extendedAttributes>
</basicinfo>
<basicinfo>
  <objectId>application-37</objectId>
  <type>
    <typeName>Application</typeName>
  </type>
  <name>SMTP</name>
  <revision>3</revision>
  <objectTypeName>Application</objectTypeName>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <extendedAttributes></extendedAttributes>
</basicinfo>
</list>

```

---

### Add a Member to the Service Group

You can add a member to the service group.

#### Example 6-32. Add member

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/applicationgroupId/members/member-moref
```

---

### Delete a Member from the Service Group

You can delete a member from the service group.

#### Example 6-33. Delete member

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup
        /applicationgroupId/members/member-moref
```

---

## Working with IP Pools

You can create a pool of IP addresses.

### Add an IP Pool

#### Example 6-34. Add IP pool

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/ipam/pools/scope/scopeId
```

where *scopeId* is globalroot-0 or datacenterId in upgrade use cases

Request Body:

```
<ipamAddressPool>
  <name>rest-ip-pool-1</name>
  <prefixLength>23</prefixLength>
  <gateway>192.168.1.1</gateway>
  <dnsSuffix>eng.vmware.com</dnsSuffix>
  <dnsServer1>10.112.0.1</dnsServer1>
  <dnsServer2>10.112.0.2</dnsServer2>
  <ipRanges>
    <ipRangeDto>
      <startAddress>192.168.1.2</startAddress>
      <endAddress>192.168.1.3</endAddress>
    </ipRangeDto>
  </ipRanges>
</ipamAddressPool>
```

---

### Query IP Pool Details

Retrieves details about the specified IP pool.

#### Example 6-35. Query IP Pool

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/ipam/pools/poolId
```

Response Body:

```
<ipamAddressPool>
  <objectId>ipaddresspool-1</objectId>
  <objectTypeName>IpAddressPool</objectTypeName>
  <vsmUuid>4237BA90-C373-A71A-9827-1673BFA29498</vsmUuid>
  <revision>1</revision>
  <type>
    <typeName>IpAddressPool</typeName>
  </type>
  <name>rest-ip-pool-1</name>
  <extendedAttributes></extendedAttributes>
  <prefixLength>23</prefixLength>
  <gateway>192.168.1.1</gateway>
  <dnsSuffix>eng.vmware.com</dnsSuffix>
  <dnsServer1>10.112.0.1</dnsServer1>
  <dnsServer2>10.112.0.2</dnsServer2>
  <ipRanges>
    <ipRangeDto>
      <id>iprange-1</id>
```

```

        <startAddress>192.168.1.2</startAddress>
        <endAddress>192.168.1.3</endAddress>
    </ipRangeDto>
</ipRanges>
<totalAddressCount>2</totalAddressCount>
<usedAddressCount>0</usedAddressCount>
<usedPercentage>0</usedPercentage>
</ipamAddressPool>

```

---

## Modify an IP Pool

To modify an IP pool, query the IP pool first. Then modify the output and send it back as the request body.

### Example 6-36. Query IP Pool

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/ipam/pools/poolId
```

Request Body:

```

<ipamAddressPool>
  <objectId>ipaddresspool-1</objectId>
  <objectTypeName>IpAddressPool</objectTypeName>
  <vsmUuid>4237BA90-C373-A71A-9827-1673BFA29498</vsmUuid>
  <revision>2</revision>  <!-- value incremented by 1-->
  <type>
    <typeName>IpAddressPool</typeName>
  </type>
  <name>rest-ip-pool-1</name>
  <extendedAttributes></extendedAttributes>
  <prefixLength>23</prefixLength>
  <gateway>192.168.1.1</gateway>
  <dnsSuffix>eng.vmware.com</dnsSuffix>
  <dnsServer1>10.112.0.1</dnsServer1>
  <dnsServer2>10.112.0.2</dnsServer2>
  <ipRanges>
    <ipRangeDto>
      <id>iprange-1</id>
      <startAddress>192.168.1.2</startAddress>
      <endAddress>192.168.1.3</endAddress>
    </ipRangeDto>
  </ipRanges>
</ipamAddressPool>

```

---

## Allocating a New IP Address

Allocates a new IP address from the specified pool.

### Example 6-37. Allocate new address

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/ipam/pools/poolId/ipaddresses
```

Request Body:

```

<ipAddressRequest>
  <allocationMode>ALLOCATE</allocationMode>
</ipAddressRequest>

```

Response Body:

```

<allocatedIpAddress>
  <id>allocatedipaddress-1</id>
  <ipAddress>192.168.1.2</ipAddress>

```

```

<gateway>192.168.1.1</gateway>
<prefixLength>23</prefixLength>
<dnsServer1>10.112.0.1</dnsServer1>
<dnsServer2>10.112.0.2</dnsServer2>
<dnsSuffix>eng.vmware.com</dnsSuffix>
<allocationNote/>sample note</allocationNote>
</allocatedIpAddress>

```

---

## Allocating a Specific IP Address

Allocates a specific IP address from the specified pool.

### Example 6-38. Allocate new address

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/ipam/pools/poolId/ipaddresses
```

Request Body:

```

<ipAddressRequest>
  <allocationMode>RESERVE</allocationMode>
  <ipAddress>192.168.1.5</ipAddress>
</ipAddressRequest>

```

Response Body:

See [Example 6-37](#).

---

## Query all IP Pools on Scope

Retrieves all IP pools on the specified scope where the *scopeID* is the reference to the desired scope. An example of the *scopeID* is globalroot-0.

### Example 6-39. Query IP pools on scope

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/ipam/pools/scope/scopeID
```

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<ipamAddressPools>
  <ipamAddressPool>
    <objectId>ipaddresspool-1</objectId>
    <objectTypeName>IpAddressPool</objectTypeName>
    <vsmUuid>42005992-319C-F762-6E8F-A0E6FBF8C9EB</vsmUuid>
    <nodeId>91efadce-09ea-4628-a23c-c2e9b6ab2b5e</nodeId>
    <revision>1</revision>
    <type>
      <typeName>IpAddressPool</typeName>
    </type>
    <name>Controller IP Pool</name>
    <scope>
      <id>globalroot-0</id>
      <objectTypeName>GlobalRoot</objectTypeName>
      <name>Global</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes/>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
    <totalAddressCount>5</totalAddressCount>
    <usedAddressCount>3</usedAddressCount>
  </ipamAddressPool>
</ipamAddressPools>

```

```

    <usedPercentage>60</usedPercentage>
    <prefixLength>24</prefixLength>
    <gateway>192.168.110.1</gateway>
    <dnsSuffix>corp.local</dnsSuffix>
    <dnsServer1>192.168.110.10</dnsServer1>
    <dnsServer2></dnsServer2>
    <ipPoolType>ipv4</ipPoolType>
    <ipRanges>
      <ipRangeDto>
        <id>iprange-1</id>
        <startAddress>192.168.110.201</startAddress>
        <endAddress>192.168.110.205</endAddress>
      </ipRangeDto>
    </ipRanges>
    <subnetId>subnet-1</subnetId>
  </ipamAddressPool>
</ipamAddressPools>

```

---

## Query Allocated IP Addresses

Retrieves all allocated IP addresses from the specified pool.

### Example 6-40. Query allocated addresses

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/ipam/pools/poolId/ipaddresses
```

Response Body:

```

<allocatedIpAddresses>
  <allocatedIpAddress>
    <id>allocatedipaddress-4</id>
    <ipAddress>192.168.1.2</ipAddress>
    <gateway>192.168.1.1</gateway>
    <prefixLength>23</prefixLength>
    <dnsServer1>10.112.0.1</dnsServer1>
    <dnsServer2>10.112.0.2</dnsServer2>
    <dnsSuffix>eng.vmware.com</dnsSuffix>
    <allocationNote>sample note</allocationNote>
  </allocatedIpAddress>
  <allocatedIpAddress>
    <id>allocatedipaddress-5</id>
    <ipAddress>192.168.1.3</ipAddress>
    <gateway>192.168.1.1</gateway>
    <prefixLength>23</prefixLength>
    <dnsServer1>10.112.0.1</dnsServer1>
    <dnsServer2>10.112.0.2</dnsServer2>
    <dnsSuffix>eng.vmware.com</dnsSuffix>
    <allocationNote>sample note</allocationNote>
  </allocatedIpAddress>
</allocatedIpAddresses>

```

---

## Release an IP Address

### Example 6-41. Release IP address

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/ipam/pools/poolId/ipaddresses/allocated-ip-address
```

---

## Delete an IP Pool

### Example 6-42. Delete IP Pool

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/ipam/pools/poolId
```

---

## Working with Tags

You can view security tags applied on a virtual machine or create a user defined security tag.

### Create Security Tag

Creates a new security tag.

#### Example 6-43. Create tag

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/securitytags/tag
```

Request Body:

```
<securityTag>
  <objectTypeName>SecurityTag</objectTypeName>
  <type>
    <typeName>SecurityTag</typeName>
  </type>
  <name>TAG_NAME</name>
  <description>description of the tag</description>
  <extendedAttributes></extendedAttributes>
</securityTag>
```

---

### Apply Tag to Virtual Machine

Applies security tag to virtual machine.

#### Example 6-44. Apply tag

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/securitytags/tag/TagIdentifierString
/vm/vmMoid
```

---

### Query Security Tags

Retrieves security tags.

#### Example 6-45. Query tag

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitytags/tag
```

Response Body:

```
<securityTags>
  <securityTag>
    <objectId>tag-id</objectId>
    <objectTypeName>SecurityTag</objectTypeName>
    <type>
```

```

    <typeName>SecurityTag</typeName>
  </type>
  <name>TAG_NAME</name>
  <description>tag description</description>
  <extendedAttributes></extendedAttributes>
</securityTag>
</securityTags>

```

---

## Query Virtual Machines Assigned to Tag

Retrieves the list of virtual machines that have the specified tag attached to them.

### Example 6-46. Query virtual machines for tag

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitytags/tag/TagIdentifierString/vm
```

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<basicinfo>
  <basicinfo>
    <objectId>vm-240</objectId>
    <objectTypeName>VirtualMachine</objectTypeName>
    <vsmUuid>4203042F-4BCB-83E8-40E7-1EA251FF7EE6</vsmUuid>
    <revision>10</revision>
    <type>
      <typeName>VirtualMachine</typeName>
    </type>
    <name>win 7</name>
    <scope>
      <id>domain-c9</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>Cluster2</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </basicinfo>
</basicinfo>

```

---

## Detach Tag from Virtual Machine

Detaches security tag from virtual machine.

### Example 6-47. Detach tag

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/securitytags/tag/TagIdentifierString/vm/vmMoid
```

---

## Delete Tag

Deletes the tag.

### Example 6-48. Delete tag

---

Request:

DELETE <https://NSX-Manager-IP-Address/api/2.0/services/securitytags/tag/TagIdentifierString>

---

## Working with Logical Switches

---

A cloud deployment or a virtual data center has a variety of applications across multiple tenants. These applications and tenants require isolation from each other for security, fault isolation, and avoiding overlapping IP addressing issues. The NSX logical switch creates logical broadcast domains or segments to which an application or tenant virtual machine can be logically wired. This allows for flexibility and speed of deployment while still providing all the characteristics of a physical network's broadcast domains (VLANs) without physical Layer 2 sprawl or spanning tree issues.

A logical switch is distributed and can span arbitrarily large compute clusters. This allows for virtual machine mobility (vMotion) within the datacenter without limitations of the physical Layer 2 (VLAN) boundary. The physical infrastructure does not have to deal with MAC/FIB table limits since the logical switch contains the broadcast domain in software.

A logical switch is mapped to a unique VXLAN, which encapsulates the virtual machine traffic and carries it over the physical IP network.

The NSX controller is the central control point for all logical switches within a network and maintains information of all virtual machines, hosts, logical switches, and VXLANs. The controller supports two new logical switch control plane modes, Unicast and Hybrid. These modes decouple NSX from the physical network. VXLANs no longer require the physical network to support multicast in order to handle the Broadcast, Unknown unicast, and Multicast (BUM) traffic within a logical switch. The unicast mode replicates all the BUM traffic locally on the host and requires no physical network configuration. In the hybrid mode, some of the BUM traffic replication is offloaded to the first hop physical switch to achieve better performance. This mode requires IGMP snooping to be turned on the first hop physical switch. Virtual machines within a logical switch can use and send any type of traffic including IPv6 and multicast.

You must be a Security Administrator in order to create VXLAN networks.

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

This chapter includes the following topics:

- [“Preparing for Logical Switches”](#) on page 118
- [“Configuring Switches”](#) on page 118
- [“Working with Segment IDs”](#) on page 120
- [“Working with Multicast Address Ranges”](#) on page 121
- [“Working with Transport Zones”](#) on page 123
- [“Working with Logical Switches”](#) on page 126
- [“Managing the Logical Switch UDP Port”](#) on page 130
- [“Querying Allocated Resources”](#) on page 130
- [“Testing Multicast Group Connectivity”](#) on page 131

- [“Performing Ping Test”](#) on page 132

## Preparing for Logical Switches

Before creating a logical switch, ensure that:

- you have installed the network virtualization components on the appropriate clusters
- you have configured VXLAN on the appropriate clusters

## Configuring Switches

You must prepare each vDS by specifying the VLAN for your L2 domain and the MTU for each vDS.

### Prepare Switch

The MTU is the maximum amount of data that can be transmitted in one packet before it is divided into smaller packets. The frames are slightly larger in size because of the traffic encapsulation, so the MTU required is higher than the standard MTU. You must set the MTU for each switch to 1600 or higher.

#### Example 7-1. Prepare switch

---

Request:

POST `https://NSX-Manager-IP-Address/api/2.0/vdn/switches`

Request Body:

```
<vdsContext>
  <switch>
    <objectId>dvs-26</objectId>
    <type><typeName>DistributedVirtualSwitch</typeName></type>
    <name></name>
    <revision>0</revision>
    <objectTypeName>DistributedVirtualSwitch</objectTypeName>
  </switch>
  <teaming>ETHER_CHANNEL</teaming>
  <mtu>mtu-value</mtu>
</vdsContext>
```

---

### Query Configured Switches

You can retrieve all configured switches.

#### Example 7-2. Get all configured switches

---

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/vdn/switches`

Response Body:

```
<vdsContexts>
  <vdsContext>
    <switch>
      <objectId>dvs-26</objectId>
      <type>
        <typeName>DistributedVirtualSwitch</typeName>
      </type>
      <name />
      <revision>0</revision>
      <objectTypeName>DistributedVirtualSwitch</objectTypeName>
    </switch>
    <teaming>LACP_PASSIVE</teaming>
    <mtu>mtu-value</mtu>
```

```

    </vdsContext>
    ...
    <vdsContext>...</vdsContext>
    ...
</vdsContexts>

```

---

## Query Configured Switches on Datacenter

You can retrieve all configured switches on a datacenter.

### Example 7-3. Get configured switches on a datacenter

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/switches/datacenter/datacenterId
```

Response Body:

```

<vdsContexts>
  <vdsContext>
    <switch>
      <objectId>dvs-26</objectId>
      <type>
        <typeName>DistributedVirtualSwitch</typeName>
      </type>
      <name />
      <revision>0</revision>
      <objectTypeName>DistributedVirtualSwitch</objectTypeName>
    </switch>
    <teaming>LACP_PASSIVE</teaming>
    <mtu>mtu-value</mtu>
  </vdsContext>
  <vdsContext>...</vdsContext>
  ...
</vdsContexts>

```

---

## Query Specific Switch

You can retrieve a specific switch by specifying its ID.

### Example 7-4. Get specific switch

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/switches/switchId
```

Response Body:

```

<vdsContext>
  <switch>
    <objectId>dvs-26</objectId>
    <type>
      <typeName>DistributedVirtualSwitch</typeName>
    </type>
    <name />
    <revision>0</revision>
    <objectTypeName>DistributedVirtualSwitch</objectTypeName>
  </switch>
  <teaming>LACP_PASSIVE</teaming>
  <mtu>mtu-value</mtu>
</vdsContext>

```

---

## Delete Switch

You can delete a switch.

### Example 7-5. Delete switch

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/switches/switchId
```

---

## Working with Segment IDs

You can specify one or more segment ID pools that is used to provide virtual network identifiers to logical switches which helps you isolate your network traffic.

### Add a new Segment ID Range

You can add a new segment ID range that provides virtual network identifiers to logical switches. More than one segment ID range is supported in the system.

#### Example 7-6. Add a segment ID range

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments
```

Request Body:

```
<segmentRange>
  <name>segment-1</name>
  <desc>Segment 1</desc>
  <begin>5000</begin>
  <end>65535</end>
</segmentRange>
```

---

The segment range is inclusive – the beginning and ending IDs are included.

### Query all Segment ID Ranges

You can retrieve all segment ID ranges.

#### Example 7-7. Get all Segment ID Ranges

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments
```

Response Body:

```
<segmentRanges>
  <segmentRange>
    <id>1</id>
    <name>segment-1</name>
    <desc>Segment 1</desc>
    <begin>5000</begin>
    <end>65535</end>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
  </segmentRange>
</segmentRanges>
```

---

## Query a Specific Segment ID Range

You can retrieve a segment ID range by specifying the segment ID.

### Example 7-8. Get a specific Segment ID Range

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments/segmentId
```

Response Body:

```
<segmentRange>
  <id>1</id>
  <name>segment-1</name>
  <desc>Segment 1</desc>
  <begin>5000</begin>
  <end>65535</end>
</segmentRange>
```

---

## Update a Segment ID Range

You can update the name, description, or end of a segment ID range.

### Example 7-9. Update a Segment ID Range

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments/segmentId
```

Request Body:

```
<segmentRange>
  <end>3000</end>
  <name></name>
  <desc></desc>
</segmentRang>
```

---

## Delete a Segment ID Range

You can delete a segment ID range.

### Example 7-10. Delete a Segment ID Range

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments/segmentId
```

---

## Working with Multicast Address Ranges

Specifying a multicast address range helps in spreading traffic across your network to avoid overloading a single multicast address. A virtualized network-ready host is assigned an IP address from this range.

### Add a new Multicast Address Range

You can add a new multicast address range.

#### Example 7-11. Add a multicast address range

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/config/multicasts
```

Request Body:

```
<multicastRange>
  <name>name</name>
  <desc>description</desc>
  <begin>239.1.1.1</begin>
  <end>239.3.3.3</end>
</multicastRange>
```

---

The address range is inclusive – the beginning and ending addresses are included.

## Query all Multicast Address Ranges

You can retrieve all multicast address ranges.

**Example 7-12.** Get all multicast ranges

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/config/multicasts
```

Response Body:

```
<multicastRanges>
  <multicastRange>
    <id>1</id>
    <name></name>
    <desc></desc>
    <begin>239.1.1.1</begin>
    <end>239.3.3.3</end>
  </multicastRange>
  <multicastRange>
    ...
  </multicastRange>
  ...
</multicastRanges>
```

---

## Get a Specific Multicast Address Range

You can retrieve a specific multicast address range.

**Example 7-13.** Get a multicast range

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/config/multicasts/multicastAddressRangeId
```

Response Body:

```
<multicastRange>
  <id>1</id>
  <name></name>
  <desc></desc>
  <begin>239.1.1.1</begin>
  <end>239.3.3.3</end>
</multicastRange>
```

---

## Update a Multicast Address Range

You can update the name, description, or end address of a multicast address range.

**Example 7-14.** Update a multicast range

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/vdn/config/multicasts/multicastAddressRangeId>

Request Body:

```
<segmentRange>
  <end>3000</end>
  <name></name>
  <desc></desc>
</segmentRange>
```

**Delete a Multicast Address Range**

You can delete a multicast address range.

**Example 7-15.** Delete multicast address range

Request:

DELETE <https://NSX-Manager-IP-Address/api/2.0/vdn/config/multicasts/multicastAddressRangeId>**Working with Transport Zones**

A transport zone is the networking infrastructure within provider virtual datacenters.

**Create a Transport Zone**

You must specify the clusters that are to be part of the transport zone. You must have the VLAN ID, UUID of the vCenter Server, and vDS ID.

**Example 7-16.** Create a transport zone

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/vdn/scopes>

Request Body:

```
<vdnScope>
  <name>tz-1</name> <!-- Required. -->
  <description>Transport Zone 1</description> <!-- Optional. -->
  <clusters>
    <cluster>
      <cluster>
        <objectId>domain-c11</objectId> <!-- One or more is required. -->
      </cluster>
    </cluster>
  </clusters>
  <controlPlaneMode>UNICAST_MODE</controlPlaneMode> <!-- Optional. -->
</vdnScope>
```

**Edit a Transport Zone**

You can add a cluster to or delete a cluster from a transport zone.

**Example 7-17.** Edit a transport zone

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId?action=patch>

Request Body:

```
<vdnScope>
  <objectId>id</objectId>
  <clusters>
    <cluster>
      <objectId>domain-c59</objectId>
    </cluster>
  </clusters>
</vdnScope>
```

---

#### Example 7-18. Recreate DVS portgroup on Logical Switch

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId?action=repair>

---

For every logical switch created, NSX will create a corresponding DVS portgroup in VCenter. If the portgroup is lost for any reason, the NSX logical switch will stop functioning. one-to-one mapping of Logical Switches map to DVS portgroups on a 1-to-1 basis, and the API is used to recreate any portgroups that have gone missing.

## Update Attributes on a Transport Zone

You can update the attributes of a transport zone.

#### Example 7-19. Update attributes of a transport zone

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId/attributes>

Request Body:

```
<vdnScope>
  <objectId>vdnScope-1</objectId>
  <name>name</name>
  <description>description</description>
</vdnScope>
```

---

## Query existing Transport Zones

You can retrieve all existing transport zones.

#### Example 7-20. Get all transport zones

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/vdn/scopes>

Response Body:

```
<vdnScopes>
  <vdnScope>
    <objectId>vdnscope-2</objectId>
    <type>
      <typeName>VdnScope</typeName>
    </type>
    <name></name>
    <description></description>
    <revision>0</revision>
```

```

<objectTypeName>VdnScope</objectTypeName>
<extendedAttributes></extendedAttributes>
<id>vdnscope-2</id>
<clusters>
  <cluster>
    <objectId>domain-c124</objectId>
    <type>
      <typeName>ClusterComputerResource</typeName>
    </type>
    <name>vxlan-cluster</name>
    <scope>
      <id>datacenter-2</id>
      <objectTypeName>Datacenter</objectTypeName>
      <name>dc1</name>
    </scope>
    <extendedAttributes></extendedAttributes>
  </cluster>
  ...
</clusters>
<virtualWireCount>10</virtualWireCount>
</VdnScope>
...
<VdnScope>...</VdnScope>
...
</VdnScopes>

```

---

## Query a Specific Transport Zone

You can retrieve a specific transport zone.

### Example 7-21. Get a transport zone

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId
```

Response Body:

```

<VdnScope>
  <objectId>vdnscope-2</objectId>
  <type>
    <typeName>VdnScope</typeName>
  </type>
  <name></name>
  <description></description>
  <revision>0</revision>
  <objectTypeName>VdnScope</objectTypeName>
  <extendedAttributes></extendedAttributes>
  <id>vdnscope-2</id>
  <clusters>
    <cluster>
      <objectId>domain-c124</objectId>
      <type>
        <typeName>ClusterComputerResource</typeName>
      </type>
      <name>vxlan-cluster</name>
      <scope>
        <id>datacenter-2</id>
        <objectTypeName>Datacenter</objectTypeName>
        <name>dc1</name>
      </scope>
      <extendedAttributes></extendedAttributes>
    </cluster>
    ...
  </clusters>
  <virtualWireCount>10</virtualWireCount>

```

---

```
</vdsScope>
```

---

## Delete a Transport Zone

You can delete a transport zone.

### Example 7-22. Delete transport zone

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId
```

---

## Working with Logical Switches

A logical switch is a collection of vDS port groups across multiple virtual distributed switches (vDS) within a transport zone.

### Create a Logical Switch

You can create a new logical switch on the specified transport zone. You must have defined a segment ID range and a multicast address range before creating a logical switch.

The default value of the controlPlaneMode parameter is the value specified for the transport zone.

### Example 7-23. Create a logical switch

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId/virtualwires
```

Request Body:

```
<virtualWireCreateSpec>
  <name>LS_vlan_tagging</name>
  <description>For guest VLAN tagging</description>
  <tenantId>virtual wire tenant</tenantId>
  <controlPlaneMode>UNICAST_MODE</controlPlaneMode>  <!-- optional. Default is the
    value specified for the transport zone. -->
  <guestVlanAllowed>true</guestVlanAllowed>
</virtualWireCreateSpec>
```

---

## Attach or Detach a Virtual Machine from a Logical Switch

You can attach or detach a virtual machine from a logical switch. To detach a virtual machine, leave the portgroupId tag empty.

### Example 7-24. Attach or detach a virtual machine

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/virtualwires/vm/vnic
```

Request Body:

```
<com.vmware.vshield.vsm.inventory.dto.VnicDto>
  <objectId>420388f4-1ce6-ef26-6643-fc6e8df99f2e.000</objectId>
  <vnicUuid>420388f4-1ce6-ef26-6643-fc6e8df99f2e.000</vnicUuid>
  <portgroupId>virtualwire-1</portgroupId>
</com.vmware.vshield.vsm.inventory.dto.VnicDto>
```

---

where the vnic uuid is generated in the following way:

vnicUUID = vm.instanceUUID + '.' + all chars except the first of the corresponding vnic virtualdevice id. The virtual device id can be retrieved from:

<https://API/mob/?moid=vm-43&doPath=config%2ehardware%2edevice%5b4000%5d>

## Query all Logical Switches on a Transport Zone

You can retrieve all logical switches on the specified transport zone.

### Example 7-25. Get all logical switches

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId/virtualwires?pagesize
=XX&startindex=YY
```

Response Body:

```
<virtualwires>
  <sortedDataPage>
    <datapart class="virtualWire">
      <objectId>virtualwire-1</objectId>
      <name>vWire1</name>
      <description>logical switch 1</description>
      <tenantId>logical switch tenant</tenantId>
      <revision>0</revision>
      <vdnScopeId>vdnscope-7</vdnScopeId>
      <vdsContextWithBacking>
        <teaming>ETHER_CHANNEL</teaming>
        <switchId>dvs-81</switchId>
        <backingType>portgroup</backingType>
        <backingValue>dvportgroup-88</backingValue>
      </vdsContextWithBacking>
      <vdnId>5002</vdnId>
      <multicastAddr>239.0.0.3</multicastAddr>
    </datapart>
    <datapart class="virtualWire">
      ....
    </datapart>
    <pagingInfo>
      <pageSize>20</pageSize>
      <startIndex>0</startIndex>
      <totalCount>3</totalCount>
      <sortOrderAscending>false</sortOrderAscending>
    </pagingInfo>
  </sortedDataPage>
</virtualwires>
```

---

In this case, ?pagesize=XX&startindex=YY specifies the query results to larger pages. The default pagesize is 20 and the default startindex is 0. "pagesize" actually refers to the number of logical switches that will be retrieved by the API call (i.e. The default pagesize value of 20 means that only 20 logical switches will be retrieved even if there are more than 20 logical switches created in NSX). To retrieve all of the logical switches, the API user should use a pagesize that's reasonably large enough.

## Query all Logical Switches on all Transport Zones

You can retrieve all logical switches across all transport zones.

### Example 7-26. Get all logical switches on all transport zones

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/virtualwires?pagesize=XX&startindex=YY
```

Response Body:

```

</virtualWires>
  <sortedDataPage>
    <datapart class="virtualWire">
      <objectId>virtualwire-1</objectId>
      <name>vwire1</name>
      <description>logical switch 1</description>
      <tenantId>logical switch tenant</tenantId>
      <revision>0</revision>
      <vdsScopeId>vdscope-7</vdsScopeId>
      <vdsContextwithBacking>
        <teaming>ETHER_CHANNEL</teaming>
        <switchId>dvs-81</switchId>
        <backingType>portgroup</backingType>
        <backingValue>dvportgroup-88</backingValue>
      </vdsContextwithBacking>
      <vdsId>5002</vdsId>
      <multicastAddr>239.0.0.3</multicastAddr>
    </datapart> ...
    <datapart class="virtualWire"> ...
  </datapart>
  <pagingInfo>
    <pageSize>20</pageSize>
    <startIndex>0</startIndex>
    <totalCount>3</totalCount>
    <sortOrderAscending>>false</sortOrderAscending>
  </pagingInfo>
</sortedDataPage>
</virtualWires>

```

---

In this case, `?pagesize=XX&startIndex=YY` specifies the query results to larger pages. The default pagesize is 20 and the default startIndex is 0. "pagesize" actually refers to the number of logical switches that will be retrieved by the API call (i.e. the default pagesize value of 20 means that only 20 logical switches will be retrieved even if there are more than 20 logical switches created in NSX). To retrieve all of the logical switches, the API user should use a pagesize that's reasonably large enough.

## Query a Specific Logical Switch

You can retrieve the definition for a logical switch.

### Example 7-27. Get a logical switch definition

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/virtualwires/virtualWireId
```

Response Body:

```

<virtualWire>
  <name>Test logical switch</name>
  <description>Test logical switch Description</description>
  <objectId>virtualwire-4</objectId>
  <vdsScopeId>vdscope-3</vdsScopeId>
  <revision>1</revision>
  <vdsContextwithBacking>
    <teaming>ETHER_CHANNEL</teaming>
    <switchId>dvs-162</switchId>
    <backingType>PortGroup</backingType>
    <backingValue>pg-moid</backingValue>
  </vdsContextwithBacking>
  <vdsId>5002</vdsId>
  <multicastAddr>239.0.0.3</multicastAddr>
</virtualWire>

```

---

## Modify Control Plane Mode

You can modify the control plane mode of a logical switch. The possible options are:

- **Multicast:** Multicast IP addresses on physical network is used for the control plane. This mode is recommended only when you are upgrading from older VXLAN deployments. Requires PIM/IGMP on physical network.
- **Unicast:** The control plane is handled by an NSX controller. All unicast traffic leverages headend replication. No multicast IP addresses or special network configuration is required.
- **Hybrid:** The optimized unicast mode. Offloads local traffic replication to physical network (L2 multicast). This requires IGMP snooping on the first-hop switch, but does not require PIM. Firsthop switch handles traffic replication for the subnet.

## Delete a Logical Switch

You can delete a logical switch.

### Example 7-28. Delete logical switch

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/virtualwires/virtualWireId
```

## Working with ARP Suppression and MAC Learning for Logical Switches

You can enable IP discovery (ARP suppression) and MAC learning for logical switches or dvPortGroup.

Enabling MAC Learning builds a VLAN - MAC pair learning table on each vNic. This table is stored as part of the dvfilter data. During vMotion, dvfilter saves/restores the table at the new location. The switch then issues RARPs for all the VLAN - MAC entries in the table.

Enabling this feature avoids possible traffic loss during vMotion in the following cases:

- the vNic is in VLAN trunk mode
- the VM is using more than one unicast MAC address. Since Etherswitch supports only one unicast MAC per vNic, RARP is not processed.

### Example 7-29. Enable ARP suppression and MAC learning

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/xvs/networks/dvpg-moid/virtualWireId/features
```

Request Body:

```
<networkFeatureConfig>
  <ipDiscoveryConfig>
    <enabled>true</enabled>
  </ipDiscoveryConfig>
  <macLearningConfig>
    <enabled>>false</enabled>
  </macLearningConfig>
</networkFeatureConfig>
```

### Example 7-30. Disable ARP suppression and MAC learning

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/xvs/networks/dvpg-moid/virtualWireId/features
```

Request Body:

```
<networkFeatureConfig>
  <ipDiscoveryConfig>
    <enabled>>false</enabled>
  </ipDiscoveryConfig>
  <macLearningConfig>
    <enabled>>false</enabled>
  </macLearningConfig>
</networkFeatureConfig>
```

---

#### Example 7-31. Query ARP suppression and MAC learning

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/xvs/networks/dvpg-moid/virtualWireId/features
```

Response Body:

```
<networkFeatureConfig>
  <ipDiscoveryConfig>
    <enabled>>true</enabled>
  </ipDiscoveryConfig>
  <macLearningConfig>
    <enabled>>true</enabled>
  </macLearningConfig>
</networkFeatureConfig>
```

---

## Managing the Logical Switch UDP Port

You can retrieve or update the UDP port.

### Get UDP Port

You can retrieve the UDP port for the logical switch.

#### Example 7-32. Get UDP port

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/config/vxlan/udp/port
```

---

### Update UDP Port

You can change the UDP port for the logical switch. If not set, the port defaults to port 8472.

#### Example 7-33. Change UDP port

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/vdn/config/vxlan/udp/port/port
```

---

## Querying Allocated Resources

You can retrieve a list of resources allocated to s in your network.

#### Example 7-34. Get resources

---

Get segment IDs allocated to s:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/config/resources/allocted?type
=segmentId&pagesize=pageSize&startIndex=startIndex
```

Get multicast address range allocated to s:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/config/resources/allocated?type
=multicastAddress&pagesize=pageSize&startIndex=startIndex
```

where

- start index is an optional parameter which specifies the starting point for retrieving the resources. If this parameter is not specified, resources are retrieved from the beginning.
- page size is an optional parameter that limits the maximum number of entries returned by the API. The default value for this parameter is 256 and the valid range is 1-1024.

## Testing Multicast Group Connectivity

You can perform a multicast group connectivity test in a transport zone.

### Test Multicast Group Connectivity in a Transport Zone

**Example 7-35.** Test multicast group connectivity in transport zone

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/scopeId/conn-check/multicast
```

Request Body:

```
<testParameters>
  <gateway>172.23.233.1</gateway>
  <packetSizeMode>0</packetSizeMode>  <!-- mode : 0 => vxlan standard packet size, 1 =>
    minimum packet size, 2 => customized packet size. --!>
  <packetSize>1600</packetSize>  <!-- applicable only if customized packet size is
    selected. --!>
  <sourceHost>
    <hostId>host-9</hostId>
    <switchId>dvs-22</switchId>
    <vlanId>54</vlanId>
  </sourceHost>
  <destinationHost>
    <hostId>host-92</hostId>
    <switchId>dvs-22</switchId>
    <vlanId>54</vlanId>
  </destinationHost>
</testParameters>
```

### Test Multicast Group Connectivity in a Logical Switch

**Example 7-36.** Test multicast group connectivity in logical switch

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/virtualWireId/conn-check/multicast
```

Request Body:

```
<testParameters>
  <gateway>172.23.233.1</gateway>
  <packetSizeMode>0</packetSizeMode>  <!-- mode : 0 => vxlan standard packet size, 1 =>
    minimum packet size, 2 => customized packet size. --!>
  <packetSize>1600</packetSize>  <!-- applicable only if customized packet size is
    selected. --!>
  <sourceHost>
```

```

    <hostId>host-9</hostId>
    <switchId>dvs-22</switchId>
    <vlanId>54</vlanId>
  </sourceHost>
  <destinationHost>
    <hostId>host-92</hostId>
    <switchId>dvs-22</switchId>
    <vlanId>54</vlanId>
  </destinationHost>
</testParameters>

```

---

## Performing Ping Test

You can perform a point to point connectivity test between two hosts across which a logical switch spans.

### Example 7-37. Perform point to point test

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/vdn/virtualwires/virtualWireId/conn-check/p2p>

Request Body:

```

<testParameters>
  <gateway>172.23.233.1</gateway>
  <packetSizeMode>0</packetSizeMode>  <!-- mode : 0 => vxlan standard packet size, 1 =>
    minimum packet size, 2 => customized packet size. --!>
  <packetSize>1600</packetSize>  <!-- applicable only if customized packet size is
    selected. --!>
  <sourceHost>
    <hostId>host-9</hostId>
    <switchId>dvs-22</switchId>
    <vlanId>54</vlanId>
  </sourceHost>
  <destinationHost>
    <hostId>host-92</hostId>
    <switchId>dvs-22</switchId>
    <vlanId>54</vlanId>
  </destinationHost>
</testParameters>

```

---

# NSX Edge Logical Router Management

# 8

NSX Edge Logical Router provides East-West distributed routing with tenant IP address space and data path isolation. Virtual machines or workloads that reside on the same host on different subnets can communicate with one another without having to traverse a traditional routing interface. A logical router can have eight uplink interfaces and up to a thousand internal interfaces.

For information on retrieving objects IDs, see [“vCenter Object IDs”](#) on page 495.

This chapter includes the following topics:

- [“Create a Logical Router”](#) on page 133
- [“Query a Logical Router”](#) on page 135
- [“Modify a Router”](#) on page 136
- [“Deleting a Router”](#) on page 137

---

All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Create a Logical Router

### Example 8-1. Create a router

---

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/edges>

Request Body:

```
<edge>
  <datacenterMoid>datacenter-2</datacenterMoid>
  <datacenterName>Datacenter</datacenterName>
  <tenant>default</tenant>
  <name>L2VPN-Client</name>
  <fqdn>vShield-edge-2</fqdn>
  <enableAesni>true</enableAesni>
  <enableFips>false</enableFips>
  <vseLogLevel>emergency</vseLogLevel>
  <vnics>
    <vnic>
      <label>vnic_0</label>
      <name>pk</name>
      <addressGroups>
        <addressGroup>
          <primaryAddress>10.112.203.19</primaryAddress>
          <subnetMask>255.255.255.0</subnetMask>
          <subnetPrefixLength>24</subnetPrefixLength>
        </addressGroup>
      </addressGroups>
    </vnic>
  </vnics>
</edge>
```

```

    <mtu>1500</mtu>
    <type>uplink</type>
    <isConnected>>true</isConnected>
    <index>0</index>
    <portgroupId>network-12</portgroupId>
    <portgroupName>VM Network</portgroupName>
    <enableProxyArp>>false</enableProxyArp>
    <enableSendRedirects>>false</enableSendRedirects>
  </vnic>
  <vnic>
    <label>vnic_1</label>
    <name>vnic1</name>
    <addressGroups />
    <mtu>1600</mtu>
    <type>trunk</type>
    <subInterfaces>
      <subInterface>
        <isConnected>>true</isConnected>
        <label>vnic_10</label>
        <name>Two</name>
        <index>10</index>
        <tunnelId>100</tunnelId>
        <vlanId>100</vlanId>
        <enableSendRedirects>>false</enableSendRedirects>
        <mtu>1500</mtu>
        <addressGroups>
          <addressGroup>
            <primaryAddress>10.10.10.1</primaryAddress>
            <subnetMask>255.255.255.0</subnetMask>
            <subnetPrefixLength>24</subnetPrefixLength>
          </addressGroup>
        </addressGroups>
      </subInterface>
    </subInterfaces>
    <isConnected>>true</isConnected>
    <index>1</index>
    <portgroupId>dvportgroup-37</portgroupId>
    <portgroupName>dvPortGroup2</portgroupName>
    <enableProxyArp>>false</enableProxyArp>
    <enableSendRedirects>>false</enableSendRedirects>
  </vnic>
</vnics>
<appliances>
  <deployAppliances>false</deployAppliances>
</appliances>
<cliSettings>
  <remoteAccess>>true</remoteAccess>
  <userName>admin</userName>
  <password>Applnumber@143</password>
</cliSettings>
<autoConfiguration>
  <enabled>>true</enabled>
  <rulePriority>high</rulePriority>
</autoConfiguration>
<type>distributedRouter</type>
<isUniversal>>false</isUniversal>
<hypervisorAssist>>false</hypervisorAssist>
<queryDaemon>
  <enabled>>false</enabled>
  <port>5666</port>
</queryDaemon>
</edge>

```

The “deployAppliances” flag in Appliance Configuration is the one which decides whether the VDR is created with or without a control VM. If the flag is *true* then VSM will deploy the appliance, and if it is *false* VSM creates a undeployed VDR. If appliances config is not given, then the flag “deployAppliances” defaults to *true* and will be validated for appliance configuration.

It is not possible to set the `<ecmp>true</ecmp>` property upon creation of a distributed logical router Edge and a subsequent API call is required to enable ECMP.

DHCP relay settings are not able to be used when creating a distributed logical router Edge and a subsequent API call is required to configure DHCP relay properties.

## Query a Logical Router

Retrieves information about the specified router.

### Example 8-2. Query a router

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId
```

Response Body:

```
<edgeSummaries>
  <edge>
    <id>edge-15</id>
    <version>21</version>
    <status>deployed</status>
    <datacenterMoid>datacenter-2</datacenterMoid>
    <datacenterName>Datacenter</datacenterName>
    <tenant>default</tenant>
    <name>vShield-edge-15</name>
    <fqdn>vShield-edge-15</fqdn>
    <enableAesni>true</enableAesni>
    <enableFips>false</enableFips>
    <vseLogLevel>info</vseLogLevel>
    <appliances>
      <applianceSize>compact</applianceSize>
      <appliance>
        <highAvailabilityIndex>0</highAvailabilityIndex>
        <vcUuid>422f63b1-bb0e-ba50-3aae-4be1263db676</vcUuid>
        <vmId>vm-62</vmId>
        <resourcePoolId>resgroup-20</resourcePoolId>
        <resourcePoolName>Resources</resourcePoolName>
        <datastoreId>datastore-23</datastoreId>
        <datastoreName>shahm-esx-storage</datastoreName>
        <hostId>host-22</hostId>
        <hostName>10.112.196.160</hostName>
        <vmFolderId>group-v3</vmFolderId>
        <vmFolderName>vm</vmFolderName>
        <vmHostname>vShield-edge-15-0</vmHostname>
        <vmName>vShield-edge-15-0</vmName>
        <deployed>true</deployed>
        <edgeId>edge-15</edgeId>
      </appliance>
    </appliances>
    <cliSettings>
      <remoteAccess>false</remoteAccess>
      <userName>admin</userName>
    </cliSettings>
    <type>distributedRouter</type>
    <mgmtInterface>
      <label>vnic_0</label>
      <name>mgmtInterface</name>
      <addressGroups>
        <addressGroup>
          <primaryAddress>10.112.196.166</primaryAddress>
```

```

        <subnetMask>255.255.252.0</subnetMask>
    </addressGroup>
</addressGroups>
<mtu>1500</mtu>
<index>0</index>
<connectedToId>dvportgroup-38</connectedToId>
<connectedToName>DvPortGroup1</connectedToName>
</mgmtInterface>
<interfaces>
  <interface>
    <label>vNic_1</label>
    <name>interface1</name>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.10.1</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
      </addressGroup>
    </addressGroups>
    <mtu>1500</mtu>
    <type>uplink</type>
    <isConnected>true</isConnected>
    <index>1</index>
    <connectedToId>dvportgroup-39</connectedToId>
    <connectedToName>dvport-vlan-1</connectedToName>
  </interface>
  <interface>
    <label>75649aea0000000a</label>
    <name>interface10</name>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.20.1</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
      </addressGroup>
    </addressGroups>
    <mtu>1500</mtu>
    <type>internal</type>
    <isConnected>true</isConnected>
    <index>10</index>
    <connectedToId>dvportgroup-40</connectedToId>
    <connectedToName>dvport-vlan-2</connectedToName>
  </interface>
  <interface>
    <label>75649aea0000000b</label>
    <name>interface-11</name>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.50.1</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
      </addressGroup>
    </addressGroups>
    <mtu>1500</mtu>
    <type>internal</type>
    <isConnected>true</isConnected>
    <index>11</index>
    <connectedToId>dvportgroup-37</connectedToId>
    <connectedToName>DvSwitch2-DVUplinks-36</connectedToName>
  </interface>
</interfaces>
<edgeAssistId>1969527530</edgeAssistId>
</edge>
</edgeSummaries>

```

## Modify a Router

Replaces the configuration of the specified router.

The location header when installing a router returns the *edgeId* of the installed router. You must use this ID to configure and manage this NSX Edge instance.

### Example 8-3. Modify router

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId>

Request Body:

```
<edge>
  <datacenterMoid>datacenter-2</datacenterMoid>
  <type>distributedRouter</type>  <!-- Mandatory to create "distributedRouter" edge.
    When absent, defaults to "gatewayServices" -->
  <appliances>  <!-- Mandatory for "distributedRouter" edge. At least one appliance
    needs to be configured -->
    <appliance>
      <resourcePoolId>resgroup-20</resourcePoolId>
      <datastoreId>datastore-23</datastoreId>
    </appliance>
  </appliances>
  <mgmtInterface>  <!-- Mandatory for "distributedRouter" edge -->
    <connectedToId>dvportgroup-38</connectedToId>
    <addressGroups>
      <addressGroup>
        <primaryAddress>10.112.196.165</primaryAddress>
        <subnetMask>255.255.252.0</subnetMask>
      </addressGroup>
    </addressGroups>
  </mgmtInterface>
  <interfaces>  <!-- Optional. Can be added later using modular APIs. Upto 999
    interfaces supported. -->
    <interface>
      <type>uplink</type>
      <mtu>1500</mtu>
      <isConnected>true</isConnected>
      <addressGroups>  <!-- Supports one or more addressGroups -->
        <addressGroup>  <!-- AddressGroup on "distributedRouter" edge can have only
          primary ipAddresses. Secondary addresses not supported -->
          <primaryAddress>192.168.10.1</primaryAddress>  <!-- "distributedRouter"
            edge only supports IPv4 addresses -->
          <subnetMask>255.255.255.0</subnetMask>
        </addressGroup>
      </addressGroups>
      <connectedToId>dvportgroup-39</connectedToId>  <!-- "distributedRouter" edge
        does not support legacy portGroups -->
    </interface>
    <interface>
      <type>internal</type>
      <mtu>1500</mtu>
      <isConnected>true</isConnected>
      <addressGroups>
        <addressGroup>
          <primaryAddress>192.168.20.1</primaryAddress>
          <subnetMask>255.255.255.0</subnetMask>
        </addressGroup>
      </addressGroups>
      <connectedToId>dvportgroup-40</connectedToId>
    </interface>
  </interfaces>
</edge>
```

---

## Deleting a Router

You can delete a logical router instance. Appliances associated with the router instance are deleted as well.

**Example 8-4.** Delete a router

Request:

DELETE <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId>

## Working with Interfaces

An NSX Edge router can have eight uplink interfaces and up to a thousand internal interfaces. It must have at least one internal interface before it can be deployed.

### Working with Management Interfaces

**Configure Management Interfaces**

Configure management interfaces for an NSX Edge router.

**Example 8-5.** Configure management interfaces

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/mgmtinterface>

Request Body:

```
<mgmtInterface>
  <addressGroups>
    <addressGroup>
      <primaryAddress>10.112.196.166</primaryAddress>
      <subnetMask>255.255.252.0</subnetMask>
    </addressGroup>
  </addressGroups>
  <mtu>1500</mtu>
  <connectedToId>dvportgroup-38</connectedToId>
</mgmtInterface>
```

**Query Management Interfaces**

Retrieves all management interfaces for the specified NSX Edge router.

**Example 8-6.** Query interfaces

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/mgmtinterface>

Response Body:

```
<mgmtInterface>
  <label>vNic_0</label>
  <name>mgmtInterface</name>
  <addressGroups>
    <addressGroup>
      <primaryAddress>10.112.196.166</primaryAddress>
      <subnetMask>255.255.252.0</subnetMask>
    </addressGroup>
  </addressGroups>
  <mtu>1500</mtu>
  <index>0</index>
  <connectedToId>dvportgroup-38</connectedToId>
  <connectedToName>DvPortGroup1</connectedToName>
</mgmtInterface>
```

## Working with all Interfaces

An NSX Edge router can have up to 8 uplink interfaces.

### Add Interfaces

Configures one or more interface for an NSX Edge Router. The specified configuration is stored in the database. If any appliance(s) is associated with this Edge instance, the specified configuration is applied to the appliance as well.

You should not define an index for the new addition of interfaces. The indexes are system-generated. To update the existing interfaces, include them in the XML with the system-generated indexes (can be obtained by a GET call).

#### Example 8-7. Add an interface

---

Request:

POST `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/interfaces/?action=patch`

Request Body:

```
<interfaces>
  <interface>
    <name>interface1</name>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.10.1</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
      </addressGroup>
    </addressGroups>
    <mtu>1500</mtu>
    <type>uplink</type>
    <isConnected>true</isConnected>
    <connectedToId>dvportgroup-39</connectedToId>
  </interface>
  <interface>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.20.1</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
      </addressGroup>
    </addressGroups>
    <mtu>1500</mtu>
    <type>internal</type>
    <isConnected>true</isConnected>
    <connectedToId>dvportgroup-40</connectedToId>
  </interface>
  <interface>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.50.1</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
      </addressGroup>
    </addressGroups>
    <mtu>1500</mtu>
    <type>internal</type>
    <isConnected>true</isConnected>
    <connectedToId>dvportgroup-37</connectedToId>
  </interface>
</interfaces>
```

---

### Query Interfaces for a NSX Edge Router

Retrieves all interfaces for the specified Edge router.

**Example 8-8.** Retrieve all interfaces

Request:

GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/interfaces

Response Body:

```

<interfaces>
  <interface>
    <label>vNic_1</label>
    <name>interface1</name>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.10.1</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
      </addressGroup>
    </addressGroups>
    <mtu>1500</mtu>
    <type>uplink</type>
    <isConnected>true</isConnected>
    <index>1</index>
    <connectedToId>dvportgroup-39</connectedToId>
    <connectedToName>dvport-vlan-1</connectedToName>
  </interface>
  <interface>
    <label>75649aea0000000a</label>
    <name>interface10</name>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.20.1</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
      </addressGroup>
    </addressGroups>
    <mtu>1500</mtu>
    <type>internal</type>
    <isConnected>true</isConnected>
    <index>10</index>
    <connectedToId>dvportgroup-40</connectedToId>
    <connectedToName>dvport-vlan-2</connectedToName>
  </interface>
  <interface>
    <label>75649aea0000000b</label>
    <name>interface-11</name>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.50.1</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
      </addressGroup>
    </addressGroups>
    <mtu>1500</mtu>
    <type>internal</type>
    <isConnected>true</isConnected>
    <index>11</index>
    <connectedToId>dvportgroup-37</connectedToId>
    <connectedToName>DvSwitch2-DvUplinks-36</connectedToName>
  </interface>
</interfaces>

```

**Delete Interfaces**

Deletes one or more interfaces for an NSX Edge Router. Stores the specified configuration in database. If any appliance(s) are associated with this edge, disconnects and deletes the interface.

**Example 8-9.** Delete interface

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/interfaces/?index
=index1&index=index2
```

**Delete all Interfaces**

Deletes all interfaces for an NSX Edge Router. Stores the specified configuration in database. If any appliance(s) are associated with this edge, disconnects and deletes the interface.

**Example 8-10.** Delete all interfaces

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/interfaces
```

**Manage an NSX Edge Router Interface**

You can manage a specific NSX Edge router interface.

**Retrieve Interface with Specific Index**

Retrieves the interface with specified index for a Edge.

**Example 8-11.** Get interface with specific index

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/interfaces/index
```

Response Body:

```
<interface>
  <label>vNic_1</label>
  <name>interface1</name>
  <addressGroups>
    <addressGroup>
      <primaryAddress>192.168.10.1</primaryAddress>
      <subnetMask>255.255.255.0</subnetMask>
    </addressGroup>
  </addressGroups>
  <mtu>1500</mtu>
  <type>uplink</type>
  <isConnected>true</isConnected>
  <index>1</index>
  <connectedToId>dvportgroup-39</connectedToId>
  <connectedToName>dvport-vlan-1</connectedToName>
</interface>
```

**Modify an Interface**

Modifies the specified interface.

**Example 8-12.** Modify interface

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/interfaces/index
```

Request Body:

```
<interface>
```

```

<name>interface1</name>
<addressGroups>
  <addressGroup>
    <primaryAddress>192.168.10.1</primaryAddress>
    <subnetMask>255.255.255.0</subnetMask>
  </addressGroup>
</addressGroups>
<mtu>1500</mtu>
<type>uplink</type>
<isConnected>true</isConnected>
<connectedToId>dvportgroup-39</connectedToId>
</interface>

```

---

### Delete Interface Configuration

Deletes the interface configuration and resets it to the factory default.

#### Example 8-13. Delete interface configuration

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/interfaces/index
```

---

## Configure Routes

Configures globalConfig, staticRouting, OSPF, and BGP.

#### Example 8-14. Configure routes

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config
```

Request Body:

```

<routing>
  <routingGlobalConfig>
    <routerId>1.1.1.1</routerId>  <!-- Required when dynamic routing protocols like
      OSPF, or BGP is configured -->
    <logging>  <!-- Optional. when absent, enable=false and logLevel=INFO -->
      <enable>false</enable>
      <logLevel>info</logLevel>
    </logging>
    <ipPrefixes>  <!-- Optional. Required only if user wants to define redistribution
      rules in dynamic routing protocols like ospf, bgp -->
      <ipPrefix>
        <name>a</name>  <!-- All the defined ipPrefix must have unique names -->
        <ipAddress>10.112.196.160/24</ipAddress>
      </ipPrefix>
      <ipPrefix>
        <name></name>
        <ipAddress>192.168.10.0/24</ipAddress>
      </ipPrefix>
    </ipPrefixes>
  </routingGlobalConfig>
  <staticRouting>
    <staticRoutes>  <!-- Optional, if no static routes needs to be configured -->
      <route>
        <description>route1</description>
        <vnic>0</vnic>
        <network>3.1.1.0/22</network>
        <nextHop>172.16.1.14</nextHop>
        <mtu>1500</mtu>  <!-- optional. valid value: smaller than the MTU set on the
          interface. Default will be the MTU of the interface on which this route is
          configured -->
      </route>
    </staticRoutes>
  </staticRouting>
</routing>

```

```

</route>
<route>
  <description>route2</description>
  <vnic>1</vnic>
  <network>4.1.1.0/22</network>
  <nextHop>10.112.196.118</nextHop>
  <mtu>1500</mtu>  <!-- Optional. Valid value:smaller than the MTU set on the
                    interface. Default will be the MTU of the interface on which this route is
                    configured -->
</route>
</staticRoutes>
<defaultRoute>  <!-- Optional, if no default routes needs to be configured -->
<description>defaultRoute</description>
<vnic>0</vnic>
<gatewayAddress>172.16.1.12</gatewayAddress>
<mtu>1500</mtu>  <!-- Optional. Valid value:smaller than the MTU set on the
                    interface. Default will be the MTU of the interface on which this route is
                    configured -->
</defaultRoute>
</staticRouting>
<ospf>  <!-- Optional, if no OSPF needs to be configured -->
<enabled>true</enabled>  <!-- Optional. Defaults to true -->
<forwardingAddress>192.168.10.2</forwardingAddress>  <!-- ipAddress on one of the
uplink interfaces -->
<protocolAddress>192.168.10.3</protocolAddress>  <!-- ipAddress on the same subnet
as the forwardingAddress -->
<ospfAreas>
  <ospfArea>
    <areaId>100</areaId>  <!-- Mandatory and unique. Valid values are 0-4294967295 -->
    <type>normal</type>  <!-- Optional. Default is normal. Valid inputs are normal,
stub -->
    <authentication>  <!-- Optional. When not specified, its "none"
authentication. -->
      <type>password</type>  <!-- Valid values are none, password , md5 -->
      <value>vmware123</value>  <!-- Value as per the type of authentication -->
    </authentication>
  </ospfArea>
</ospfAreas>
<ospfInterfaces>
  <ospfInterface>
    <vnic>0</vnic>
    <areaId>100</areaId>
    <helloInterval>10</helloInterval>  <!-- Optional. Default 10 sec. Valid values are
1-255-->
    <deadInterval>40</deadInterval>  <!-- Optional. Default 40 sec. Valid values are
1-65535 -->
    <priority>128</priority>  <!-- Optional. Default 128. Valid values are 0-255 -->
    <cost>10</cost>  <!-- Optional. Auto based on interface speed. Valid values are
1-65535 -->
    <mtuIgnore>true|false</mtuIgnore>
  </ospfInterface>
</ospfInterfaces>
<redistribution>
  <enabled>true</enabled>  <!-- Optional. Defaults to false. -->
  <rules>
    <rule>
      <prefixName>a</prefixName>  <!-- Optional. Default is "any". prefixName used
here should be defined in the routingGlobalConfig->ipPrefixes -->
      <from>
        <isis>true</isis>  <!-- Optional. Defaults to false -->
        <ospf>false</ospf>  <!-- Optional. Defaults to false -->
        <bgp>false</bgp>  <!-- Optional. Defaults to false -->
        <static>false</static>  <!-- Optional. Defaults to false -->
        <connected>true</connected>  <!-- Optional. Defaults to false -->
      </from>
      <action>deny</action>  <!-- Mandatory. Valid values are deny|permit -->
    </rule>
  </rules>
</redistribution>

```

```

    <prefixName>b</prefixName>  <!-- Optional. Default is "any". prefixName used
    here should be defined in the routingGlobalConfig->ipPrefixes -->
    <from>
      <isis>>false</isis>        <!-- Optional. Defaults to false -->
      <ospf>>false</ospf>        <!-- Optional. Defaults to false -->
      <bgp>>true</bgp>           <!-- Optional. Defaults to false -->
      <static>>false</static>    <!-- Optional. Defaults to false -->
      <connected>>false</connected> <!-- Optional. Defaults to false -->
    </from>
    <action>permit</action>    <!-- Mandatory. Valid values are deny|permit -->
  </rule>
</rules>
</redistribution>
</ospf>
<bgp>      <!-- Optional, if no BGP needs to be configured -->
<enabled>>true</enabled>    <!-- Optional. Default is true -->
<localAS>65535</localAS>  <!-- Valid values are : 0-65535 -->
<bgpNeighbours>
  <bgpNeighbour>
    <ipAddress>192.168.10.10</ipAddress> <!-- Peer's IP. IPv4 only. Should not be
    same as any of interfaces's IPs, forwardingAddress, protocolAddress -->
    <forwardingAddress>192.168.1.10</forwardingAddress> <!-- Address defined on
    one of the uplink interfaces's -->
    <protocolAddress>192.168.1.11</protocolAddress>    <!-- Address in the above
    same subnet as the forwardingAddress -->
    <remoteAS>65500</remoteAS>    <!-- Valid values are 1-65534 -->
    <weight>60</weight>          <!-- Optional. Default is 60. Valid
    values are 0-65535 -->
    <holdDownTimer>180</holdDownTimer> <!-- Optional. Default is 180 seconds.
    Valid values are : 2-65535. -->
    <keepAliveTimer>60</keepAliveTimer> <!-- Optional. Default is 60 seconds.
    Valid values are : 1-65534. -->
    <password>vmware123</password> <!-- Optional -->
    <bgpFilters>                  <!-- Optional -->
      <bgpFilter>
        <direction>in</direction>    <!-- Valid values are in/out -->
        <action>permit</action>      <!-- Valid values are permit/deny -->
        <network>10.0.0.0/8</network> <!-- Valid values are CIDR networks. IPv4
        only. IPv6 support not supported -->
        <ipPrefixGe>17</ipPrefixGe> <!-- Optional. "Greater than or equal to" &
        used for filtering based on prefix length. Valid IPv4 prefixes -->
        <ipPrefixLe>32</ipPrefixLe> <!-- Optional. "Less than or equal to" &
        used for filtering based on prefix length. Valid IPv4 prefixes -->
      </bgpFilter>
    </bgpFilters>
  </bgpNeighbour>
</bgpNeighbours>
<redistribution>
  <enabled>true</enabled> <!-- Optional. Defaults to false. -->
  <rules>
    <rule>
      <prefixName>a</prefixName>  <!-- Optional. Default is "any". prefixName used
      here should be defined in the routingGlobalConfig->ipPrefixes -->
      <from>
        <isis>true</isis>          <!-- Optional. Defaults to false -->
        <ospf>true</ospf>          <!-- Optional. Defaults to false -->
        <bgp>>false</bgp>           <!-- Optional. Defaults to false -->
        <static>true</static>      <!-- Optional. Defaults to false -->
        <connected>>false</connected> <!-- Optional. Defaults to false -->
      </from>
      <action>deny</action>        <!-- Mandatory. Valid values are deny|permit -->
    </rule>
  </rule>
  <from>
    <isis>>false</isis>            <!-- Optional. Defaults to false -->
    <ospf>>false</ospf>            <!-- Optional. Defaults to false -->
    <bgp>>false</bgp>              <!-- Optional. Defaults to false -->
    <static>>false</static>        <!-- Optional. Defaults to false -->
  </from>

```

```

        <connected>true</connected> <!-- Optional. Defaults to false -->
    </from>
    <action>permit</action> <!-- Mandatory. valid values are deny|permit -->
</rule>
</rules>
</redistribution>
</bgp>
</routing>

```

---

## Query Routes

Retrieves global, static, OSPF, and BGP configurations.

### Example 8-15. Retrieve routes

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config>

Response Body:

```

<routing>
  <routingGlobalConfig>
    <routerId>1.1.1.1</routerId>
    <logging>
      <enable>>false</enable>
      <logLevel>info</logLevel>
    </logging>
    <ipPrefixes>
      <ipPrefix>
        <name>a</name>
        <ipAddress>10.112.196.160/24</ipAddress>
      </ipPrefix>
      <ipPrefix>
        <name>b</name>
        <ipAddress>192.168.10.0/24</ipAddress>
      </ipPrefix>
    </ipPrefixes>
  </routingGlobalConfig>
  <staticRouting>
    <staticRoutes>
      <route>
        <description>route1</description>
        <vnic>0</vnic>
        <network>3.1.1.0/22</network>
        <nextHop>172.16.1.14</nextHop>
        <mtu>1500</mtu>
        <type>user</type>
      </route>
      <route>
        <description>route2</description>
        <vnic>1</vnic>
        <network>4.1.1.0/22</network>
        <nextHop>10.112.196.118</nextHop>
        <mtu>1500</mtu>
        <type>user</type>
      </route>
    </staticRoutes>
    <defaultRoute>
      <description>defaultRoute</description>
      <vnic>0</vnic>
      <gatewayAddress>172.16.1.12</gatewayAddress>
      <mtu>1500</mtu>
    </defaultRoute>
  </staticRouting>
  <ospf>
    <enabled>true</enabled>

```

```

<forwardingAddress>192.168.10.2</forwardingAddress>
<protocolAddress>192.168.10.3</protocolAddress>
<ospfAreas>
  <ospfArea>
    <areaId>100</areaId>
    <type>normal</type>
    <authentication>
      <type>password</type>
      <value>vmware123</value>
    </authentication>
  </ospfArea>
</ospfAreas>
<ospfInterfaces>
  <ospfInterface>
    <vnic>0</vnic>
    <areaId>100</areaId>
    <helloInterval>10</helloInterval>
    <deadInterval>40</deadInterval>
    <priority>128</priority>
    <cost>10</cost>
    <mtuIgnore>true|false</mtuIgnore>
  </ospfInterface>
</ospfInterfaces>
<redistribution>
  <enabled>true</enabled>
  <rules>
    <rule>
      <id>1</id>
      <prefixName>a</prefixName>
      <from>
        <isis>true</isis>
        <ospf>false</ospf>
        <bgp>false</bgp>
        <static>false</static>
        <connected>true</connected>
      </from>
      <action>deny</action>
    </rule>
    <rule>
      <id>0</id>
      <prefixName>b</prefixName>
      <from>
        <isis>false</isis>
        <ospf>false</ospf>
        <bgp>true</bgp>
        <static>false</static>
        <connected>false</connected>
      </from>
      <action>permit</action>
    </rule>
  </rules>
</redistribution>
</ospf>
<bgp>
  <enabled>true</enabled>
  <localAS>65535</localAS>
  <bgpNeighbours>
    <bgpNeighbour>
      <ipAddress>192.168.10.10</ipAddress>
      <forwardingAddress>192.168.1.10</forwardingAddress>
      <protocolAddress>192.168.1.11</protocolAddress>
      <remoteAS>65500</remoteAS>
      <weight>60</weight>
      <holdDownTimer>180</holdDownTimer>
      <keepAliveTimer>60</keepAliveTimer>
      <password>vmware123</password>
      <bgpFilters>
        <bgpFilter>

```

```

        <direction>in</direction>
        <action>permit</action>
        <network>10.0.0.0/8</network>
        <ipPrefixGe>17</ipPrefixGe>
        <ipPrefixLe>32</ipPrefixLe>
    </bgpFilter>
    <bgpFilter>
        <direction>out</direction>
        <action>deny</action>
        <network>20.0.0.0/26</network>
    </bgpFilter>
</bgpFilters>
</bgpNeighbour>
</bgpNeighbours>
<redistribution>
    <enabled>true</enabled>
    <rules>
        <rule>
            <id>1</id>
            <prefixName>a</prefixName>
            <from>
                <isis>true</isis>
                <ospf>true</ospf>
                <bgp>false</bgp>
                <static>true</static>
                <connected>false</connected>
            </from>
            <action>deny</action>
        </rule>
        <rule>
            <id>0</id>
            <from>
                <isis>false</isis>
                <ospf>false</ospf>
                <bgp>false</bgp>
                <static>false</static>
                <connected>true</connected>
            </from>
            <action>permit</action>
        </rule>
    </rules>
</redistribution>
</bgp>
</routing>

```

---

## Delete Routes

Deletes the routing configuration stored in the NSX Manager database and the default routes from the specified NSX Edge router.

### Example 8-16. Delete routing

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config
```

---

## Manage Global Routing Configuration

Configures the default gateway for static routes and dynamic routing details.

## Specify Global Configuration

### Example 8-17. Configure global route

---

Request:

PUT `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/global`

Request Body:

```
<routingGlobalConfig>
  <routerId>1.1.1.1</routerId>  <!-- Required when dynamic routing protocols like OSPF,
    or BGP is configured -->
  <ecmp>>false</ecmp> <!-- Optional. Defaults to false. -->
  <logging>  <!-- Optional. When absent, enable=false and logLevel=INFO -->
    <enable>>false</enable>
    <logLevel>info</logLevel>
  </logging>
  <ipPrefixes>  <!-- Optional. Required only if user wants to define redistribution
    rules in dynamic routing protocols like ospf, isis, bgp -->
    <ipPrefix>
      <name>a</name>  <!-- All the defined ipPrefix must have unique names -->
      <ipAddress>10.112.196.160/24</ipAddress>
    </ipPrefix>
    <ipPrefix>
      <name>b</name>
      <ipAddress>192.168.10.0/24</ipAddress>
    </ipPrefix>
  </ipPrefixes>
</routingGlobalConfig>
```

---

## Query Global Route

Retrieves routing information from the NSX Manager database for an edge which includes the following:

- Default route settings
- Static route configurations

### Example 8-18. Query global route

---

Request:

GET `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/global`

```
<routingGlobalConfig>
  <ecmp>>false</ecmp>
  <logging>
    <enable>>false</enable>
    <logLevel>info</logLevel>
  </logging>
</routingGlobalConfig>
```

---

## Manage Static Routing

Add or query static and default routes for specified Edge.

### Configure Static Routes

Configures static and default routes.

### Example 8-19. Configure static routes

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/static>

Request Body:

```
<staticRouting>
  <staticRoutes>
    <route>
      <description>route1</description>  <!-- Optional-->
      <vnic>0</vnic>  <!-- Optional-->
      <network>3.1.1.4/22</network>
      <nextHop>172.16.1.14</nextHop>
      <mtu>1500</mtu>  <!-- Optional. Valid value:smaller than the MTU set on the
        interface. Default will be the MTU of the interface on which this route is
        configured -->
      <adminDistance></adminDistance>  <!-- Optional. Default value is 1-->
    </route>
    <route>
      <description>route2</description>
      <vnic>1</vnic>
      <network>4.1.1.4/22</network>
      <nextHop>10.112.196.118</nextHop>
      <mtu>1500</mtu>  <!-- Optional. Valid value:smaller than the MTU set on the
        interface. Default will be the MTU of the interface on which this route is
        configured -->
      <adminDistance />  <!-- Optional. Default value is 1-->
    </route>
  </staticRoutes>
  <defaultRoute>
    <description>defaultRoute</description>
    <vnic>0</vnic>
    <gatewayAddress>172.16.1.12</gatewayAddress>
    <mtu>1500</mtu>  <!-- Optional. Valid value:smaller than the MTU set on the
      interface. Default will be the MTU of the interface on which this route is
      configured -->
    <adminDistance />  <!-- Optional. Default value is 1-->
  </defaultRoute>
</staticRouting>
```

---

## Query Static Routes

Retrieves static and default routes.

### Example 8-20. Configure static routes

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/static>

Response Body:

```
<staticRouting>
  <staticRoutes>
    <route>
      <description>route1</description>
      <vnic>0</vnic>
      <network>3.1.1.4/22</network>
      <nextHop>172.16.1.14</nextHop>
      <mtu>1500</mtu>
      <type>user</type>
    </route>
    <route>
      <description>route2</description>
      <vnic>1</vnic>
      <network>4.1.1.4/22</network>
      <nextHop>10.112.196.118</nextHop>
      <mtu>1500</mtu>
      <type>user</type>
    </route>
```

```

</staticRoutes>
<defaultRoute>
  <description>defaultRoute</description>
  <vnic>0</vnic>
  <gatewayAddress>172.16.1.12</gatewayAddress>
  <mtu>1500</mtu>
</defaultRoute>
</staticRouting>

```

---

### Delete Static Routes

Deletes both static and default routing configuration stored in the NSX Manager database.

#### Example 8-21. Delete static routes

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/static
```

---

## Manage OSPF Routes for NSX Edge

NSX Edge supports OSPF, an interior gateway protocol that routes IP packets only within a single routing domain. It gathers link state information from available routers and constructs a topology map of the network. The topology determines the routing table presented to the Internet Layer, which makes routing decisions based on the destination IP address found in IP packets.

OSPF routing policies provide a dynamic process of traffic load balancing between routes of equal cost. An OSPF network is divided into routing areas to optimize traffic. An area is a logical collection of OSPF networks, routers, and links that have the same area identification.

Areas are identified by an Area ID.

### Configure OSPF

#### Example 8-22. Configure OSPF

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/ospf
```

Request Body:

```

<ospf>
  <enabled>true</enabled>  <!-- when not specified, it will be treated as false, when
                           false, it will delete the existing config -->
  <ospfAreas>
    <ospfArea>
      <areaId>100</areaId> <!-- Mandatory and unique. Valid values are 0-4294967295 -->
      <type>normal</type> <!-- Optional. Default is normal. Valid inputs are normal,
                           nssa -->
      <authentication> <!-- Optional. When not specified, its "none" authentication.
                       -->
      <type>password</type>  <!-- Valid values are none, password , md5 -->
      <value>vmware123</value>  <!-- Value as per the type of authentication -->
    </ospfArea>
  </ospfAreas>
  <ospfInterfaces>
    <ospfInterface>
      <vnic>0</vnic>
      <areaId>100</areaId>
      <helloInterval>10</helloInterval> <!-- optional. Default 10 sec. Valid values are
                                           1-255-->
    </ospfInterface>
  </ospfInterfaces>
</ospf>

```

```

<deadInterval>40</deadInterval> <!-- Optional. Default 40 sec. Valid values are
1-65535 -->
<priority>128</priority> <!-- Optional. Default 128. valid values are 0-255 -->
<cost>10</cost> <!-- Optional. Auto based on interface speed. Valid values are
1-65535 -->
<mtuIgnore>true|false</mtuIgnore>
</ospfInterface>
</ospfInterfaces>
<redistribution>
<enabled>true</enabled> <!-- Optional. Defaults to false. -->
<rules>
<rule>
<prefixName>a</prefixName> <!-- Optional. Default is "any". prefixName used
here should be defined in the routingGlobalConfig->ipPrefixes -->
<from>
<isis>true</isis> <!-- Optional. Defaults to false -->
<ospf>false</ospf> <!-- Optional. Defaults to false -->
<bgp>false</bgp> <!-- Optional. Defaults to false -->
<static>false</static> <!-- Optional. Defaults to false -->
<connected>true</connected> <!-- Optional. Defaults to false -->
</from>
<action>deny</action> <!-- Mandatory. Valid values are deny|permit -->
</rule>
<rule>
<prefixName>b</prefixName> <!-- Optional. Default is "any". prefixName used
here should be defined in the routingGlobalConfig->ipPrefixes -->
<from>
<isis>false</isis> <!-- Optional. Defaults to false -->
<ospf>false</ospf> <!-- Optional. Defaults to false -->
<bgp>true</bgp> <!-- Optional. Defaults to false -->
<static>false</static> <!-- Optional. Defaults to false -->
<connected>false</connected> <!-- Optional. Defaults to false -->
</from>
<action>permit</action> <!-- Mandatory. Valid values are deny|permit -->
</rule>
</rules>
</redistribution>
</ospf>

```

---

## Query OSPF

### Example 8-23. Query OSPF

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/ospf>

Response Body:

```

<ospf>
<enabled>true</enabled>
<ospfAreas>
<ospfArea>
<areaId>100</areaId>
<type>normal</type>
<authentication>
<type>password</type>
<value>vmware123</value>
</authentication>
</ospfArea>
</ospfAreas>
<ospfInterfaces>
<ospfInterface>
<vnic>0</vnic>
<areaId>100</areaId>
<helloInterval>10</helloInterval>
<deadInterval>40</deadInterval>

```

```

    <priority>128</priority>
    <cost>10</cost>
    <mtuIgnore>true|false</mtuIgnore>
  </ospfInterface>
</ospfInterfaces>
<redistribution>
  <enabled>true</enabled>
  <rules>
    <rule>
      <id>1</id>
      <prefixName>a</prefixName>
      <from>
        <isis>true</isis>
        <ospf>false</ospf>
        <bgp>false</bgp>
        <static>false</static>
        <connected>true</connected>
      </from>
      <action>deny</action>
    </rule>
    <rule>
      <id>0</id>
      <prefixName>b</prefixName>
      <from>
        <isis>false</isis>
        <ospf>false</ospf>
        <bgp>true</bgp>
        <static>false</static>
        <connected>false</connected>
      </from>
      <action>permit</action>
    </rule>
  </rules>
</redistribution>
</ospf>

```

---

### Delete OSPF

Deletes OSPF routing.

#### Example 8-24. Delete OSPF

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/ospf
```

---

## Manage BGP Routes for NSX Edge

Border Gateway Protocol (BGP) makes core routing decisions. It includes a table of IP networks or prefixes which designate network reachability among autonomous systems. An underlying connection between two BGP speakers is established before any routing information is exchanged. Keep alive messages are sent out by the BGP speakers in order to keep this relationship alive. Once the connection is established, the BGP speakers exchange routes and synchronize their tables.

### Configure BGP

#### Example 8-25. Configure BGP

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/bgp
```

Request Body:

```

<bgp>
  <enabled>true</enabled>  <!-- Optional. Default is false -->
  <localAS>65534</localAS>  <!-- Valid values are : 1-65534 -->
  <bgpNeighbours>
    <bgpNeighbour>
      <ipAddress>192.168.1.10</ipAddress>  <!-- IPv4 only. IPv6 support not
        supported -->
      <remoteAS>65500</remoteAS>  <!-- Valid values are 0-65535 -->
      <weight>60</weight>  <!-- Optional. Default is 60. Valid values are 0-65535 -->
      <holdDownTimer>180</holdDownTimer>  <!-- Optional. Default is 180 seconds.
        Valid values are : 2-65535. -->
      <keepAliveTimer>60</keepAliveTimer>  <!-- Optional. Default is 60 seconds.
        Valid values are : 1-65534. -->
      <password>vmware123</password>  <!-- Optional -->
      <bgpFilters>  <!-- Optional -->
        <bgpFilter>
          <direction>in</direction>  <!-- Valid values are in/out -->
          <action>permit</action>  <!-- Valid values are permit/deny -->
          <network>10.0.0.0/8</network>  <!-- Valid values are CIDR networks. IPv4
            only. IPv6 support not supported -->
          <ipPrefixGe>17</ipPrefixGe>  <!-- Optional. "Greater than or equal to" &
            used for filtering based on prefix length. Valid IPv4 prefixes -->
          <ipPrefixLe>32</ipPrefixLe>  <!-- Optional. "Less than or equal to" &
            used for filtering based on prefix length. Valid IPv4 prefixes -->
        </bgpFilter>
      </bgpFilters>
    </bgpNeighbour>
  </bgpNeighbours>
  <redistribution>
    <enabled>true</enabled>  <!-- Optional. Defaults to false. -->
    <rules>
      <rule>
        <prefixName>a</prefixName>  <!-- Optional. Default is "any". prefixName used
          here should be defined in the routingGlobalConfig->ipPrefixes -->
        <from>
          <isis>true</isis>  <!-- Optional. Defaults to false -->
          <ospf>true</ospf>  <!-- Optional. Defaults to false -->
          <bgp>false</bgp>  <!-- Optional. Defaults to false -->
          <static>true</static>  <!-- Optional. Defaults to false -->
          <connected>>false</connected>  <!-- Optional. Defaults to false -->
        </from>
        <action>deny</action>  <!-- Mandatory. Valid values are deny|permit -->
      </rule>
      <rule>
        <from>
          <isis>>false</isis>  <!-- Optional. Defaults to false -->
          <ospf>>false</ospf>  <!-- Optional. Defaults to false -->
          <bgp>>false</bgp>  <!-- Optional. Defaults to false -->
          <static>>false</static>  <!-- Optional. Defaults to false -->
          <connected>true</connected>  <!-- Optional. Defaults to false -->
        </from>
        <action>permit</action>  <!-- Mandatory. Valid values are deny|permit -->
      </rule>
    </rules>
  </redistribution>
</bgp>

```

---

## Query BGP

### Example 8-26. Query BGP

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/bgp>

Response Body:

```

<bgp>
  <enabled>true</enabled>
  <localAS>65535</localAS>
  <bgpNeighbours>
    <bgpNeighbour>
      <ipAddress>192.168.1.10</ipAddress>
      <remoteAS>65500</remoteAS>
      <weight>60</weight>
      <holdDownTimer>180</holdDownTimer>
      <keepAliveTimer>60</keepAliveTimer>
      <password>vmware123</password>
      <bgpFilters>
        <bgpFilter>
          <direction>in</direction>
          <action>permit</action>
          <network>10.0.0.0/8</network>
          <ipPrefixGe>17</ipPrefixGe>
          <ipPrefixLe>32</ipPrefixLe>
        </bgpFilter>
        <bgpFilter>
          <direction>out</direction>
          <action>deny</action>
          <network>20.0.0.0/26</network>
        </bgpFilter>
      </bgpFilters>
    </bgpNeighbour>
  </bgpNeighbours>
  <redistribution>
    <enabled>true</enabled>
    <rules>
      <rule>
        <id>1</id>
        <prefixName>a</prefixName>
        <from>
          <isis>true</isis>
          <ospf>true</ospf>
          <bgp>false</bgp>
          <static>true</static>
          <connected>false</connected>
        </from>
        <action>deny</action>
      </rule>
      <rule>
        <id>0</id>
        <from>
          <isis>false</isis>
          <ospf>false</ospf>
          <bgp>false</bgp>
          <static>false</static>
          <connected>true</connected>
        </from>
        <action>permit</action>
      </rule>
    </rules>
  </redistribution>
</bgp>

```

---

## Delete BGP

Deletes BGP routing.

### Example 8-27. Delete BGP

---

Request:

DELETE <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/bgp>

---

## Working with Bridging

You can create an L2 bridge between a logical switch and a VLAN, which enables you to migrate virtual workloads to physical devices with no impact on IP addresses. A logical network can leverage a physical gateway and access existing physical network and security resources by bridging the logical switch broadcast domain to the VLAN broadcast domain.

The L2 bridge runs on the host that has the NSX Edge logical router virtual machine. An L2 bridge instance maps to a single VLAN, but there can be multiple bridge instances. The logical router cannot be used as a gateway for devices connected to a bridge.

If High Availability is enabled on the Logical Router and the primary NSX Edge virtual machine goes down, the bridge is automatically moved over to the host with the secondary virtual machine. For this seamless migration to happen, VLAN must have been configured on the host that has the secondary NSX Edge virtual machine.

## Configure a Bridge

Configures a bridge.

### Example 8-28. Configure bridge

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/bridging/config>

Request Body:

```
<bridges>
  <version>9</version>
  <enabled>>false</enabled>
  <bridge>
    <name>test1</name>
    <virtualWire>virtualwire-1</virtualWire>
    <dvportGroup>dvportgroup-36</dvportGroup>
  </bridge>
  <bridge>
    <name>test2</name>
    <virtualWire>virtualwire-2</virtualWire>
    <dvportGroup>dvportgroup-37</dvportGroup>
  </bridge>
</bridges>
```

---

The enabled parameter in the above Request Body is ineffective and does not enable or disable the bridge. To disable a bridge, you must delete it.

## Query Bridge Configuration

Retrieves bridge configuration.

### Query BGP

### Example 8-29. Query bridges

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/bridging/config>

Response Body:

```
<bridges>
```

```
<version>4</version>
<enabled>>true</enabled>
<bridge>
  <bridgeId>1</bridgeId>
  <name>bridge1</name>
  <virtualWire>dvportgroup-23</virtualWire>
  <dvportGroup>dvportgroup-25</dvportGroup>
</bridge>
</bridges>
```

---

## Delete Bridge Configuration

Deletes bridges.

### **Example 8-30.** Delete bridges

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/bridging/config
```

---

# NSX Edge Services Gateway Management

---

# 9

NSX Edge Services Gateway gives you access to all NSX Edge services such as firewall, NAT, DHCP, VPN, load balancing, and high availability. You can install multiple NSX Edge services gateway virtual appliances in a datacenter. Each NSX Edge virtual appliance can have a total of ten uplink and internal network interfaces.

The internal interfaces connect to secured port groups and act as the gateway for all protected virtual machines in the port group. The subnet assigned to the internal interface can be a publicly routed IP space or a NATed/routed RFC 1918 private space. Firewall rules and other NSX Edge services are enforced on traffic between network interfaces.

Uplink interfaces of NSX Edge connect to uplink port groups that have access to a shared corporate network or a service that provides access layer networking. Multiple external IP addresses can be configured for load balancer, site-to-site VPN, and NAT services.

After you install network virtualization components and one or more logical switches in your environment, you can secure internal networks by installing a Edge Services gateway.

This chapter includes the following topics:

- [“Query Installed Edges”](#) on page 158
- [“Modifying NSX Edge Configuration”](#) on page 162
- [“System Control Edge Configuration”](#) on page 166
- [“Deleting NSX Edge”](#) on page 168
- [“Configuring Edge Services in Async Mode”](#) on page 168
- [“Configuring Certificates”](#) on page 169
- [“Working with NSX Edge Firewall”](#) on page 172
- [“Working with Routing”](#) on page 186
- [“Working with Load Balancer”](#) on page 200
- [“Configure DNS Servers”](#) on page 222
- [“Working with DHCP Service”](#) on page 224
- [“Working with DHCP Relay”](#) on page 228
- [“Working with High Availability \(HA\)”](#) on page 229
- [“Working with Syslog”](#) on page 231
- [“Managing SSL VPN”](#) on page 232
- [“Working with L2 VPN”](#) on page 262
- [“Working with IPSEC VPN”](#) on page 266

- [“Managing an NSX Edge”](#) on page 270

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Query Installed Edges

You can retrieve a list of NSX Edges in your inventory or filter the results by datacenter or port group.

### Example 9-1. Retrieve Edges

---

Retrieve all Edges Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/
```

Retrieve Edges by datacenter:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/?datacenter=datacenterMoid
```

Retrieve Edges on specified tenant:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/?tenant=tenantId
```

Retrieve Edges with one interface on specified port group:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/?pg=pgMoid
```

Retrieve Edges with specified tenant and port group:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/?tenant=tenant&pg=pgMoid
```

---

### Example 9-2. Retrieve Edge details

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId
```

Response Body:

```
<edge>
  <id>edge-79</id>
  <version>5</version>
  <description>testEdge</description>
  <status>deployed</status>
  <datacenterMoid>datacenter-2</datacenterMoid>
  <datacenterName>datacenterForEdge</datacenterName>
  <name>testEdge</name>
  <fqdn>testEdge</fqdn>
  <enableAesni>true</enableAesni>
  <enableFips>false</enableFips>
  <vseLogLevel>info</vseLogLevel>
  <edgeAssistId>1460487509</edgeAssistId>
  <vnics>
    <vnic>
      <index>0</index>
      <name>uplink-vnic-network-2581</name>
      <type>uplink</type>
      <portgroupId>network-2581</portgroupId>
      <portgroupName>Mgmt</portgroupName>
      <addressGroups>
        <addressGroup>
          <primaryAddress>192.168.3.1</primaryAddress>
          <secondaryAddresses>
            <ipAddress>192.168.3.2</ipAddress>
            <ipAddress>192.168.3.3</ipAddress>
          </secondaryAddresses>
          <subnetMask>255.255.255.0</subnetMask>
        </addressGroup>
      </addressGroups>
    </vnic>
  </vnics>
</edge>
```

```

<addressGroup>
  <primaryAddress>192.168.4.1</primaryAddress>
  <secondaryAddresses>
    <ipAddress>192.168.4.2</ipAddress>
    <ipAddress>192.168.4.3</ipAddress>
  </secondaryAddresses>
  <subnetMask>255.255.255.0</subnetMask>
  <!-- GET will always have subnetMask field for ipv4 and subnetPrefixLength
  for ipv6 -->
</addressGroup>
<addressGroup>
  <primaryAddress>ffff::1</primaryAddress>
  <secondaryAddresses>
    <ipAddress>ffff::2</ipAddress>
  </secondaryAddresses>
  <subnetPrefixLength>64</subnetPrefixLength>
</addressGroup>
</addressGroups>
<mtu>1500</mtu>
<enableProxyArp>>false</enableProxyArp>
<enableSendRedirects>>true</enableSendRedirects>
<isConnected>>true</isConnected>
</vnic>
.....
</vnics>
<appliances>
  <applianceSize>compact</applianceSize>
  <appliance>
    <highAvailabilityIndex>0</highAvailabilityIndex>
    <vcUuid>4208f392-1693-11db-6355-4affd859ef33</vcUuid>
    <vmId>vm-4021</vmId>
    <resourcePoolId>resgroup-2454</resourcePoolId>
    <resourcePoolName>Resources</resourcePoolName>
    <datastoreId>datastore-2457</datastoreId>
    <datastoreName>shahm-esx-storage</datastoreName>
    <hostId>host-2455</hostId>
    <hostName>10.112.196.160</hostName>
    <vmFolderId>group-v3</vmFolderId>
    <vmFolderName>vm</vmFolderName>
    <vmHostname>vShieldEdge-network-2264-0</vmHostname>
    <vmName>vShield-edge-79-0</vmName>
    <deployed>>true</deployed>
    <edgeId>edge-79</edgeId>
  </appliance>
</appliances>
<cliSettings>
  <remoteAccess>>false</remoteAccess>
  <userName>admin</userName>
</cliSettings>
<features>
  <featureConfig/>
  <firewall>
    <version>1</version>
    <enabled>>true</enabled>
    <defaultPolicy>
      <action>deny</action>
      <loggingEnabled>>false</loggingEnabled>
    </defaultPolicy>
    <rules>
      <rule>
        <id>131078</id>
        <ruleTag>131078</ruleTag>
        <name>rule1</name>
        <ruleType>user</ruleType>
        <source>
          <exclude>
            false
          </exclude>
        </source>
      </rule>
    </rules>
  </firewall>
</features>

```

```

        <groupingObjectId>ipset-938</groupingObjectId>
    </source>
    <destination/>
    <exclude>
        false
    </exclude>
    <application>
        <applicationId>application-666</applicationId>
    </application>
    <action>accept</action>
    <enabled>true</enabled>
    <loggingEnabled>>false</loggingEnabled>
    <matchTranslated>>false</matchTranslated>
</rule>
    ...
</rules>
</firewall>
<routing>
    <version>1</version>
    <enabled>true</enabled> <!-- A read-only field. Cannot be set to false as
        routing cannot be disabled -->
    <staticRouting>
        <defaultRoute>
            <vnic>0</vnic>
            <gatewayAddress>10.112.3.253</gatewayAddress>
            <description>defaultGw on the external interface</description>
        </defaultRoute>
        <staticRoutes>
            <route>
                <vnic>0</vnic>
                <network>192.168.30.0/24</network>
                <nextHop>10.112.2.41</nextHop>
                <type>user</type>
            </route>
            ...
        </staticRoutes>
    </staticRouting>
    <ospf>
        <enabled>>false</enabled>
    </ospf>
</routing>
<highAvailability>
    <version>1</version>
    <enabled>>false</enabled>
    <declareDeadTime>6</declareDeadTime>
    <logging>
        <enable>>false</enable>
        <logLevel>info</logLevel>
    </logging>
</highAvailability>
<syslog>
    <version>1</version>
    <enabled>true</enabled>
    <protocol>udp</protocol>
    <serverAddresses>
        <ipAddress>1.1.1.1</ipAddress>
        <ipAddress>1.1.1.2</ipAddress>
    </serverAddresses>
</syslog>
<ipsec>
    <version>1</version>
    <enabled>true</enabled>
    <logging>
        <enable>>false</enable>
        <logLevel>info</logLevel>
    </logging>
    <sites>
        <site>

```

```

    <enabled>true</enabled>
    <name>site1</name>
    <localId>10.112.2.40</localId>
    <localIp>10.112.2.40</localIp>
    <peerId>10.112.2.41</peerId>
    <peerIp>10.112.2.41</peerIp>
    <encryptionAlgorithm>aes256</encryptionAlgorithm>
    <mtu>1500</mtu>
    <enablePfs>true</enablePfs>
    <dhGroup>dh2</dhGroup>
    <localSubnets>
      <subnet>192.168.10.0/24</subnet>
    </localSubnets>
    <peerSubnets>
      <subnet>192.168.40.0/24</subnet>
    </peerSubnets>
    <psk>1234</psk>
    <authenticationMode>psk</authenticationMode>
  </site>
  ....
</sites>
<global>
  <caCertificates/>
  <crlCertificates/>
</global>
</ipsec>
<dhcp>
  <version>1</version>
  <enabled>>false</enabled>
  <staticBindings>
    <staticBinding>
      <autoConfigureDNS>true</autoConfigureDNS>
      <bindingId>binding-1</bindingId>
      <vmId>vm-2460</vmId>
      <vnicId>1</vnicId>
      <hostname>test</hostname>
      <ipAddress>192.168.10.6</ipAddress>
      <defaultGateway>192.168.10.1</defaultGateway>
      <leaseTime>86400</leaseTime>
    </staticBinding>
    ....
  </staticBindings>
  <ipPools>
    <ipPool>
      <autoConfigureDNS>true</autoConfigureDNS>
      <poolId>pool-1</poolId>
      <ipRange>192.168.10.2-192.168.10.5</ipRange>
      <defaultGateway>192.168.10.1</defaultGateway>
      <leaseTime>86400</leaseTime>
    </ipPool>
    ....
  </ipPools>
  <logging>
    <enable>>false</enable>
    <logLevel>info</logLevel>
  </logging>
</dhcp>
<nat>
  <version>1</version>
  <enabled>true</enabled>
  <natRules>
    <natRule>
      <ruleId>196610</ruleId>
      <ruleTag>196610</ruleTag>
      <ruleType>user</ruleType>
      <action>dnat</action>
      <vnic>1</vnic>
      <originalAddress>10.112.196.162</originalAddress>

```

```

        <translatedAddress>192.168.10.3</translatedAddress>
        <loggingEnabled>>false</loggingEnabled>
        <enabled>>true</enabled>
        <protocol>tcp</protocol>
        <originalPort>80</originalPort>
        <translatedPort>80</translatedPort>
    </natRule>
    ....
</natRules>
</nat>
<featureConfig/>
</features>
<autoConfiguration>
    <enabled>>true</enabled>
    <rulePriority>high</rulePriority>
</autoConfiguration>
<dnsClient>
    <primaryDns>10.117.0.1</primaryDns>
    <secondaryDns>10.117.0.2</secondaryDns>
    <domainName>vmware.com</domainName>
    <domainName>foo.com</domainName>
</dnsClient>
<queryDaemon>
    <enabled>>true</enabled>
    <port>5666</port>
</queryDaemon>
</edge>

```

---

## Modifying NSX Edge Configuration

Replaces current NSX Edge configuration.

### Example 9-3. Modify Edge configuration

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId>

Request Body:

```

<edge>
  <id>edge-79</id>
  <description>testEdge</description>
  <datacenterMoid>datacenter-2</datacenterMoid>
  <name>testEdge</name>
  <fqdn>testEdge</fqdn>
  <enableAesni>true</enableAesni>
  <enableFips>false</enableFips>
  <vseLogLevel>info</vseLogLevel>
  <vnics>
    <vnic>
      <index>0</index>
      <name>uplink-vnic-network-2581</name>
      <type>uplink</type>
      <portgroupId>network-2581</portgroupId>
      <addressGroups>
        <addressGroup> <!-- Vnic can be configured to have more than one
          addressGroup/subnets -->
          <primaryAddress>192.168.3.1</primaryAddress> <!-- This is mandatory for
          an addressGroup -->
          <secondaryAddresses> <!-- Optional. Should be used to add/defined other
          IPs used for NAT, LB, VPN, etc -->
            <ipAddress>192.168.3.2</ipAddress>
            <ipAddress>192.168.3.3</ipAddress> <!-- Optional. This way multiple
          IP Addresses can be assigned to a vnic/interface -->
          </secondaryAddresses>
        </addressGroup>
      </addressGroups>
    </vnic>
  </vnics>
</edge>

```

```

    <subnetMask>255.255.255.0</subnetMask> <!-- either subnetMask or
    subnetPrefixLength should be provided. If both then subnetprefixLength is
    ignored -->
  </addressGroup>
  <addressGroup> <!-- vnic can be configured to have more than one
  addressGroup/subnets -->
    <primaryAddress>192.168.4.1</primaryAddress> <!-- This is mandatory for
    an addressGroup -->
    <secondaryAddresses> <!-- Optional. Should be used to add/defined other
    IPs used for NAT, LB, VPN, etc -->
      <ipAddress>192.168.4.2</ipAddress>
      <ipAddress>192.168.4.3</ipAddress> <!-- Optional. This way multiple
      IP Addresses can be assigned to a vnic/interface -->
    </secondaryAddresses>
    <subnetPrefixLength>24</subnetPrefixLength> <!-- subnetPrefixLength
    valid values for ipv4 1-32 -->
  </addressGroup>
  <addressGroup> <!-- ipv6 addressGroup -->
    <primaryAddress>ffff::1</primaryAddress> <!-- This is mandatory for an
    addressGroup -->
    <secondaryAddresses> <!-- Optional. Should be used to add/defined other
    IPs used for NAT, LB, VPN, etc -->
      <ipAddress>ffff::2</ipAddress>
    </secondaryAddresses>
    <subnetPrefixLength>64</subnetPrefixLength> <!-- subnetPrefixLength
    valid values 1-128 -->
  </addressGroup>
</addressGroups>
<mtu>1500</mtu>
<enableProxyArp>false</enableProxyArp>
<enableSendRedirects>true</enableSendRedirects>
<isConnected>true</isConnected>
<inShapingPolicy> <!-- optional -->
  <averageBandwidth>200000000</averageBandwidth>
  <peakBandwidth>200000000</peakBandwidth>
  <burstSize>0</burstSize>
  <enabled>true</enabled>
  <inherited>false</inherited>
</inShapingPolicy>
<outShapingPolicy> <!-- optional -->
  <averageBandwidth>400000000</averageBandwidth>
  <peakBandwidth>400000000</peakBandwidth>
  <burstSize>0</burstSize>
  <enabled>true</enabled>
  <inherited>false</inherited>
</outShapingPolicy>
</vnic>
</vnic>
.....
</vnics>
<appliances>
  <applianceSize>compact</applianceSize>
  <appliance>
    <resourcePoolId>resgroup-2454</resourcePoolId>
    <datastoreId>datastore-2457</datastoreId>
    <vmFolderId>group-v3</vmFolderId>
  </appliance>
</appliances>
<cliSettings>
  <remoteAccess>false</remoteAccess>
  <userName>admin</userName>
</cliSettings>
<features>
  <firewall>
    <defaultPolicy>
      <action>deny</action>
      <loggingEnabled>false</loggingEnabled>
    </defaultPolicy>

```

```

<rules>
  <rule>
    <id>131078</id>
    <ruleTag>131078</ruleTag>
    <name>rule1</name>
    <ruleType>user</ruleType>
    <source>
      <exclude>
        false
      </exclude>
      <groupingObjectId>ipset-938</groupingObjectId>
    </source>
    <destination/>
    <application>
      <applicationId>application-666</applicationId>
    </application>
    <action>accept</action>
    <enabled>true</enabled>
    <loggingEnabled>>false</loggingEnabled>
    <matchTranslated>>false</matchTranslated>
  </rule>
  ...
</rules>
</firewall>
<routing>
  <staticRouting>
    <defaultRoute>
      <vnic>0</vnic>
      <gatewayAddress>10.112.3.253</gatewayAddress>
      <description>defaultGw on the external interface</description>
    </defaultRoute>
    <staticRoutes>
      <route>
        <vnic>0</vnic>
        <network>192.168.30.0/24</network>
        <nextHop>10.112.2.41</nextHop>
        <type>user</type>
      </route>
      ...
    </staticRoutes>
  </staticRouting>
  <ospf>
    <enabled>>false</enabled>
  </ospf>
</routing>
<highAvailability>
  <enabled>>false</enabled>
  <declareDeadTime>6</declareDeadTime>
  <logging>
    <enable>>false</enable>
    <logLevel>info</logLevel>
  </logging>
</highAvailability>
<syslog>
  <protocol>udp</protocol>
  <serverAddresses>
    <ipAddress>1.1.1.1</ipAddress>
    <ipAddress>1.1.1.2</ipAddress>
  </serverAddresses>
</syslog>
<ipsec>
  <enabled>true</enabled>
  <logging>
    <enable>>false</enable>
    <logLevel>info</logLevel>
  </logging>
<sites>
  <site>

```

```

    <enabled>true</enabled>
    <name>site1</name>
    <localId>10.112.2.40</localId>
    <localIp>10.112.2.40</localIp>
    <peerId>10.112.2.41</peerId>
    <peerIp>10.112.2.41</peerIp>
    <encryptionAlgorithm>aes256</encryptionAlgorithm>
    <mtu>1500</mtu>
    <enablePfs>true</enablePfs>
    <dhGroup>dh2</dhGroup>
    <localSubnets>
      <subnet>192.168.10.0/24</subnet>
    </localSubnets>
    <peerSubnets>
      <subnet>192.168.40.0/24</subnet>
    </peerSubnets>
    <psk>1234</psk>
    <authenticationMode>psk</authenticationMode>
  </site>
  ....
</sites>
<global>
  <caCertificates/>
  <crlCertificates/>
</global>
</ipsec>
<dhcp>
  <enabled>true</enabled>
  <staticBindings>
    <staticBinding>
      <autoConfigureDNS>true</autoConfigureDNS>
      <bindingId>binding-1</bindingId>
      <vmId>vm-2460</vmId>
      <vnicId>1</vnicId>
      <hostname>test</hostname>
      <ipAddress>192.168.10.6</ipAddress>
      <defaultGateway>192.168.10.1</defaultGateway>
      <leaseTime>86400</leaseTime>
    </staticBinding>
    ....
  </staticBindings>
  <ipPools>
    <ipPool>
      <autoConfigureDNS>true</autoConfigureDNS>
      <poolId>pool-1</poolId>
      <ipRange>192.168.10.2-192.168.10.5</ipRange>
      <defaultGateway>192.168.10.1</defaultGateway>
      <leaseTime>86400</leaseTime>
    </ipPool>
    ....
  </ipPools>
  <logging>
    <enable>false</enable>
    <logLevel>info</logLevel>
  </logging>
</dhcp>
<nat>
  <natRules>
    <natRule>
      <ruleId>196610</ruleId>
      <ruleTag>196610</ruleTag>
      <ruleType>user</ruleType>
      <action>dnat</action>
      <vnic>1</vnic> <!-- Optional -->
      <originalAddress>10.112.196.162</originalAddress>
      <translatedAddress>192.168.10.3</translatedAddress>
      <loggingEnabled>false</loggingEnabled>
      <enabled>true</enabled>
    </natRule>
  </natRules>

```

```

        <protocol>tcp</protocol>
        <originalPort>80</originalPort>
        <translatedPort>80</translatedPort>
    </natRule>
    . . .
</natRules>
</nat>
</features>
<autoConfiguration>
    <enabled>true</enabled>
    <rulePriority>high</rulePriority>
</autoConfiguration>
</edge>

```

---

where *groupingObjectId* can be cluster, network, etc.

## System Control Edge Configuration

### RP\_FILTER:

RP\_FILTER stands for Reverse Path Filter and has values

- 0 - DISABLE (Disabled on UI)
- 1 - ENABLE (Enabled on UI)
- 2 - Loose (Loose on UI)

Reverse Path Filtering is generally used to enable/disable asymmetric routing.

When an interface has `rp_filter` Enabled (value 1), the packet is accepted only if the source IP mentioned in the packet is routable on the same interface, otherwise the packet is dropped. This is helpful to prevent IP address spoofing. On the other hand, legitimate packets may also be dropped whenever asymmetric routes are present in the network, with `rp_filter` enabled.

When the interface has the `rp_filter` value set to Loose (value 2), the packet is accepted if the source IP mentioned in the packet is routable on any of the configured interface, otherwise the packet is dropped. This mode is normally set whenever the network has asymmetric routes.

No validations for the source IP are carried out when `rp_filter` is set in Disabled mode (value 0).

**Allowed System Control parameters** (inside the `<property></property>` tags in the APIs below):

#### **arp\_announce to decide the IP address to go out in ARP:**

```

sysctl.net.ipv4.conf.all.arp_announce
sysctl.net.ipv4.conf.default.arp_announce

```

#### **tcp timeout values for conntrack to fine tune NAT perf:**

```

sysctl.net.netfilter.nf_conntrack_tcp_timeout_fin_wait
sysctl.net.netfilter.nf_conntrack_tcp_timeout_close_wait
sysctl.net.netfilter.nf_conntrack_tcp_timeout_max_retrans
sysctl.net.netfilter.nf_conntrack_tcp_timeout_unacknowledged
sysctl.net.netfilter.nf_conntrack_tcp_max_retrans

```

#### **rp\_filter to disable uRPF check:**

```

sysctl.net.ipv4.conf.all.rp_filter
sysctl.net.ipv4.conf.default.rp_filter

```

**sysctl.net.ipv4.conf.vNic\_[0-4094].rp\_filter is handled by regex in validation logic and is not specified explicitly here.**

#### **Tweaking ARP limits in cache:**

```

sysctl.net.ipv4.neigh.default.gc_thresh1

```

```

sysctl.net.ipv6.neigh.default.gc_thresh1
sysctl.net.ipv4.neigh.default.gc_thresh2
sysctl.net.ipv6.neigh.default.gc_thresh2
sysctl.net.ipv4.neigh.default.gc_thresh3
sysctl.net.ipv6.neigh.default.gc_thresh3

```

#### TIME\_WAIT socket connections Reuse Recycle configuration.

```

sysctl.net.ipv4.tcp_tw_reuse
sysctl.net.ipv4.tcp_tw_recycle

```

#### Load balancer tuning parameters

```

lb.global.tune.bufsize
lb.global.tune.maxrewrite
sysctl.net.ipv4.vs.expire_nodest_conn

```

#### Example 9-4. Update System Control (sysctl) Configuration

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/systemcontrol/config>

Request Body:

```

<systemControl>
  <property>sysctl.net.ipv4.conf.vNic_1.rp_filter=2</property>
  <property>sysctl.net.ipv4.conf.vNic_2.rp_filter=2</property>
  <property>sysctl.net.ipv4.conf.vNic_3.rp_filter=2</property>
  <property>sysctl.net.netfilter.nf_conntrack_tcp_timeout_syn_sent=30</property>
  <property>sysctl.net.netfilter.nf_conntrack_tcp_timeout_syn_recv=20</property>
  <property>sysctl.net.netfilter.nf_conntrack_tcp_timeout_established=3660</property>
  <property>sysctl.net.netfilter.nf_conntrack_tcp_timeout_time_wait=25</property>
  <property>sysctl.net.netfilter.nf_conntrack_udp_timeout=30</property>
  <property>sysctl.net.netfilter.nf_conntrack_udp_timeout_stream=40</property>
  <property>sysctl.net.netfilter.nf_conntrack_icmp_timeout=20</property>
  <property>sysctl.net.netfilter.nf_conntrack_icmpv6_timeout=20</property>
  <property>sysctl.net.netfilter.nf_conntrack_generic_timeout=180</property>
</systemControl>

```

---

#### Example 9-5. Query System Control Configuration

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/systemcontrol/config>

Response Body:

```

<systemControl>
  <property>sysctl.net.ipv4.conf.vNic_1.rp_filter=2</property>
  <property>sysctl.net.ipv4.conf.vNic_2.rp_filter=2</property>
  <property>sysctl.net.ipv4.conf.vNic_3.rp_filter=2</property>
  <property>sysctl.net.netfilter.nf_conntrack_tcp_timeout_syn_sent=30</property>
  <property>sysctl.net.netfilter.nf_conntrack_tcp_timeout_syn_recv=20</property>
  <property>sysctl.net.netfilter.nf_conntrack_tcp_timeout_established=3660</property>
  <property>sysctl.net.netfilter.nf_conntrack_tcp_timeout_time_wait=25</property>
  <property>sysctl.net.netfilter.nf_conntrack_udp_timeout=30</property>
  <property>sysctl.net.netfilter.nf_conntrack_udp_timeout_stream=40</property>
  <property>sysctl.net.netfilter.nf_conntrack_icmp_timeout=20</property>
  <property>sysctl.net.netfilter.nf_conntrack_icmpv6_timeout=20</property>
  <property>sysctl.net.netfilter.nf_conntrack_generic_timeout=180</property>
</systemControl>

```

---

NOTE : The above API will give an empty list if user has never modified any property using the PUT API.

**Example 9-6.** Delete System Control Configuration

Request:

DELETE <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/systemcontrol/config?rebootNow>

## Deleting NSX Edge

Deletes specified Edge from database. Associated appliances are also deleted.

**Example 9-7.** Delete Edge

Request:

DELETE <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId>

## Configuring Edge Services in Async Mode

You can configure Edge to work in async mode. In the async mode, accepted commands return an Accepted status and a taskId. To know the status of the task, you can check the status of that taskId.

The advantage of the async mode is that APIs are returned very fast and actions like vm deployment, reboots, publish to Edge appliance, etc are done behind the scene under the taskId .

To configure async mode, `?async=true` at the end of any 4.0 service configuration URL for POST, PUT, and DELETE calls. Without async mode, the location header in HTTP response has the resource ID whereas in async mode, location header has the job ID.

### Query Async Job Status

Retrieves job status (SUCCESS/FAILED/QUEUED/RUNNING/ROLLBACK), URI of the resource, and ID of the resource as shown in output representation.

**Example 9-8.** Query job status

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/jobs/jobId>

Response Body:

```

<edgeJob>
  <jobId>jobdata-2128</jobId>
  <message>Deploying vShield Edge Virtual Machine TestEdge11-0</message>
  <status>RUNNING</status>
  <result>
    <key>ResultURI</key>
    <value>/api/4.0/edges/edge-4</value>
  </result>
  <result>
    <key>edgeId</key>
    <value>edge-4</value>
  </result>
</edgeJob>

```

### Query all Jobs

**Example 9-9.** Query all jobs

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/jobs?status=all
```

Request Body:

```
<edgeJobs>
  <edgeJob>
    <jobId>jobdata-917</jobId>
    <status>COMPLETED</status>
    <result>
      <key>edgeId</key>
      <value>edge-4</value>
    </result>
  </edgeJob>
  <edgeJob>
    <jobId>jobdata-915</jobId>
    <status>COMPLETED</status>
    <result>
      <key>edgeId</key>
      <value>edge-4</value>
    </result>
  </edgeJob>
</edgeJobs>
```

---

## Query active Jobs

**Example 9-10.** Query active jobs

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/jobs?status=active
```

Request Body:

```
<edgeJobs>
  <edgeJob>
    <jobId>jobdata-917</jobId>
    <message>Publishing configurations on vShield Edge Virtual Machine vm-65</message>
    <status>RUNNING</status>
    <result>
      <key>edgeId</key>
      <value>edge-4</value>
    </result>
  </edgeJob>
</edgeJobs>
```

---

## Configuring Certificates

vShield Edge supports self-signed certificates, certificates signed by a Certification Authority (CA), and certificates generated and signed by a CA.

### Working with Certificates

Allows you to manage self signed certificates.

#### Create Certificate

Creates a single or multiple certificates.

**Example 9-11.** Create self signed certificate

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/truststore/certificate/edgeId
```

Request Body:

```
<trustObject>
  <pemEncoding></pemEncoding>
  <privateKey></privateKey>
  <passphrase></passphrase>
</trustObject>
```

---

### Create Certificate or Certificate Chain for CSR

Imports a certificate or a certificate chain against a certificate signing request.

#### Example 9-12. Create certificate for CSR

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/truststore/certificate?csrId=csrId
```

Request Body:

```
<trustObject>
  <pemEncoding></pemEncoding>
</trustObject>
```

---

### Query Certificates

Retrieves the certificate object for the specified certificate ID. If the certificate ID is a chain, multiple certificate objects are retrieved.

#### Example 9-13. Query specific certificate

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/truststore/certificate/certificateId
```

---

#### Example 9-14. Query all certificates for a scope

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/truststore/certificate/scope/scopeId
```

---

In this case, *scopeId* can be either *globalroot-0* or the *edgeId* for the relevant NSX Edge.

### Delete Certificate

Deletes the specified certificate.

#### Example 9-15. Delete certificate

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/truststore/certificate
/certificateId
```

---

## Working with Certificate Signing Requests (CSRs)

Allows you to manage CSRs.

## Create CSR

### Example 9-16. Create CSR

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/services/truststore/csr/edgeId>

Request Body:

```

<csr>
  <subject>
    <attribute>
      <key>CN</key>
      <value>VSM</value>
    </attribute>
    <attribute>
      <key>O</key>
      <value>VMware</value>
    </attribute>
    <attribute>
      <key>OU</key>
      <value>IN</value>
    </attribute>
    <attribute>
      <key>C</key>
      <value>IN</value>
    </attribute>
  </subject>
  <algorithm>RSA</algorithm>
  <keySize>1024</keySize>
</csr>

```

---

## Create Self Signed Certificate for CSR

### Example 9-17. Create self signed certificate for CSR

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/services/truststore/csr/csrId?noOfDays=value>

---

## Query CSRs

Retrieves specified CSR or all CSRs for specified scope.

### Example 9-18. Query specific CSR

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/truststore/csr/csrId>

---

### Example 9-19. Query CSRs for specific scope

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/truststore/csr/scope/edgeId>

Request Body:

```

<csrs>
  <csr>
    ...
  </csr>
  <csr>

```

```

    ...
  </csr>
  ...
</csrs>

```

---

## Working with Certificate Revocation List (CRL)

Allows you to manage CRLs.

### Create a CRL

Creates a CRL on the specified scope.

#### Example 9-20. Create CRL

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/truststore/crl/edgeId
```

Request Body:

```

<trustObject>
  <pemEncoding></pemEncoding>
</trustObject>

```

---

### Query CRL

Retrieves all CRLs certificates for the specified certificate or scope.

#### Example 9-21. Query CRL

---

Retrieve certificate object for the specified certificate ID:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/truststore/crl/crId
```

Retrieve all certificates for the specified scope:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/truststore/crl/scope/edgeId
```

---

### Delete CRL

Deletes the specified CRL.

#### Example 9-22. Delete CRL

---

Request

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/truststore/crl/crId
```

---

## Working with NSX Edge Firewall

Edge Firewall provides perimeter security functionality including firewall, Network Address Translation (NAT) as well as Site to site IPSec and SSL VPN functionality. This solution is available in the virtual machine form factor and can be deployed in a High Availability mode.

Rules created at an NSX Edge level are referred to as local rules. These rules are not displayed at the globalroot-0 level.

## Configure Firewall

Configures firewall for an Edge and stores the specified configuration in database. If any appliance(s) are associated with this Edge, applies the configuration to these. While using this API, you should send the `globalConfig`, `defaultPolicy` and the rules. If either of them are not sent, the previous config if any on those fields will be removed and will be changed to the system defaults.

Starting in NSX 6.2.3, NSX Edge has the ability to protect against SYN flood attacks by detecting bogus TCP connections and terminating them without consuming firewall state tracking resources. This feature is disabled (false) by default. Set `enableSynFloodProtection` to true to enable.

### Example 9-23. Configure firewall

Request:

PUT `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config`

Request Body:

```
<firewall>
  <defaultPolicy>  <!-- Optional. default is deny -->
    <action>deny</action>
    <loggingEnabled>false</loggingEnabled>  <!-- Optional. Defaults to false -->
  </defaultPolicy>
  <globalConfig>  <!-- Optional -->
    <tcpPickOngoingConnections>false</tcpPickOngoingConnections>  <!-- Optional.
      Defaults to false -->
    <tcpAllowOutOfWindowPackets>false</tcpAllowOutOfWindowPackets>  <!-- Optional.
      Defaults to false -->
    <tcpSendResetForClosedVsePorts>true</tcpSendResetForClosedVsePorts>  <!-- Optional.
      Defaults to true -->
    <dropInvalidTraffic>true</dropInvalidTraffic>  <!-- Optional. Defaults to true -->
    <logInvalidTraffic>false</logInvalidTraffic>  <!-- Optional. Defaults to false -->
    <tcpTimeoutOpen>30</tcpTimeoutOpen>  <!-- Optional. Defaults to 30 -->
    <tcpTimeoutEstablished>21600</tcpTimeoutEstablished>  <!-- Optional. Defaults to
      21600 (6 hours) -->
    <tcpTimeoutClose>30</tcpTimeoutClose>  <!-- Optional. Defaults to 30 -->
    <udpTimeout>60</udpTimeout>  <!-- Optional. Defaults to 60 -->
    <icmpTimeout>10</icmpTimeout>  <!-- Optional. Defaults to 10 -->
    <icmp6Timeout>10</icmp6Timeout>  <!-- Optional. Defaults to 10 -->
    <ipGenericTimeout>120</ipGenericTimeout>  <!-- Optional. Defaults to 120 -->
    <enableSynFloodProtection>false</enableSynFloodProtection> <!-- Defaults to false
      -->
  </globalConfig>
  <firewallRules>
    <firewallRule>
      <ruleTag>1</ruleTag>  <!-- Optional. This can be used to specify user
        controlled ids on VSE. The inputs here should be 1-65536. If not specified,
        VSM will generate ruleId -->
      <name>rule1</name>  <!-- Optional -->
      <source>  <!-- Optional. Default behaviour is like "any". ipsetId or
        predefined-vnicGroupIds can be used -->
        <vnicGroupId>vnic-index-5</vnicGroupId>  <!-- Possible values are
          "vnic-index-[0-9]", "vse", "external" or "internal". Can define multiple of
          these -->
        <groupingObjectId>ipset-128</groupingObjectId>  <!-- Id of IPAddresses
          grouping Objects available to the edge. Can define multiple of these -->
        <ipAddress>1.1.1.1</ipAddress>  <!-- Possible formats are IP, IP1-IPn, CIDR.
          Can define multiple of these -->
      </source>
      <destination>  <!-- Optional. Default behaviour is like "any". ipsetId or
        predefined-vnicGroupIds can be used -->
        <groupingObjectId>ipset-126</groupingObjectId>  <!-- Id of IPAddresses
          grouping Objects available to the edge. Can define multiple of these -->
        <vnicGroupId>vnic-index-5</vnicGroupId>  <!-- Possible values are
          "vnic-index-[0-9]", "vse", "external" or "internal". Can define multiple of
          these -->
    </firewallRule>
  </firewallRules>
</firewall>
```

```

    <groupingObjectId>ipset-128</groupingObjectId>  <!-- Id of IPAddresses
    grouping Objects available to the edge. Can define multiple of these -->
    <ipAddress>192.168.10.0/24</ipAddress>  <!-- Possible formats are IP,
    IP1-IPn, CIDR. Can define multiple of these -->
  </destination>
  <application>  <!-- Optional. Default behaviour is like "any". applicationsetId
  or applicationgroupId can be used -->
    <applicationId>application-155</applicationId>  <!-- Id of service available
    to the edge. Can define multiple of these -->
    <service>  <!-- Can define multiple of these -->
      <protocol>tcp</protocol>
    <port>80</port>  <!-- Default is "any". Can define multiple of these -->
      <sourcePort>1500</sourcePort>  <!-- Default is "any". Can define multiple
      of these -->
    </service>
  </application>
  <matchTranslated>true</matchTranslated>  <!-- Optional. Default behaviour is
  like "false" -->
  <direction>in</direction>  <!-- Optional. Default behaviour is like "any".
  Possible values are in|out -->
  <action>accept</action>  <!-- Mandatory. Possible values are
  accept|deny|reject-->
  <enabled>true</enabled>  <!-- Optional. Defaults to true -->
  <loggingEnabled>true</loggingEnabled>  <!-- Optional. Defaults to false -->
  <description>comments</description>  <!-- Optional -->
</firewallRule>
<firewallRule>
  ...
</firewallRule>
  .....
</firewallRules>
</firewall>

```

---

where *ruleId* uniquely identifies a rule and should be specified only for rules that are being updated.

If *ruleTag* is specified, the rules on Edge are configured using this user input. Otherwise, Edge is configured using *ruleIds* generated by NSX Manager.

## Query Firewall Configuration

Retrieves firewall configuration on specified Edge.

### Example 9-24. Query firewall

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config
```

Response Body:

```

<firewall>
  <version>1</version>
  <enabled>true</enabled>
  <defaultPolicy>
    <action>deny</action>
    <loggingEnabled>>false</loggingEnabled>
  </defaultPolicy>
  <globalConfig>
    <tcpPickOngoingConnections>>false</tcpPickOngoingConnections>
    <tcpAllowOutOfWindowPackets>>false</tcpAllowOutOfWindowPackets>
    <tcpSendResetForClosedVsePorts>true</tcpSendResetForClosedVsePorts>
    <dropInvalidTraffic>true</dropInvalidTraffic>
    <logInvalidTraffic>>false</logInvalidTraffic>
    <tcpTimeoutOpen>30</tcpTimeoutOpen>
    <tcpTimeoutEstablished>21600</tcpTimeoutEstablished>
    <tcpTimeoutClose>30</tcpTimeoutClose>
    <udpTimeout>60</udpTimeout>

```

```

<icmpTimeout>10</icmpTimeout>
<icmp6Timeout>10</icmp6Timeout>
<ipGenericTimeout>120</ipGenericTimeout>
<enableSynFloodProtection>false</enableSynFloodProtection>
</globalConfig>
<firewallRules>
  <firewallRule>
    <id>131079</id>
    <ruleTag>131079</ruleTag>
    <name>firewall</name>
    <ruleType>internal_high</ruleType>
    <source>
      <vnicGroupId>vse</vnicGroupId>
    </source>
    <action>accept</action>
    <enabled>true</enabled>
    <loggingEnabled>false</loggingEnabled>
    <description>firewall</description>
  </firewallRule>
  <firewallRule>
    <id>131080</id>
    <ruleTag>131080</ruleTag>
    <name>ipsec</name>
    <ruleType>internal_high</ruleType>
    <source>
      <groupingObjectId>ipset-934</groupingObjectId>
      <groupingObjectId>ipset-933</groupingObjectId>
    </source>
    <destination>
      <groupingObjectId>ipset-934</groupingObjectId>
      <groupingObjectId>ipset-933</groupingObjectId>
    </destination>
    <application>
      <applicationId>application-661</applicationId>
      <applicationId>application-662</applicationId>
    </application>
    <action>accept</action>
    <enabled>true</enabled>
    <loggingEnabled>false</loggingEnabled>
    <description>ipsec</description>
  </firewallRule>
  <firewallRule>
    <id>131077</id>
    <ruleTag>131077</ruleTag>
    <name>name1</name>
    <ruleType>user</ruleType>
    <source>
      <groupingObjectId>ipset-940</groupingObjectId>
      <ipAddress>1.1.1.1</ipAddress>  <!-- IP -->
      <ipAddress>2.2.2.2/24</ipAddress>  <!-- CIDR -->
      <ipAddress>1.1.1.1-1.1.1.10</ipAddress>  <!-- IP Range -->
    </source>
    <destination>
      <groupingObjectId>ipset-941</groupingObjectId>
      <vnicGroupId>vse</vnicGroupId>
      <vnicGroupId>external</vnicGroupId>
    </destination>
    <application>  <!-- Optional. Default behaviour is "any:any". Can define
      multiple of these -->
      <applicationId>application-667</applicationId>
      <service>  <!-- Optional. Can define multiple of these -->
        <protocol>tcp</protocol>
        <port>80</port>
      </service>
    </application>
    <action>deny</action>
    <direction>in</direction>
    <enabled>true</enabled>

```

```

    <loggingEnabled>>false</loggingEnabled>
    <matchTranslated>>true</matchTranslated>
  </firewallRule>
</firewallRules>
<firewallRule>
  <id>131078</id>
  <ruleTag>131078</ruleTag>
  <name>name2</name>
  <ruleType>user</ruleType>
  <source>
    <groupingObjectId>ipset-938</groupingObjectId>
  </source>
  <destination/>
  <application>
    <applicationId>application-666</applicationId>
  </application>
  <action>accept</action>
  <enabled>true</enabled>
  <loggingEnabled>>false</loggingEnabled>
  <matchTranslated>>false</matchTranslated>
</firewallRule>
<firewallRule>
  <id>131075</id>
  <ruleTag>131075</ruleTag>
  <name>default rule for ingress traffic</name>
  <ruleType>default_policy</ruleType>
  <action>deny</action>
  <enabled>true</enabled>
  <loggingEnabled>>false</loggingEnabled>
  <description>default rule for ingress traffic</description>
</firewallRule>
</firewallRules>
</firewall>

```

---

## Query Pre Rules

Retrieves rules that were added to the global firewall configuration and applied to the specified Edge. These rules are read only at the Edge level.

### Example 9-25. Query firewall

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config
```

---

## Append Firewall Rules

Adds one or more rules below the existing rules in the rules table.

### Example 9-26. Add firewall rule

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config/rules
```

Request Body:

```

<firewallRules>
  <firewallRule>
    <ruleTag>1</ruleTag> <!-- Optional. This can be used to specify user controlled
      ids on VSE. The inputs here should be 1-65536. If not specified, VSM will
      generate ruleId -->
    <name>rule1</name> <!-- Optional -->
    <source> <!-- Optional. Default behaviour is like "any". ipsetId or
      predefined-vnicGroupIds can be used -->
  </firewallRule>
</firewallRules>

```

```

    <vnicGroupId>vnic-index-5</vnicGroupId>  <!-- Possible values are
      "vnic-index-[0-9]", "vse", "external" or "internal". Can define multiple of
      these -->
    <groupingObjectId>ipset-128</groupingObjectId>  <!-- Id of IPAddresses grouping
      Objects available to the edge. Can define multiple of these -->
  </source>
  <destination>  <!-- Optional. Default behaviour is like "any". ipsetId or
    predefined-vnicGroupIds can be used -->
    <groupingObjectId>ipset-126</groupingObjectId>  <!-- Id of IPAddresses grouping
      Objects available to the edge. Can define multiple of these -->
    <vnicGroupId>vnic-index-5</vnicGroupId>  <!-- Possible values are
      "vnic-index-[0-9]", "vse", "external" or "internal". Can define multiple of
      these -->
    <groupingObjectId>ipset-128</groupingObjectId>  <!-- Id of IPAddresses grouping
      Objects available to the edge. Can define multiple of these -->
  </destination>
  <application>  <!-- Optional. Default behaviour is like "any". applicationsetId or
    applicationgroupId can be used -->
    <applicationId>application-155</applicationId>  <!-- Id of Service available to
      the edge. Can define multiple of these -->
  </application>
  <matchTranslated>true</matchTranslated>  <!-- Optional. Default behaviour is like
    "false" -->
  <direction>in</direction>  <!-- Optional. Default behaviour is like "any".
    Possible values are in|out -->
  <action>accept</action>  <!-- Mandatory. Possible values are accept|deny -->
  <enabled>true</enabled>  <!-- Optional. Defaults to true -->
  <loggingEnabled>true</loggingEnabled>  <!-- Optional. Defaults to false -->
  <description>comments</description>  <!-- Optional -->
</firewallRule>
</firewallRules>

```

---

### Add a Firewall Rule Above a Specific Rule

You can add a rule above a specific rule by indicating its *ruleID*. If no user-rules exist in the firewall rules table, you can specify *ruleId=0*. If you do not specify a *ruleID* or the specified *ruleID* does not exist, Edge Manager displays an error.

#### Example 9-27. Add a rule above a specific rule

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config
/rules?aboveRuleId=ruleId
```

Request Body:

```

<firewallRule>
  <ruleTag>1</ruleTag>  <!-- Optional. This can be used to specify user controlled ids
    on VSE. The inputs here should be 1-65536. If not specified, VSM will
    generate ruleId -->
  <name>rule1</name>  <!-- Optional -->
  <source>  <!-- Optional. Default behaviour is like "any". ipsetId or
    predefined-vnicGroupIds can be used -->
    <vnicGroupId>vnic-index-5</vnicGroupId>  <!-- Possible values are
      "vnic-index-[0-9]", "vse", "external" or "internal". Can define multiple of
      these -->
    <groupingObjectId>ipset-128</groupingObjectId>  <!-- Id of IPAddresses grouping
      Objects available to the edge. Can define multiple of these -->
  </source>
  <destination>  <!-- Optional. Default behaviour is like "any". ipsetId or
    predefined-vnicGroupIds can be used -->
    <groupingObjectId>ipset-126</groupingObjectId>  <!-- Id of IPAddresses grouping
      Objects available to the edge. Can define multiple of these -->
    <vnicGroupId>vnic-index-5</vnicGroupId>  <!-- Possible values are
      "vnic-index-[0-9]", "vse", "external" or "internal". Can define multiple of
      these -->

```

```

    <groupingObjectId>ipset-128</groupingObjectId>  <!-- Id of IPAddresses grouping
      Objects available to the edge. Can define multiple of these -->
  </destination>
  <application>  <!-- Optional. Default behaviour is like "any". applicationsetId or
    applicationgroupId can be used -->
    <applicationId>application-155</applicationId>  <!-- Id of Service available to
      the edge. Can define multiple of these -->
  </application>
  <matchTranslated>true</matchTranslated>  <!-- Optional. Default behaviour is like
    "false" -->
  <direction>in</direction>  <!-- Optional. Default behaviour is like "any". Possible
    values are in|out -->
  <action>accept</action>  <!-- Mandatory. Possible values are accept|deny -->
  <enabled>true</enabled>  <!-- Optional. Defaults to true -->
  <loggingEnabled>true</loggingEnabled>  <!-- Optional. Defaults to false -->
  <description>comments</description>  <!-- Optional -->
</firewallRule>

```

---

## Query Specific Rule

**Example 9-28.** Retrieve specific rule

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config/rules/ruleId
```

Response Body:

```

<firewallRule>
  <name>new rule</name>
  <source>
    <vnicGroupId>vnic-index-5</vnicGroupId>
  </source>
  <destination>
    <groupingObjectId>ipset-127</groupingObjectId>
  </destination>
  <action>accept</action>
  <enabled>true</enabled>
  <loggingEnabled>true</loggingEnabled>
  <description></description>
</firewallRule>

```

---

## Modify Firewall Rule

You can modify a rule by specifying its *ruleID*. Note that only local rules can be modified at the NSX Edge level.

**Example 9-29.** Update specific rule

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config/rules/ruleId
```

Request Body:

```

<firewallRule>
  <ruleTag>1</ruleTag>  <!-- Optional. This can be used to specify user controlled ids
    on VSE. The inputs here should be 1-65536. If not specified, VSM will
    generate ruleId -->
  <name>rule1</name>  <!-- Optional -->
  <source>  <!-- Optional. Default behaviour is like "any". ipsetId or
    predefined-vnicGroupIds can be used -->
    <vnicGroupId>vnic-index-5</vnicGroupId>  <!-- Possible values are
      "vnic-index-[0-9]", "vse", "external" or "internal". Can define multiple of
      these -->

```

```

    <groupingObjectId>ipset-128</groupingObjectId>  <!-- Id of IPAddresses grouping
      Objects available to the edge. Can define multiple of these -->
  </source>
  <destination>  <!-- Optional. Default behaviour is like "any". ipsetId or
    predefined-vnicGroupIds can be used -->
    <groupingObjectId>ipset-126</groupingObjectId>  <!-- Id of IPAddresses grouping
      Objects available to the edge. Can define multiple of these -->
    <vnicGroupId>vnic-index-5</vnicGroupId>  <!-- Possible values are
      "vnic-index-[0-9]", "vse", "external" or "internal". Can define multiple of
      these -->
    <groupingObjectId>ipset-128</groupingObjectId>  <!-- Id of IPAddresses grouping
      Objects available to the edge. Can define multiple of these -->
  </destination>
  <application>  <!-- Optional. Default behaviour is like "any". applicationsetId or
    applicationgroupId can be used -->
    <applicationId>application-155</applicationId>  <!-- Id of Service available to
      the edge. Can define multiple of these -->
  </application>
  <matchTranslated>true</matchTranslated>  <!-- Optional. Default behaviour is like
    "false" -->
  <direction>in</direction>  <!-- Optional. Default behaviour is like "any". Possible
    values are in|out -->
  <action>accept</action>  <!-- Mandatory. Possible values are accept|deny -->
  <enabled>true</enabled>  <!-- Optional. Defaults to true -->
  <loggingEnabled>true</loggingEnabled>  <!-- Optional. Defaults to false -->
  <description>comments</description>  <!-- Optional -->
</firewallRule>

```

---

## Delete a Firewall Rule

Deletes the rule with the specified rule ID. Note that only local rules can be deleted at the NSX Edge level.

**Example 9-30.** Delete firewall rule

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config/rules/ruleId
```

---

## Delete Firewall Configuration

Deletes firewall configuration for Edge.

**Example 9-31.** Delete firewall configuration

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config
```

---

## Manage Global Firewall Configuration

Global firewall configuration allows fine grained tuning of firewall behavior and its stateful session timeouts.

The default settings of these parameters are set for normal stateful firewall operation. Administrators are not expected to modify these default settings unless to support a specific custom scenario.

### Query Global Firewall Configuration

Retrieves the firewall default policy for an Edge.

**Example 9-32.** Query global firewall configuration

Request:

GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config/global

Response Body:

```
<globalConfig>
  <tcpPickOngoingConnections>false</tcpPickOngoingConnections>
  <tcpAllowOutOfWindowPackets>false</tcpAllowOutOfWindowPackets>
  <tcpSendResetForClosedVsePorts>true</tcpSendResetForClosedVsePorts>
  <dropInvalidTraffic>true</dropInvalidTraffic>
  <logInvalidTraffic>false</logInvalidTraffic>
  <tcpTimeoutOpen>30</tcpTimeoutOpen>
  <tcpTimeoutEstablished>3600</tcpTimeoutEstablished>
  <tcpTimeoutClose>30</tcpTimeoutClose>
  <udpTimeout>60</udpTimeout>
  <icmpTimeout>10</icmpTimeout>
  <icmp6Timeout>10</icmp6Timeout>
  <ipGenericTimeout>120</ipGenericTimeout>
</globalConfig>
```

**Modify Global Configuration**

Configures firewall global config for an Edge. Stores the specified configuration in database. If any appliance(s) are associated with this Edge, applies the configuration to these. Does not change the defaultPolicy and rules.

**Example 9-33.** Modify global firewall configuration

Request:

PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config/global

Request Body:

```
<globalConfig> <!-- Optional -->
  <tcpPickOngoingConnections>false</tcpPickOngoingConnections> <!-- Optional. Defaults
    to false -->
  <tcpAllowOutOfWindowPackets>false</tcpAllowOutOfWindowPackets> <!-- Optional.
    Defaults to false -->
  <tcpSendResetForClosedVsePorts>true</tcpSendResetForClosedVsePorts> <!-- Optional.
    Defaults to true -->
  <dropInvalidTraffic>true</dropInvalidTraffic> <!-- Optional. Defaults to true -->
  <logInvalidTraffic>false</logInvalidTraffic> <!-- Optional. Defaults to false -->
  <tcpTimeoutOpen>30</tcpTimeoutOpen> <!-- Optional. Defaults to 30 -->
  <tcpTimeoutEstablished>21600</tcpTimeoutEstablished> <!-- Optional. Defaults
    to 3600 -->
  <tcpTimeoutClose>30</tcpTimeoutClose> <!-- Optional. Defaults to 30 -->
  <udpTimeout>60</udpTimeout> <!-- Optional. Defaults to 60 -->
  <icmpTimeout>10</icmpTimeout> <!-- Optional. Defaults to 10 -->
  <icmp6Timeout>10</icmp6Timeout> <!-- Optional. Defaults to 10 -->
  <ipGenericTimeout>120</ipGenericTimeout> <!-- Optional. Defaults to 120 -->
  <enableSynFloodProtection>false</enableSynFloodProtection> <!-- Optional. Defaults to
    false -->
</globalConfig>
```

**Manage Default Firewall Policy**

Default firewall settings apply to traffic that does not match any of the user-defined firewall rules. The default Edge firewall policy blocks all incoming traffic.

## Query Default Firewall Policy

Retrieves default firewall policy for the specified Edge.

### Example 9-34. Query default firewall configuration

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config/defaultpolicy
```

Response Body:

```
<firewallDefaultPolicy>
  <action>ACCEPT</action>
  <loggingEnabled>>true</loggingEnabled>
</firewallDefaultPolicy>
```

---

## Modify Default Firewall Policy

Configures default firewall policy for the specified Edge.

### Example 9-35. Modify default firewall configuration

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config/defaultpolicy
```

Request Body:

```
<firewallDefaultPolicy>
  <action>ACCEPT</action>
  <loggingEnabled>>true</loggingEnabled>
</firewallDefaultPolicy>
```

---

## Query Firewall Statistics

Retrieves number of ongoing connections for the firewall configuration.

### Example 9-36. Query firewall statistics

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/statistics
/firewall?interval=range
```

Response Body:

```
<dashboardStatistics>
  <meta>
    <startTime>1336068000</startTime> <!-- in seconds -->
    <endTime>1336100700</endTime> <!-- in seconds -->
    <interval>300</interval>
  </meta>
  <data>
    <firewall>
    </firewall>
  </data>
</dashboardStatistics>
```

---

where input range can be given in query parameter:

Default (when not specified): 60 mins (One hour)

This input is either 1 - 60 minutes or oneDay|oneWeek|oneMonth|oneYear'

## Query Firewall Statistics for Rule

Retrieves statistics for a rule.

---

### Example 9-37. Query statistics for a rule

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/statistics/ruleId
```

Response Body:

```
<firewallRuleStats>
  <timestamp>1342317563</timestamp>
  <connectionCount>0</connectionCount>
  <packetCount>0</packetCount>
  <byteCount>0</byteCount>
</firewallRuleStats>
```

---

## Disable Firewall

Firewall can be disabled only on an xlarge Edge.

---

### Example 9-38. Disable Firewall

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/firewall/config
```

Request Body:

```
<firewall>
  <enabled>false</enabled>
</firewall>
```

---

## Working with NAT

NSX Edge provides network address translation (NAT) service to protect the IP addresses of internal (private) networks from the public network.

### Configure NAT

You can configure NAT rules to provide access to services running on privately addressed virtual machines. There are two types of NAT rules that can be configured: SNAT and DNAT. When you post a NAT configuration, all the rules (both SNAT and DNAT) must be posted together. Otherwise, only the posted rules are retained, and unposted rules are deleted.

All SNAT and DNAT rules configured by using REST requests appear under the **NAT** tab for the appropriate Edge in the vSphere Client plug-in.

---

### Example 9-39. Configure SNAT and DNAT rules for a Edge

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/nat/config
```

Request Body:

```
<nat>
  <natRules>
    <natRule>
      <ruleTag>65537</ruleTag>  <!-- Optional. Can be used to specify
                                user-controlled ids on VSE. Valid inputs 65537-131072. If not
                                specified, vshield manager will generate ruleId -->
```

```

<action>dnat</action>
<vnic>0</vnic> <!-- Optional-->
<originalAddress>10.112.196.116</originalAddress>
<translatedAddress>172.16.1.10</translatedAddress>
<loggingEnabled>true</loggingEnabled> <!-- Optional. Default is false -->
<enabled>true</enabled> <!-- Optional. Default is true -->
<description>my comments</description> <!-- Optional -->
<protocol>tcp</protocol> <!-- Optional. Default is "any". This tag is not
supported for SNAT rule -->
<translatedPort>3389</translatedPort> <!-- Optional. Default is "any". This
tag is not supported for SNAT rule -->
<originalPort>3389</originalPort> <!-- Optional. Default is "any". This tag
is not supported for SNAT rule -->
</natRule>
<natRule>
  <ruleTag>65538</ruleTag> <!-- Optional. Can be used to specify
user-controlled ids on VSE. Valid inputs 65537-131072. If not
specified, VSM will generate ruleId -->
  <action>snat</action>
  <vnic>1</vnic> <!-- Optional-->
  <originalAddress>172.16.1.10</originalAddress>
  <translatedAddress>10.112.196.116</translatedAddress>
  <loggingEnabled>false</loggingEnabled> <!-- Optional. Default is "false" -->
  <enabled>true</enabled> <!-- Optional. Default is "true" -->
  <description>no comments</description> <!-- Optional. Default is "any" -->
</natRule>
</natRules>
</nat>

```

For the data path to work, you need to add firewall rules to allow the required traffic for IP addresses and port per the NAT rules.

Rules:

- You must add *icmpType* if you configure icmp as the protocol.
- The *originalAddress* and *translatedAddress* elements can be entered in either of these methods:
  - *ipAddress* specified as a single IP address, a hyphen-separated IP address range (for example, 192.168.10.1-192.168.10.255) or a subnet in CIDR notation (198.168.10.1/24).
  - the keyword *any*
- The *originalPort* and *translatedPort* parameters can be entered in one of the following formats: the keyword *any*, the port number as an integer, or a range of port number, for example *portX-portY*.
- You can add multiple SNAT rules by entering multiple `<type>snat</type>` sections in the body.
- SNAT does not support port or protocol parameters.
- Logging is disabled by default. To enable logging, add an `<enableLog>` element set to true.

## Query NAT Rules for an Edge

**Example 9-40.** Query SNAT and DNAT rules for a Edge

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/nat/config
```

Response Body:

```

<nat>
  <natRules>
    <natRule>
      <ruleTag>196609</ruleTag>
      <ruleId>196609</ruleId>
      <action>dnat</action>

```

```

    <vnic>0</vnic><!-- Optional -->
    <originalAddress>10.112.196.116</originalAddress>
    <translatedAddress>172.16.1.10</translatedAddress>
    <loggingEnabled>true</loggingEnabled>
    <enabled>true</enabled>
    <description>my comments</description>
    <protocol>tcp</protocol>
    <translatedPort>3389</translatedPort>
    <originalPort>3389</originalPort>
    <ruleType>user</ruleType>
  </natRule>
  <natRule>
    <ruleTag>196609</ruleTag>
    <ruleId>196609</ruleId>
    <action>snat</action>
    <vnic>1</vnic><!-- Optional -->
    <originalAddress>172.16.1.10</originalAddress>
    <translatedAddress>10.112.196.116</translatedAddress>
    <loggingEnabled>false</loggingEnabled>
    <enabled>true</enabled>
    <description>no comments</description>
    <protocol>any</protocol>
    <originalPort>any</originalPort>
    <translatedPort>any</translatedPort>
    <ruleType>user</ruleType>
  </natRule>
</natRules>
</nat>

```

---

### Delete all NAT Rules

Deletes all SNAT and DNAT rules for a Edge. The auto plumbed rules continue to exist.

#### Example 9-41. Delete NAT rules

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/nat/config
```

---

### Add a NAT Rule above a Specific Rule

Adds a NAT rule above the specified rule ID. If no NAT rules exist in the NAT rules table, you can specify ruleId=0. If you do not specify a ruleID or the specified ruleID does not exist, Edge Manager displays an error.

#### Example 9-42. Add a NAT rule above a specific rule

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/nat/config
     /rules?aboveRuleId=ruleId
```

Request Body:

```

<natRule>
  <action>dnat</action>
  <vnic>0</vnic><!-- Optional -->
  <originalAddress>10.112.196.116</originalAddress>
  <translatedAddress>172.16.1.10</translatedAddress>
  <loggingEnabled>true</loggingEnabled>
  <enabled>true</enabled>
  <description>my comments</description>
  <protocol>tcp</protocol>
  <translatedPort>3389</translatedPort>
  <originalPort>3389</originalPort>

```

```
</natRule>
```

---

## Append NAT Rules

Appends one or more rules to the bottom of the NAT rules table.

**Example 9-43.** Add NAT rules to the bottom of the rules table

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/nat/config/rules
```

Request Body:

```
<natRules>
  <natRule>
    <action>dnat</action>
    <vnic>0</vnic><!-- Optional -->
    <originalAddress>10.112.196.116</originalAddress>
    <translatedAddress>172.16.1.10</translatedAddress>
    <loggingEnabled>true</loggingEnabled>
    <enabled>true</enabled>
    <description>my comments</description>
    <protocol>tcp</protocol>
    <translatedPort>3389</translatedPort>
    <originalPort>3389</originalPort>
  </natRule>
</natRules>
```

---

where vnic is the internal or uplink interface of the Edge (0-9).

## Modify a NAT Rule

Replaces the NAT rule with the specified rule ID.

**Example 9-44.** Replaces a NAT rule

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/nat/config/rules/ruleId
```

Request Body:

```
<natRule>
  <action>dnat</action>
  <vnic>0</vnic><!-- Optional -->
  <originalAddress>10.112.196.116</originalAddress>
  <translatedAddress>172.16.1.10</translatedAddress>
  <loggingEnabled>true</loggingEnabled>
  <enabled>true</enabled>
  <description>my comments</description>
  <protocol>tcp</protocol>
  <translatedPort>3389</translatedPort>
  <originalPort>3389</originalPort>
</natRule>
```

---

where vnic is the internal or uplink interface of the Edge (0-9).

## Delete a NAT Rule

Deletes the rule with the specified *ruleID*.

**Example 9-45.** Delete NAT rule

Request:

DELETE <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/nat/config/rules/ruleId>

## Working with Routing

You can specify static and dynamic routing for each NSX Edge.

Dynamic routing provides the necessary forwarding information between layer 2 broadcast domains, thereby allowing you to decrease layer 2 broadcast domains and improve network efficiency and scale. NSX extends this intelligence to where the workloads reside for doing East-West routing. This allows more direct virtual machine to virtual machine communication without the costly or timely need to extend hops. At the same time, NSX also provides North-South connectivity, thereby enabling tenants to access public networks.

## Configure Routes

Configures globalConfig, staticRouting, OSPF, and BGP.

**Example 9-46.** Configure routes

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config>

Request Body:

```
<routing>
  <routingGlobalConfig>
    <ecmp>false</ecmp>  <!-- Optional. Default is false -->
    <routerId>1.1.1.1</routerId>  <!-- Required when dynamic routing protocols like
      OSPF, or BGP is configured -->
    <logging>  <!-- Optional. When absent, enable=false and logLevel=INFO -->
      <enable>false</enable>
      <logLevel>info</logLevel>
    </logging>
    <ipPrefixes>  <!-- Optional. Required only if user wants to define redistribution
      rules in dynamic routing protocols like ospf, isis, bgp -->
      <ipPrefix>
        <name>a</name>  <!-- All the defined ipPrefix must have unique names -->
        <ipAddress>10.112.196.160/24</ipAddress>
      </ipPrefix>
      <ipPrefix>
        <name>b</name>
        <ipAddress>192.168.10.0/24</ipAddress>
      </ipPrefix>
    </ipPrefixes>
  </routingGlobalConfig>
  <staticRouting>
    <staticRoutes>  <!-- Optional, if no static routes needs to be configured -->
      <route>
        <description>route1</description>
        <vnic>0</vnic>
        <network>3.1.1.4/22</network>
        <nextHop>172.16.1.14</nextHop>
        <mtu>1500</mtu>  <!-- Optional. Valid value: smaller than the MTU set on the
          interface. Default will be the MTU of the interface on which this route is
          configured -->
      </route>
      <route>
        <description>route2</description>
        <vnic>1</vnic>
        <network>4.1.1.4/22</network>
        <nextHop>10.112.196.118</nextHop>
      </route>
    </staticRoutes>
  </staticRouting>
</routing>
```

```

        <mtu>1500</mtu>  <!-- Optional. Valid value:smaller than the MTU set on the
        interface. Default will be the MTU of the interface on which this route is
        configured -->
    </route>
</staticRoutes>
<defaultRoute>  <!-- Optional, if no default routes needs to be configured -->
    <description>defaultRoute</description>
    <vnic>0</vnic>
    <gatewayAddress>172.16.1.12</gatewayAddress>
    <mtu>1500</mtu>  <!-- Optional. Valid value:smaller than the MTU set on the
    interface. Default will be the MTU of the interface on which this route is
    configured -->
</defaultRoute>
</staticRouting>
<ospf>  <!-- Optional, if no OSPF needs to be configured -->
    <enabled>true</enabled>  <!-- Optional. Defaults to true -->
    <defaultOriginate>false</defaultOriginate>  <!-- default is false, user can
    configure edge router to publish default route by setting it to true. -->
    <gracefulRestart>false</gracefulRestart>  <!-- default is false, user can enable
    graceful restart by setting it to true. Its a newly added optional
    field.-->
    <ospfAreas>
        <ospfArea>
            <areaId>100</areaId>  <!-- Mandatory and unique. Valid values are
            0-4294967295 -->
            <type>normal</type>  <!-- Optional. Default is normal. Valid inputs are
            normal, nssa -->
            <authentication>  <!-- Optional. When not specified, its "none"
            authentication. -->
                <type>password</type>  <!-- valid values are none, password , md5 -->
                <value>vmware123</value>  <!-- Value as per the type of authentication -->
            </authentication>
        </ospfArea>
    </ospfAreas>
    <ospfInterfaces>
        <ospfInterface>
            <vnic>0</vnic>
            <areaId>100</areaId>
            <helloInterval>10</helloInterval>  <!-- Optional. Default 10 sec. Valid
            values are 1-255-->
            <deadInterval>40</deadInterval>  <!-- Optional. Default 40 sec. Valid
            values are 1-65535 -->
            <priority>128</priority>  <!-- Optional. Default 128. Valid values are 0-255
            -->
            <cost>10</cost>  <!-- Optional. Auto based on interface speed. Valid values
            are 1-65535 -->
        </ospfInterface>
    </ospfInterfaces>
    <redistribution>
        <enabled>true</enabled>  <!-- Optional. Defaults to false. -->
        <rules>
            <rule>
                <prefixName>a</prefixName>  <!-- Optional. Default is "any". prefixName
                used here should be defined in the routingGlobalConfig->ipPrefixes -->
                <from>
                    <isis>true</isis>  <!-- Optional. Defaults to false -->
                    <ospf>false</ospf>  <!-- Optional. Defaults to false -->
                    <bgp>false</bgp>  <!-- Optional. Defaults to false -->
                    <static>false</static>  <!-- Optional. Defaults to false -->
                    <connected>true</connected>  <!-- Optional. Defaults to false -->
                </from>
                <action>deny</action>  <!-- Mandatory. Valid values are deny|permit -->
            </rule>
            <rule>
                <prefixName>b</prefixName>  <!-- Optional. Default is "any". prefixName
                used here should be defined in the routingGlobalConfig->ipPrefixes -->
                <from>
                    <isis>>false</isis>  <!-- Optional. Defaults to false -->

```

```

        <ospf>>false</ospf>    <!-- Optional. Defaults to false -->
        <bgp>>true</bgp>      <!-- Optional. Defaults to false -->
        <static>>false</static> <!-- Optional. Defaults to false -->
        <connected>>false</connected> <!-- Optional. Defaults to false -->
    </from>
    <action>permit</action> <!-- Mandatory. Valid values are deny|permit -->
</rule>
</rules>
</redistribution>
</ospf>
<isis> <!-- Optional, if no ISIS needs to be configured -->
    <enabled>>true</enabled> <!-- Optional. Defaults to true -->
    <systemId>0004.c150.f1c0</systemId> <!-- Optional. 6 byte length & specified in
        HEX. When not specified, derived routingGlobalConfig.routerId -->
    <areaIds> <!-- At least one is required. Max supported is 3 -->
        <areaId>49.0005.8000.ab7c.0000.ffe9.0001</areaId>
        <areaId>49.0005.8000.ab7c.0000.ffe9.0002</areaId> <!-- Variable length between
            1 and 13 bytes & specified in HEX. -->
        <areaId>49.0005.8000.ab7c.0000.ffe9.0003</areaId>
    </areaIds>
    <istype>level-1-2</istype> <!-- Optional. Default is 'level-1-2'. Valid values
        are level-1, level-2, level-1-2 -->
    <domainPassword>vshield</domainPassword> <!-- Optional. Domain level
        authentication. Used when type is level-2 -->
    <areaPassword>edge</areaPassword> <!-- Optional. Area level authentication. Used
        when type is level-1 -->
    <isisInterfaces>
        <isisInterface>
            <vnic>1</vnic>
            <meshGroup>10</meshGroup> <!-- Optional. Valid values are : 0-4294967295
                -->
            <helloInterval>10000</helloInterval> <!-- Optional. Default is 10000
                millisecond . Valid values are : 10-600000 -->
            <helloMultiplier>3</helloMultiplier> <!-- Optional. Default is 3. Valid
                values are : 2-100 -->
            <lspInterval>33</lspInterval> <!-- Optional. Default is 33 milliseconds.
                Valid values are : 1-65535 -->
            <metric>10</metric> <!-- Optional. Default is 10. Valid values are :
                1-16777215 -->
            <priority>64</priority> <!-- Optional. Default is 64. Valid values are :
                0-127 -->
            <circuitType>level-1-2</circuitType> <!-- Optional. Valid values are
                level-1, level-2, level-1-2. If absent, 'type' from above is used -->
            <password>msr</password> <!-- Optional. Per interface authentication -->
        </isisInterface>
    </isisInterfaces>
</redistribution>
<enabled>true</enabled> <!-- Optional. Defaults to false. -->
<rules>
    <rule>
        <prefixName>a</prefixName> <!-- Optional. Default is "any". prefixName
            used here should be defined in the routingGlobalConfig->ipPrefixes -->
        <from>
            <isis>>false</isis> <!-- Optional. Defaults to false -->
            <ospf>true</ospf> <!-- Optional. Defaults to false -->
            <bgp>>false</bgp> <!-- Optional. Defaults to false -->
            <static>true</static> <!-- Optional. Defaults to false -->
            <connected>>false</connected> <!-- Optional. Defaults to false -->
        </from>
        <action>deny</action> <!-- Mandatory. Valid values are deny|permit -->
    </rule>
    <rule>
        <prefixName>b</prefixName> <!-- Optional. Default is "any". prefixName
            used here should be defined in the routingGlobalConfig->ipPrefixes -->
        <from>
            <isis>>false</isis> <!-- Optional. Defaults to false -->
            <ospf>>false</ospf> <!-- Optional. Defaults to false -->
            <bgp>true</bgp> <!-- Optional. Defaults to false -->

```

```

        <static>>false</static>  <!-- Optional. Defaults to false -->
        <connected>>true</connected>  <!-- Optional. Defaults to false -->
    </from>
    <action>permit</action>  <!-- Mandatory. Valid values are deny|permit -->
</rule>
</rules>
</redistribution>
</isis>
<bgp>  <!-- Optional, if no BGP needs to be configured -->
    <enabled>>true</enabled>  <!-- Optional. Default is true -->
    <localAS>1</localAS>  <!-- valid values are : 0-65535 -->
    <bgpNeighbours>
        <bgpNeighbour>
            <ipAddress>192.168.1.10</ipAddress>  <!-- IPv4 only. IPv6 support not
                supported -->
            <remoteAS>65500</remoteAS>  <!-- valid values are 0-65535 -->
            <weight>60</weight>  <!-- Optional. Default is 60. Valid values are
                0-65535 -->
            <holdDownTimer>180</holdDownTimer>  <!-- Optional. Default is 180 seconds.
                Valid values are : 2-65535 . -->
            <keepAliveTimer>60</keepAliveTimer>  <!-- Optional. Default is 60 seconds.
                Valid values are : 1-65534 . -->
            <password>vmware123</password>  <!-- Optional -->
            <bgpFilters>  <!-- Optional -->
                <bgpFilter>
                    <direction>in</direction>  <!-- valid values are in/out -->
                    <action>permit</action>  <!-- valid values are permit/deny -->
                    <network>10.0.0.0/8</network>  <!-- Valid values are CIDR networks.
                        IPv4 only. IPv6 support not supported -->
                    <ipPrefixGe>17</ipPrefixGe>  <!-- Optional. "Greater than or equal to"
                        & used for filtering based on prefix length. Valid IPv4 prefixes -->
                    <ipPrefixLe>32</ipPrefixLe>  <!-- Optional. "Less than or equal to" &
                        used for filtering based on prefix length. Valid IPv4 prefixes -->
                </bgpFilter>
            </bgpFilters>
        </bgpNeighbour>
    </bgpNeighbours>
</redistribution>
    <enabled>true</enabled>  <!-- Optional. Defaults to false. -->
    <rules>
        <rule>
            <prefixName>a</prefixName>  <!-- Optional. Default is "any". prefixName
                used here should be defined in the routingGlobalConfig->ipPrefixes -->
            <from>
                <isis>true</isis>  <!-- Optional. Defaults to false -->
                <ospf>true</ospf>  <!-- Optional. Defaults to false -->
                <bgp>>false</bgp>  <!-- Optional. Defaults to false -->
                <static>true</static>  <!-- Optional. Defaults to false -->
                <connected>>false</connected>  <!-- Optional. Defaults to false -->
            </from>
            <action>deny</action>  <!-- Mandatory. Valid values are deny|permit -->
        </rule>
        <rule>
            <from>
                <isis>>false</isis>  <!-- Optional. Defaults to false -->
                <ospf>>false</ospf>  <!-- Optional. Defaults to false -->
                <bgp>>false</bgp>  <!-- Optional. Defaults to false -->
                <static>>false</static>  <!-- Optional. Defaults to false -->
                <connected>true</connected>  <!-- Optional. Defaults to false -->
            </from>
            <action>permit</action>  <!-- Mandatory. Valid values are deny|permit -->
        </rule>
    </rules>
</redistribution>
</bgp>
</routing>

```

## Query Routes

### Example 9-47. Retrieve routes

---

Request:

GET `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config`

Response Body:

```
<staticRouting>
  <staticRoutes>
    <route>
      <vnic>0</vnic>
      <network>3.1.1.4/22</network>
      <nextHop>172.16.1.14</nextHop>
      <mtu>1500</mtu>  <!-- Optional. Valid value:smaller than the MTU set on the
                        interface. Default is MTU of the interface on which this route
                        is configured -->
      <type>user</type>
    </route>
    <route>
      <vnic>1</vnic>
      <network>4.1.1.4/22</network>
      <nextHop>10.112.196.118</nextHop>
      <mtu>1500</mtu>  <!-- Optional. Valid value:smaller than the MTU set on the
                        interface. Default is MTU of the interface on which this route
                        is configured -->
      <type>user</type>
    </route>
  </staticRoutes>
  <defaultRoute>
    <vnic>0</vnic>
    <gatewayAddress>172.16.1.12</gatewayAddress>
    <mtu>1500</mtu>  <!-- Optional. Valid value:smaller than the MTU set on the
                      interface. Default is MTU of the interface on which this route is
                      configured -->
  </defaultRoute>
</staticRouting>
```

---

## Delete Routes

Deletes the routing configuration stored in the NSX Manager database and the default routes from the specified NSX Edge appliance.

### Example 9-48. Delete routing

---

Request:

DELETE `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config`

---

## Manage Global Routing Configuration

Configures the default gateway for static routes and dynamic routing details.

### Specify Global Configuration

#### Example 9-49. Configure global route

---

Request:

PUT `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/global`

Request Body:

```

<routingGlobalConfig>
  <routerId>1.1.1.1</routerId>  <!-- Required when dynamic routing protocols like OSPF,
    or BGP is configured -->
  <ecmp>>false</ecmp> <!-- Optional. Defaults to false. -->
  <logging>  <!-- Optional. When absent, enable=false and logLevel=INFO -->
    <enable>>false</enable>
    <logLevel>info</logLevel>
  </logging>
  <ipPrefixes>  <!-- Optional. Required only if user wants to define redistribution
    rules in dynamic routing protocols like ospf, isis, bgp -->
    <ipPrefix>
      <name>a</name>  <!-- All the defined ipPrefix must have unique names -->
      <ipAddress>10.112.196.160/24</ipAddress>
    </ipPrefix>
    <ipPrefix>
      <name>b</name>
      <ipAddress>192.168.10.0/24</ipAddress>
    </ipPrefix>
  </ipPrefixes>
</routingGlobalConfig>

```

---

### Query Global Route

Retrieves routing information from the NSX Manager database for an edge which includes the following:

- Default route settings
- Static route configurations

#### Example 9-50. Query global route

---

Request Body:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/global
```

---

## Manage Static Routing

Add or query static and default routes for specified Edge.

### Configure Static Routes

Configures static and default routes.

#### Example 9-51. Configure static routes

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/static
```

Request Body:

```

<staticRouting>
  <staticRoutes>
    <route>
      <description>route1</description>
      <vnic>0</vnic>
      <network>3.1.1.4/22</network>
      <nextHop>172.16.1.14</nextHop>
      <mtu>1500</mtu>  <!-- Optional. Valid value: smaller than the MTU set on the
        interface. Default will be the MTU of the interface on which this route is
        configured -->
    </route>
    <route>
      <description>route2</description>
      <vnic>1</vnic>
    </route>
  </staticRoutes>
</staticRouting>

```

```

    <network>4.1.1.4/22</network>
    <nextHop>10.112.196.118</nextHop>
    <mtu>1500</mtu>    <!-- Optional. Valid value:smaller than the MTU set on the
                        interface. Default will be the MTU of the interface on which this route is
                        configured -->
  </route>
</staticRoutes>
<defaultRoute>
  <description>defaultRoute</description>
  <vnic>0</vnic>
  <gatewayAddress>172.16.1.12</gatewayAddress>
  <mtu>1500</mtu>    <!-- Optional. Valid value:smaller than the MTU set on the
                    interface. Default will be the MTU of the interface on which this route is
                    configured -->
</defaultRoute>
</staticRouting>

```

---

## Query Static Routes

Retrieves static and default routes.

### Example 9-52. Query static routes

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/static>

Response Body:

```

<staticRouting>
  <staticRoutes>
    <route>
      <description>route1</description>
      <vnic>0</vnic>
      <network>3.1.1.4/22</network>
      <nextHop>172.16.1.14</nextHop>
      <mtu>1500</mtu>
      <type>user</type>
    </route>
    <route>
      <description>route2</description>
      <vnic>1</vnic>
      <network>4.1.1.4/22</network>
      <nextHop>10.112.196.118</nextHop>
      <mtu>1500</mtu>
      <type>user</type>
    </route>
  </staticRoutes>
  <defaultRoute>
    <description>defaultRoute</description>
    <vnic>0</vnic>
    <gatewayAddress>172.16.1.12</gatewayAddress>
    <mtu>1500</mtu>
  </defaultRoute>
</staticRouting>

```

---

## Delete Static Routes

Deletes both static and default routing configuration stored in the NSX Manager database.

### Example 9-53. Delete static routes

---

Request

DELETE <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/static>

---

## Manage OSPF Routes for NSX Edge

NSX Edge supports OSPF, an interior gateway protocol that routes IP packets only within a single routing domain. It gathers link state information from available routers and constructs a topology map of the network. The topology determines the routing table presented to the Internet Layer, which makes routing decisions based on the destination IP address found in IP packets.

OSPF routing policies provide a dynamic process of traffic load balancing between routes of equal cost. An OSPF network is divided into routing areas to optimize traffic. An area is a logical collection of OSPF networks, routers, and links that have the same area identification.

Areas are identified by an Area ID.

### Configure OSPF

#### Example 9-54. Configure OSPF

---

Request

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/ospf>

Request Body:

```
<ospf>
  <enabled>true</enabled>  <!-- when not specified, it will be treated as false, when
                           false, it will delete the existing config -->
  <ospfAreas>
    <ospfArea>
      <areaId>100</areaId>  <!-- Mandatory and unique. Valid values are
                           0-4294967295 -->
      <type>normal</type>  <!-- Optional. Default is normal. Valid inputs are normal,
                           nssa -->
      <authentication>  <!-- Optional. When not specified, its "none"
                        authentication. -->
        <type>password</type>  <!-- valid values are none, password , md5 -->
        <value>vmware123</value>  <!-- value as per the type of authentication -->
      </authentication>
    </ospfArea>
  </ospfAreas>
  <ospfInterfaces>
    <ospfInterface>
      <vnic>0</vnic>
      <areaId>100</areaId>
      <helloInterval>10</helloInterval>  <!-- Optional. Default 10 sec. Valid values
                                           are 1-255-->
      <deadInterval>40</deadInterval>  <!-- Optional. Default 40 sec. Valid values are
                                           1-65535 -->
      <priority>128</priority>  <!-- Optional. Default 128. Valid values are 0-255 -->
      <cost>10</cost>  <!-- Optional. Auto based on interface speed. Valid values are
                       1-65535 -->
    </ospfInterface>
  </ospfInterfaces>
  <redistribution>
    <enabled>true</enabled>  <!-- optional. Defaults to false. -->
    <rules>
      <rule>
        <prefixName>a</prefixName>  <!-- Optional. Default is "any". prefixName used
                                     here should be defined in the routingGlobalConfig->ipPrefixes -->
        <from>
          <isis>true</isis>  <!-- optional. Defaults to false -->
          <ospf>false</ospf>  <!-- Optional. Defaults to false -->
          <bgp>false</bgp>  <!-- Optional. Defaults to false -->
          <static>false</static>  <!-- optional. Defaults to false -->
          <connected>true</connected>  <!-- Optional. Defaults to false -->
        </from>
      </rule>
    </rules>
  </redistribution>
</ospf>
```

```

    </from>
    <action>deny</action>  <!-- Mandatory. valid values are deny|permit -->
  </rule>
  <rule>
    <prefixName>b</prefixName>  <!-- Optional. Default is "any". prefixName used
      here should be defined in the routingGlobalConfig->ipPrefixes -->
    <from>
      <isis>>false</isis>  <!-- Optional. Defaults to false -->
      <ospf>>false</ospf>  <!-- Optional. Defaults to false -->
      <bgp>>true</bgp>  <!-- Optional. Defaults to false -->
      <static>>false</static>  <!-- Optional. Defaults to false -->
      <connected>>false</connected>  <!-- Optional. Defaults to false -->
    </from>
    <action>permit</action>  <!-- Mandatory. Valid values are deny|permit -->
  </rule>
</rules>
</redistribution>
</ospf>

```

---

## Query OSPF

### Example 9-55. Query OSPF

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/ospf>

Request Body:

```

<ospf>
  <enabled>>true</enabled>
  <ospfAreas>
    <ospfArea>
      <areaId>100</areaId>
      <type>normal</type>
      <authentication>
        <type>password</type>
        <value>vmware123</value>
      </authentication>
    </ospfArea>
  </ospfAreas>
  <ospfInterfaces>
    <ospfInterface>
      <vnic>0</vnic>
      <areaId>100</areaId>
      <helloInterval>10</helloInterval>
      <deadInterval>40</deadInterval>
      <priority>128</priority>
      <cost>10</cost>
    </ospfInterface>
  </ospfInterfaces>
  <redistribution>
    <enabled>>true</enabled>
    <rules>
      <rule>
        <id>1</id>
        <prefixName>a</prefixName>
        <from>
          <isis>>true</isis>
          <ospf>>false</ospf>
          <bgp>>false</bgp>
          <static>>false</static>
          <connected>>true</connected>
        </from>
        <action>deny</action>
      </rule>
    </rules>
  </redistribution>
</ospf>

```

```

<id>0</id>
<prefixName>b</prefixName>
<from>
  <isis>>false</isis>
  <ospf>>false</ospf>
  <bgp>>true</bgp>
  <static>>false</static>
  <connected>>false</connected>
</from>
<action>permit</action>
</rule>
</rules>
</redistribution>
</ospf>

```

---

## Delete OSPF

Deletes OSPF routing.

### Example 9-56. Delete OSPF

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/ospf
```

---

## Manage ISIS Routes for NSX Edge

Intermediate System to Intermediate System (IS-IS) is a routing protocol designed to move information by determining the best route for datagrams through a packet-switched network. A two-level hierarchy is used to support large routing domains. A large domain may be divided into areas. Routing within an area is referred to as Level 1 routing. Routing between areas is referred to as Level 2 routing. A Level 2 Intermediate System (IS) keeps track of the paths to destination areas. A Level 1 IS keeps track of the routing within its own area. For a packet going to another area, a Level 1 IS sends the packet to the nearest Level 2 IS in its own area, regardless of what the destination area is. Then the packet travels via Level 2 routing to the destination area, where it may travel via Level 1 routing to the destination. This is referred to as Level-1-2.

## Configure ISIS

### Example 9-57. Configure ISIS

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/isis
```

Request Body:

```

<isis>
  <enabled>>true</enabled>
  <systemId>0004.c150.f1c0</systemId>  <!-- Optional. 6 byte length & specified in HEX.
    when not specified, derived routingGlobalConfig.routerId -->
  <areaIds>  <!-- At least one is required. Max supported is 3 -->
    <areaId>49.0005.8000.ab7c.0000.ffe9.0001</areaId>
    <areaId>49.0005.8000.ab7c.0000.ffe9.0002</areaId>  <!-- Variable length between 1
    and 13 bytes & specified in HEX. -->
    <areaId>49.0005.8000.ab7c.0000.ffe9.0003</areaId>
  </areaIds>
  <isType>level-1-2</isType>  <!-- Optional. Default is 'level-1-2'. valid values are
    level-1, level-2, level-1-2 -->
  <domainPassword>vshield</domainPassword>  <!-- Optional. Domain level authentication.
    Used when type is level-2 -->
  <areaPassword>edge</areaPassword>  <!-- Optional. Area level authentication. Used
    when type is level-1 -->
  <isisInterfaces>

```

```

<isisInterface>
  <vnic>0</vnic>
  <meshGroup>10</meshGroup>  <!-- Optional. Valid values are : 0-4294967295 -->
  <helloInterval>10000</helloInterval>  <!-- Optional. Default is 10000 millisecond
    . Valid values are : 10-600000 -->
  <helloMultiplier>3</helloMultiplier>  <!-- Optional. Default is 3. Valid values
    are : 2-100 -->
  <lspInterval>33</lspInterval>  <!-- Optional. Default is 33 milliseconds. Valid
    values are : 1-65535 -->
  <metric>10</metric>  <!-- Optional. Default is 10. Valid values are :
    1-16777215 -->
  <priority>64</priority>  <!-- Optional. Default is 64. Valid values are :
    0-127 -->
  <circuitType>level-1-2</circuitType>  <!-- Optional. Valid values are level-1,
    level-2, level-1-2. If absent, 'type' from above is used -->
  <password>msr</password>  <!-- Optional. Per interface authentication -->
</isisInterface>
</isisInterfaces>
<redistribution>
  <enabled>true</enabled>  <!-- Optional. Defaults to false. -->
  <rules>
    <rule>
      <prefixName>a</prefixName>  <!-- Optional. Default is "any". prefixName used
        here should be defined in the routingGlobalConfig->ipPrefixes -->
      <from>
        <isis>false</isis>  <!-- Optional. Defaults to false -->
        <ospf>true</ospf>  <!-- Optional. Defaults to false -->
        <bgp>false</bgp>  <!-- Optional. Defaults to false -->
        <static>true</static>  <!-- Optional. Defaults to false -->
        <connected>false</connected>  <!-- Optional. Defaults to false -->
      </from>
      <action>deny</action>  <!-- Mandatory. Valid values are deny|permit -->
    </rule>
    <rule>
      <prefixName>b</prefixName>  <!-- Optional. Default is "any". prefixName used
        here should be defined in the routingGlobalConfig->ipPrefixes -->
      <from>
        <isis>false</isis>  <!-- Optional. Defaults to false -->
        <ospf>false</ospf>  <!-- Optional. Defaults to false -->
        <bgp>true</bgp>  <!-- Optional. Defaults to false -->
        <static>false</static>  <!-- Optional. Defaults to false -->
        <connected>true</connected>  <!-- Optional. Defaults to false -->
      </from>
      <action>permit</action>  <!-- Mandatory. Valid values are deny|permit -->
    </rule>
  </rules>
</redistribution>
</isis>

```

---

## Query ISIS

### Example 9-58. Query ISIS

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/isis>

Response Body:

```

<isis>
  <enabled>true</enabled>
  <systemId>0004.c150.f1c0</systemId>
  <areaIds>
    <areaId>49.0005.8000.ab7c.0000.ffe9.0001</areaId>
    <areaId>49.0005.8000.ab7c.0000.ffe9.0002</areaId>
    <areaId>49.0005.8000.ab7c.0000.ffe9.0003</areaId>
  </areaIds>

```

```

<isType>level-1-2</isType>
<domainPassword>vshield</domainPassword>
<areaPassword>edge</areaPassword>
<isisInterfaces>
  <isisInterface>
    <vnic>0</vnic>
    <meshGroup>10</meshGroup>
    <helloInterval>10000</helloInterval>
    <helloMultiplier>3</helloMultiplier>
    <lspInterval>33</lspInterval>
    <metric>10</metric>
    <priority>64</priority>
    <circuitType>level-1-2</circuitType>
    <password>msr</password>
  </isisInterface>
</isisInterfaces>
<redistribution>
  <enabled>true</enabled>
  <rules>
    <rule>
      <id>1</id>
      <prefixName>a</prefixName>
      <from>
        <isis>false</isis>
        <ospf>true</ospf>
        <bgp>false</bgp>
        <static>true</static>
        <connected>false</connected>
      </from>
      <action>deny</action>
    </rule>
    <rule>
      <id>0</id>
      <prefixName>b</prefixName>
      <from>
        <isis>false</isis>
        <ospf>false</ospf>
        <bgp>true</bgp>
        <static>false</static>
        <connected>true</connected>
      </from>
      <action>permit</action>
    </rule>
  </rules>
</redistribution>
</isis>

```

---

### Delete ISIS

Deletes ISIS routing.

#### Example 9-59. Delete ISIS

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/isis
```

---

## Manage BGP Routes for NSX Edge

Border Gateway Protocol (BGP) makes core routing decisions. It includes a table of IP networks or prefixes which designate network reachability among autonomous systems. An underlying connection between two BGP speakers is established before any routing information is exchanged. Keep alive messages are sent out by the BGP speakers in order to keep this relationship alive. Once the connection is established, the BGP speakers exchange routes and synchronize their tables.

## Configure BGP

### Example 9-60. Configure BGP

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/bgp>

Request Body:

```
<bgp>
  <enabled>true</enabled>    <!-- Optional. Default is false -->
  <localAS>65534</localAS>  <!-- Valid values are : 1-65534 -->
  <bgpNeighbours>
    <bgpNeighbour>
      <ipAddress>192.168.1.10</ipAddress>  <!-- IPv4 only. IPv6 support not supported
      -->
      <remoteAS>65500</remoteAS>  <!-- Valid values are 0-65535 -->
      <weight>60</weight>  <!-- Optional. Default is 60. Valid values are 0-65535 -->
      <holdDownTimer>180</holdDownTimer>  <!-- Optional. Default is 180 seconds.
      Valid values are : 2-65535. -->
      <keepAliveTimer>60</keepAliveTimer>  <!-- Optional. Default is 60 seconds.
      Valid values are : 1-65534. -->
      <password>vmware123</password>  <!-- Optional -->
      <bgpFilters>  <!-- Optional -->
        <bgpFilter>
          <direction>in</direction>  <!-- Valid values are in/out -->
          <action>permit</action>  <!-- Valid values are permit/deny -->
          <network>10.0.0.0/8</network>  <!-- Valid values are CIDR networks. IPv4
          only. IPv6 support not supported -->
          <ipPrefixGe>17</ipPrefixGe>  <!-- Optional. "Greater than or equal to" &
          used for filtering based on prefix length. Valid IPv4 prefixes -->
          <ipPrefixLe>32</ipPrefixLe>  <!-- Optional. "Less than or equal to" &
          used for filtering based on prefix length. Valid IPv4 prefixes -->
        </bgpFilter>
      </bgpFilters>
    </bgpNeighbour>
  </bgpNeighbours>
  <redistribution>
    <enabled>true</enabled>  <!-- Optional. Defaults to false. -->
    <rules>
      <rule>
        <prefixName>a</prefixName>  <!-- Optional. Default is "any". prefixName used
        here should be defined in the routingGlobalConfig->ipPrefixes -->
        <from>
          <isis>true</isis>  <!-- Optional. Defaults to false -->
          <ospf>true</ospf>  <!-- Optional. Defaults to false -->
          <bgp>false</bgp>  <!-- Optional. Defaults to false -->
          <static>true</static>  <!-- Optional. Defaults to false -->
          <connected>false</connected>  <!-- Optional. Defaults to false -->
        </from>
        <action>deny</action>  <!-- Mandatory. Valid values are deny|permit -->
      </rule>
      <rule>
        <from>
          <isis>false</isis>  <!-- Optional. Defaults to false -->
          <ospf>false</ospf>  <!-- Optional. Defaults to false -->
          <bgp>false</bgp>  <!-- Optional. Defaults to false -->
          <static>false</static>  <!-- Optional. Defaults to false -->
          <connected>true</connected>  <!-- Optional. Defaults to false -->
        </from>
        <action>permit</action>  <!-- Mandatory. Valid values are deny|permit -->
      </rule>
    </rules>
  </redistribution>
</bgp>
```

---

**Query BGP****Example 9-61.** Query BGP

Request:

GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/bgp

Response Body:

```

<bgp>
  <enabled>true</enabled>
  <localAS>65535</localAS>
  <bgpNeighbours>
    <bgpNeighbour>
      <ipAddress>192.168.1.10</ipAddress>
      <remoteAS>65500</remoteAS>
      <weight>60</weight>
      <holdDownTimer>180</holdDownTimer>
      <keepAliveTimer>60</keepAliveTimer>
      <password>vmware123</password>
      <bgpFilters>
        <bgpFilter>
          <direction>in</direction>
          <action>permit</action>
          <network>10.0.0.0/8</network>
          <ipPrefixGe>17</ipPrefixGe>
          <ipPrefixLe>32</ipPrefixLe>
        </bgpFilter>
        <bgpFilter>
          <direction>out</direction>
          <action>deny</action>
          <network>20.0.0.0/26</network>
        </bgpFilter>
      </bgpFilters>
    </bgpNeighbour>
  </bgpNeighbours>
  <redistribution>
    <enabled>true</enabled>
    <rules>
      <rule>
        <id>1</id>
        <prefixName>a</prefixName>
        <from>
          <isis>true</isis>
          <ospf>true</ospf>
          <bgp>false</bgp>
          <static>true</static>
          <connected>false</connected>
        </from>
        <action>deny</action>
      </rule>
      <rule>
        <id>0</id>
        <from>
          <isis>false</isis>
          <ospf>false</ospf>
          <bgp>false</bgp>
          <static>false</static>
          <connected>true</connected>
        </from>
        <action>permit</action>
      </rule>
    </rules>
  </redistribution>
</bgp>

```

## Delete BGP

Deletes BGP routing.

### Example 9-62. Delete BGP

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/routing/config/bgp
```

---

## Working with Load Balancer

The NSX Edge load balancer enables network traffic to follow multiple paths to a specific destination. It distributes incoming service requests evenly among multiple servers in such a way that the load distribution is transparent to users. Load balancing thus helps in achieving optimal resource utilization, maximizing throughput, minimizing response time, and avoiding overload. NSX Edge provides load balancing up to Layer 7.

You map an external, or public, IP address to a set of internal servers for load balancing. The load balancer accepts TCP, HTTP, or HTTPS requests on the external IP address and decides which internal server to use. Port 8090 is the default listening port for TCP, port 80 is the default port for HTTP, and port 443 is the default port for HTTPS.

## Configure Load Balancer

The input contains five parts: application profile, virtual server, pool, monitor and application rule.

### Example 9-63. Configure load balancer

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
```

Request Body:

```
<loadBalancer>
  <enabled>true</enabled>    <!-- optional, default is true -->
  <enableServiceInsertion>>false</enableServiceInsertion>  <!-- optional, default is
    false-->
  <accelerationEnabled>true</accelerationEnabled>  <!-- optional, default is false-->
  <logging>  <!-- optional, default is false/INFO -->
    <enable>true</enable>
    <logLevel>debug</logLevel>  <!-- valid values include: emergency, alert, critical,
      error, warning, notice, info, debug -->
  </logging>
  <virtualServer>  <!-- 0-64 virtualServer items could be added -->
    <virtualServerId>virtualServer-1</virtualServerId>  <!-- optional, virtualServerId
      should match virtualServer-X pattern -->
    <name>http_vip</name>  <!-- required, unique virtualServer name per edge -->
    <description>http virtualServer</description>  <!-- optional -->
    <enabled>true</enabled>  <!-- optional, default is true -->
    <ipAddress>10.117.35.172</ipAddress>  <!-- required, a valid Edge vNic ip
      address(ipv4/ipv6) -->
    <protocol>http</protocol>  <!-- required, valid values are http/https/tcp -->
    <port>80</port>  <!-- required, 1~65535 -->
    <connectionLimit>123</connectionLimit>  <!-- optional, default is 0 -->
    <connectionRateLimit>123</connectionRateLimit>  <!-- optional, default is null -->
    <applicationProfileId>applicationProfile-1</applicationProfileId>  <!-- required,
      a valid applicationProfileId -->
    <defaultPoolId>pool-1</defaultPoolId>  <!-- optional, a valid poolId -->
    <enableServiceInsertion>>false</enableServiceInsertion>  <!-- optional, default is
      false -->
    <accelerationEnabled>true</accelerationEnabled>  <!-- optional, default is
      false -->
  </virtualServer>
  <vendorProfile>
```

```

    <vendorTemplateId>577</vendorTemplateId>    <!-- required, a valid
            vendorTemplateId -->
    <vendorTemplateName>F5</vendorTemplateName>    <!-- optional -->
    <profileAttributes>    <!-- optional -->
        <attribute>
            <key>abcd</key>
            <name>abcd</name>
            <value>1234</value>
        </attribute>
    </profileAttributes>
</vendorProfile>    <!-- optional, it is required when per virtualServer
                    enableServiceInsertion flag and global enableServiceInsertion flag are set
                    to true, the VIP would be offloaded to vendor devices instead of Edge -->
</virtualServer>
<virtualServer>
    <virtualServerId>virtualServer-2</virtualServerId>
    <name>https_vip</name>
    <description>https virtualServer</description>
    <enabled>true</enabled>
    <ipAddress>10.117.35.172</ipAddress>
    <protocol>https</protocol>
    <port>443</port>
    <connectionLimit>123</connectionLimit>
    <connectionRateLimit>123</connectionRateLimit>
    <applicationProfileId>applicationProfile-2</applicationProfileId>
    <defaultPoolId>pool-2</defaultPoolId>
    <enableServiceInsertion>false</enableServiceInsertion>
    <accelerationEnabled>false</accelerationEnabled>
</virtualServer>
<virtualServer>
    <virtualServerId>virtualServer-3</virtualServerId>
    <name>tcp_transparent_vip</name>
    <description>tcp virtualServer</description>
    <enabled>true</enabled>
    <ipAddress>10.117.35.172</ipAddress>
    <protocol>tcp</protocol>
    <port>1234</port>
    <connectionLimit>123</connectionLimit>
    <applicationProfileId>applicationProfile-3</applicationProfileId>
    <defaultPoolId>pool-3</defaultPoolId>
    <enableServiceInsertion>false</enableServiceInsertion>
    <accelerationEnabled>true</accelerationEnabled>
</virtualServer>
<virtualServer>
    <virtualServerId>virtualServer-4</virtualServerId>
    <name>tcp_snat_vip</name>
    <description>tcp snat virtualServer</description>
    <enabled>true</enabled>
    <ipAddress>10.117.35.172</ipAddress>
    <protocol>tcp</protocol>
    <port>1235</port>
    <connectionLimit>123</connectionLimit>
    <applicationProfileId>applicationProfile-3</applicationProfileId>
    <defaultPoolId>pool-4</defaultPoolId>
    <enableServiceInsertion>false</enableServiceInsertion>
    <accelerationEnabled>true</accelerationEnabled>
</virtualServer>
<applicationProfile>
    <applicationProfileId>applicationProfile-1</applicationProfileId>
    <name>http_application_profile</name>
    <insertXForwardedFor>true</insertXForwardedFor>
    <sslPassthrough>true</sslPassthrough>
    <persistence>
        <method>cookie</method>    <!-- required, cookie is used for http protocol,
                                    ssl_sessionid for https -->
        <cookieName>JSESSIONID</cookieName>    <!-- optional, required when method is
                                                cookie -->
    </persistence>

```

```

        <cookieMode>insert</cookieMode>    <!-- optional, valid values are
            insert/prefix/app, required when method is cookie -->
    </persistence>
</applicationProfile>
<applicationProfile>
    <applicationProfileId>applicationProfile-2</applicationProfileId>    <!-- optional,
        it should match "applicationProfile-X" patter and required when it is
        referenced -->
    <name>https_application_profile</name>    <!-- required -->
    <insertXForwardedFor>true</insertXForwardedFor>    <!-- optional, default is
        false -->
    <sslPassthrough>true</sslPassthrough>    <!-- optional, default is false -->
    <persistence>    <!-- optional -->
        <method>ssl_sessionid</method>    <!-- required, valid values are ssl_sessionid,
            cookie, sourceip, msrdp -->
    </persistence>
</applicationProfile>
<applicationProfile>
    <applicationProfileId>applicationProfile-3</applicationProfileId>
    <name>tcp_application_profile</name>
    <insertXForwardedFor>false</insertXForwardedFor>
    <sslPassthrough>true</sslPassthrough>
</applicationProfile>
<pool>    <!-- 0-64 pool items could be added -->
    <poolId>pool-1</poolId>    <!-- optional, it should match "pool-X" pattern, this
        item is required when it has reference -->
    <name>pool-http</name>    <!-- required, unique pool name per edge -->
    <description>pool-http</description>    <!-- optional -->
    <transparent>false</transparent>    <!-- optional, default is false -->
    <algorithm>round-robin</algorithm>    <!-- optional, valid values are round-robin,
        ip-hash, uri, leastconn, default is round-robin -->
    <monitorId>monitor-1</monitorId>    <!-- optional, it should be a valid monitorId,
        it is an array -->
    <member>    <!-- 0-32 pool member items could be added -->
        <memberId>member-1</memberId>    <!-- optional, it should match "member-X"
            pattern, this item is required when it has reference -->
        <ipAddress>192.168.101.201</ipAddress>    <!-- optional, a valid ip
            address(ipv4/ipv6), it is required when groupingObjectId is not
            specified -->
        <groupingObjectId>vm-24</groupingObjectId>    <!-- optional, groupingObject id
            such as vm-24, network-25, dvportgroup-26 -->
        <weight>1</weight>    <!-- optional, default is 1 -->
        <port>80</port>    <!-- required -->
        <minConn>10</minConn>    <!-- optional, default is 0 -->
        <maxConn>100</maxConn>    <!-- optional, default is 0 -->
        <name>m1</name>    <!-- optional, it is required when it is used in ACL rule -->
    </member>
    <member>
        <memberId>member-2</memberId>
        <ipAddress>192.168.101.202</ipAddress>
        <weight>1</weight>
        <port>80</port>
        <minConn>10</minConn>
        <maxConn>100</maxConn>
        <name>m2</name>
        <condition>enabled</condition>    <!-- optional, default is enabled, valid values
            are enabled/disabled -->
    </member>
</pool>
<pool>
    <poolId>pool-2</poolId>
    <name>pool-https</name>
    <description>pool-https</description>
    <transparent>false</transparent>
    <algorithm>round-robin</algorithm>
    <monitorId>monitor-2</monitorId>
    <member>
        <memberId>member-3</memberId>

```

```

    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <port>443</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m3</name>
  </member>
  <member>
    <memberId>member-4</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>443</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m4</name>
  </member>
</pool>
<pool>
  <poolId>pool-3</poolId>
  <name>pool-tcp</name>
  <description>pool-tcp</description>
  <transparent>true</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-3</monitorId>
  <member>
    <memberId>member-5</memberId>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <port>1234</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m5</name>
    <monitorPort>80</monitorPort>
  </member>
  <member>
    <memberId>member-6</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>1234</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m6</name>
    <monitorPort>80</monitorPort>
  </member>
</pool>
<pool>
  <poolId>pool-4</poolId>
  <name>pool-tcp-snat</name>
  <description>pool-tcp-snat</description>
  <transparent>false</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-3</monitorId>
  <member>
    <memberId>member-7</memberId>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <port>1234</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m7</name>
    <monitorPort>80</monitorPort>
  </member>
  <member>
    <memberId>member-8</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>1234</port>
    <minConn>10</minConn>

```

```

        <maxConn>100</maxConn>
        <name>m8</name>
        <monitorPort>80</monitorPort>
    </member>
</pool>
<monitor>
    <monitorId>monitor-1</monitorId>  <!-- optional, this item should follow
        "monitor-x" pattern, it is required when it is referenced -->
    <type>http</type>  <!-- required, valid values are http/https/tcp -->
    <interval>5</interval>  <!-- optional, default is 5 -->
    <timeout>15</timeout>  <!-- optional, default is 15 -->
    <maxRetries>3</maxRetries>  <!-- optional, default is 3 -->
    <method>GET</method>  <!-- optional, valid value is
        OPTIONS/GET/HEAD/POST/PUT/DELETE/TRACE/CONNECT -->
    <url>/</url>  <!-- optional -->
    <name>http-monitor</name>  <!-- required -->
    <expected>HTTP/1</expected>  <!-- optional, Expected response string. Default is
        "HTTP/1" for http(s) protocol -->
    <send>hello</send> -->  <!-- optional, URL encoded http POST data for http(s)
        protocol -->
    <receive>ok</received> -->  <!-- optional, String to expect in the content for
        http(s) protocol -->
    <extension>no-bodymax-age=3hcontent-type=Application/xml</extension>
        <!-- Above tag is optional, advanced setting for monitor to fill more
        customized parameters -->
</monitor>
<monitor>
    <monitorId>monitor-2</monitorId>
    <type>https</type>
    <interval>5</interval>
    <timeout>15</timeout>
    <maxRetries>3</maxRetries>
    <method>GET</method>
    <url>/</url>
    <name>https-monitor</name>
</monitor>
<monitor>
    <monitorId>monitor-3</monitorId>
    <type>tcp</type>
    <interval>5</interval>
    <timeout>15</timeout>
    <maxRetries>3</maxRetries>
    <name>tcp-monitor</name>
</monitor>
</loadBalancer>

```

Configuration example to show HTTP/HTTPS Redirection, SSL Offloading, Content Switching, HTTP HealthMonitor:

```

<loadBalancer>
    <enabled>true</enabled>
    <accelerationEnabled>true</accelerationEnabled>
    <logging>
        <enable>true</enable>
        <logLevel>debug</logLevel>
    </logging>
    <applicationRule>
        <applicationRuleId>applicationRule-1</applicationRuleId>  <!-- optional, it should
            follow "applicationRule-x" pattern, required when it is referenced -->
        <name>traffic_ctrl_rule</name>  <!-- required, unique applicationRule name per
            Edge -->
        <script>acl srv1_full srv_conn(pool-http/m1) gt 50
            acl srv2_full srv_conn(pool-http/m2) gt 50 use_backend pool-backup if
            srv1_full or srv2_full</script>  <!-- required, one ACL rule -->
    </applicationRule>
    <applicationRule>
        <applicationRuleId>applicationRule-2</applicationRuleId>

```

```

    <name>redirection_rule</name>
    <script>acl google_page url_beg /google
redirect location https://www.google.com/ if google_page</script>
</applicationRule>
</applicationRule>
    <applicationRuleId>applicationRule-3</applicationRuleId>
    <name>l7_rule</name>
    <script>acl backup_page url_beg /backup
use_backend pool-backup if backup_page</script>
</applicationRule>
</virtualServer>
    <virtualServerId>virtualServer-1</virtualServerId>
    <name>http_redirection_vip</name>
    <description>http redirection virtualServer</description>
    <enabled>true</enabled>
    <ipAddress>10.117.35.171</ipAddress>
    <protocol>http</protocol>
    <port>80</port>
    <connectionLimit>123</connectionLimit>
    <connectionRateLimit>123</connectionRateLimit>
    <applicationProfileId>applicationProfile-1</applicationProfileId>
    <enableServiceInsertion>>false</enableServiceInsertion>
    <accelerationEnabled>true</accelerationEnabled>
</virtualServer>
</virtualServer>
    <virtualServerId>virtualServer-2</virtualServerId>
    <name>https_vip</name>
    <description>https virtualServer</description>
    <enabled>true</enabled>
    <ipAddress>10.117.35.171</ipAddress>
    <protocol>https</protocol>
    <port>443</port>
    <connectionLimit>123</connectionLimit>
    <connectionRateLimit>123</connectionRateLimit>
    <defaultPoolId>pool-1</defaultPoolId>
    <applicationProfileId>applicationProfile-2</applicationProfileId>
    <applicationRuleId>applicationRule-1</applicationRuleId> <!-- optional, it is
        applicationRuleId list, each item should be a valid applicationRuleId -->
    <applicationRuleId>applicationRule-2</applicationRuleId>
    <applicationRuleId>applicationRule-3</applicationRuleId>
    <enableServiceInsertion>>false</enableServiceInsertion>
    <accelerationEnabled>true</accelerationEnabled>
</virtualServer>
</applicationProfile>
    <applicationProfileId>applicationProfile-1</applicationProfileId>
    <name>https_redirection_application_profile</name>
    <insertXForwardedFor>>false</insertXForwardedFor>
    <sslPassthrough>>false</sslPassthrough>
    <httpRedirect> <!-- optional -->
        <to>https://10.117.35.171</to> <!-- required, a uri -->
    </httpRedirect>
</applicationProfile>
</applicationProfile>
    <applicationProfileId>applicationProfile-2</applicationProfileId>
    <name>ssl_offloading_application_profile</name>
    <insertXForwardedFor>>false</insertXForwardedFor>
    <serverSslEnabled>true</serverSslEnabled> <!-- optional, default is true, it is a
        switch flag to enable/disable serverSsl offloading -->
    <sslPassthrough>>false</sslPassthrough>
    <clientSsl> <!-- optional -->
        <clientAuth>ignore</clientAuth> <!-- optional, valid values are ignore/required
        -->
        <ciphers>AES:ALL:!NULL:!eNULL:+RC4:@STRENGTH</ciphers> <!-- required, ciphers
        -->
        <serviceCertificate>certificate-4</serviceCertificate> <!-- required, a
        serviceCertificate List -->
        <caCertificate>certificate-3</caCertificate> <!-- required, a ca list -->
        <crlCertificate>crl-1</crlCertificate> <!-- optional, a crl list -->

```

```

    </clientSsl>
  <serverSsl>
    <ciphers>AES:ALL:!aNULL:!eNULL:+RC4:@STRENGTH</ciphers>
    <serviceCertificate>certificate-4</serviceCertificate>
    <caCertificate>certificate-3</caCertificate>
    <crlCertificate>crl-1</crlCertificate>
  </serverSsl>
</applicationProfile>
<pool>
  <poolId>pool-1</poolId>
  <name>pool-http</name>
  <description>pool-http</description>
  <transparent>>false</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-1</monitorId>
  <member>
    <memberId>member-1</memberId>
    <ipAddress>192.168.101.101</ipAddress>
    <weight>1</weight>
    <port>80</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m1</name>
  </member>
  <member>
    <memberId>member-2</memberId>
    <ipAddress>192.168.101.102</ipAddress>
    <weight>1</weight>
    <port>80</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m2</name>
  </member>
</pool>
<pool>
  <poolId>pool-2</poolId>
  <name>pool-backup</name>
  <description>pool backup</description>
  <transparent>>false</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-1</monitorId>
  <member>
    <memberId>member-3</memberId>
    <ipAddress>192.168.102.101</ipAddress>
    <weight>1</weight>
    <port>80</port>
    <name>m3</name>
  </member>
  <member>
    <memberId>member-4</memberId>
    <ipAddress>192.168.102.102</ipAddress>
    <weight>1</weight>
    <port>80</port>
    <name>m4</name>
  </member>
</pool>
<monitor>
  <monitorId>monitor-1</monitorId>
  <type>http</type>
  <interval>5</interval>
  <timeout>15</timeout>
  <maxRetries>3</maxRetries>
  <method>GET</method>
  <url>/</url>
  <name>http-monitor</name>
</monitor>

```

```
</loadBalancer>
```

---

For the data path to work, you need to add firewall rules to allow required traffic as per the load balancer configuration.

## Query Load Balancer Configuration

Gets current load balancer configuration.

**Example 9-64.** Retrieve load balancer configuration

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
```

Response Body:

See [Example 9-63](#).

---

## Delete Load Balancer Configuration

**Example 9-65.** Delete load balancer configuration

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
```

---

## Manage Application profiles

You create an application profile to define the behavior of a particular type of network traffic. After configuring a profile, you associate the profile with a virtual server. The virtual server then processes traffic according to the values specified in the profile. Using profiles enhances your control over managing network traffic, and makes traffic-management tasks easier and more efficient.

### Append Application Profile

Adds an application profile to the current set of application profiles.

**Example 9-66.** Append profile

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/applicationprofiles
```

Request Body:

```
<applicationProfile>
  <name>http_application_profile_2</name>
  <insertXForwardedFor>true</insertXForwardedFor>
  <sslPassthrough>true</sslPassthrough>
  <persistence>
    <method>cookie</method>
    <cookieName>JSESSIONID</cookieName>
    <cookieMode>insert</cookieMode>
  </persistence>
</applicationProfile>
```

---

## Modify Application Profile

Modifies an application profile.

### Example 9-67. Modify profile

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
/applicationprofiles/applicationProfileId
```

Request Body:

```
<applicationProfile>
  <name>http_application_profile_2</name>
  <insertXForwardedFor>true</insertXForwardedFor>
  <sslPassthrough>true</sslPassthrough>
  <persistence>
    <method>cookie</method>
    <cookieName>JSESSIONID</cookieName>
    <cookieMode>insert</cookieMode>
  </persistence>
</applicationProfile>
```

---

## Query Application Profile

Retrieves an application profile.

### Example 9-68. Query profile

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
/applicationprofiles/applicationProfileId
```

Response Body:

```
<applicationProfile>
  <applicationProfileId>applicationProfile-2</applicationProfileId>
  <name>HTTPS-Application-Profile</name>
  <insertXForwardedFor>true</insertXForwardedFor>
  <sslPassthrough>false</sslPassthrough>
  <template>HTTPS</template>
  <serverSslEnabled>false</serverSslEnabled>
</applicationProfile>
```

---

## Query all Application Profiles

Retrieves all application profiles on Edge.

### Example 9-69. Query profiles

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
/applicationprofiles/
```

Response Body:

```
<loadBalancer>
  <applicationProfile>
    <applicationProfileId>applicationProfile-2</applicationProfileId>
    <name>HTTPS-Application-Profile</name>
    <insertXForwardedFor>true</insertXForwardedFor>
    <sslPassthrough>false</sslPassthrough>
    <template>HTTPS</template>
    <serverSslEnabled>false</serverSslEnabled>
  </applicationProfile>
</loadBalancer>
```

```

</applicationProfile>
<applicationProfile>
  <applicationProfileId>applicationProfile-3</applicationProfileId>
  <persistence>
    <method>cookie</method>
    <cookieName>JSESSIONID</cookieName>
    <cookieMode>insert</cookieMode>
  </persistence>
  <name>HTTP-Application-Profile</name>
  <insertXForwardedFor>true</insertXForwardedFor>
  <sslPassthrough>>false</sslPassthrough>
  <template>HTTP</template>
  <serverSslEnabled>>false</serverSslEnabled>
</applicationProfile>
<applicationProfile>
  <applicationProfileId>applicationProfile-4</applicationProfileId>
  <persistence>
    <method>sourceip</method>
  </persistence>
  <name>TCP-Application-Profile</name>
  <insertXForwardedFor>>false</insertXForwardedFor>
  <sslPassthrough>>false</sslPassthrough>
  <template>TCP</template>
  <serverSslEnabled>>false</serverSslEnabled>
</applicationProfile>
</loadBalancer>

```

---

### Delete Application Profile

Deletes an application profile.

#### Example 9-70. Delete profile

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
/applicationprofiles/applicationProfileId
```

---

### Delete all Application Profiles

Deletes all application profile.

#### Example 9-71. Delete profiles

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
/applicationprofiles
```

---

## Manage Application Rules

You can write an application rule to directly manipulate and manage IP application traffic.

### Append Application Rule

Adds an application rule.

#### Example 9-72. Append rule

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
/applicationrules
```

Request Body:

```
<applicationRule>
  <name>redirection_rule</name>
  <script>acl vmware_page url_beg /vmware redirect location https://www.vmware.com/ if
    vmware_page</script>
</applicationRule>
```

---

### Modify Application Rule

Modifies an application rule.

#### Example 9-73. Modify rule

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
    /applicationrules/applicationruleId
```

Request Body:

See [Example 9-72](#).

---

### Query Application Rule

Retrieves an application rule.

#### Example 9-74. Query rule

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
    /applicationrules/applicationruleId
```

Response Body:

See [Example 9-72](#).

---

### Query all Application Rules

Retrieves all application rules on Edge.

#### Example 9-75. Query rules

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
    /applicationrules
```

Response Body:

See [Example 9-72](#).

---

### Delete Application Rule

Deletes an application rule.

#### Example 9-76. Delete rule

---

Request:

---

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
        /applicationrules/applicationruleId
```

---

### Delete all Application Rules

Deletes all application rules.

#### Example 9-77. Delete rules

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config
        /applicationrules
```

---

## Manage Load Balancer Monitors

You create a service monitor to define health check parameters for a particular type of network traffic. When you associate a service monitor with a pool, the pool members are monitored according to the service monitor parameters.

### Append Monitor

Adds a load balancer monitor.

#### Example 9-78. Append monitor

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/monitors
```

Request Body:

```
<monitor>
  <type>http</type>
  <interval>5</interval>
  <timeout>15</timeout>
  <maxRetries>3</maxRetries>
  <method>GET</method>
  <url>/</url>
  <name>http-monitor-2</name>
</monitor>
```

---

### Modify Monitor

Modifies a load balancer monitor.

#### Example 9-79. Modify monitor

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/monitors
    /monitorId
```

Request Body:

```
<monitor>
  <type>http</type>
  <interval>5</interval>
  <timeout>15</timeout>
  <maxRetries>3</maxRetries>
  <method>GET</method>
  <url>/</url>
  <name>http-monitor-2</name>
```

```
</monitor>
```

---

## Query Monitor

Retrieves a load balancer monitor.

### Example 9-80. Query monitor

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/monitors
/monitorId
```

Response Body:

```
<monitor>
  <type>http</type>
  <interval>5</interval>
  <timeout>15</timeout>
  <maxRetries>3</maxRetries>
  <method>GET</method>
  <url>/</url>
  <name>http-monitor-2</name>
</monitor>
```

## Query all Monitors

Retrieves all load balancer monitors.

### Example 9-81. Query monitors

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/monitors
```

Request Body:

```
<loadBalancer>
  <monitor>
    <monitorId>monitor-1</monitorId>
    <type>http</type>
    <interval>5</interval>
    <timeout>15</timeout>
    <maxRetries>3</maxRetries>
    <method>GET</method>
    <url>/</url>
    <name>http-monitor</name>
  </monitor>
  <monitor>
    <monitorId>monitor-2</monitorId>
    <type>https</type>
    <interval>5</interval>
    <timeout>15</timeout>
    <maxRetries>3</maxRetries>
    <method>GET</method>
    <url>/</url>
    <name>https-monitor</name>
  </monitor>
  <monitor>
    <monitorId>monitor-3</monitorId>
    <type>tcp</type>
    <interval>5</interval>
    <timeout>15</timeout>
    <maxRetries>3</maxRetries>
    <name>tcp-monitor</name>
  </monitor>
</loadBalancer>
```

```
</loadBalancer>
```

---

### Delete Monitor

Deletes a load balancer monitor.

#### Example 9-82. Delete monitor

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/monitors
/monitorId
```

---

### Delete all Monitors

Deletes all load balancer monitors.

#### Example 9-83. Delete monitors

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/monitors
```

---

## Manage Virtual Servers

You can add an NSX Edge internal or uplink interface as a virtual server.

### Append Virtual Server

Adds a virtual server.

#### Example 9-84. Append virtual server

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/
virtualservers
```

Request Body:

```
<virtualServer>
  <name>http_vip_2</name>
  <description>http virtualServer 2</description>
  <enabled>true</enabled>
  <ipAddress>10.117.35.172</ipAddress>
  <protocol>http</protocol>
  <port>443,6000-7000</port>  <!-- port field changed from a single integer to a
    string with format where numbers are separated by "," or a range expressed
    by "-" -->
  <connectionLimit>123</connectionLimit>
  <connectionRateLimit>123</connectionRateLimit>
  <applicationProfileId>applicationProfile-1</applicationProfileId>
  <defaultPoolId>pool-1</defaultPoolId>
  <enableServiceInsertion>false</enableServiceInsertion>
  <accelerationEnabled>true</accelerationEnabled>
</virtualServer>
```

---

### Query a Virtual Server

Retrieves specified virtual server details.

**Example 9-85.** Query virtual server

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/
    virtualservers/virtualserverId
```

Response Body:

See [Example 9-84](#).**Query all Virtual Servers**

Retrieves all virtual servers.

**Example 9-86.** Query virtual servers

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/
    virtualservers
```

Response Body:

```
<loadBalancer>
  <virtualServer>
    <virtualServerId>virtualServer-1</virtualServerId>
    <name>http_vip</name>
    <description>http virtualServer</description>
    <enabled>true</enabled>
    <ipAddress>10.117.35.172</ipAddress>
    <protocol>http</protocol>
    <port>443,6000-7000</port> <!-- port field changed from a single integer to a
      string with format where numbers are separated by "," or a range expressed
      by "-" -->
    <connectionLimit>123</connectionLimit>
    <connectionRateLimit>123</connectionRateLimit>
    <defaultPoolId>pool-1</defaultPoolId>
    <applicationProfileId>applicationProfile-1</applicationProfileId>
    <enableServiceInsertion>false</enableServiceInsertion>
    <accelerationEnabled>true</accelerationEnabled>
  </virtualServer>
  <virtualServer>
    <virtualServerId>virtualServer-2</virtualServerId>
    <name>https_vip</name>
    <description>https virtualServer</description>
    <enabled>true</enabled>
    <ipAddress>10.117.35.172</ipAddress>
    <protocol>https</protocol>
    <port>443,6000-7000</port> <!-- port field changed from a single integer to a
      string with format where numbers are separated by "," or a range expressed
      by "-" -->
    <connectionLimit>123</connectionLimit>
    <connectionRateLimit>123</connectionRateLimit>
    <defaultPoolId>pool-2</defaultPoolId>
    <applicationProfileId>applicationProfile-2</applicationProfileId>
    <enableServiceInsertion>false</enableServiceInsertion>
    <accelerationEnabled>false</accelerationEnabled>
  </virtualServer>
  <virtualServer>
    <virtualServerId>virtualServer-3</virtualServerId>
    <name>tcp_transparent_vip</name>
    <description>tcp virtualServer</description>
    <enabled>true</enabled>
    <ipAddress>10.117.35.172</ipAddress>
    <protocol>tcp</protocol>
```

```

<port>443,6000-7000</port>  <!-- port field changed from a single integer to a
      string with format where numbers are separated by "," or a range expressed
      by "-" -->
<connectionLimit>123</connectionLimit>
<defaultPoolId>pool-3</defaultPoolId>
<applicationProfileId>applicationProfile-3</applicationProfileId>
<enableServiceInsertion>>false</enableServiceInsertion>
<accelerationEnabled>>true</accelerationEnabled>
</virtualServer>
<virtualServer>
  <virtualServerId>virtualServer-4</virtualServerId>
  <name>tcp_snat_vip</name>
  <description>tcp snat virtualServer</description>
  <enabled>>true</enabled>
  <ipAddress>10.117.35.172</ipAddress>
  <protocol>tcp</protocol>
  <port>443,6000-7000</port>  <!-- port field changed from a single integer to a
      string with format where numbers are separated by "," or a range expressed
      by "-" -->
  <connectionLimit>123</connectionLimit>
  <defaultPoolId>pool-4</defaultPoolId>
  <applicationProfileId>applicationProfile-3</applicationProfileId>
  <enableServiceInsertion>>false</enableServiceInsertion>
  <accelerationEnabled>>true</accelerationEnabled>
</virtualServer>
</loadBalancer>

```

---

### Delete a Virtual Server

Deletes specified virtual server.

#### Example 9-87. Delete virtual server

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/
virtualservers/virtualserverId
```

---

### Delete all Virtual Server

Deletes all virtual servers.

#### Example 9-88. Delete all virtual server

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/
virtualservers
```

---

## Manage Backend Pools

You can add a server pool to manage and share backend servers flexibly and efficiently. A pool manages load balancer distribution methods and has a service monitor attached to it for health check parameters.

### Append Backend Pool

Adds a load balancer server pool to the specified NSX Edge.

#### Example 9-89. Append backend pool

---

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/pools>

Request Body:

```
<pool>
  <name>pool-tcp-snat-2</name>
  <description>pool-tcp-snat-2</description>
  <transparent>>false</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-3</monitorId>
  <member>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <port>80</port>  <!-- Becomes optional, in which case monitorPort needs to be
                     configured. In other words - either port or monitorPort - one of them must
                     be specified-->
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m5</name>
    <monitorPort>80</monitorPort>  <!-- Becomes optional, in which case port needs to
                                   be configured. In other words - either port or monitorPort - one of them
                                   must be specified-->
  </member>
  <member>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>80</port>  <!-- Becomes optional, in which case monitorPort needs to be
                     configured. In other words - either port or monitorPort - one of them must
                     be specified-->
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m6</name>
    <monitorPort>80</monitorPort>  <!-- Becomes optional, in which case port needs to
                                   be configured. In other words - either port or monitorPort - one of them
                                   must be specified-->
  </member>
</pool>
```

---

## Modify a Backend Pool

Updates the specified pool.

**Example 9-90.** Modify backend pool

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/pools/poolId>

Request Body:

```
<pool>
  <name>pool-tcp-snat-2</name>
  <description>pool-tcp-snat-3</description>
  <transparent>>false</transparent>
  <algorithm>round-robin</algorithm>
  <monitorId>monitor-3</monitorId>
  <member>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <port>1234</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m5</name>
    <condition>enabled\disabled</condition>
    <monitorPort>80</monitorPort>
  </member>
  <member>
    <ipAddress>192.168.101.202</ipAddress>
```

```

    <weight>1</weight>
    <port>1234</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m6</name>
    <condition>enabled\disabled</condition>
    <monitorPort>80</monitorPort>
  </member>
</pool>

```

Alternatively, the following call can be used to achieve the same result:

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/members/
memberId?enable=true\false
```

---

## Query Backend Pool Details

Retrieves information about the specified pool.

**Example 9-91.** Get backend pool details

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/pools/poolId
```

Response Body:

See Example [Example 9-90](#).

---

## Query all Backend Pools

Gets all backend pools configured for the specified NSX Edge.

**Example 9-92.** Query all backend pools

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/pools
```

Response Body:

```

<loadBalancer>
  <pool>
    <type>slb</type>
    <poolId>pool-1</poolId>
    <name>pool-http</name>
    <description>pool-http</description>
    <algorithm>round-robin</algorithm>
    <transparent>true</transparent>
    <monitorId>monitor-1</monitorId>
    <member>
      <memberId>member-1</memberId>
      <ipAddress>192.168.101.201</ipAddress>
      <weight>1</weight>
      <port>80</port>
      <maxConn>100</maxConn>
      <minConn>10</minConn>
      <condition>enabled</condition>
      <name>m1</name>
    </member>
    <member>
      <memberId>member-2</memberId>
      <ipAddress>192.168.101.202</ipAddress>
      <weight>1</weight>
      <port>80</port>

```

```

    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m2</name>
  </member>
</pool>
<pool>
  <type>slb</type>
  <poolId>pool-2</poolId>
  <name>pool-https</name>
  <description>pool-https</description>
  <algorithm>round-robin</algorithm>
  <transparent>false</transparent>
  <monitorId>monitor-2</monitorId>
  <member>
    <memberId>member-11</memberId>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <port>443</port>
    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m3</name>
  </member>
  <member>
    <memberId>member-4</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>443</port>
    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m4</name>
  </member>
</pool>
<pool>
  <type>slb</type>
  <poolId>pool-3</poolId>
  <name>pool-tcp</name>
  <description>pool-tcp</description>
  <algorithm>round-robin</algorithm>
  <transparent>true</transparent>
  <monitorId>monitor-3</monitorId>
  <member>
    <memberId>member-5</memberId>
    <ipAddress>192.168.101.201</ipAddress>
    <weight>1</weight>
    <monitorPort>80</monitorPort>
    <port>1234</port>
    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m5</name>
  </member>
  <member>
    <memberId>member-6</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <monitorPort>80</monitorPort>
    <port>1234</port>
    <maxConn>100</maxConn>
    <minConn>10</minConn>
    <condition>enabled</condition>
    <name>m6</name>
  </member>
</pool>
<pool>
  <type>slb</type>

```

```

<poolId>pool-4</poolId>
<name>pool-tcp-snat</name>
<description>pool-tcp-snat</description>
<algorithm>round-robin</algorithm>
<transparent>false</transparent>
<monitorId>monitor-3</monitorId>
<member>
  <memberId>member-7</memberId>
  <ipAddress>192.168.101.201</ipAddress>
  <weight>1</weight>
  <monitorPort>80</monitorPort>
  <port>1234</port>
  <maxConn>100</maxConn>
  <minConn>10</minConn>
  <condition>enabled</condition>
  <name>m7</name>
</member>
<member>
  <memberId>member-8</memberId>
  <ipAddress>192.168.101.202</ipAddress>
  <weight>1</weight>
  <monitorPort>80</monitorPort>
  <port>1234</port>
  <maxConn>100</maxConn>
  <minConn>10</minConn>
  <condition>enabled</condition>
  <name>m8</name>
</member>
</pool>
</loadBalancer>

```

---

### Delete a Backend Pool

Deletes the specified pool.

#### Example 9-93. Delete backend pool

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/pools
/poolId
```

---

### Delete all Backend Pools

Deletes all backend pools configured for the specified NSX Edge.

#### Example 9-94. Delete backend pool

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/pools
```

---

## Query Statistics

Retrieves load balancer statistics.

#### Example 9-95. Retrieve load balancer statistics

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/statistics
```

Response Body:

```

<loadBalancerStatusAndStats>
  <timeStamp>1359722922</timeStamp>
  <pool>
    <poolId>pool-1</poolId>
    <name>pool-http</name>
    <member>
      <memberId>member-1</memberId>
      <name>m1</name>
      <ipAddress>192.168.101.201</ipAddress>
      <status>UP</status>
      <bytesIn>70771</bytesIn>
      <bytesOut>74619</bytesOut>
      <curSessions>0</curSessions>
      <maxSessions>1</maxSessions>
      <rate>0</rate>
      <rateMax>17</rateMax>
      <totalSessions>142</totalSessions>
    </member>
    <member>
      <memberId>member-2</memberId>
      <name>m2</name>
      <ipAddress>192.168.101.202</ipAddress>
      <status>UP</status>
      <bytesIn>70823</bytesIn>
      <bytesOut>70605</bytesOut>
      <curSessions>0</curSessions>
      <maxSessions>1</maxSessions>
      <rate>0</rate>
      <rateMax>17</rateMax>
      <totalSessions>141</totalSessions>
    </member>
    <status>UP</status>
    <bytesIn>141594</bytesIn>
    <bytesOut>145224</bytesOut>
    <curSessions>0</curSessions>
    <maxSessions>2</maxSessions>
    <rate>0</rate>
    <rateMax>34</rateMax>
    <totalSessions>283</totalSessions>
  </pool>
  <virtualServer>
    <virtualServerId>virtualServer-9</virtualServerId>
    <name>http_vip</name>
    <ipAddress>10.117.35.172</ipAddress>
    <status>OPEN</status>
    <bytesIn>141594</bytesIn>
    <bytesOut>145224</bytesOut>
    <curSessions>1</curSessions>
    <httpReqTotal>283</httpReqTotal>
    <httpReqRate>0</httpReqRate>
    <httpReqRateMax>34</httpReqRateMax>
    <maxSessions>2</maxSessions>
    <rate>0</rate>
    <rateLimit>0</rateLimit>
    <rateMax>2</rateMax>
    <totalSessions>13</totalSessions>
  </virtualServer>
  <globalSite>
    <name>BJ site</name>
    <globalSiteId>site-3</globalSiteId>
    <msgSent>3</msgSent>
    <msgRecv>747</msgRecv>
    <msgRate>0</msgRate>
    <dnsReq>0</dnsReq>
    <dnsResolved>0</dnsResolved>
  </globalSite>
  <globalIp>
    <fqdn>www.company.com</fqdn>
  </globalIp>

```

```

    <globalIpId>gip-3</globalIpId>
    <dnsReq>0</dnsReq>
    <dnsResolved>0</dnsResolved>
    <dnsMiss>0</dnsMiss>
  </globalIp>
  <globalPool>
    <name>www-primary</name>
    <poolId>pool-1</poolId>
    <dnsReq>0</dnsReq>
    <dnsResolved>0</dnsResolved>
    <dnsMiss>0</dnsMiss>
    <member>
      <name>10.117.7.110</name>
      <memberId>member-3</memberId>
      <status>up</status>
      <dnsHit>0</dnsHit>
      <cpuUsage>3</cpuUsage>
      <memUsage>91</memUsage>
      <sessions>0</sessions>
      <curConn>14</curConn>
      <sessLimit>0</sessLimit>
      <sessRate>0</sessRate>
      <totalThroughput>0</totalThroughput>
      <packagesPerSec>0</packagesPerSec>
    </member>
  </globalPool>
  <globalPool>
    <name>www-primary</name>
    <poolId>pool-1</poolId>
    <dnsReq>0</dnsReq>
    <dnsResolved>0</dnsResolved>
    <dnsMiss>0</dnsMiss>
    <member>
      <name>10.117.7.110</name>
      <memberId>member-3</memberId>
      <status>up</status>
      <dnsHit>0</dnsHit>
      <cpuUsage>3</cpuUsage>
      <memUsage>91</memUsage>
      <sessions>0</sessions>
      <curConn>14</curConn>
      <sessLimit>0</sessLimit>
      <sessRate>0</sessRate>
      <totalThroughput>0</totalThroughput>
      <packagesPerSec>0</packagesPerSec>
    </member>
  </globalPool>
</loadBalancerStatusAndStats>

```

---

## Update Load Balancer Acceleration Mode

**Example 9-96.** Update acceleration mode

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/acceleration?enable=true\false
```

---

## Update Load Balancer Member Condition

**Example 9-97.** Update member condition

---

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/loadbalancer/config/members/memberId?enable=true>false>

---

## Configure DNS Servers

You can configure external DNS servers to which vShield Edge can relay name resolution requests from clients. vShield Edge will relay client application requests to the DNS servers to fully resolve a network name and cache the response from the servers.

### Configure DNS

Updates the DNS server configuration. DNS server list allows two addresses – primary and secondary. The default cache size is 16 MB where the minimum can be 1 MB, and the maximum 8196 MB.

The default listeners is any, which means listen on all VSE interfaces. If provided, the listener's IP address must be assigned to an internal interface.

Logging is disabled by default.

#### Example 9-98. Configure DNS servers

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dns/config>

Request Body:

```
<dns>
  <enabled>true</enabled>  <!-- optional. default is true-->
  <dnsServers>
    <ipAddress>10.117.0.1</ipAddress> <!-- Max is 2 external dns server -->
  </dnsServers>
  <cacheSize>128</cacheSize>  <!-- optional. default is 16, max to 8192 -->
  <listeners>  <!-- optiona. if provided, IPs must be defined on Edge interfaces. -->
    <ipAddress>192.168.100.1</ipAddress>
    <ipAddress>192.168.100.2</ipAddress>
  </listeners>
  <logging>  <!-- optinal. default is disabled. -->
    <logLevel>info</logLevel>  <!-- optional. default is "info" -->
    <enable>true</enable>  <!-- optional. default is "false" -->
  </logging>
</dns>
```

---

### Retrieve DNS Configuration

Gets details of DNS configuration, including the service status.

#### Example 9-99. Get DNS server configuration

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dns/config>

Response Body:

```
<dns>
  <enabled>true</enabled>
  <dnsServers>
    <ipAddress>10.117.0.1</ipAddress>
  </dnsServers>
  <cacheSize>128</cacheSize>
  <listeners>
    <ipAddress>192.168.100.1</ipAddress>
    <ipAddress>192.168.100.2</ipAddress>
  </listeners>
</dns>
```

```

</listeners>
<logging>
  <logLevel>info</logLevel>
  <enable>true</enable>
</logging>
</dns>

```

---

## Delete DNS Configuration

Deletes DNS servers.

**Example 9-100.** Delete DNS servers

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dns/config
```

---

## Retrieve DNS Statistics

Gets DNS server statistics.

**Example 9-101.** Get DNS server statistics

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dns/statistics
```

Response Body:

```

<dns>
  <stats>
    <timestamp>2011-10-10 12:12:12</timestamp>
    <requests>
      <total>120000</total>
      <queries>110000</queries>
    </requests>
    <responses>
      <total>108000</total>
      <success>105000</success>
      <nxdomain>1000</nxdomain>
      <servFail>400</servFail>
      <formErr>300</formErr>
      <nxdomain>1000</nxdomain>
      <others>300</others>
    </responses>
    <cachedDBRRSet>15000</cachedDBRRSet>
  </stats>
</dns>

```

---

where

- requests.total indicates all the incoming requests to the DNS server, including DNS query and other types of request (e.g. transfer, updates)
- requests.queries indicates all the DNS queries the server received.
- responses.total indicates all responses the server returned to requests. It could be different from the requests.total because some requests could be rejected. total = success + nxrrset + servFail + formErr + nxdomain + others
- responses.success indicates all the successful DNS answers.
- responses.nxdomain indicates the count of no existent resource record set

- `responses.servFail` indicates the count of SERVFAIL answer
- `responses.formErr` indicates the count of format error answer
- `responses.nxdomain` indicates the count of no-suhc-domain answer
- `responses.others` indicates the count of other type of answers.

## Working with DHCP Service

NSX Edge provides DHCP service to bind assigned IP addresses to MAC addresses, helping to prevent MAC spoofing attacks. All virtual machines protected by a NSX Edge can obtain IP addresses dynamically from the NSX Edge DHCP service.

NSX Edge supports IP address pooling and one-to-one static IP address allocation based on the vCenter managed object ID (`vmId`) and interface ID (`interfaceId`) of the requesting client.

If either bindings or pools are not included in the PUT call, existing bindings or pools are deleted.

All DHCP settings configured by REST requests appear under the **NSX Edge > DHCP** tab for the appropriate NSX Edge in the NSX Manager user interface and in vSphere Client plug-in.

NSX Edge DHCP service adheres to the following rules:

- Listens on the NSX Edge internal interface (non-uplink interface) for DHCP discovery.
- As stated above, `vmId` specifies the `vc-moref-id` of the virtual machine, and `vnid` specifies the index of the vNic for the requesting client. The `hostname` is an identification of the binding being created. This `hostName` is not pushed as the specified host name of the virtual machine.
- By default, all clients use the IP address of the internal interface of the NSX Edge as the default gateway address. To override it, specify `defaultGateway` per binding or per pool. The client's broadcast and `subnetMask` values are from the internal interface for the container network.
- `leaseTime` can be infinite, or a number of seconds. If not specified, the default lease time is 1 day.
- Logging is disabled by default.
- Setting the parameter `enable=true` starts the DHCP service while `enable=false` stops the service.
- Both `staticBinding` and `ipPools` must be part of the request body. Else, they will be deleted if configured earlier.

## Configure DHCP

### Example 9-102. Configure DHCP service

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/config
```

Request Body:

```
<dhcp>
  <enabled>true</enabled>    <!-- optional, default is "true". -->
  <staticBindings>
    <staticBinding>    <!-- NOTE: user can either specify macAddress directly, or
      specify vmId and vnid. In case both are specified, only macAddress will
      be used; vmId and vnid will be ignored.-->
      <macAddress>12:34:56:78:90:AB</macAddress> <!-- optional. -->
      <vmId>vm-111</vmId>    <!-- optional. the vm must be connected to the given
        vNic below. -->
      <vnid>1</vnid>    <!-- optional. possible values 0 to 9 -->
      <hostname>abcd</hostname>    <!-- optional. disallow duplicate. -->
```

```

<ipAddress>192.168.4.2</ipAddress>  <!--the IP can either belong to a a
  subnet of one of Edge's vNics or it can be any valid IP address, but the IP
  must not overlap with any primary/secondary IP addresses associated with
  any of Edge's vNICs. If the IP does not belong to any Edge vNic subnets,
  you must ensure that the default gateway and subnetMask are configured via
  this API call. -->
<subnetMask>255.255.255.0</subnetMask>  <!-- optional. If the assigned IP
  belongs to one of Edge vNICs' subnets, the default is the subnetMask of
  this vNic subnet, otherwise the default is 255.255.255.0.-->
<defaultGateway>192.168.4.1</defaultGateway>  <!-- optional. If the assigned
  IP belongs to one of Edge vNICs' subnets, the default is the primary IP of
  this vNic subnet. Otherwise, you must include a correct gateway IP address
  in this API call. If the IP address is not provided, the client host may
  not get default gateway IP from the DHCP server. -->
<domainName>eng.vmware.com</domainName>  <!-- optional. -->
<primaryNameServer>192.168.4.1</primaryNameServer>  <!-- optional. if
  autoConfiguredDNS=true, the DNS primary/secondary ips will be generated from
  DNS service(if configured). -->
<secondaryNameServer>4.2.2.4</secondaryNameServer>  <!-- ditto. -->
<leaseTime>infinite</leaseTime>  <!-- optional. in seconds, default is
  "86400". valid leaseTime is a valid number or "infinite". -->
<autoConfiguredDNS>true</autoConfiguredDNS>  <!-- optional. default is true.
  -->
</staticBinding>
</staticBindings>
<ipPools>
  <ipPool>
    <ipRange>192.168.4.192-192.168.4.220</ipRange>  <!-- required. The IP range
      can either fall entirely within one of the Edge vNIC subnets, or it can be
      a valid IP range outside any Edge subnets. The IP range, however, cannot
      contain an IP that is defined as a vNic primary secondary IP. If the range
      does not fall entirely within one of the Edge vNIC subnets, you must
      provide correct subnetMask and default gateway. -->
    <defaultGateway>192.168.4.1</defaultGateway>  <!-- optional. If the ipRange
      falls entirely within one of the Edge vNIC subnets, defaultGateway is set
      to the primary IP of the vNIC configured with the matching subnet.
      Otherwise, you must provide the correct gateway IP. If an IP is not
      provided, the client host may not get default gateway IP from the DHCP
      server.-->
    <subnetMask>255.255.255.0</subnetMask>  <!-- optional. If not specified, and
      the the ipRange belongs to an Edge vnic subnet, it is defaulted to a minimum
      subnet mask which is figured out with the ip-range itself, e.g. the mask of
      range 192.168.5.2-192.168.5.20 is 255.255.255.224. You can edit this range,
      if required.-->
    <domainName>eng.vmware.com</domainName>  <!-- optional. -->
    <primaryNameServer>192.168.4.1</primaryNameServer>  <!-- optional. if
      autoConfiguredDNS=true, the dns primary/secondary ips will be generated from
      DNS service(if configured). -->
    <secondaryNameServer>4.2.2.4</secondaryNameServer>  <!-- ditto. -->
    <leaseTime>3600</leaseTime>  <!-- optional. in seconds, default is "86400".
      valid leaseTime is a valid number or "infinite". -->
    <autoConfiguredDNS>true</autoConfiguredDNS>  <!-- optional. default is true.
      -->
  </ipPool>
</ipPools>
<logging>  <!-- optional. logging is disabled by default. -->
  <enable>>false</enable>  <!-- optional, default is false. -->
  <logLevel>info</logLevel>  <!-- optional, default is "info". -->
</logging>
</dhcp>

```

---

**NOTE** If the NSX Edge autoConfiguration flag and autoConfigureDNS is true, and the primaryNameServer or secondaryNameServer parameters are not specified, NSX Manager applies the DNS settings to the DHCP configuration.

## Query DHCP Configuration

Gets the DHCP configuration on a NSX Edge including IP pool and static binding assignments.

### Example 9-103. Get DHCP configuration

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/config>

Response Body:

```
<dhcp>
  <enabled>true</enabled>
  <staticBindings>
    <staticBinding>
      <vmId>vm-111</vmId>
      <vnicId>1</vnicId>
      <hostname>abcd</hostname>
      <ipAddress>192.168.4.2</ipAddress>
      <subnetMask>255.255.255.0</subnetMask>
      <defaultGateway>192.168.4.1</defaultGateway>
      <domainName>eng.vmware.com</domainName>
      <primaryNameServer>192.168.4.1</primaryNameServer>
      <secondaryNameServer>4.2.2.4</secondaryNameServer>
      <leaseTime>infinite</leaseTime>
      <autoConfiguredDNS>true</autoConfiguredDNS>
    </staticBinding>
  </staticBindings>
  <ipPools>
    <ipPool>
      <ipRange>192.168.4.192-192.168.4.220</ipRange>
      <defaultGateway>192.168.4.1</defaultGateway>
      <subnetMask>255.255.255.0</subnetMask>
      <domainName>eng.vmware.com</domainName>
      <primaryNameServer>192.168.4.1</primaryNameServer>
      <secondaryNameServer>4.2.2.4</secondaryNameServer>
      <leaseTime>3600</leaseTime>
      <autoConfiguredDNS>true</autoConfiguredDNS>
    </ipPool>
  </ipPools>
  <logging>
    <enable>false</enable>
    <logLevel>info</logLevel>
  </logging>
</dhcp>
```

---

## Delete DHCP Configuration

Deletes the DHCP configuration and reverse the configuration back to factory defaults.

### Example 9-104. Delete DHCP configuration

---

Request:

DELETE <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/config>

---

## Retrieve DHCP Lease Information

### Example 9-105. Get DHCP lease information

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/leaseInfo>

Response Body:

```
<dhcpLeases>
  <timeStamp>1326950787</timeStamp>
  <dhcpLeaseInfo>
    <leaseInfo>
      <uid>\001\000Pv\265\204\207</uid>
      <macAddress>00:50:56:b5:84:87</macAddress>
      <ipAddress>192.168.4.2</ipAddress>
      <clientHostname>vto-suse-dev</clientHostname>
      <bindingState>active</bindingState>
      <nextBindingState>free</nextBindingState>
      <cltt>4 2012/01/19 05:24:50</cltt>
      <starts>4 2012/01/19 05:24:50</starts>
      <ends>4 2012/01/19 17:24:50</ends>
      <hardwareType>ethernet</hardwareType>
    </leaseInfo>
  </dhcpLeaseInfo>
</dhcp>
```

---

### Append IP Pool to DHCP Configuration

Appends an IP pool to the DHCP configuration. Returns a pool ID within a Location HTTP header.

#### Example 9-106. Add IP pool

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/config/ippools
```

Request Body:

```
<ipPool>
  <ipRange>192.168.5.2-192.168.5.20</ipRange>
  <defaultGateway>192.168.5.1</defaultGateway>
  <domainName>eng.vmware.com</domainName>
  <primaryNameServer>1.2.3.4</primaryNameServer>
  <secondaryNameServer>4.3.2.1</secondaryNameServer>
  <leaseTime>3600</leaseTime>
  <autoConfiguredDNS>true</autoConfiguredDNS>
</ipPool>
```

---

### Append Static Binding to DHCP Configuration

Appends a static-binding to the DHCP configuration. A static-binding ID is returned within a Location HTTP header.

#### Example 9-107. Add static binding

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/config/bindings
```

Request Body:

```
<staticBinding>
  <vmId>vm-157</vmId>
  <vnicId>3</vnicId> <!-- possible values 0 to 9 -->
  <hostname>vShield-edge-2-0</hostname>
  <ipAddress>192.168.6.66</ipAddress>
  <defaultGateway>192.168.6.1</defaultGateway>
  <domainName>eng.vmware.com</domainName>
  <primaryNameServer>1.2.3.4</primaryNameServer>
  <secondaryNameServer>4.3.2.1</secondaryNameServer>
  <leaseTime>infinite</leaseTime>
  <autoConfiguredDNS>true</autoConfiguredDNS>
```

---

```
</staticBinding>
```

---

### Delete DHCP Pool

Deletes a pool specified by pool-id.

**Example 9-108.** Delete DHCP pool

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/config/ippools/poolId
```

---

### Delete DHCP Static Binding

Deletes the static-binding specified by binding-id.

**Example 9-109.** Delete DHCP static binding

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/config/bindings/bindingId
```

---

## Working with DHCP Relay

Dynamic Host Configuration Protocol (DHCP) relay enables you to leverage your existing DHCP infrastructure from within NSX without any interruption to the IP address management in your environment. DHCP messages are relayed from virtual machine(s) to the designated DHCP server(s) in the physical world. This enables IP addresses within NSX to continue to be in synch with IP addresses in other environments.

DHCP configuration is applied on the logical router port and can list several DHCP servers. Requests are sent to all listed servers. While relaying the DHCP request from the client, the relay adds a Gateway IP Address to the request. The external DHCP server uses this gateway address to match a pool and allocate an IP address for the request. The gateway address must belong to a subnet of the NSX port on which the relay is running.

You can specify a different DHCP server for each logical switch and can configure multiple DHCP servers on each logical router to provide support for multiple IP domains.

**NOTE** DHCP relay does not support overlapping IP address space (option 82).

DHCP Relay and DHCP service cannot run on a port/vNic at the same time. If a relay agent is configured on a port, a DHCP pool cannot be configured on the subnet(s) of this port.

**Example 9-110.** Configure DHCP relay

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/config/relay
```

Request Body:

```
<relay> <!-- You can configure ipPool, static-binding and relay at the same time if
      there is no any overlap on vnic. -->
  <relayServer> <!-- required. at lease one external server. -->
    <groupingObjectId>IPset1</groupingObjectId> <!-- a list of dhcp server IP addresses
      required, there can be multiple sever
      group objects, the maximum groupObject is 4 the maximum number of server IP
      addresses is 16 -->
    <groupingObjectId>IPset2</groupingObjectId>
    <ipAddress>10.117.35.202</ipAddress> <!-- support both IP address and FQDN -->
    <fqdn>www.dhcpserver</fqdn> <!-- Specify the IP of the fqdn, and add a Firewall
      rule to allow the response from the server represented by the fqdn such as:
      src - the IP; dest - any; service - udp:67:any. -->
```

```

</relayServer>
<relayAgents>  <!-- required. at least one relayAgent. -->
  <relayAgent>  <!-- NOTE: in case this is a node in the middle of
                 chained-relayed-dhcp, you must create a route rule to dispatch the response
                 from upper relay/server to the lower relay such as:
                 net:giAddress-of-the-tail-node, mask:xxx, gw:
                 uplink-ip-of-the-lower-relay. -->
    <vnicIndex>1</vnicIndex>  <!-- required. No default. specify the vnic that
                             proxy the dhcp request. -->
    <giAddress>192.168.1.254</giAddress>  <!-- optional. Defaults to the vnic
                                         primary address. Only one giAddress allowed. -->
  </relayAgent>
  <relayAgent>
    <vnicIndex>3</vnicIndex>
    <giAddress>192.168.3.254</giAddress>
  </relayAgent>
</relayAgents>
</relay>

```

---

## Query DHCP Relay

### Example 9-111. Query DHP Relay

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/config/relay
```

Response Body:

```

<relay>
  <relayServer>
    <groupingObjectId>IPset1</groupingObjectId>
    <groupingObjectId>IPset2</groupingObjectId>
  </relayServer>
  <relayAgents>
    <relayAgent>
      <vnicIndex>1</vnicIndex>
      <giAddress>
        192.168.1.254</giAddress>
    </relayAgent>
    <relayAgent>
      <vnicIndex >3</vnicIndex>
      <giAddress>192.168.3.254</giAddress>
    </relayAgent>
  </relayAgents>
</relay>

```

---

## Delete DHCP Relay Configuration

### Example 9-112. Delete DHP relay configuration

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dhcp/config/relay
```

---

## Working with High Availability (HA)

High Availability (HA) ensures that a NSX Edge appliance is always available on your virtualized network. You can enable HA either when installing NSX Edge or on an installed NSX Edge instance.

If a single appliance is associated with NSX Edge, the appliance configuration is cloned for the standby appliance. If two appliances are associated with NSX Edge and one of them is deployed, this REST call deploys the remaining appliance and push HA configuration to both.

HA relies on an internal interface. If an internal interface does not exist, this call will not deploy the secondary appliance, or push HA config to appliance. The enabling of HA will be done once an available internal interface is added.

If the PUT call includes an empty xml `<highAvailability />` or `enabled=false`, it acts as a DELETE call.

---

#### Example 9-113. Configure high availability

Request:

PUT `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/highavailability/config`

Request Body:

```
<highAvailability>
  <ipAddresses>  <!-- Optional. It is a pair of ipAddresses with /30 subnet mandatory,
                 one for each appliance. If provided, they must NOT overlap with any
                 subnet defined on the Edge vNics. If not specified, a pair of ips will
                 be picked up from reserved subnet 169.254.0.0/16. -->
    <ipAddress>192.168.10.1/30</ipAddress>
    <ipAddress>192.168.10.2/30</ipAddress>
  </ipAddresses>
  <declareDeadTime>6</declareDeadTime>  <!-- Optional. Default is 6 seconds -->
  <enabled>true</enabled>  <!-- optional, defaults to true. The enabled flag will cause
                           the HA appliance be deployed or destroyed. -->
</highAvailability>
```

---

## Retrieve High Availability Configuration

---

#### Example 9-114. Get high availability configuration

Request:

GET `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/highavailability/config`

Response Body:

```
<highAvailability>
  <vnic>1</vnic>
  <ipAddresses>
    <ipAddress>192.168.10.1/30</ipAddress>
    <ipAddress>192.168.10.2/30</ipAddress>
  </ipAddresses>
  <declareDeadTime>6</declareDeadTime>  <!-- Optional. Default is 6 seconds -->
</highAvailability>
```

---

## Delete High Availability Configuration

NSX Manager deletes the standby appliance and removes the HA config from the active appliance.

You can also delete the HA configuration by using a PUT call with empty xml `<highAvailability />` or with `<highAvailability><enabled>>false</enabled></highAvailability>`.

---

#### Example 9-115. Delete high availability configuration

Request:

DELETE `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/highavailability/config`

## Force High Availability Failover

Starting in NSX 6.2.3, you can trigger a high availability failover on the active NSX Edge appliance by setting the value of `haAdminState` to `down` in the NSX Edge appliance configuration. The `haAdminState` determines whether or not an NSX Edge appliance is participating in high availability. Both appliances in an NSX Edge high availability configuration normally have an `haAdminState` of `up`. When you set the `haAdminState` of the active appliance to be `down`, it will stop participating in high availability, and will inform the standby appliance of its status. The standby appliance will become active immediately. See [“Working with Appliances”](#) on page 277 for information on modifying NSX Edge appliance configuration.

## Working with Syslog

You can configure one or two remote syslog servers. NSX Edge events and logs related to firewall events that flow from NSX Edge appliances are sent to the syslog servers.

### Configure Syslog

Configures syslog servers.

#### Example 9-116. Configure syslog servers

---

Request:

PUT `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/syslog/config`

Request Body:

```
<syslog>
  <protocol>udp</protocol>  <!-- Optional. Default is "udp". Valid values : tcp|udp -->
  <serverAddresses>  <!-- Maximum 2 remote IPs can be configured. -->
    <ipAddress>1.1.1.1</ipAddress>
    <ipAddress>1.1.1.2</ipAddress>
  </serverAddresses>
</syslog>
```

---

### Query Syslog

Retrieves syslog server information.

#### Example 9-117. Query syslog servers

---

Request:

GET `https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/syslog/config`

Response Body:

```
<syslog>
  <protocol>udp</protocol>  <!-- Optional. Default is "udp". Valid values : tcp|udp -->
  <serverAddresses>  <!-- Maximum 2 remote IPs can be configured. -->
    <ipAddress>1.1.1.1</ipAddress>
    <ipAddress>1.1.1.2</ipAddress>
  </serverAddresses>
</syslog>
```

---

### Delete Syslog

Deletes syslog servers.

#### Example 9-118. Delete syslog servers

---

Request:

---

DELETE <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/syslog/config>

---

## Managing SSL VPN

With SSL VPN-Plus, remote users can connect securely to private networks behind a NSX Edge gateway. Remote users can access servers and applications in the private networks.

### Enable or Disable SSL VPN

Enables or disables SSL VPN on the NSX Edge appliance.

**Example 9-119.** Enable or disable SSL VPN

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config
/?enableService=true|false
```

---

### Query SSL VPN Details

Retrieves SSL VPN details.

**Example 9-120.** Get SSL VPN details

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/
```

---

## Manage Server Settings

### Apply Server Settings

Configures SSL VPN server on port 443 using the certificate named server-cert that is already uploaded on the NSX Edge appliance and the specified cipher.

**Example 9-121.** Apply server settings

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/server/
```

Request Body:

```
<serverSettings>
  <serverAddresses>
    <ipAddress>10.112.243.109</ipAddress>  <!-- Ipv4 or IPV6 address of any of the
      external vnic. ipv4 and ipv6 both can not configured. -->
  </serverAddresses>
  <port>443</port>  <!--optional. Default is 60003 -->
  <certificateId>certificate-1</certificateId>  <!-- Certificate has to be generated
    using certificate REST API and id returned should be mentioned here-->
  <cipherList>  <!-- any one or more of the following ciphers can be part of
    configuration: RC4-MD5|AES128-SHA|AES256-SHA|DES-CBC3-SHA-->
    <cipher>RC4-MD5</cipher>
    <cipher>AES128-SHA</cipher>
    <cipher>AES256-SHA</cipher>
    <cipher>DES-CBC3-SHA</cipher>
  </cipherList>
</serverSettings>
```

---

## Query Server Settings

Gets server settings.

### Example 9-122. Apply server settings

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/server/
```

Response Body:

```
<serverSettings>
  <serverAddresses>
    <ipAddress>10.110.12.249</ipAddress>
  </serverAddresses>
  <port>60003</port>
  <certificateId>certificate-1</certificateId>
  <cipherList>
    <cipher>RC4-MD5</cipher>
  </cipherList>
</serverSettings>
```

---

## Configure Private Networks

### Add Private Network

Configures a private network that the administrator wants to expose to remote users over the SSL VPN tunnel.

### Example 9-123. Add private network

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/privatenetworks/
```

Request Body:

```
<privateNetwork>
  <description>This is a private network for UI-team</description>
  <network>192.168.1.0/24</network>
  <sendOverTunnel> <!--optional. -->
    <ports>20-40</ports> <!-- optional. Default is 0-0 -->
    <optimize>>false</optimize> <!--optional. Default is true -->
  </sendOverTunnel>
  <enabled>>true</enabled> <!--optional. Default is true-->
</privateNetwork>
```

---

### Modify Private Network

Modifies the specified private network in the SSL VPN service on NSX Edge.

### Example 9-124. Add private network

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/privatenetworks/objectId
```

Request Body:

```
<privateNetwork>
  <description>This is a private network for UI-team</description>
  <network>192.168.1.0/24</network>
  <sendOverTunnel>
    <ports>20-40</ports>
```

```

    <optimize>false</optimize>
  </sendOverTunnel>
  <enabled>true</enabled>
</privateNetwork>

```

---

### Query Specific Private Network

Gets the specified private network profile in the SSL VPN instance on NSX Edge.

#### Example 9-125. Query private network

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/privatenetworks/objectId
```

Response Body:

```

<privateNetwork>
  <description>This is a private network for UI-team</description>
  <network>192.168.1.0/24</network>
  <sendOverTunnel>
    <ports>20-40</ports>
    <optimize>false</optimize>
  </sendOverTunnel>
  <enabled>true</enabled>
</privateNetwork>

```

---

### Query all Private Networks

Gets all private network profiles in the SSL VPN instance on NSX Edge.

#### Example 9-126. Query private network

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/privatenetworks/
```

Response Body:

```

<privateNetwork>
  <privateNetwork>
    <objectId>privatenetwork-1</objectId>
    <description>This is a private network for pune-qa-team</description>
    <network>192.168.1.0/24</network>
    <sendOverTunnel>
      <ports>10-20</ports>
      <optimize>true</optimize>
    </sendOverTunnel>
    <enabled>true</enabled>
  </privateNetwork>
</privateNetwork>

```

---

### Delete Private Network

Deletes the specified dynamic IP address configuration from the SSL VPN instance on NSX Edge.

#### Example 9-127. Delete private network

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/privatenetworks/objectId
```

---

### Delete all Private Networks

Deletes all dynamic IP address configurations from the SSL VPN instance on NSX Edge.

**Example 9-128.** Delete private network

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/privatenetworks/
```

---

### Apply All Private Networks

Updates all private network configurations of NSX Edge with the given list of private network configurations. If the configuration is present, it is updated; if it is not present, a new private network configuration is created. Existing configurations not included in the REST call are deleted.

**Example 9-129.** Apply all private networks

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/privatenetworks/
```

---

## Configure Web Resource

### Add Portal Web Resource

Adds a web access server that the remote user can connect to via a web browser.

**Example 9-130.** Add portal web resource

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/webresources/
```

Request Body:

```
<webResource>
  <name>VMware</name>
  <url>http://www.vmware.com</url>
  <method name="POST">
    <data>username=stalin</data>
  </method>
  <description>Click here to visit the corporate intranet Homepage </description>
  <enabled>true</enabled>  <!--optional. Default is true-->
</webResource>
```

---

### Modify Portal Web Resource

Modifies the specified web access server.

**Example 9-131.** Modify portal web resource

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/webresources
/objectId
```

Request Body:

```
<webResource>
  <name>VMware</name>
  <url>http://www.vmware.com</url>
  <method name="POST">
    <data>username=stalin</data>
  </method>
  <description>Click here to visit the corporate intranet Homepage</description>
  <enabled>true</enabled>  <!--optional. Default is true-->
</webResource>
```

---

### Query Portal Web Resource

Gets the specified web access server.

**Example 9-132.** Get specific portal web resource

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/webresources/
/objectId
```

Response Body:

```
<webResource>
  <name>VMware</name>
  <url>http://www.vmware.com</url>
  <method name="POST">
    <data>username=stalin</data>
  </method>
  <description>Click here to visit the corporate intranet Homepage</description>
  <enabled>true</enabled>  <!--optional. Default is true-->
</webResource>
```

---

### Query all Web Resources

Gets all web resources on the SSL VPN instance.

**Example 9-133.** Get portal web resource

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/webresources/
```

Response Body:

```
<webResources>
  <webResource>
    <objectId>webresource-1</objectId>
    <name>VMware</name>
    <url>http://www.vmware.com</url>
    <method name="POST">
      <data>username=stalin </data>
    </method>
    <description>Click here to visit the corporate intranet Homepage </description>
    <enabled>true</enabled>
  </webResource>
</webResources>
```

---

### Delete Portal Web Resource

Deletes the specified web access server.

**Example 9-134.** Delete specific portal web resource

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/webresources/
/objectId
```

**Deletes all Web Resources**

Deletes all web resources on the SSL VPN instance.

**Example 9-135.** Deletes all portal web resources

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/webresources/
```

**Apply All Web Resources**

Updates web resource configurations of NSX Edge with the given list of web resource configurations. If the configuration is present, it is updated; if it is not present, a new web resource configuration is created. Existing configurations not included in the REST call are not deleted.

**Example 9-136.** Apply all private networks

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/privatenetworks/
```

## Configure Users

**Add User**

Adds a new portal user.

**Example 9-137.** Add a user

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/auth/localserver
/users/
```

Request Body:

```
<user>
  <userId>stalin</userId>
  <password>apple@123</password>
  <firstName>STALIN</firstName>
  <lastName>RAJAKILLI</lastName>
  <description>This user belong to vsm team</description>
  <disableUserAccount>>false</disableUserAccount> <!--optional. Default is false-->
  <passwordNeverExpires>true</passwordNeverExpires> <!--optional. Default is false-->
  <allowChangePassword>
    <changePasswordOnNextLogin>>false</changePasswordOnNextLogin> <!--optional.
      Default is false-->
  </allowChangePassword>
</user>
```

## Modify User

Modifies the specified portal user.

### Example 9-138. Modify user

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/auth/localserver/users/
```

Request Body:

```
<user>
  <userId>stalin</userId>
  <password>apple@123</password>
  <firstName>STALIN</firstName>
  <lastName>RAJAKILLI</lastName>
  <description>This user belong to vsm team</description>
  <disableUserAccount>>false</disableUserAccount> <!--optional. Default is false-->
  <passwordNeverExpires>>true</passwordNeverExpires> <!--optional. Default is false-->
  <allowChangePassword>
    <changePasswordOnNextLogin>>false</changePasswordOnNextLogin> <!--optional.
      Default is false-->
  </allowChangePassword>
</user>
```

---

## Query User Details

Gets information about the specified user.

### Example 9-139. Query user

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/auth/localserver/users/userId
```

Response Body:

```
<users>
  <user>
    <userId>stalin</userId>
    <firstName>Bob</firstName>
    <lastName>weber</lastName>
    <disableUserAccount>>false</disableUserAccount> <!--optional. Default is false-->
    <passwordNeverExpires>>true</passwordNeverExpires> <!--optional. Default is
      false-->
    <allowChangePassword>
      <changePasswordOnNextLogin>>false</changePasswordOnNextLogin> <!--optional.
        Default is false-->
    </allowChangePassword>
  </user>
</users>
```

---

## Delete User

Deletes specified user.

### Example 9-140. Delete user

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/auth
/localserver/users/userId
```

---

### Delete all Users

Deletes all users on the specified SSL VPN instance.

#### Example 9-141. Delete all user

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/auth
/localserver/users/
```

---

### Apply all Users

Updates all users of NSX Edge with the given list of users. If the user is present, it is updated; if it is not present, a new user is created. Existing users not included in the REST call are not deleted.

#### Example 9-142. Apply all users

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/auth/localusers/users
```

Request Body:

```
<users>
  <user>
    <userId>stalin</userId>
    <password>apple@123</password>
    <firstName>Bob</firstName>
    <lastName>weber</lastName>
    <description>This user belong to vsm team</description>
    <disableUserAccount>>false</disableUserAccount>
    <passwordNeverExpires>>true</passwordNeverExpires>
    <allowChangePassword>
      <changePasswordOnNextLogin>>false</changePasswordOnNextLogin>
    </allowChangePassword>
  </user>
</users>
```

---

## Configure IP Pool

You can add, edit, or delete an IP pool.

### Add IP Pool

Creates an IP pool that will be used to assign IP address to remote users.

#### Example 9-143. Add IP pool

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/ippools/
```

Request Body:

```
<ipAddressPool>
  <description>description</description>
  <ipRange>10.112.243.11-10.112.243.57</ipRange>
  <netmask>255.0.0.0</netmask>
  <gateway>192.168.1.1</gateway>
```

```

    <primaryDns>192.168.10.1</primaryDns> <!--optional. -->
    <secondaryDns>4.2.2.2</secondaryDns> <!--optional. -->
    <dnsSuffix></dnsSuffix>
    <winsServer>10.112.243.201</winsServer>
    <enabled>>true</enabled> <!--optional. Default is true-->
</ipAddressPool>

```

---

## Modify IP Pool

Modifies the specified IP pool.

### Example 9-144. Modify IP pool

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/ippools/ippoolId
```

Request Body:

```

<ipAddressPool>
  <description>description</description>
  <ipRange>10.112.243.11-10.112.243.57</ipRange>
  <netmask>255.0.0.0</netmask>
  <gateway>192.168.1.1</gateway>
  <primaryDns>192.168.10.1</primaryDns>
  <secondaryDns>4.2.2.2</secondaryDns>
  <dnsSuffix></dnsSuffix>
  <winsServer>10.112.243.201</winsServer>
  <enabled>>true</enabled>
</ipAddressPool>

```

---

## Query IP Pool

Gets details of the IP pool.

### Example 9-145. Get IP pool

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/ippools/ippoolId
```

Response Body:

```

<ipAddressPool>
  <objectId>ipPool-1</objectId>
  <description>description</description>
  <ipRange>10.112.243.11-10.112.243.57</ipRange>
  <netmask>255.0.0.0</netmask>
  <gateway>192.168.1.1</gateway>
  <primaryDns>192.168.10.1</primaryDns> <!--optional. -->
  <secondaryDns>4.2.2.2</secondaryDns> <!--optional. -->
  <dnsSuffix></dnsSuffix>
  <winsServer>10.112.243.201</winsServer>
  <enabled>>true</enabled> <!--optional. Default is true-->
</ipAddressPool>

```

---

## Query all IP Pools

Gets all IP pools configured on the SSL VPN instance.

**Example 9-146.** Gets all IP pools

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/ippools/
```

Response Body:

```
<ipAddressPool>
  <objectId>ipPool-1</objectId>
  <description>description</description>
  <ipRange>10.112.243.11-10.112.243.57</ipRange>
  <netmask>255.0.0.0</netmask>
  <gateway>192.168.1.1</gateway>
  <primaryDns>192.168.10.1</primaryDns  <!--optional. -->
  <secondaryDns>4.2.2.2</secondaryDns>  <!--optional. -->
  <dnsSuffix></dnsSuffix>
  <winsServer>10.112.243.201</winsServer>
  <enabled>true</enabled>  <!--optional. Default is true-->
</ipAddressPool>
```

**Delete IP Pool**

Deletes the specified IP pool.

**Example 9-147.** Delete IP pool

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/ippools/ippoolId
```

**Deletes all IP Pools**

Deletes all IP pools on the SSL VPN instance.

**Example 9-148.** Deletes all IP pools

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/ippools/
```

**Apply all IP Pools**

Updates all IP pools of NSX Edge with the given list of users. If the IP pool is present, it is updated; if it is not present, a new IP pool is created. Existing pools not included in the REST call are not deleted.

**Example 9-149.** Apply IP pools

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/ippools/
```

Request Body:

```
<ipAddressPools>
  <ipAddressPool>
    <description>description</description>
    <ipRange>10.112.243.11-10.112.243.57</ipRange>
    <netmask>255.0.0.0</netmask>
    <gateway>192.168.1.1</gateway>
```

```

    <primaryDns>192.168.10.1</primaryDns>
    <secondaryDns>4.2.2.2</secondaryDns>
    <dnsSuffix></dnsSuffix>
    <winsServer>10.112.243.201</winsServer>
    <enabled>true</enabled>
  </ipAddressPool>
</ipAddressPools>

```

---

## Configure Network Extension Client Parameters

### Apply Client Configuration

Sets advanced parameters for full access client configurations – such as whether client should auto-reconnect in case of network failures or network unavailability, or whether the client should be uninstalled after logout.

#### Example 9-150. Apply IP pools

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/clientconfig/
```

Request Body:

```

<clientConfiguration>
  <autoReconnect>true</autoReconnect>  <!--optional. Default is false-->
  <fullTunnel>  <!--optional. Default Tunnel mode is SPLIT-->
  <excludeLocalSubnets>true</excludeLocalSubnets>  <!--optional. Default is false-->
    <gatewayIp>10.112.243.11</gatewayIp>
  </fullTunnel>
  <upgradeNotification>false</upgradeNotification>  <!--optional. Default is false-->
</clientConfiguration>

```

### Get Client Configuration

Gets information about the specified client.

#### Example 9-151. Get client configuration

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/clientconfig/
```

Response Body:

```

<clientConfiguration>
  <autoReconnect>true</autoReconnect>  <!--optional. Default is false-->
  <tunnelConfiguration>
    <excludeLocalSubnets>true</excludeLocalSubnets>  <!--optional. Default is false-->
    <gatewayIp>10.112.243.11</gatewayIp>
  </tunnelConfiguration>
  <upgradeNotification>false</upgradeNotification>  <!--optional. Default is false-->
</clientConfiguration>

```

---

## Configure Network Extension Client Installation Package

You can add, delete, or edit an installation package for the SSL client.

## Add Client Installation Package

Creates setup executables (installers) for full access network clients. These setup binaries are later downloaded by remote clients and installed on their systems. The primary parameters needed to configure this setup are - hostname of the gateway, and its port and a profile name which is shown to the user to identify this connection. Administrator can also set few other parameters such as whether to automatically start the application on windows login, hide the system tray icon etc.

### Example 9-152. Add installation package

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/installpackages/
```

Request Body:

```
<clientInstallPackage>
  <profileName>client</profileName>
  <gatewayList>
    <gateway>
      <hostName>10.112.243.123</hostName>
      <port>443</port>  <!--optional. Default is 443-->
    </gateway>
  </gatewayList>
  <startClientOnLogon>false</startClientOnLogon>  <!--optional. Default is false-->
  <hideSystrayIcon>true</hideSystrayIcon>  <!--optional. Default is false-->
  <rememberPassword>true</rememberPassword>  <!--optional. Default is false-->
  <silentModeOperation>true</silentModeOperation>  <!--optional. Default is false-->
  <silentModeInstallation>false</silentModeInstallation>  <!--optional. Default is
  false-->
  <hideNetworkAdaptor>false</hideNetworkAdaptor>  <!--optional. Default is false-->
  <createDesktopIcon>true</createDesktopIcon>  <!--optional. Default is true-->
  <enforceServerSecurityCertValidation>false</enforceServerSecurityCertValidation>
  <!--Above tag is optional. Default is true-->
  <createLinuxClient>false</createLinuxClient>  <!--optional. Default is false-->
  <createMacClient>false</createMacClient>  <!--optional. Default is false-->
  <description>windows client</description>
  <enabled>true</enabled>  <!--optional. Default is true-->
</clientInstallPackage>
```

---

## Modify Client Installation Package

Modifies the specified installation package.

### Example 9-153. Modify installation package

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/installpackages/objectId
```

Request Body:

```
<clientInstallPackage>
  <profileName>client</profileName>
  <gatewayList>
    <gateway>
      <hostName>10.112.243.123</hostName>
      <port>443</port>  <!--optional. Default is 443-->
    </gateway>
  </gatewayList>
  <startClientOnLogon>false</startClientOnLogon>  <!--optional. Default is false-->
  <hideSystrayIcon>true</hideSystrayIcon>  <!--optional. Default is false-->
  <rememberPassword>true</rememberPassword>  <!--optional. Default is false-->
  <silentModeOperation>true</silentModeOperation>  <!--optional. Default is false-->
```

```

<silentModeInstallation>>false</silentModeInstallation> <!--optional. Default is
false-->
<hideNetworkAdaptor>>false</hideNetworkAdaptor> <!--optional. Default is false-->
<createDesktopIcon>>true</createDesktopIcon> <!--optional. Default is true-->
<enforceServerSecurityCertValidation>>false</enforceServerSecurityCertValidation>
    <!-- Above tag is optional. Default is true-->
<createLinuxClient>>false</createLinuxClient> <!--optional. Default is false-->
<createMacClient>>false</createMacClient> <!--optional. Default is false-->
<description>>windows client</description>
<enabled>>true</enabled> <!--optional. Default is true-->
</clientInstallPackage>

```

---

## Query Client Installation Package

Gets information about the specified installation package.

### Example 9-154. Query installation package

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/installpackages/objectId
```

Response Body:

```

<clientInstallPackage>
  <objectId>clientinstallpackage-1</objectId>
  <profileName>client</profileName> <gatewayList>
  <gatewayList>
    <gateway>
      <hostName>10.112.243.123</hostName>
      <port>443</port> <!--optional. Default is 443-->
    </gateway>
  </gatewayList>
  <startClientOnLogon>>false</startClientOnLogon>
  <hideSystrayIcon>>true</hideSystrayIcon>
  <rememberPassword>>true</rememberPassword>
  <silentModeOperation>>true</silentModeOperation>
  <silentModeInstallation>>false</silentModeInstallation>
  <hideNetworkAdaptor>>false</hideNetworkAdaptor>
  <createDesktopIcon>>true</createDesktopIcon>
  <enforceServerSecurityCertValidation>>false</enforceServerSecurityCertValidation>
  <createLinuxClient>>false</createLinuxClient>
  <createMacClient>>false</createMacClient>
  <description>>windows client</description>
  <enabled>>true</enabled>
</clientInstallPackage>

```

---

## Query all Client Installation Packages

Gets information about all installation packages.

### Example 9-155. Query all installation package

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/installpackages/
```

Response Body:

```

<clientInstallPackages>
  <clientInstallPackage>
    <objectId>clientinstallpackage-1</objectId>
    <profileName>client</profileName> <gatewayList>
    <gatewayList>

```

```

    <gateway>
      <hostName>10.112.243.123</hostName>
      <port>443</port>
    </gateway>
  </gatewayList>
</startClientOnLogon>false</startClientOnLogon>
<hideSystrayIcon>true</hideSystrayIcon>
<rememberPassword>true</rememberPassword>
<silentModeOperation>true</silentModeOperation>
<silentModeInstallation>false</silentModeInstallation>
<hideNetworkAdaptor>false</hideNetworkAdaptor>
<createDesktopIcon>true</createDesktopIcon>
<enforceServerSecurityCertValidation>false</enforceServerSecurityCertValidation>
<createLinuxClient>false</createLinuxClient>
<createMacClient>false</createMacClient>
<description>windows client</description>
<enabled>true</enabled>
</clientInstallPackage>
<clientInstallPackage>

```

---

### Delete Client Installation Package

Deletes the specified installation package.

**Example 9-156.** Delete installation package

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/installpackages/objectId
```

---

### Delete all Client Installation Packages

Deletes all installation packages.

**Example 9-157.** Delete all installation packages

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/installpackages/
```

---

### Apply all Installation Packages

Updates all installation packages on NSX Edge with the given list of installation packages. If the installation package is present, it is updated; if it is not present, a new installation package is created. Existing installation packages not included in the REST call are deleted.

**Example 9-158.** Apply installation packages

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/client
/networkextension/installpackages/
```

Request Body:

```

<clientInstallPackages>
  <clientInstallPackage>
    <objectId>clientinstallpackage-1</objectId>
    <profileName>client</profileName>
    <gatewayList>
      <gateway>

```

```

        <hostName>10.112.243.123</hostName>
        <port>443</port>
    </gateway>
</gatewayList>
<startClientOnLogon>>false</startClientOnLogon>
<hideSystrayIcon>>true</hideSystrayIcon>
<rememberPassword>>true</rememberPassword>
<silentModeOperation>>true</silentModeOperation>
<silentModeInstallation>>false</silentModeInstallation>
<hideNetworkAdaptor>>false</hideNetworkAdaptor>
<createDesktopIcon>>true</createDesktopIcon>
<enforceServerSecurityCertValidation>>false</enforceServerSecurityCertValidation>
<createLinuxClient>>false</createLinuxClient>
<createMacClient>>false</createMacClient>
<description>windows client</description>
<enabled>>true</enabled>
</clientInstallPackage>
</clientInstallPackages>

```

---

## Configure Portal Layouts

You can configure the web layout bound to the SSL VPN client.

### Upload Portal Logo

Uploads the portal logo from the given local path.

#### Example 9-159. Upload portal logo

---

Request:

```

/usr/bin/curl -v -k -i -F layoutFile=@/tmp/portalLogo.jpg -H 'Authorization: Basic
YWRtaW46ZGVmYXVsdA=='
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/layout/images
/portallogo/

```

---

### Upload Phat Banner

Uploads the phat client banner from the given local path. The phat banner image must in the bmp format.

#### Example 9-160. Upload phat banner

---

Request:

```

/usr/bin/curl -v -k -i -F "banner=@/tmp/phatBanner.bmp" -H 'Authorization: Basic
YWRtaW46ZGVmYXVsdA=='
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/layout/images
/phatbanner

```

---

### Upload Client Connected Icon

Uploads the client connected icon from the given local path. The icon image must be of type ico.

#### Example 9-161. Upload client connected icon

---

Request:

```

POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/layout/images
/connecticon/

```

---

## Upload Client Disconnected Icon

Uploads the client disconnected icon from the given local path. The icon image must be of type icon.

### Example 9-162. Upload client disconnected icon

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/layout/images/disconnecticon/
```

---

## Upload Client Desktop Icon

Uploads the client desktop icon from the given local path. The icon image must be of type ico.

### Example 9-163. Upload client desktop icon

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/layout/images/desktopicon/
```

---

## Upload Error Connected Icon

Uploads the client error connected icon from the given local path. The icon image must be of type ico.

### Example 9-164. Upload client desktop icon

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/layout/images/erroricon/
```

---

## Apply Layout Configuration

Sets the portal layout.

### Example 9-165. Apply layout configuration

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/layout/portal
```

Request Body:

```
<layout>  <!-- portal layout configuration-->
  <portalTitle>Pepsi Remote Access</portalTitle>  <!--optional. Default value is
    VMware -->
  <companyName>pepsi, Inc.</companyName>  <!--optional. Default value is VMware -->
  <logoBackgroundColor>FFFFFF</logoBackgroundColor>  <!-- Portal Color Configuration;
    Default value is FFFFFFFF-->
  <titleColor>996600</titleColor>  <!--optional. Default value is 996600 -->
  <topFrameColor>000000</topFrameColor>  <!--optional. Default value is 000000 -->
  <menuBarColor>999999</menuBarColor>  <!--optional. Default value is 999999 -->
  <rowAlternativeColor>FFFFFF</rowAlternativeColor>  <!--optional. Default value is
    FFFFFFFF -->
  <bodyColor>FFFFFF</bodyColor>  <!--optional. Default value is FFFFFFFF -->
  <rowColor>F5F5F5</rowColor>  <!--optional. Default value is F5F5F5 -->
</layout>
```

---

## Query Portal Layout

Gets the portal layout configuration.

### Example 9-166. Query layout configuration

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/layout/portal
```

Response Body:

```
<layout> <!-- portal layout configuration-->
  <portalTitle>Pepsi Remote Access</portalTitle> <!--optional. Default value is VMware
  -->
  <companyName>pepsi, Inc.</companyName> <!--optional. Default value is VMware -->
  <logoBackgroundColor>FFFFFF</logoBackgroundColor> <!-- Portal Color Configuration;
  Default value is FFFFFFF-->
  <titleColor>996600</titleColor> <!--optional. Default value is 996600 -->
  <topFrameColor>000000</topFrameColor> <!--optional. Default value is 000000 -->
  <menuBarColor>999999</menuBarColor> <!--optional. Default value is 999999 -->
  <rowAlternativeColor>FFFFFF</rowAlternativeColor> <!--optional. Default value is
  FFFFFFF -->
  <bodyColor>FFFFFF</bodyColor> <!--optional. Default value is FFFFFFF -->
  <rowColor>F5F5F5</rowColor> <!--optional. Default value is F5F5F5 -->
</layout>
```

---

## Configure Authentication Parameters

You can add an external authentication server (AD, LDAP, Radius, or RSA) which is bound to the SSL gateway. All users in the bounded authenticated server will be authenticated.

### Upload RSA Config File

Uploads the RSA configuration file to NSX Manager.

See the “Generate the Authentication Manager Configuration File” section of the RSA Authentication Manager Administrator’s Guide for instructions on how to configure and download the RSA configuration file from RSA Authentication Manager.

### Example 9-167. Upload RSA config file

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/auth/settings/
  /rsaconfigfile/
```

---

### Apply Authentication Configuration

Sets authentication process for remote users. The administrator specifies whether username password based authentication should be enabled and the list and details of authentication servers such as active directory, ldap, radius etc. The administrator can also enable client certificate based authentication.

### Example 9-168. Apply Authentication Configuration

---

Request:edgeId

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/auth/settings/
```

Request Body:

```
<authenticationConfig>
  <passwordAuthentication>
    <authenticationTimeout>1</authenticationTimeout> <!--optional. Default value is
    1 mins-->
```

```

<primaryAuthServers>  <!-- Only four auth servers can be part of authentication
                        configuration including secondary auth server and can be of type
                        AD,LDAP,RADIUS,LOCAL and RSA -->
<com.vmware.vshield.edge.sslvpn.dto.LdapAuthServerDto>
  <ip>1.1.1.1</ip>
  <port>90</port>    <!--optional. Default value is 639 if ssl enabled or 389
                      for normal cfg-->
  <timeOut>20</timeOut>  <!--optional. Default value is 10 secs-->
  <enableSsl>>false</enableSsl>  <!--optional. Default is false-->
  <searchBase>searchbasevalue</searchBase>
  <bindDomainName>binddnvalue</bindDomainName>
  <bindPassword>password</bindPassword>  <!--optional.-->
  <loginAttributeName>cain</loginAttributeName>  <!--optional. Default is
            SAMAccountName -->
  <searchFilter>found</searchFilter>  <!--optional. Default is
            'objectClass=*'-->
  <enabled>>true</enabled>  <!--optional. Default is ture-->
</com.vmware.vshield.edge.sslvpn.dto.LdapAuthServerDto>
<com.vmware.vshield.edge.sslvpn.dto.RadiusAuthServerDto>
  <ip>3.3.3.3</ip>
  <port>90</port>    <!--optional. Default value is 1812-->
  <timeOut>20</timeOut>  <!--optional. Default value is 10 secs-->
  <secret>struct9870</secret>
  <nasIp>1.1.1.9</nasIp>  <!--optional. Default value is 0.0.0.0-->
  <retryCount>10</retryCount>  <!--optional. Default value is 3-->
</com.vmware.vshield.edge.sslvpn.dto.RadiusAuthServerDto>
<com.vmware.vshield.edge.sslvpn.dto.LocalAuthServerDto>  <!--Only one Local
                  auth server can be part of authentication configuration -->
  <enabled>true</enabled>
  <passwordPolicy>  <!-- optional. -->
    <minLength>1</minLength>  <!--optional. Default value is 1-->
    <maxLength>1</maxLength>  <!--optional. Default value is 63-->
    <minAlphabets>0</minAlphabets>  <!--optional -->
    <minDigits>0</minDigits>  <!--optional -->
    <minSpecialChar>1</minSpecialChar>  <!--optional -->
    <allowUserIdwithinPassword>>false</allowUserIdwithinPassword>
      <!-- Above tag is optional. Default value is false -->
    <passwordLifeTime>20</passwordLifeTime>  <!--optional. Default value
      is 30 days-->
    <expiryNotification>1</expiryNotification>  <!--optional. Default
      value is 25 days-->
  </passwordPolicy>
  <accountLockoutPolicy>  <!--optional -->
    <retryCount>3</retryCount>  <!--optional. Default value is 3-->
    <retryDuration>3</retryDuration>  <!--optional. Default value is 2
      days -->
    <lockoutDuration>3</lockoutDuration>  <!--optional. Default value is
      2 days -->
  </accountLockoutPolicy>
</com.vmware.vshield.edge.sslvpn.dto.LocalAuthServerDto>
<com.vmware.vshield.edge.sslvpn.dto.RsaAuthServerDto>  <!-- Only one RSA auth
                  server can be configured. RSA configuration file has to be
                  uploaded prior to config RSA auth server RSA timeOut is
                  optional. Default value is 60 secs-->
  <timeOut>20</timeOut>
  <sourceIp>1.2.2.3</sourceIp>
</com.vmware.vshield.edge.sslvpn.dto.RsaAuthServerDto>
</primaryAuthServers>
<secondaryAuthServer>  <!--Any of one of the auth server AD, LDAP, RSA, LOCAL or
                        RADIUS can be sec auth server -->
<com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
  <ip>1.1.1.1</ip>
  <port>90</port>    <!--optional. Default value is 639 if ssl enabled or 389
                      for normal cfg-->
  <timeOut>20</timeOut>  <!--optional. Default value is 10 secs-->
  <enableSsl>>false</enableSsl>  <!--optional. Default is false-->
  <searchBase>searchbasevalue</searchBase>
  <bindDomainName>binddnvalue</bindDomainName>

```

```

    <bindPassword>password</bindPassword> <!--optional. -->
    <loginAttributeName>cain</loginAttributeName> <!--optional. Default is
        SAMAccountName -->
    <searchFilter>found</searchFilter> <!--optional. Default is
        'objectClass=*'-->
    <terminateSessionOnAuthFails>>false</terminateSessionOnAuthFails>
        <!--Above tag is optional. Default is false-->
    <enabled>>true</enabled>
  </com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
</secondaryAuthServer>
</passwordAuthentication>
</authenticationConfig>

```

---

## Query Authentication Configuration

Gets information about the specified authentication server.

### Example 9-169. Query Authentication Configuration

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/auth/settings/>

Response Body:

```

<com.vmware.vshield.edge.sslvpn.dto.AuthenticationConfigurationDto>
  <passwordAuthentication>
    <authenticationTimeout>1</authenticationTimeout>
    <primaryAuthServers>
      <com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
        <ip>1.1.1.1</ip>
        <port>90</port>
        <timeOut>20</timeOut>
        <enableSsl>>false</enableSsl>
        <searchBase>searchbasevalue</searchBase>
        <bindDomainName>binddnvalue</bindDomainName>
        <bindPassword>password</bindPassword>
        <loginAttributeName>cain</loginAttributeName>
        <searchFilter>found</searchFilter>
        <enabled>>true</enabled>
      </com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
    </primaryAuthServers>
    <secondaryAuthServer>
      <com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
        <ip>1.1.1.1</ip>
        <port>90</port>
        <timeOut>20</timeOut>
        <enableSsl>>false</enableSsl>
        <searchBase>searchbasevalue</searchBase>
        <bindDomainName>binddnvalue</bindDomainName>
        <bindPassword>password</bindPassword>
        <loginAttributeName>cain</loginAttributeName>
        <searchFilter>found</searchFilter>
        <terminateSessionOnAuthFails>>false</terminateSessionOnAuthFails>
        <enabled>>true</enabled>
      </com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
    </secondaryAuthServer>
  </passwordAuthentication>
</authenticationConfig>

```

---

## Configure SSL VPN Advanced Configuration

### Apply advanced configuration

Applies advanced configuration.

**Example 9-170.** Apply advanced configuration

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/advancedconfig/>

Request Body:

```
<advancedConfig>
  <enableCompression>>false</enableCompression>  <!--optional. Default is false-->
  <forceVirtualKeyboard>>false</forceVirtualKeyboard>  <!--optional. Default is
    false-->
  <preventMultipleLogon>>true</preventMultipleLogon>  <!--optional. Default is false-->
  <randomizeVirtualKeys>>false</randomizeVirtualKeys>  <!--optional. Default is false-->
  <timeout>  <!--optional. -->
    <forcedTimeout>16</forcedTimeout>  <!--optional. Value is in minute(s)-->
    <sessionIdTimeout>10</sessionIdTimeout>  <!--optional. Default is 10 mins-->
  </timeout>
  <clientNotification></clientNotification>
  <enablePublicUrlAccess>>false</enablePublicUrlAccess>  <!--optional. Default is
    false-->
  <enableLogging>>false</enableLogging>  <!--optional. Default is false-->
</advancedConfig>
```

**Query Advanced Configuration**

Retrieves SSL VPN advanced configuration.

**Example 9-171.** Query advanced configuration

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/advancedconfig/>

Response Body:

```
<advancedConfig>
  <enableCompression>>false</enableCompression>  <!--optional. Default is false-->
  <forceVirtualKeyboard>>false</forceVirtualKeyboard>  <!--optional. Default is false-->
  <preventMultipleLogon>>true</preventMultipleLogon>  <!--optional. Default is false-->
  <randomizeVirtualKeys>>false</randomizeVirtualKeys>  <!--optional. Default is
    false-->
  <timeout>  <!--optional. -->
    <forcedTimeout>16</forcedTimeout>  <!--optional. Value is in minute(s)-->
    <sessionIdTimeout>10</sessionIdTimeout>  <!--optional. Default is 10 mins-->
  </timeout>
  <clientNotification></clientNotification>
  <enablePublicUrlAccess>>false</enablePublicUrlAccess>  <!--optional. Default is
    false-->
  <enableLogging>>false</enableLogging>  <!--optional. Default is false-->
</advancedConfig>
```

**Working with Active Clients**

You can retrieve a list of active clients for the SSL VPN session and disconnect a specific client.

**Query Active Clients**

Retrieves a list of active clients for the SSL VPN session.

**Example 9-172.** Query active clients

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/activesessions/>

Response Body:

```
<activeSessions>
  <activeSession>
    <sessionId>488382</sessionId>
    <sessionType>PHAT</sessionType>
    <userName>demo</userName>
    <startTime>2011-09-24-06:00</startTime>
    <upTime>101400</upTime>
    <idleTime>2</idleTime>
    <totalNonTcpBytesReceived>6576</totalNonTcpBytesReceived>
    <totalTcpBytesReceived>30816</totalTcpBytesReceived>
    <totalNonTcpBytesSent>0</totalNonTcpBytesSent>
    <totalTcpBytesSent>152722</totalTcpBytesSent>
    <clientInternalIp>1.0.192.10</clientInternalIp>
    <clientVirtualIP>192.168.27.20</clientVirtualIP>
    <clientExternalNatIp>10.112.243.227</clientExternalNatIp>
    <clientExternalNatPort>50498</clientExternalNatPort>
    <totalConnections>2</totalConnections>
    <totalActiveConnection>4</totalActiveConnection>
  </activeSession>
</activeSessions>
```

---

### Disconnect Active Client

Disconnects an active client.

**Example 9-173.** Disconnect active client

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/activesessions
/sessionId
```

---

## Manage Logon and Logoff scripts

You can bind a login or logoff script to the NSX Edge gateway.

### Upload Script

You can add multiple login or logoff scripts. For example, you can bind a login script for starting Internet Explorer with gmail.com. When the remote user logs in to the SSL client, Internet Explorer opens up gmail.com.

The upload script returns a script file ID which is used to configure the file parameters.

**Example 9-174.** Upload script

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/script/file/
```

---

### Configure Script Parameters

Configures parameters associated with the uploaded script file.

**Example 9-175.** Add script parameters

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/script/
```

Request Body:

```

<logonLogoffScript>
  <scriptId>logonlogoffscriptfile-12</scriptId>  <!-- Script file id generated
            using upload script file REST API-->
  <type>BOTH</type>
  <description>Testing modify script</description>
  <enabled>>false</enabled>  <!--optional. Default is true -->
</logonLogoffScript>

```

---

### Modify Script Configuration

Modifies the parameters associated with the specified *script file ID*.

#### Example 9-176. Modify script parameters

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/script/scriptFileId
```

Request Body:

```

<logonLogoffScript>
  <scriptId>logonlogoffscriptfile-12</scriptId>
  <type>BOTH</type>
  <description>Testing modify sscript</description>
  <enabled>>false</enabled>
</logonLogoffScript>

```

---

### Query Script Configuration

Retrieves parameters associated with the specified *script file ID*.

#### Example 9-177. Get script parameters

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/script/scriptFileId
```

Response Body:

```

<logonLogoffScript>
  <objectId>logonlogoffscript-1</objectId>
  <scriptId>logonlogoffscriptfile-12</scriptId>
  <type>BOTH</type>
  <description>Testing modify script</description>
  <scriptIdUri>https://nsxmgr-ip/api/4.0/edges/edge-id/sslvpn/config/script/file
                /scriptFileId/</scriptIdUri>
  <enabled>>false</enabled>
</logonLogoffScript>

```

---

### Query All Script Configurations

Retrieves all script configurations for the specified NSX Edge.

#### Example 9-178. Get all script parameters

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/script/
```

Response Body:

```

<logonLogoffScript>
  <logonLogoffScript>
    <scriptId>logonlogoffscriptfile-12</scriptId>
    <type>BOTH</type>

```

```

    <description>Testing modify sscript</description>
    <enabled>>false</enabled>
  </logonLogoffScript>
</logonLogoffScript>

```

---

### Delete Script Configuration

Deletes the parameters associated with the specified script file ID.

**Example 9-179.** Delete script parameters

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/script/
      /scriptId
```

---

### Delete All Script Configuragtions

Deletes all script configurations for the specified NSX Edge.

**Example 9-180.** Delete script parameters

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/script/
```

---

### Apply All Script Configurations

Updates all script configurations on the specified NSX Edge with the given list of configurations. If the configuration is present, it is updated; if it is not present, a new configuration is created. Existing configurations not included in the REST call are deleted.

**Example 9-181.** Apply script configurations

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/script/
```

Request Body:

```

<logonLogoffScript>
  <logonLogoffScript>
    <objectId>logonlogoffscript-1</objectId>
    <scriptId>logonlogoffscriptfile-12</scriptId>
    <type>BOTH</type>
    <enabled>>false</enabled>
    <description>This script will run on both login and logoff of phat
      client</description>
  </logonLogoffScript>
</logonLogoffScript>

```

### Reconfigure SSL VPN

Pushes the entire SSL VPN configuration to the specified NSX Edge in a single call.

**Example 9-182.** Reconfigure SSL VPN

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/
```

Request Body:

```

<sslvpnConfig>
  <enabled>true</enabled>
  <logging> <!-- optional . -->
    <enable>>false</enable>
    <logLevel>debug</logLevel>
  </logging>
  <serverSettings>
    <ip>10.112.243.109</ip>
    <port>443</port> <!--optional. Default is 443 -->
      <!-- Certificate has to be generated using certificate REST API
      and id returned should be mentioned here-->
    <certificateId>certificate-1</certificateId> <!-- optional; Certificate has to
    be generated using certificate REST API and id returned should be
    mentioned here-->
    <cipherList> <!-- any one or more of the following ciphers can be part of
    configuration -->
      <cipher>RC4-MD5</cipher>
      <cipher>AES128-SHA</cipher>
      <cipher>AES256-SHA</cipher>
      <cipher>DES-CBC3-SHA</cipher>
    </cipherList>
  </serverSettings>
  <privateNetworks>
    <privateNetwork>
      <description>This is a private network for UI-team</description>
      <network>192.168.1.0/24</network>
      <sendOverTunnel>
        <ports>20-40</ports> <!-- optional. Default is 0-0 -->
        <optimize>>false</optimize> <!--optional. Default is true -->
      </sendOverTunnel>
      <enabled>true</enabled> <!--optional. Default is true-->
    </privateNetwork>
  </privateNetworks>
  <users>
    <user>
      <userId>stalin</userId>
      <password>apple@123</password>
      <firstName>STALIN</firstName>
      <lastName>RAJAKILLI</lastName>
      <description>This user belong to vsm team</description>
      <disableUserAccount>>false</disableUserAccount> <!--optional. Default is
      false-->
      <passwordNeverExpires>true</passwordNeverExpires> <!--optional. Default is
      false-->
      <allowChangePassword>
      <changePasswordOnNextLogin>>false</changePasswordOnNextLogin> <!--optional.
      Default is false-->
    </allowChangePassword>
    </user>
  </users>
  <ipAddressPools>
    <ipAddressPool>
      <description>description</description>
      <ipRange>10.112.243.11-10.112.243.57</ipRange>
      <netmask>255.0.0.0</netmask>
      <gateway>192.168.1.1</gateway>
      <primaryDns>192.168.10.1</primaryDns>
      <secondaryDns>4.2.2.2</secondaryDns>
      <dnsSuffix></dnsSuffix>
      <winsServer>10.112.243.201</winsServer>
      <enabled>true</enabled> <!--optional. Default is true-->
    </ipAddressPool>
  </ipAddressPools>
  <clientInstallPackages>
    <clientInstallPackage>
      <profileName>client</profileName>
      <gatewayList>
        <gateway>

```

```

        <hostName>10.112.243.123</hostName>
        <port>443</port> <!--optional. Default is 443-->
    </gateway>
</gatewayList>
<startClientOnLogon>>false</startClientOnLogon> <!--optional. Default is
    false-->
<hideSystrayIcon>>true</hideSystrayIcon> <!--optional. Default is false-->
<rememberPassword>>true</rememberPassword> <!--optional. Default is false-->
<silentModeOperation>>true</silentModeOperation> <!--optional. Default is
    false-->
<silentModeInstallation>>false</silentModeInstallation> <!--optional. Default
    is false-->
<hideNetworkAdaptor>>false</hideNetworkAdaptor> <!--optional. Default is
    false-->
<createDesktopIcon>>true</createDesktopIcon> <!--optional. Default is true-->
<enforceServerSecurityCertValidation>>false</enforceServerSecurity
    CertValidation> <!--optional. Default is true-->
<createLinuxClient>>false</createLinuxClient> <!--optional. Default is
    false-->
<createMacClient>>false</createMacClient> <!--optional. Default is false-->
<description>windows client</description>
<enabled>>true</enabled> <!--optional. Default is true-->
</clientInstallPackage>
</clientInstallPackages>
<webResources>
    <webResource>
        <name>VMware</name>
        <url>http://www.vmware.com</url>
        <method name="POST">
            <data>username=stalin </data>
        </method>
        <description>Click here to visit the corporate intranet Homepage</description>
        <enabled>>true</enabled> <!--optional. Default is true-->
    </webResource>
</webResources>
<clientConfiguration>
    <autoReconnect>>true</autoReconnect> <!--optional. Default is false-->
    <fullTunnel> <!--optional. Default Tunnel mode is SPLIT-->
        <excludeLocalSubnets>>true</excludeLocalSubnets> <!--optional. Default is
            false-->
        <gatewayIp>10.112.243.11</gatewayIp>
    </fullTunnel>
    <upgradeNotification>>false</upgradeNotification> <!--optional. Default is
        false-->
</clientConfiguration>
<advancedConfig>
    <enableCompression>>false</enableCompression> <!--optional. Default is false-->
    <forceVirtualKeyboard>>false</forceVirtualKeyboard> <!--optional. Default is
        false-->
    <preventMultipleLogon>>true</preventMultipleLogon> <!--optional. Default is
        false-->
    <randomizeVirtualKeys>>false</randomizeVirtualKeys> <!--optional. Default is
        false-->
    <timeout><!--optional. -->
        <forcedTimeout>16</forcedTimeout> <!--optional. -->
        <sessionIdleTimeout>10</sessionIdleTimeout> <!--optional. Default value is
            10 mins-->
    </timeout>
    <clientNotification></clientNotification>
    <enablePublicUrlAccess>>false</enablePublicUrlAccess> <!--optional. Default is
        false-->
    <enableLogging>>false</enableLogging> <!--optional. Default is false-->
</advancedConfig>
<authenticationConfiguration>
    <passwordAuthentication>
        <authenticationTimeout>1</authenticationTimeout> <!--optional. Default value
            is 1 mins-->

```

```

<primaryAuthServers>  <!-- Only four auth servers can be part of
                        authentication configuration including secondary auth server and
                        can be of type AD,LDAP,RADIUS,LOCAL and RSA -->
  <com.vmware.vshield.edge.sslvpn.dto.LdapAuthServerDto>
    <ip>1.1.1.1</ip>
    <port>90</port>  <!--optional. Default value is 639 if ssl enabled or
                     389 for normal cfg-->
    <timeOut>20</timeOut>  <!--optional. Default value is 10 secs-->
    <enableSsl>>false</enableSsl>  <!--optional. Default is false-->
    <searchBase>searchbasevalue</searchBase>
    <bindDomainName>binddnvalue</bindDomainName>
    <bindPassword>password</bindPassword>  <!--optional.-->
    <loginAttributeName>cain</loginAttributeName>  <!--optional. Default
            is SAMAccountName -->
    <searchFilter>found</searchFilter>  <!--optional. Default is
            'objectClass=*'-->
    <enabled>>true</enabled>  <!--optional. Default is ture-->
  </com.vmware.vshield.edge.sslvpn.dto.LdapAuthServerDto>
  <com.vmware.vshield.edge.sslvpn.dto.RadiusAuthServerDto>
    <ip>3.3.3.3</ip>
    <port>90</port>  <!--optional. Default value is 1812-->
    <timeOut>20</timeOut>  <!--optional. Default value is 10 secs-->
    <secret>struct9870</secret>
    <nasIp>1.1.1.9</nasIp>  <!--optional. Default value is 0.0.0.0-->
    <retryCount>10</retryCount>  <!--optional. Default value is 3-->
  </com.vmware.vshield.edge.sslvpn.dto.RadiusAuthServerDto>
  <com.vmware.vshield.edge.sslvpn.dto.LocalAuthServerDto>  <!--Only one Local
                    auth server can be part of authentication configuration -->
    <enabled>>true</enabled>
    <passwordPolicy>  <!-- optional. -->
      <minLength>1</minLength>  <!--optional. Default value is 1-->
      <maxLength>63</maxLength>  <!--optional. Default value is 63-->
      <minAlphabets>0</minAlphabets>  <!--optional -->
      <minDigits>0</minDigits>  <!--optional -->
      <minSpecialChar>1</minSpecialChar>  <!--optional -->
      <allowUserIdwithinPassword>>false</allowUserIdwithinPassword>  <!--
                            optional. Default value is false -->
      <passwordLifeTime>20</passwordLifeTime>  <!--optional. Default value
                            is 30 days-->
      <expiryNotification>1</expiryNotification>  <!--optional. Default
                            value is 25 days-->
    </passwordPolicy>
    <accountLockoutPolicy>  <!--optional -->
      <retryCount>3</retryCount>  <!--optional. Default value is 3-->
      <retryDuration>3</retryDuration>  <!--optional. Default value is 2
                            days -->
      <lockoutDuration>3</lockoutDuration>  <!--optional. Default value is
                            2 days -->
    </accountLockoutPolicy>
  </com.vmware.vshield.edge.sslvpn.dto.LocalAuthServerDto>
  <!--com.vmware.vshield.edge.sslvpn.dto.RsaAuthServerDto>  <!-- Only one RSA
                    auth server can be configured.RSA configuration file has to be
                    uploaded prior to config RSA auth server RSA timeOut is
                    optional. Default value is 60 secs -->
    <timeOut>20</timeOut>
    <sourceIp>1.2.2.3</sourceIp>
  </com.vmware.vshield.edge.sslvpn.dto.RsaAuthServerDto>
</primaryAuthServers>
<secondaryAuthServer>  <!--Any of one of the auth server AD, LDAP, RSA, LOCAL
                        or RADIUS can be sec auth server -->
  <com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
    <ip>1.1.1.1</ip>
    <port>90</port>  <!--optional. Default value is 639 if ssl enabled or
                     389 for normal cfg-->
    <timeOut>20</timeOut>  <!--optional. Default value is 10 secs-->
    <enableSsl>>false</enableSsl>  <!--optional. Default is false-->
    <searchBase>searchbasevalue</searchBase>
    <bindDomainName>binddnvalue</bindDomainName>

```

```

    <bindPassword>password</bindPassword> <!--optional. -->
    <loginAttributeName>cain</loginAttributeName> <!--optional. Default
        is SAMAccountName -->
    <searchFilter>found</searchFilter> <!--optional. Default is
        'objectClass=*'-->
    <terminateSessionOnAuthFails>>false</terminateSessionOnAuthFails>
        <!--Above tag is optional. Default is false-->
    <enabled>>true</enabled>
    </com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
</secondaryAuthServer>
</passwordAuthentication>
</authenticationConfiguration>
</sslvpnConfig>

```

---

## Query SSL VPN Configuration

Retrieves the SSL VPN configurations of the specified NSX Edge.

### Example 9-183. Query SSL VPN Configuration

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/
```

Response Body:

```

<sslvpnConfig>
  <version>32</version>
  <enabled>true</enabled>
  <logging> <!-- optional. -->
    <enable>>false</enable>
    <logLevel>debug</logLevel>
  </logging>
  <serverSettings>
    <ip>10.112.243.109</ip>
    <port>443</port>
    <certificateId>certificate-1</certificateId>
    <cipherList>
      <cipher>RC4-MD5</cipher>
      <cipher>AES128-SHA</cipher>
      <cipher>AES256-SHA</cipher>
      <cipher>DES-CBC3-SHA</cipher>
    </cipherList>
  </serverSettings>
  <privateNetworks>
    <privateNetwork>
      <description>This is a private network for UI-team</description>
      <network>192.168.1.0/24</network>
      <sendOverTunnel>
        <ports>20-40</ports>
        <optimize>>false</optimize>
      </sendOverTunnel>
      <enabled>true</enabled>
    </privateNetwork>
  </privateNetworks>
  <users>
    <user>
      <userId>stalin</userId>
      <password>apple@123</password>
      <firstName>STALIN</firstName>
      <lastName>RAJAKILLI</lastName>
      <description>This user belong to vsm team</description>
      <disableUserAccount>>false</disableUserAccount>
      <passwordNeverExpires>true</passwordNeverExpires>
      <allowChangePassword>
        <changePasswordOnNextLogin>>false</changePasswordOnNextLogin>
      </allowChangePassword>
    </user>
  </users>
</sslvpnConfig>

```

```

    </user>
  </users>
  <ipAddressPools>
    <ipAddressPool>
      <description>description</description>
      <ipRange>10.112.243.11-10.112.243.57</ipRange>
      <netmask>255.0.0.0</netmask>
      <gateway>192.168.1.1</gateway>
      <primaryDns>192.168.10.1</primaryDns>
      <secondaryDns>4.2.2.2</secondaryDns>
      <dnsSuffix></dnsSuffix>
      <winsServer>10.112.243.201</winsServer>
      <enabled>true</enabled>
    </ipAddressPool>
  </ipAddressPools>
  <clientInstallPackages>
    <clientInstallPackage>
      <profileName>client</profileName>
      <gatewayList>
        <gateway>
          <hostName>10.112.243.123</hostName>
          <port>443</port>
        </gateway>
      </gatewayList>
      <!-- Optional Parameters-->
      <startClientOnLogon>false</startClientOnLogon>
      <hideSystrayIcon>true</hideSystrayIcon>
      <rememberPassword>true</rememberPassword>
      <silentModeOperation>true</silentModeOperation>
      <silentModeInstallation>false</silentModeInstallation>
      <hideNetworkAdaptor>false</hideNetworkAdaptor>
      <createDesktopIcon>true</createDesktopIcon>
      <enforceServerSecurityCertValidation>false</enforceServerSecurity
        CertValidation>
      <createLinuxClient>false</createLinuxClient>
      <createMacClient>false</createMacClient>
      <description>windows client</description>
      <enabled>true</enabled>
    </clientInstallPackage>
  </clientInstallPackages>
  <webResources>
    <webResource>
      <name>VMware</name>
      <url>http://www.vmware.com</url>
      <method name="POST">
        <data>username=stalin </data>
      </method>
      <description>Click here to visit the corporate intranet Homepage</description>
      <enabled>true</enabled>
    </webResource>
  </webResources>
  <clientConfiguration>
    <autoReconnect>true</autoReconnect>
    <fullTunnel>
      <excludeLocalSubnets>true</excludeLocalSubnets>
      <gatewayIp>10.112.243.11</gatewayIp>
    </fullTunnel>
    <upgradeNotification>false</upgradeNotification>
  </clientConfiguration>
  <advancedConfig>
    <enableCompression>false</enableCompression>
    <forceVirtualKeyboard>false</forceVirtualKeyboard>
    <preventMultipleLogon>true</preventMultipleLogon>
    <randomizeVirtualKeys>false</randomizeVirtualKeys>
    <timeout>
      <forcedTimeout>16</forcedTimeout>
      <sessionIdleTimeout>10</sessionIdleTimeout>
    </timeout>
  </advancedConfig>

```

```

    <clientNotification></clientNotification>
    <enablePublicUrlAccess>>false</enablePublicUrlAccess>
    <enableLogging>>false</enableLogging>
  </advancedConfig>
  <authenticationConfiguration>
    <passwordAuthentication>
      <authenticationTimeout>1</authenticationTimeout>
      <primaryAuthServers>
        <com.vmware.vshield.edge.sslvpn.dto.LdapAuthServerDto>
          <ip>1.1.1.1</ip>
          <port>90</port>
          <timeOut>20</timeOut>
          <enableSsl>>false</enableSsl>
          <searchBase>searchbasevalue</searchBase>
          <bindDomainName>binddnvalue</bindDomainName>
          <bindPassword>password</bindPassword>
          <loginAttributeName>cain</loginAttributeName>
          <searchFilter>found</searchFilter>
          <enabled>>true</enabled>
        </com.vmware.vshield.edge.sslvpn.dto.LdapAuthServerDto>
        <com.vmware.vshield.edge.sslvpn.dto.RadiusAuthServerDto>
          <ip>3.3.3.3</ip>
          <port>90</port>
          <timeOut>20</timeOut>
          <secret>struct9870</secret>
          <nasIp>1.1.1.9</nasIp>
          <retryCount>10</retryCount>
        </com.vmware.vshield.edge.sslvpn.dto.RadiusAuthServerDto>
        <com.vmware.vshield.edge.sslvpn.dto.LocalAuthServerDto>
          <enabled>>true</enabled>
          <passwordPolicy>
            <minLength>1</minLength>
            <maxLength>63</maxLength>
            <minAlphabets>0</minAlphabets>
            <minDigits>0</minDigits>
            <minSpecialChar>1</minSpecialChar>
            <allowUserIdwithinPassword>>false</allowUserIdwithinPassword>
            <passwordLifeTime>20</passwordLifeTime>
            <expiryNotification>1</expiryNotification>
          </passwordPolicy>
          <accountLockoutPolicy>
            <retryCount>3</retryCount>
            <retryDuration>3</retryDuration>
            <lockoutDuration>3</lockoutDuration>
          </accountLockoutPolicy>
        </com.vmware.vshield.edge.sslvpn.dto.LocalAuthServerDto>
        <timeOut>20</timeOut>
        <sourceIp>1.2.2.3</sourceIp>
      </com.vmware.vshield.edge.sslvpn.dto.RsaAuthServerDto>
    </primaryAuthServers>
    <secondaryAuthServer>
      <com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
        <ip>1.1.1.1</ip>
        <port>90</port>
        <timeOut>20</timeOut>
        <enableSsl>>false</enableSsl>
        <searchBase>searchbasevalue</searchBase>
        <bindDomainName>binddnvalue</bindDomainName>
        <bindPassword>password</bindPassword>
        <loginAttributeName>cain</loginAttributeName>
        <searchFilter>found</searchFilter>
        <terminateSessionOnAuthFails>>false</terminateSessionOnAuthFails>
        <enabled>>true</enabled>
      </com.vmware.vshield.edge.sslvpn.dto.AdAuthServerDto>
    </secondaryAuthServer>
  </passwordAuthentication>
</authenticationConfiguration>

```

```
</sslvpnConfig>
```

---

## Delete SSL VPN Configuration

Deletes the SSL VPN configurations on the specified NSX Edge.

### Example 9-184. Delete SSL VPN Configuration

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/
```

---

## Query SSL VPN Statistics

Retrieves SSL VPN statistics on the specified NSX Edge.

### Example 9-185. Get SSL VPN statistics

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/statistics/dashboard
    /sslvpn?interval=range <!--range can be 1 - 60 minutes or
    oneDay|oneweek|onemonth|oneyear. Default is 60 minutes -->
```

Response Body:

```
<dashboardStatistics>
  <meta>
    <startTime>1344809160</startTime> <!-- in seconds -->
    <endTime>1344809460</endTime> <!-- in seconds -->
    <interval>300</interval>
  </meta>
  <data>
    <sslvpn>
      <sslvpnBytesOut>
        <dashboardStatistic>
          <timestamp>1344809160</timestamp>
          <value>0.0</value>
        </dashboardStatistic>
        <dashboardStatistic>
          <timestamp>1344809460</timestamp>
          <value>0.0</value>
        </dashboardStatistic>
      </sslvpnBytesOut>
      <sslvpnBytesIn>
        <dashboardStatistic>
          <timestamp>1344809160</timestamp>
          <value>0.0</value>
        </dashboardStatistic>
        <dashboardStatistic>
          <timestamp>1344809460</timestamp>
          <value>0.0</value>
        </dashboardStatistic>
      </sslvpnBytesIn>
      <activeClients>
        <dashboardStatistic>
          <timestamp>1344809160</timestamp>
          <value>4.0</value>
        </dashboardStatistic>
        <dashboardStatistic>
          <timestamp>1344809460</timestamp>
          <value>4.0</value>
        </dashboardStatistic>
      </activeClients>
      <authFailures>
```

```

    <dashboardStatistic>
      <timestamp>1344809160</timestamp>
      <value>2.0</value>
    </dashboardStatistic>
    <dashboardStatistic>
      <timestamp>1344809460</timestamp>
      <value>2.0</value>
    </dashboardStatistic>
  </authFailures>
  <sessionsCreated>
    <dashboardStatistic>
      <timestamp>1344809160</timestamp>
      <value>4.0</value>
    </dashboardStatistic>
    <dashboardStatistic>
      <timestamp>1344809460</timestamp>
      <value>4.0</value>
    </dashboardStatistic>
  </sessionsCreated>
</sslvpn>
</data>
</dashboardStatistics>

```

## Enable or Disable SSLv3

Enables or disables SSLv3.

### Example 9-186. Enable or Disable SSLv3

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/sslvpn/config/server/>

Request Body:

```

<serverSettings>
  <serverAddresses>
    <ipAddress>
      <ipAddress>10.117.81.69</ipAddress>
    </ipAddress>
  </serverAddresses>
  <port>443</port>
  <sslVersionList>
    <version>SSLv3</version>
    <version>TLSv1</version>
    <version>TLSv1_2</version>
    <version>TLSv1_1</version>
  </sslVersionList>
</serverSettings>

```

---

## Working with L2 VPN

L2 VPN allows you to configure a tunnel between two sites. Virtual machines remain on the same subnet in spite of being moved between these sites, which enables you to extend your datacenter. An NSX Edge at one site can provide all services to virtual machines on the other site.

In order to create the L2 VPN tunnel, you configure an L2 VPN server and L2 VPN client.

### Configure L2VPN

You first enable the L2 VPN service on the NSX Edge instance and then configure a server and a client.

#### Example 9-187. Configure L2VPN for Server

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/l2vpn/config/>

Request Body:

```
<l2vpn>
  <enabled>true</enabled>  <!-- Optional, true by default -->
  <logging>  <!-- optional. Disable by default. -->
    <enable>>false</enable>  <!-- optional, false by default. -->
    <logLevel>info</logLevel>  <!-- optional, default is INFO. -->
  </logging>
  <l2vpnsites>
    <l2vpnsite>
      <server>
        <configuration>
          <listenerIp>192.168.15.65</listenerIp>  <!-- Required. IP of external
            interface on which L2VPN service to listen on -->
          <listenerPort>443</listenerPort>  <!-- optional. 443 by default. Port on
            which L2VPN service to listen on -->
          <encryptionAlgorithm>RC4-MD5</encryptionAlgorithm>  <!-- Mandatory.
            Supported ciphers are "RC4-MD5", "AES128-SHA", "AES256-SHA",
            "DES-CBC3-SHA", "AES128-GCM-SHA256" and "NULL-MD5"-->
          <serverCertificate>certificate-4</serverCertificate>  <!-- Optional. If
            not specified server will use its default(selfsigned) certificate-->
          <peerSites>  <!-- Required. Minimum one peer site must be configured to
            enable l2vpn server service-->
            <peerSite>  <!-- Required. Minimum one peer site must be configured to
              enable l2vpn server service-->
              <name>PeerSite1</name>  <!-- Required. Unique site name given to
                the site getting configured -->
              <description>description</description>  <!-- optional. Description
                about the site -->
              <l2vpnUser>  <!-- Required. Every peer site must have a user
                configuration -->
                <userId>apple</userId>
                <password>apple</password>
              </l2vpnUser>
              <vnics>  <!-- Required. List of vnics to be stretched over the
                tunnel -->
                <index>10</index>
              </vnics> <egressOptimization>  <!-- optional. To block the internet
                requests over tunnel-->
                <gatewayIpAddress>192.168.15.1</gatewayIpAddress>
              </egressOptimization>
              <enabled>true</enabled>  <!-- optional. true by default-->
            </peerSite>
          </peerSites>
        </configuration>
      </server>
    </l2vpnsite>
  </l2vpnsites>
</l2vpn>
```

---

#### Example 9-188. Configure L2VPN for Client

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/l2vpn/config/>

Request Body:

```
<l2vpn>
  <enabled>true</enabled>  <!-- Optional, true by default -->
  <logging>  <!-- optional. Disable by default. -->
    <enable>>false</enable>  <!-- optional, false by default. -->
    <logLevel>info</logLevel>  <!-- optional, default is INFO. -->
  </logging>
  <l2vpnsites>
    <l2vpnsite>
```

```

<client>
  <configuration>
    <serverAddress>192.168.15.23</serverAddress>  <!-- Required. IP/Hostname
to connect -->
    <serverPort>443</serverPort>  <!-- optional. 443 by default. Port to
connect on -->
    <vnic>10</vnic>  <!-- Required. Traffic from this internal vnic
interfaces will be forwarded to L2VPN tunnel -->
    <vnic>11</vnic>
    <encryptionAlgorithm>AES128-SHA</encryptionAlgorithm>  <!-- Mandatory.
Supported ciphers are "RC4-MD5", "AES128-SHA", "AES256-SHA",
"DES-CBC3-SHA", "AES128-GCM-SHA256" and "NULL-MD5"-->
    <caCertificate>certificate-4</caCertificate>  <!-- Optional. Validate
server certificate sent from server against this certificate-->
    <egressOptimization>  <!-- optional. To block the internet requests over
tunnel-->
      <gatewayIpAddress>192.168.15.1</gatewayIpAddress>
    </egressOptimization>
  </configuration>
</client>
</l2VpnSite>
</l2VpnSites>
</l2Vpn>

```

## Query L2VPN

Retrieves the current L2VPN configuration for NSX Edge.

### Example 9-189. Query L2VPN

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/l2vpn/config/
```

Response Body:

```

<l2Vpn>
  <version>4</version>
  <enabled>true</enabled>
  <logging>
    <enable>false</enable>
    <logLevel>info</logLevel>
  </logging>
  <l2VpnSites>
    <l2VpnSite>
      <client>
        <configuration>
          <serverAddress>192.168.15.23</serverAddress>
          <serverPort>443</serverPort>
          <caCertificate>certificate-4</caCertificate>
          <vnic>10</vnic>
          <egressOptimization>
            <gatewayIpAddress>192.168.15.1</gatewayIpAddress>
          </egressOptimization>
          <encryptionAlgorithm>AES128-SHA</encryptionAlgorithm>

```

```

</configuration>
<l2VpnUser>
  <userId>apple</userId>
</l2VpnUser>
<proxySetting>
  <type>https</type>
  <address>10.112.243.202</address>
  <port>443</port>
  <userName>root</userName>
</proxySetting>
</client>
</l2VpnSite>
</l2VpnSites>
</l2Vpn>

```

---

## Query L2VPN Statistics

Retrieves L2VPN statistics which has information such as tunnel status, sent bytes, received bytes etc. for the given edge.

### Example 9-190. Query L2VPN statistics

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/l2vpn/config/statistics
```

Response Body:

```

<l2vpnStatusAndStats>
  <timeStamp>1403285853</timeStamp>
  <siteStats>
    <l2vpnStats>
      <name>site-1</name>
      <tunnelStatus>up</tunnelStatus>
      <establishedDate>1403285827</establishedDate>
      <txBytesFromLocalSubnet>478</txBytesFromLocalSubnet>
      <encryptionAlgorithm>RC4-MD5</encryptionAlgorithm>
      <rxBytesOnLocalSubnet>42</rxBytesOnLocalSubnet>
    </l2vpnStats>
    <l2vpnStats>
      <name>site-2</name>
      <tunnelStatus>up</tunnelStatus>
      <establishedDate>1403285829</establishedDate>
      <txBytesFromLocalSubnet>408</txBytesFromLocalSubnet>
      <encryptionAlgorithm>RC4-MD5</encryptionAlgorithm>
      <rxBytesOnLocalSubnet>450</rxBytesOnLocalSubnet>
    </l2vpnStats>
  </siteStats>
</l2vpnStatusAndStats>

```

---

## Enable L2VPN

Enables or disables the L2VPN service on edge appliance according to the value of the query parameter "enableService".

### Example 9-191. Enable L2VPN

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/l2vpn/config/?enableService=true
```

---

Result Codes:

On Success: 204 No Content

On Failure:

- 400 Bad Request
- 403 Forbidden if the user is not having appropriate role and scope
- 404 Not found

## Delete L2VPN

### Example 9-192. Delete L2VPN

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/l2vpn/config/
```

---

## Working with IPSEC VPN

NSX Edge supports site-to-site IPsec VPN between an NSX Edge instance and remote sites. NSX Edge supports certificate authentication, preshared key mode, IP unicast traffic, and no dynamic routing protocol between the NSX Edge instance and remote VPN routers. Behind each remote VPN router, you can configure multiple subnets to connect to the internal network behind an NSX Edge through IPsec tunnels. These subnets and the internal network behind a NSX Edge must have address ranges that do not overlap.

You can deploy an NSX Edge agent behind a NAT device. In this deployment, the NAT device translates the VPN address of an NSX Edge instance to a publicly accessible address facing the Internet. Remote VPN routers use this public address to access the NSX Edge instance.

You can place remote VPN routers behind a NAT device as well. You must provide the VPN native address and the VPN Gateway ID to set up the tunnel. On both ends, static one-to-one NAT is required for the VPN address.

You can have a maximum of 64 tunnels across a maximum of 10 sites.

### Example 9-193. Configure IPSEC VPN

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/ipsec/config
```

Request Body:

```
<ipsec>
  <enabled>true</enabled>  <!-- optional, true by default -->
  <logging>  <!-- optional. logging is disable by default. -->
    <logLevel>debug</logLevel>  <!-- optional, default is info. -->
    <enable>true</enable>  <!-- optional, default is false. -->
  </logging>
  <global>
    <psk>hello123</psk>  <!-- Required only when peerIp is specified as any in
      siteConfig -->
    <serviceCertificate>certificate-4</serviceCertificate>  <!-- Required when x.509
      certificate mode is selected -->
    <caCertificates>  <!-- Optional, CA list -->
      <caCertificate>certificate-3</caCertificate>
    </caCertificates>
    <crlCertificates>  <!-- optional, CRL list -->
      <crlCertificate>crl-1</crlCertificate>
    </crlCertificates>
  </global>
  <sites>
    <site>
      <enabled>true</enabled>  <!-- optional, true by default -->
```

```

<name>VPN to edge-pa-1</name>  <!-- Optional -->
<description>psk VPN to edge-pa-1 192.168.11.0/24 ==
    192.168.1.0/24</description>  <!-- Optional -->
<localId>11.0.0.11</localId>
<localIp>11.0.0.11</localIp>
<peerId>11.0.0.1</peerId>
<peerIp>any</peerIp>  <!-- Can be a Ipv4Address such as 11.0.0.3 -->
<encryptionAlgorithm>aes256</encryptionAlgorithm>  <!-- Optional, default
    aes256-->
<authenticationMode>psk</authenticationMode>  <!-- Possible values are psk
    and x.509 -->
<!-- <psk>hello123</psk> -->  <!-- Required if peerIp is not any -->
<enablePfs>true</enablePfs>  <!-- Optional, true by default -->
<dhGroup>dh2</dhGroup>  <!-- Optional, dh2 by default -->
<localSubnets>
  <subnet>192.168.11.0/24</subnet>
</localSubnets>
<peerSubnets>
  <subnet>192.168.1.0/24</subnet>
</peerSubnets>
</site>
<site>
  <name>VPN to edge-right</name>
  <description>certificate VPN to edge-right 192.168.22.0/24 ==
    192.168.2.0/24</description>
  <localId>11.0.0.12</localId>
  <localIp>11.0.0.12</localIp>
  <peerId>C=CN, ST=BJ, L=BJ, O=VMware, OU=DEV, CN=Right</peerId>  <!-- Should
    be a DN if authenticationMode is x.509 -->
  <peerIp>11.0.0.2</peerIp>
  <encryptionAlgorithm>aes256</encryptionAlgorithm>
  <authenticationMode>x.509</authenticationMode>
  <enablePfs>true</enablePfs>
  <dhGroup>dh2</dhGroup>
  <localSubnets>
    <subnet>192.168.22.0/24</subnet>
  </localSubnets>
  <peerSubnets>
    <subnet>192.168.2.0/24</subnet>
  </peerSubnets>
  <extension>securelocaltrafficbyip=192.168.11.1</extension>  <!-- Default
    value. To disable this extension, replace with
    securelocaltrafficbyip=0-->
</site>
</sites>
</ipsec>

```

## Retrieve IPSec Configuration

### Example 9-194. Get IPSec Configuration

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/ipsec/config
```

Response Body when IPSec is not configured:

```

<ipsec>
  <enabled>true</enabled>
  <logging>
    <enable>true</enable>
    <logLevel>debug</logLevel>
  </logging>
  <sites></sites>  <!-- No site to site config present -->
</ipsec>

```

Response Body when IPSec is configured for site-to-site:

```

<ipsec>
  <enabled>true</enabled>
  <logging>
    <logLevel>debug</logLevel>
    <enable>true</enable>
  </logging>
  <global>
    <psk>hello123</psk>
    <serviceCertificate>certificate-4</serviceCertificate>
    <caCertificates> <!-- optional, CA list -->
      <caCertificate>certificate-3</caCertificate>
    </caCertificates>
    <crlCertificates>
      <crlCertificate>crl-1</crlCertificate>
    </crlCertificates>
  </global>
  <sites>
    <site>
      <enabled>true</enabled>
      <name>VPN to edge-pa-1</name>
      <description>psk VPN to edge-pa-1 192.168.11.0/24 ==
        192.168.1.0/24</description>
      <localId>11.0.0.11</localId>
      <localIp>11.0.0.11</localIp>
      <peerId>11.0.0.1</peerId>
      <peerIp>any</peerIp>
      <encryptionAlgorithm>aes256</encryptionAlgorithm>
      <authenticationMode>psk</authenticationMode>
      <enablePfs>true</enablePfs>
      <dhGroup>dh2</dhGroup>
      <localSubnets>
        <subnet>192.168.11.0/24</subnet>
      </localSubnets>
      <peerSubnets>
        <subnet>192.168.1.0/24</subnet>
      </peerSubnets>
    </site>
    <site>
      <name>VPN to edge-right</name>
      <description>certificate VPN to edge-right 192.168.22.0/24 ==
        192.168.2.0/24</description>
      <localId>11.0.0.12</localId>
      <localIp>11.0.0.12</localIp>
      <peerId>C=CN, ST=BJ, L=BJ, O=VMware, OU=DEV, CN=Right</peerId>
      <peerIp>11.0.0.2</peerIp>
      <encryptionAlgorithm>aes256</encryptionAlgorithm>
      <authenticationMode>x.509</authenticationMode>
      <enablePfs>true</enablePfs>
      <dhGroup>dh2</dhGroup>
      <localSubnets>
        <subnet>192.168.22.0/24</subnet>
      </localSubnets>
      <peerSubnets>
        <subnet>192.168.2.0/24</subnet>
      </peerSubnets>
    </site>
  </sites>
</ipsec>

```

---

## Retrieve IPSec Statistics

**Example 9-195.** Get IPSEC statistics

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/ipsec/statistics>

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<ipsecStatusAndStats>
  <siteStatistics>
    <ikeStatus>
      <channelStatus>up</channelStatus>
      <channelState>STATE_MAIN_I4 (ISAKMP SA established)</channelState>
      <lastInformationalMessage></lastInformationalMessage>
      <localIpAddress>10.0.0.12</localIpAddress>
      <peerId>11.0.0.12</peerId>
      <peerIpAddress>10.0.0.2</peerIpAddress>
    </ikeStatus>
    <tunnelStats>
      <tunnelStatus>up</tunnelStatus>
      <tunnelState>STATE_QUICK_I2 (sent QI2, IPsec SA established)</tunnelState>
      <lastInformationalMessage></lastInformationalMessage>
      <localSubnet>192.168.2.0/24</localSubnet>
      <peerSubnet>192.168.22.0/24</peerSubnet>
    </tunnelStats>
  </siteStatistics>
  <siteStatistics>
    <ikeStatus>
      <channelStatus>up</channelStatus>
      <channelState>STATE_MAIN_I4 (ISAKMP SA established)</channelState>
      <lastInformationalMessage></lastInformationalMessage>
      <localIpAddress>10.0.0.11</localIpAddress>
      <peerId>11.0.0.11</peerId>
      <peerIpAddress>10.0.0.1</peerIpAddress>
    </ikeStatus>
    <tunnelStats>
      <tunnelStatus>up</tunnelStatus>
      <tunnelState>STATE_QUICK_I2 (sent QI2, IPsec SA established)</tunnelState>
      <lastInformationalMessage></lastInformationalMessage>
      <localSubnet>192.168.1.0/24</localSubnet>
      <peerSubnet>192.168.11.0/24</peerSubnet>
    </tunnelStats>
  </siteStatistics>
  <timestamp>1325766138</timestamp>
</ipsecStatusAndStats>
```

## Query Tunnel Traffic Statistics

Retrieves tunnel traffic statistics for the specified time interval. Default interval is 1 hour. Other possible values are 1-60 minutes|one day|one week|one month|one year.

### Example 9-196. Get tunnel traffic statistics

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/statistics/dashboard
/ipsec?interval=range
```

Response Body:

```
<dashboardStatistics>
  <meta>
    <startTime>1344809160</startTime>  <!-- in seconds -->
    <endTime>1344809460</endTime>  <!-- in seconds -->
    <interval>300</interval>
  </meta>
  <data>
    <ipsec>
      <ipsecTunnels>
        <dashboardStatistic>
          <timestamp>1344809160</timestamp>
          <value>0.0</value>
```

```

</dashboardStatistic>
<dashboardStatistic>
  <timestamp>1344809460</timestamp>
  <value>0.0</value>
</dashboardStatistic>
</ipsecTunnels>
<ipsecBytesIn>
  <dashboardStatistic>
    <timestamp>1344809160</timestamp>
    <value>0.0</value>
  </dashboardStatistic>
  <dashboardStatistic>
    <timestamp>1344809460</timestamp>
    <value>0.0</value>
  </dashboardStatistic>
</ipsecBytesIn>
<ipsecBytesOut>
  <dashboardStatistic>
    <timestamp>1344809160</timestamp>
    <value>0.0</value>
  </dashboardStatistic>
  <dashboardStatistic>
    <timestamp>1344809460</timestamp>
    <value>0.0</value>
  </dashboardStatistic>
</ipsecBytesOut>
</ipsec>
</data>
</dashboardStatistics>

```

---

### Delete IPsec Configuration

Deletes the IPSEC configuration for the specified NSX Edge.

#### Example 9-197. Delete IPsec

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/ipsec/config/
```

---

## Managing an NSX Edge

### Force Sync Edge

Re-synchronizes the NSX Edge virtual machines.

#### Example 9-198. Force sync Edge

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId?action=forcesync
```

---

### Redeploy Edge

Redeploys NSX Edge virtual machines.

#### Example 9-199. Redeploy Edge

---

Request:

---

POST <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId?action=redeploy>

---

## Update DNS Settings

Update dns settings (primary/secondary and search domain) of an Edge.

### Example 9-200. Update DNS

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/dnsclient>

Request Body:

```
<dnsClient>
  <primaryDns>10.117.0.1</primaryDns>
  <secondaryDns>10.117.0.2</secondaryDns>
  <domainName>vmware.com</domainName>
  <domainName>foo.com</domainName>
</dnsClient>
```

---

## Modify AESNI Setting

Redeploys NSX Edge virtual machines.

### Example 9-201. Modify AESNI

---

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/aesni?enable=true/false>

---

## Modify Edge Appliance Core Dump Setting

Enabling the advanced debugging feature redeploys the Edge Service Gateway (ESG), enables coredump, deploys an inbuilt extra disk to save the core-dump files (The extra disk consumes 1GB for compact edge and 8GB for other edge types, and disabling this feature detaches the disk), and runs the binary in debug mode.

### Example 9-202. Modify core dump setting

---

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/coredump?enable=true/false>

---

## Modify Log Setting

### Example 9-203. Modify log setting

---

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/logging?level=logLevel>

---

## Query Edge Summary

Retrieves details about the specified Edge.

### Example 9-204. Retrieve Edge details

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/summary>

Response Body:

```
<edgeSummary>
  <objectId>edge-32</objectId>
  <type>
    <typeName>Edge</typeName>
  </type>
  <name>vShield-edge-32</name>
  <revision>16</revision>
  <objectTypeName>Edge</objectTypeName>
  <id>edge-32</id>
  <state>deployed</state>
  <datacenterMoid>datacenter-2</datacenterMoid>
  <datacenterName>Datacenter</datacenterName>
  <apiVersion>4.0</apiVersion>
  <numberOfConnectedVnics>2</numberOfConnectedVnics>
  <appliancesSummary>
    <vmVersion>5.1.0</vmVersion>
    <applianceSize>compact</applianceSize>
    <fqdn>vShield-edge-32</fqdn>
    <numberOfDeployedVms>1</numberOfDeployedVms>
    <activeVseHaIndex>0</activeVseHaIndex>
    <vmMoidOfActiveVse>vm-301</vmMoidOfActiveVse>
    <vmNameOfActiveVse>vShield-edge-32-0</vmNameOfActiveVse>
    <hostMoidOfActiveVse>host-159</hostMoidOfActiveVse>
    <hostNameOfActiveVse>10.20.114.8</hostNameOfActiveVse>
    <resourcePoolMoidOfActiveVse>resgroup-208</resourcePoolMoidOfActiveVse>
    <resourcePoolNameOfActiveVse>Resources</resourcePoolNameOfActiveVse>
    <dataStoreMoidOfActiveVse>datastore-160</dataStoreMoidOfActiveVse>
    <dataStoreNameOfActiveVse>storage1</dataStoreNameOfActiveVse>
    <statusFromVseUpdatedOn>1310625858000</statusFromVseUpdatedOn>
  </appliancesSummary>
  <featureCapabilities>
    <timestamp>1337956125602</timestamp>
    <featureCapability>
      <service>nat</service>
      <isSupported>true</isSupported>
      <configurationLimit>
        <key>MAX_RULES_PER_ACTION</key>
        <value>2048</value>
      </configurationLimit>
    </featureCapability>
    <featureCapability>
      <service>syslog</service>
      <isSupported>true</isSupported>
      <configurationLimit>
        <key>MAX_SERVER_IPS</key>
        <value>2</value>
      </configurationLimit>
    </featureCapability>
    <featureCapability>
      <service>staticRouting</service>
      <isSupported>true</isSupported>
      <configurationLimit>
        <key>MAX_ROUTES</key>
        <value>2048</value>
      </configurationLimit>
    </featureCapability>
    <featureCapability>
      <service>ipsec</service>
      <isSupported>true</isSupported>
      <configurationLimit>
        <key>MAX_TUNNELS</key>
        <value>64</value>
      </configurationLimit>
    </featureCapability>
  </featureCapabilities>
</edgeSummary>
```

```

<service>loadBalancer</service>
<isSupported>true</isSupported>
<configurationLimit>
  <key>MAX_POOLS</key>
  <value>10</value>
</configurationLimit>
<configurationLimit>
  <key>MAX_VIRTUAL_SERVERS</key>
  <value>10</value>
</configurationLimit>
<configurationLimit>
  <key>MAX_MEMBERS_IN_POOL</key>
  <value>32</value>
</configurationLimit>
</featureCapability>
<featureCapability>
  <service>fw</service>
  <isSupported>true</isSupported>
  <configurationLimit>
    <key>MAX_RULES</key>
    <value>2048</value>
  </configurationLimit>
</featureCapability>
<featureCapability>
  <service>dns</service>
  <isSupported>true</isSupported>
  <configurationLimit>
    <key>MAX_SERVER_IPS</key>
    <value>2</value>
  </configurationLimit>
</featureCapability>
<featureCapability>
  <service>sslvpn</service>
  <isSupported>true</isSupported>
  <configurationLimit>
    <key>MAX_CONCURRENT_USERS</key>
    <value>25</value>
  </configurationLimit>
</featureCapability>
<featureCapability>
  <service>edge</service>
  <isSupported>true</isSupported>
  <configurationLimit>
    <key>MAX_APPLIANCES</key>
    <value>2</value>
  </configurationLimit>
  <configurationLimit>
    <key>MAX_VNICs</key>
    <value>10</value>
  </configurationLimit>
</featureCapability>
<featureCapability>
  <service>firewall</service>
  <isSupported>true</isSupported>
  <configurationLimit>
    <key>MAX_RULES</key>
    <value>2048</value>
  </configurationLimit>
</featureCapability>
<featureCapability>
  <service>dhcp</service>
  <isSupported>true</isSupported>
  <configurationLimit>
    <key>MAX_POOL_AND_BINDINGS</key>
    <value>2048</value>
  </configurationLimit>
</featureCapability>
</featureCapability>

```

```

    <service>highAvailability</service>
    <isSupported>true</isSupported>
    <configurationLimit>
      <key>MAX_MANAGEMENT_IPS</key>
      <value>2</value>
    </configurationLimit>
  </featureCapability>
</featureCapabilities>
</edgeSummary>

```

---

## Query Edge Status

Retrieves the status of the specified Edge.

### Example 9-205. Query status

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/status
```

Response Body:

```

<edgeStatus>
  <timestamp>1343739873000</timestamp>
  <systemStatus>good</systemStatus>
  <activeVseHaIndex>0</activeVseHaIndex>
  <edgeStatus>GREEN</edgeStatus>  <!-- {GREY,RED,YELLOW,GREEN}. GREY => unknown status.
    RED => None of appliance in serving state. YELLOW => Intermittent health
    check failures. If health check fails for 5 consecutive times for all
    appliance (2 for HA else 1) then status will turn to RED. GREEN => Good -->
  <publishStatus>APPLIED</publishStatus>  <!-- Applied or persisted i.e., not applied
    to vse yet-->
  <version>8</version>  <!-- Current configuration version -->
  <edgeVmStatus>
    <edgeVmStatus>
      <edgeVMStatus>GREEN</edgeVMStatus>  <!-- individual vm status -->
      <haState>active</haState>  <!-- active / standby -->
      <index>0</index>
      <id>vm-358</id>
      <name>test2-0</name>
    </edgeVmStatus>
    <edgeVmStatus>
      <edgeVMStatus>GREEN</edgeVMStatus>
      <haState>active</haState>
      <index>1</index>
      <id>vm-362</id>
      <name>test2-1</name>
    </edgeVmStatus>
  </edgeVmStatus>
  <featureStatuses>
    <featureStatus>
      <service>loadBalancer</service>
      <configured>>false</configured>
      <serverStatus>down</serverStatus>
    </featureStatus>
    <featureStatus>
      <service>dhcp</service>
      <configured>true</configured>
      <publishStatus>Applied</publishStatus>
      <serverStatus>up</serverStatus>
    </featureStatus>
    <featureStatus>
      <service>sslvpn</service>
      <configured>>false</configured>
      <serverStatus>down</serverStatus>
    </featureStatus>
  </featureStatuses>
</edgeStatus>

```

```

    <service>syslog</service>
    <configured>>false</configured>
    <serverStatus>up</serverStatus>
  </featureStatus>
</featureStatus>
  <service>nat</service>
  <configured>>false</configured>
</featureStatus>
</featureStatus>
  <service>dns</service>
  <configured>>false</configured>
  <serverStatus>down</serverStatus>
</featureStatus>
</featureStatus>
  <service>ipsec</service>
  <configured>>false</configured>
  <serverStatus>down</serverStatus>
</featureStatus>
</featureStatus>
  <service>firewall</service>
  <configured>>true</configured>
  <publishStatus>Applied</publishStatus>
</featureStatus>
</featureStatus>
  <service>staticRouting</service>
  <configured>>false</configured>
</featureStatus>
</featureStatus>
  <service>highAvailability</service>
  <configured>>true</configured>
  <publishStatus>Applied</publishStatus>
  <serverStatus>up</serverStatus>
</featureStatus>
</featureStatuses>
</edgeStatus>

```

This call can be used with the following query parameters:

- `getlatest`: fetches the status live from NSX Edge when set to true (default). When false, fetches the latest available status from database.
- `detailed`: fetches the detailed status per feature when set to true. When false (default), gives an aggregated summary of the status per feature.
- `preRulesStatus=true`: fetches detailed output for pre rules in addition to the regular output. Default value is false.

```

<preRulesExists>>true</preRulesExists>
<lastPublishedPreRulesGenerationNumber>1404824989200</lastPublishedPreRulesGeneration
Number>

```

Sample calls include:

```

GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/status?getlatest=false&detailed
=true
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/status?getlatest=true&detailed
=true
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/status?getlatest=false&detailed
=false
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/status?detailed=true
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/status?getlatest=false
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/status?preRulesStatus=true

```

## Query Edge Tech Support Logs

Retrieves the tech support logs for the specified Edge.

**Example 9-206.** Query tech support logs

Request:

---

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/techsupportlogs
```

---

## Manage CLI Credentials and Access

You can modify the CLI credentials and enable or disable SSH services for a Edge.

### Modify CLI Credentials

You can use this API to:

- Modify the password and password expiry for an existing CLI user.
- Change the CLI login (ssh) banner text.
- Modify both the username and password for Edge CLI User. This results in:
  - deletion of the old user.
  - creation of the new user with specified username and password.

**Example 9-207.** Modify CLI credentials

Request:

---

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/clisettings
```

Request Body:

```
<cliSettings>  <!-- optional. Default user/pass is admin/random, and remoteAccess is
                false (i.e. disabled) -->
  <userName>test</userName>
  <password>testpass</password>
  <remoteAccess>true</remoteAccess>
  <passwordExpiry>30</passwordExpiry>  <!-- optional. in days. defaults to 90.-->
  <sshLoginBannerText>Hello</sshLoginBannerText>  <!-- user configurable banner -->
</cliSettings>
```

---

### Change CLI Remote Access

Enables or disables the SSH service on the specified Edge.

**Example 9-208.** Change CLI remote access

Request:

---

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/cliRemoteaccess?enable
      =true\false
```

---

## Manage Auto Configuration Settings

Auto configuration default setting is enabled by default and the priority is high.

If you disable auto configuration settings, you must add the required NAT, firewall, routing rules to enable control-channel traffic for other services such as load balancing, VPN, etc.

If you change the priority of the auto configuration settings to low, the internal/auto configured rules are placed in lower precedence than the rules you create. With this, you can again control special allow/deny rules for these services too. For example, you can block specific IP addresses from accessing the VPN services.

## Modify Auto Configuration Settings

Changes the auto configuration settings for the NSX Edge.

### Example 9-209. Modify auto configuration settings

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/autoconfiguration
```

Request Body:

```
<autoConfiguration>
  <enabled>true</enabled>
  <rulePriority>high</rulePriority>
</autoConfiguration>
```

---

## Query Auto Configuration Settings

Retrieves auto configuration settings for the NSX Edge.

### Example 9-210. Retrieve auto configuration settings

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/autoconfiguration
```

Response Body:

```
<autoConfiguration>
  <enabled>true</enabled>
  <rulePriority>high</rulePriority>
</autoConfiguration>
```

---

## Working with Appliances

You can manage the Edge appliances with these REST calls.

**NOTE** Do not use hidden/system resource pool IDs as they are not supported on the UI.

### Query Appliance Configuration

Retrieves configuration of both appliances.

Starting in NSX 6.2.3, the output of the edge API methods includes both configured and current information for the NSX Edge appliance placement attributes of resource pool, datastore, host, and VM folder. These attributes are not used for configuration, but allow you to compare the configured attributes with the current attributes to see if the NSX Edge appliance placement has changed since configuration, for example, due to storage DRS or manual migration.

### Example 9-211. Get appliance configuration

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/appliances
```

Response Body:

```
<appliances>
  <applianceSize>compact</applianceSize>
  <appliance>
    <highAvailabilityIndex>0</highAvailabilityIndex>
    <vcUuid>502e2dd9-3df7-4820-6925-29832a1c0b79</vcUuid>
    <vmId>vm-417</vmId>
    <haAdminState>up</haAdminState>
    <resourcePoolId>domain-c41</resourcePoolId>
```

```

<resourcePoolName>Management & Edge Cluster</resourcePoolName>
<datastoreId>datastore-29</datastoreId>
<datastoreName>ds-site-a-nfs01</datastoreName>
<hostId>host-202</hostId>
<hostName>esxmgmt-01a.corp.local</hostName>
<vmFolderId>group-v242</vmFolderId>
<vmFolderName>NSX Edges</vmFolderName>
<vmHostName>Perimeter-Gateway-02-0</vmHostName>
<vmName>Perimeter-Gateway-02-0</vmName>
<deployed>true</deployed>
<cpuReservation>
  <reservation>1000</reservation>
</cpuReservation>
<memoryReservation>
  <reservation>512</reservation>
</memoryReservation>
<edgeId>edge-5</edgeId>
<configuredResourcePool>
  <id>domain-c41</id>
  <name>Management & Edge Cluster</name>
  <isValid>true</isValid>
</configuredResourcePool>
<configuredDataStore>
  <id>datastore-29</id>
  <name>ds-site-a-nfs01</name>
  <isValid>true</isValid>
</configuredDataStore>
<configuredHost>
  <id>host-202</id>
  <name>esxmgmt-01a.corp.local</name>
  <isValid>true</isValid>
</configuredHost>
<configuredVmFolder>
  <id>group-v242</id>
  <name>NSX Edges</name>
  <isValid>true</isValid>
</configuredVmFolder>
</appliance>
<appliance>
  <highAvailabilityIndex>1</highAvailabilityIndex>
  <vcUuid>502e3ebf-02cb-dcd2-9701-91db1e0e3bd8</vcUuid>
  <vmId>vm-429</vmId>
  <haAdminState>up</haAdminState>
  <resourcePoolId>domain-c41</resourcePoolId>
  <resourcePoolName>Management & Edge Cluster</resourcePoolName>
  <datastoreId>datastore-29</datastoreId>
  <datastoreName>ds-site-a-nfs01</datastoreName>
  <hostId>host-202</hostId>
  <hostName>esxmgmt-01a.corp.local</hostName>
  <vmFolderId>group-v242</vmFolderId>
  <vmFolderName>NSX Edges</vmFolderName>
  <vmHostName>Perimeter-Gateway-02-1</vmHostName>
  <vmName>Perimeter-Gateway-02-1</vmName>
  <deployed>true</deployed>
  <edgeId>edge-5</edgeId>
  <configuredResourcePool>
    <id>domain-c41</id>
    <name>Management & Edge Cluster</name>
    <isValid>true</isValid>
  </configuredResourcePool>
  <configuredDataStore>
    <id>datastore-29</id>
    <name>ds-site-a-nfs01</name>
    <isValid>true</isValid>
  </configuredDataStore>
  <configuredHost>
    <id>host-202</id>
    <name>esxmgmt-01a.corp.local</name>

```

```

        <isValid>true</isValid>
      </configuredHost>
    <configuredVmFolder>
      <id>group-v242</id>
      <name>NSX Edges</name>
      <isValid>true</isValid>
    </configuredVmFolder>
  </appliance>
  <deployAppliances>true</deployAppliances>
</appliances>

```

### Modify Appliance Configuration

You can retrieve the configuration of both appliances by using the GET call in [Example 9-211](#) and replace the size, resource pool, datastore, and custom parameters of the appliances by using a PUT call. If there were two appliances earlier you PUT only one appliance, the other appliance is deleted.

#### Example 9-212. Modify appliance configuration

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/appliances>

Request Body:

```

<appliances>
  <applianceSize>COMPACT</applianceSize>
  <appliance>
    <resourcePoolId>resgroup-1610</resourcePoolId>
    <datastoreId>datastore-5288</datastoreId>
  </appliance>
  <appliance>
    <resourcePoolId>resgroup-1610</resourcePoolId>
    <datastoreId>datastore-5288</datastoreId>
  </appliance>
</appliances>

```

---

### Change Appliance Size

Changes the size of both appliances.

#### Example 9-213. Change appliance size

---

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/appliances/?size=compact|large|xlarge>

---

### Manage an Appliance

You can manage an appliance by specifying its HA index.

#### Query Appliance

Retrieves the configuration of the appliance with the specified *haIndex*.

#### Example 9-214. Get configuration of appliance with specified haIndex

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/appliances/haIndex>

Response Body:

```

<appliance>
  <highAvailabilityIndex>0</highAvailabilityIndex>

```

```

<vcUuid>502e2dd9-3df7-4820-6925-29832a1c0b79</vcUuid>
<vmId>vm-417</vmId>
<haAdminState>up</haAdminState>
<resourcePoolId>domain-c41</resourcePoolId>
<resourcePoolName>Management & Edge Cluster</resourcePoolName>
<datastoreId>datastore-29</datastoreId>
<datastoreName>ds-site-a-nfs01</datastoreName>
<hostId>host-202</hostId>
<hostName>esxmgmt-01a.corp.local</hostName>
<vmFolderId>group-v242</vmFolderId>
<vmFolderName>NSX Edges</vmFolderName>
<vmHostname>Perimeter-Gateway-02-0</vmHostname>
<vmName>Perimeter-Gateway-02-0</vmName>
<deployed>>true</deployed>
<cpuReservation>
  <reservation>1000</reservation>
</cpuReservation>
<memoryReservation>
  <reservation>512</reservation>
</memoryReservation>
<edgeId>edge-5</edgeId>
<configuredResourcePool>
  <id>domain-c41</id>
  <name>Management & Edge Cluster</name>
  <invalid>true</invalid>
</configuredResourcePool>
<configuredDataStore>
  <id>datastore-29</id>
  <name>ds-site-a-nfs01</name>
  <invalid>true</invalid>
</configuredDataStore>
<configuredHost>
  <id>host-202</id>
  <name>esxmgmt-01a.corp.local</name>
  <invalid>true</invalid>
</configuredHost>
<configuredVmFolder>
  <id>group-v242</id>
  <name>NSX Edges</name>
  <invalid>true</invalid>
</configuredVmFolder>
</appliance>

```

---

## Modify Appliance

Modifies the configuration of the appliance with the specified *haIndex*.

**Example 9-215.** Modify configuration of appliance with specified *haIndex*

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/appliances/haIndex>

Request Body:

```

<appliance>
  <resourcePoolId>resgroup-53</resourcePoolId>
  <datastoreId>datastore-29</datastoreId>
  <hostId>host-28</hostId>
  <vmFolderId>group-v38</vmFolderId>
  <customField>
    <key>system.service.vmware.vsla.main01</key>
    <value>string</value>
  </customField>
  <cpuReservation>
    <limit>2399</limit>
    <reservation>500</reservation>

```

```

    <shares>500</shares>
  </cpuReservation>
  <memoryReservation>
    <limit>5000</limit>
    <reservation>500</reservation>
    <shares>20480</shares>
  </memoryReservation>
</appliance>

```

---

You can force an HA failover on an active NSX Edge appliance by changing its `haAdminState` value to `down`. See [“Force High Availability Failover”](#) on page 231 for more information.

**Example 9-216.** Change `haAdminState` configuration of appliance with specified `haIndex`

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/appliances/haIndex
```

Request Body:

```

<appliance>
  <highAvailabilityIndex>haIndex</highAvailabilityIndex>
  <vmId>vm-93</vmId>
  <haAdminState>down</haAdminState>
  <resourcePoolId>domain-c7</resourcePoolId>
  ...
</appliance>

```

---

### Delete Appliance

Deletes the appliance with the specified `haIndex`.

**Example 9-217.** Delete appliance configuration

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/appliances/haIndex
```

---

## Working with Interfaces

You can add up to ten internal or uplink interfaces to each Edge instance. A Edge must have at least one internal interface before it can be deployed.

### Add Interfaces or Sub Interfaces

You can configure one or more interface for an NSX Edge. The specified configuration is stored in the database. If any appliance(s) is associated with this Edge instance, the specified configuration is applied to the appliance as well.

**Example 9-218.** Add an interface or sub interface

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/vnics/?action=patch
```

Request Body:

```

<vnics> <!-- mamimum 10 interfaces index:0-9 can be configured. Until one connected vnic
      is configured, none of the configured features will serve the network -->
  <vnic>
    <name>EXT</name> <!-- optional. System has default Names.
      format vNic0 ... vNic7 -->
    <addressGroups>

```

```

<addressGroup>  <!-- Vnic can be configured to have more than one
  addressGroup/subnets -->
  <primaryAddress>192.168.12.240</primaryAddress>  <!-- This is mandatory for
  an addressGroup -->
  <secondaryAddresses>  <!-- Optional. Should be used to add/defined other IPs
  used for NAT, LB, VPN, etc -->
    <ipAddress>192.168.3.2</ipAddress>
    <ipAddress>192.168.3.3</ipAddress>  <!-- Optional. This way multiple IP
  Addresses can be assigned to a vnic/interface -->
  </secondaryAddresses>
  <subnetMask>255.255.255.0</subnetMask>  <!-- either subnetMask or
  subnetPrefixLength should be provided. If both then subnetprefixLength is
  ignored -->
</addressGroup>
<addressGroup>  <!-- Vnic can be configured to have more than one
  addressGroup/subnets -->
  <primaryAddress>192.168.4.1</primaryAddress>  <!-- This is mandatory for an
  addressGroup -->
  <secondaryAddresses>  <!-- Optional. Should be used to add/defined other IPs
  used for NAT, LB, VPN, etc -->
    <ipAddress>192.168.4.2</ipAddress>
    <ipAddress>192.168.4.3</ipAddress>  <!-- Optional. This way multiple IP
  Addresses can be assigned to a vnic/interface -->
  </secondaryAddresses>
  <subnetPrefixLength>24</subnetPrefixLength>
</addressGroup>
<addressGroup>  <!-- ipv6 addressGroup -->
  <primaryAddress>ffff::1</primaryAddress>  <!-- This is mandatory for an
  addressGroup -->
  <secondaryAddresses>  <!-- Optional. Should be used to add/defined other IPs
  used for NAT, LB, VPN, etc -->
    <ipAddress>ffff::2</ipAddress>
  </secondaryAddresses>
  <subnetPrefixLength>64</subnetPrefixLength>  <!-- prefixLength valid values
  1-128 -->
</addressGroup>
</addressGroups>
<mtu>1500</mtu>  <!-- optional. Default is 1600 for type "TRUNK" and 1500 for
  others-->
<type>uplink</type>  <!-- optional. Default is internal. Other possible value is
  "uplink" and "TRUNK" -->
<index>0</index>
<portgroupId>network-12</portgroupId>  <!-- Possible values here are portgroupIds
  or virtualwire-id. portgroupId needs to be defined if isConnected=true. For
  vnic of type "TRUNK" logical switch cannot be used -->
<portgroupName>VM Network</portgroupName>
<macAddress>  <!-- optional. When not specified, macAddresses will be managed by
  VC -->
  <edgeVmHaIndex>0</edgeVmHaIndex>  <!-- possible values 0 or 1 when HA is
  enabled -->
  <value>00:50:56:01:03:23</value>  <!-- optional. User must ensure that
  macAddresses provided are unique within the given layer 2 domain. -->
</macAddress>
<fenceParameter>  <!-- optional -->
  <key>ethernet0.filter1.param1</key>
  <value>1</value>
</fenceParameter>
<enableProxyArp>false</enableProxyArp>  <!-- optional. Default is false -->
<enableSendRedirects>true</enableSendRedirects>  <!-- optional. Default is true -->
<enableBridgeMode>false</enableBridgeMode>  <!-- optional. Default is false -->
<isConnected>true</isConnected>  <!-- optional. Default is false -->
<inShapingPolicy>  <!-- optional -->
  <averageBandwidth>200000000</averageBandwidth>
  <peakBandwidth>200000000</peakBandwidth>
  <burstSize>0</burstSize>
  <enabled>true</enabled>
  <inherited>false</inherited>
</inShapingPolicy>

```

```

<outShapingPolicy> <!-- optional -->
  <averageBandwidth>40000000</averageBandwidth>
  <peakBandwidth>40000000</peakBandwidth>
  <burstSize>0</burstSize>
  <enabled>true</enabled>
  <inherited>>false</inherited>
</outShapingPolicy>
</vnic>
<vnic>
  <label>vNic_1</label>
  <name>vnic1</name>
  <addressGroups />
  <mtu>1600</mtu>
  <type>trunk</type>
  <subInterfaces> <!--optional. Can be used only for sub-interface creation -->
    <subInterface>
      <isConnected>true</isConnected> <!-- optional. Default is false -->
      <name>sub1</name> <!-- optional. System has default Names. format vNic0 ...
        vNic7 -->
      <index>10</index> <!-- optional. Used only for modification of
        subInterfaces -->
      <tunnelId>12</tunnelId> <!-- Required. can be between 1-4094 but must be
        unique -->
      <vlanId>12</vlanId> <!-- Optional. Used only when vlan is to be specified.
        Both vlanId and logicalSwitchId cannot be specified . can be between 0-4094
        but must be unique -->
      <enableSendRedirects>>false</enableSendRedirects> <!-- optional. Default is
        false -->
      <mtu>1500</mtu> <!-- optional. will be defaulted to the least of all the
        trunked vnics mtu -->
      <addressGroups>
        <addressGroup>
          <primaryAddress>1.2.3.4</primaryAddress>
          <subnetMask>255.255.255.0</subnetMask>
          <subnetPrefixLength>24</subnetPrefixLength>
        </addressGroup>
      </addressGroups>
    </subInterface>
    <subInterface>
      <isConnected>true</isConnected>
      <label>vNic_11</label>
      <name>sub2</name>
      <index>11</index>
      <tunnelId>2</tunnelId>
      <logicalSwitchId>virtualwire-9</logicalSwitchId> <!-- Optional. Used only
        when network is to be specified. Both vlanId and logicalSwitchId cannot be
        specified . Any pgmoid of a network other than standard port group can be
        specified-->
      <enableSendRedirects>>false</enableSendRedirects>
      <mtu>1500</mtu>
      <addressGroups>
        <addressGroup>
          <primaryAddress>1.2.2.3</primaryAddress>
          <subnetMask>255.255.255.0</subnetMask>
          <subnetPrefixLength>24</subnetPrefixLength>
        </addressGroup>
      </addressGroups>
    </subInterface>
    <subInterface> <!-- If both vlanId and logicalSwitchId are not provided then
      it is of type NONE-->
      <isConnected>true</isConnected>
      <label>vNic_12</label>
      <name>sub3</name>
      <index>12</index>
      <tunnelId>3</tunnelId>
      <enableSendRedirects>>false</enableSendRedirects>
      <mtu>1500</mtu>
      <addressGroups />

```

```

    </subInterface>
  </subInterfaces>
  <isConnected>true</isConnected>
  <index>1</index>
  <portgroupId>dvportgroup-23</portgroupId>
  <portgroupName>Trunk</portgroupName>
  <enableProxyArp>false</enableProxyArp>
  <enableSendRedirects>false</enableSendRedirects>
</vnic>
</vnics>

```

---

where:

- `inShapingPolicy`, `outShapingPolicy` are optional. Can only be specified for a vnic connected to a distributed portgroup.
- `averageBandwidth` is a required field. Other fields are optional. If not specified, `peakBandwidth` is defaulted to `averageBandwidth`, `burstSize` is defaulted to '0', `enabled` is defaulted to 'true', inherited is defaulted to 'false'. `averageBandwidth`, `peakBandwidth` and `burstSize` values are in 'bits per second'.
- `addressGroups` contains IP addresses for the interface with each `addressGroup` representing the IP addresses within the same subnet. For each subnet, you can specify a `primaryAddress` (required), `secondaryAddress` (optional), and the `subnetMask` (required).

## Retrieve Interfaces for a Edge

Retrieves all interfaces for the specified Edge.

### Example 9-219. Retrieve all interfaces

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/vnics/
```

Response Body:

```

<vnics>
  <vnic>
    <label>vNic_0</label>
    <name>EXT</name>
    <addressGroups>
      <addressGroup>
        <primaryAddress>192.168.12.240</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
        <subnetPrefixLength>24</subnetPrefixLength>
      </addressGroup>
    </addressGroups>
    <mtu>1500</mtu>
    <type>uplink</type>
    <isConnected>true</isConnected>
    <index>0</index>
    <portgroupId>network-12</portgroupId>
    <portgroupName>VM Network</portgroupName>
    <enableProxyArp>false</enableProxyArp>
    <enableSendRedirects>false</enableSendRedirects>
  </vnic>
  <vnic>
    <label>vNic_1</label>
    <name>vnic1</name>
    <addressGroups />
    <mtu>1500</mtu>
    <type>trunk</type>
    <subInterfaces>
      <subInterface>
        <isConnected>true</isConnected>
        <label>vNic_10</label>
        <name>sub1</name>

```

```

<index>10</index>
<tunnelId>12</tunnelId>
<vlanId>12</vlanId>
<enableSendRedirects>>false</enableSendRedirects>
<mtu>1500</mtu>
<addressGroups>
  <addressGroup>
    <primaryAddress>1.2.3.4</primaryAddress>
    <subnetMask>255.255.255.0</subnetMask>
    <subnetPrefixLength>24</subnetPrefixLength>
  </addressGroup>
</addressGroups>
</subInterface>
<subInterface>
  <isConnected>>true</isConnected>
  <label>vNic_11</label>
  <name>sub2</name>
  <index>11</index>
  <tunnelId>2</tunnelId>
  <logicalSwitchId>virtualwire-9</logicalSwitchId>
  <logicalSwitchName>vw-1</logicalSwitchName>
  <enableSendRedirects>>false</enableSendRedirects>
  <mtu>1500</mtu>
  <addressGroups>
    <addressGroup>
      <primaryAddress>1.2.2.3</primaryAddress>
      <subnetMask>255.255.255.0</subnetMask>
      <subnetPrefixLength>24</subnetPrefixLength>
    </addressGroup>
  </addressGroups>
</subInterface>
<subInterface>
  <isConnected>>true</isConnected>
  <label>vNic_12</label>
  <name>sub3</name>
  <index>12</index>
  <tunnelId>3</tunnelId>
  <vlanId>0</vlanId>
  <enableSendRedirects>>false</enableSendRedirects>
  <mtu>1500</mtu>
  <addressGroups />
</subInterface>
</subInterfaces>
<isConnected>>true</isConnected>
<index>1</index>
<portgroupId>dvportgroup-23</portgroupId>
<portgroupName>Trunk</portgroupName>
<enableProxyArp>>false</enableProxyArp>
<enableSendRedirects>>false</enableSendRedirects>
</vnic>
<vnic>
  <label>vNic_2</label>
  <name>vnic2</name>
  <addressGroups />
  <mtu>1500</mtu>
  <type>internal</type>
  <isConnected>>false</isConnected>
  <index>2</index>
  <enableProxyArp>>false</enableProxyArp>
  <enableSendRedirects>true</enableSendRedirects>
</vnic>
. . .
. . .
. . .
</vnics>

```

## Retrieve Specified Interface

Retrieves the interface with specified index for an Edge.

### Example 9-220. Retrieve interface

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/vnics/index>

Response Body:

```
<vnic>
  <label>vNic_1</label>
  <name>vnic1</name>
  <addressGroups />
  <mtu>1500</mtu>
  <type>trunk</type>
  <subInterfaces>
    <subInterface>
      <isConnected>true</isConnected>
      <label>vNic_10</label>
      <name>sub1</name>
      <index>10</index>
      <tunnelId>12</tunnelId>
      <vlanId>12</vlanId>
      <enableSendRedirects>false</enableSendRedirects>
      <mtu>1500</mtu>
      <addressGroups>
        <addressGroup>
          <primaryAddress>1.2.3.4</primaryAddress>
          <subnetMask>255.255.255.0</subnetMask>
          <subnetPrefixLength>24</subnetPrefixLength>
        </addressGroup>
      </addressGroups>
    </subInterface>
    <subInterface>
      <isConnected>true</isConnected>
      <label>vNic_11</label>
      <name>sub2</name>
      <index>11</index>
      <tunnelId>2</tunnelId>
      <logicalSwitchId>virtualwire-9</logicalSwitchId>
      <logicalSwitchName>vw-1</logicalSwitchName>
      <enableSendRedirects>false</enableSendRedirects>
      <mtu>1500</mtu>
      <addressGroups>
        <addressGroup>
          <primaryAddress>1.2.2.3</primaryAddress>
          <subnetMask>255.255.255.0</subnetMask>
          <subnetPrefixLength>24</subnetPrefixLength>
        </addressGroup>
      </addressGroups>
    </subInterface>
    <subInterface>
      <isConnected>true</isConnected>
      <label>vNic_12</label>
      <name>sub3</name>
      <index>12</index>
      <tunnelId>3</tunnelId>
      <vlanId>0</vlanId>
      <enableSendRedirects>false</enableSendRedirects>
      <mtu>1500</mtu>
      <addressGroups />
    </subInterface>
  </subInterfaces>
  <isConnected>true</isConnected>
  <index>1</index>
  <portgroupId>dvportgroup-23</portgroupId>
```

```

    <portgroupName>Trunk</portgroupName>
    <enableProxyArp>>false</enableProxyArp>
    <enableSendRedirects>>false</enableSendRedirects>
  </vnic>

```

---

## Modify Specified Interface

Modifies the interface with specified index for an Edge.

### Example 9-221. Modify interface

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/vnics/index
```

Request Body:

```

<vnic>
  <name>trunk</name>  <!-- optional. System has default Names.
                      format vnic0 ... vnic7 -->
  <addressGroups>
    <addressGroup>  <!-- vnic can be configured to have more than one
                   addressGroup/subnets -->
      <primaryAddress>192.168.12.240</primaryAddress>  <!-- This is mandatory for an
      addressGroup -->
      <secondaryAddresses>  <!-- Optional. Should be used to add/defined other IPs
      used for NAT, LB, VPN, etc -->
        <ipAddress>192.168.3.2</ipAddress>
        <ipAddress>192.168.3.3</ipAddress>  <!-- Optional. This way multiple IP
      Addresses can be assigned to a vnic/interface -->
      </secondaryAddresses>
      <subnetMask>255.255.255.0</subnetMask>  <!-- either subnetMask or
      subnetPrefixLength should be provided. If both then subnetprefixLength is
      ignored -->
    </addressGroup>
    <addressGroup>  <!-- vnic can be configured to have more than one
                   addressGroup/subnets -->
      <primaryAddress>192.168.4.1</primaryAddress>  <!-- This is mandatory for an
      addressGroup -->
      <secondaryAddresses>  <!-- optional. Should be used to add/defined other IPs
      used for NAT, LB, VPN, etc -->
        <ipAddress>192.168.4.2</ipAddress>
        <ipAddress>192.168.4.3</ipAddress>  <!-- Optional. This way multiple IP
      Addresses can be assigned to a vnic/interface -->
      </secondaryAddresses>
      <subnetPrefixLength>24</subnetPrefixLength>
    </addressGroup>
    <addressGroup>  <!-- ipv6 addressGroup -->
      <primaryAddress>ffff::1</primaryAddress>  <!-- This is mandatory for an
      addressGroup -->
      <secondaryAddresses>  <!-- Optional. Should be used to add/defined other IPs
      used for NAT, LB, VPN, etc -->
        <ipAddress>ffff::2</ipAddress>
      </secondaryAddresses>
      <subnetPrefixLength>64</subnetPrefixLength>  <!-- prefixLength valid values
      1-128 -->
    </addressGroup>
  </addressGroups>
  <subInterfaces>
    <subInterface>  <!--optional. Can be used only for sub-interface creation -->
      <isConnected>>true</isConnected>  <!-- optional. Default is false -->
      <name>sub1</name>  <!-- optional. System has default Names. format vnic0 ...
      vnic7 -->
      <index>10</index>  <!-- optional. Used only for modification of
      subInterfaces -->
      <tunnelId>12</tunnelId>  <!-- Required. can be between 1-4094 but must be
      unique -->

```

```

    <vlanId>12</vlanId>  <!-- Optional. Used only when vlan is to be specified.
      Both vlanId and logicalSwitchId cannot be specified . can be between 0-4094
      but must be unique -->
    <enableSendRedirects>>false</enableSendRedirects>  <!-- optional. Default is
      false -->
    <mtu>1500</mtu>  <!-- optional. Will be defaulted to the least of all the
      trunked vnics mtu -->
    <addressGroups>
      <addressGroup>
        <primaryAddress>1.2.3.4</primaryAddress>
        <subnetMask>255.255.255.0</subnetMask>
        <subnetPrefixLength>24</subnetPrefixLength>
      </addressGroup>
    </addressGroups>
  </subInterface>
</subInterface>
  <isConnected>>true</isConnected>
  <label>vNic_11</label>
  <name>sub2</name>
  <index>11</index>
  <tunnelId>2</tunnelId>
  <logicalSwitchId>virtualwire-9</logicalSwitchId>  <!-- Optional. Used only when
    network is to be specified. Both vlanId and logicalSwitchId cannot be
    specified . Any pgmoid of a network other than standard port group can be
    specified-->
  <enableSendRedirects>>false</enableSendRedirects>
  <mtu>1500</mtu>
  <addressGroups>
    <addressGroup>
      <primaryAddress>1.2.2.3</primaryAddress>
      <subnetMask>255.255.255.0</subnetMask>
      <subnetPrefixLength>24</subnetPrefixLength>
    </addressGroup>
  </addressGroups>
</subInterface>
<subInterface>  <!-- If both vlanId and logicalSwitchId are not provided then it
  is of type NONE-->
  <isConnected>>true</isConnected>
  <label>vNic_12</label>
  <name>sub3</name>
  <index>12</index>
  <tunnelId>3</tunnelId>
  <enableSendRedirects>>false</enableSendRedirects>
  <mtu>1500</mtu>
  <addressGroups />
</subInterface>
</subInterfaces>
<mtu>1500</mtu>  <!-- optional. Default is 1600 for type "TRUNK" and 1500 for
  others-->
<type>trunk</type>  <!-- optional. Default is internal. Other possible value is
  "uplink" and "TRUNK" -->
<index>1</index>
<portgroupId>network-12</portgroupId>  <!-- Possible values here are portgroupIds or
  virtualwire-id. portgroupId needs to be defined if isConnected=true. For
  vnic of type "TRUNK" logical switch cannot be used -->
<portgroupName>VM Network</portgroupName>
<macAddress>  <!-- optional. When not specified, macAddresses will be managed
  by VC -->
  <edgeVmHaIndex>0</edgeVmHaIndex>  <!-- possible values 0 or 1 when HA is enabled
  -->
  <value>00:50:56:01:03:23</value>  <!-- optional. User must ensure that
  macAddresses provided are unique within the given layer 2 domain. -->
</macAddress>
<fenceParameter>  <!-- optional -->
  <key>ethernet0.filter1.param1</key>
  <value>1</value>
</fenceParameter>
<enableProxyArp>>false</enableProxyArp>  <!-- optional. Default is false -->

```

```

<enableSendRedirects>true</enableSendRedirects>  <!-- optional. Default is true -->
<enableBridgeMode>>false</enableBridgeMode>  <!-- optional. Default is false -->
<isConnected>true</isConnected>  <!-- optional. Default is false -->
<inShapingPolicy>  <!-- optional -->
  <averageBandwidth>200000000</averageBandwidth>
  <peakBandwidth>200000000</peakBandwidth>
  <burstSize>0</burstSize>
  <enabled>true</enabled>
  <inherited>>false</inherited>
</inShapingPolicy>
<outShapingPolicy>  <!-- optional -->
  <averageBandwidth>400000000</averageBandwidth>
  <peakBandwidth>400000000</peakBandwidth>
  <burstSize>0</burstSize>
  <enabled>true</enabled>
  <inherited>>false</inherited>
</outShapingPolicy>
</vnic>

```

---

### Delete Interfaces

Resets the interface with the specified index to factory defaults.

#### Example 9-222. Delete interface

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/vnics/index
```

---

## Manage a Edge Interface

You can manage a specific Edge interface.

### Retrieve Interface with Specific Index

Retrieves the interface with specified index for a Edge.

#### Example 9-223. Get interface with specific index

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/vnics/index
```

Response Body:

```

<vnic>
  <index>0</index>
  <name>uplink-vnic-network-2581</name>
  <type>uplink</type>
  <portgroupId>network-2581</portgroupId>
  <portgroupName>Mgmt</portgroupName>
  <addressGroups>
    <addressGroup>
      <primaryAddress>192.168.3.1</primaryAddress>
      <secondaryAddresses>
        <ipAddress>192.168.3.2</ipAddress>
        <ipAddress>192.168.3.3</ipAddress>
      </secondaryAddresses>
      <subnetMask>255.255.255.0</subnetMask>
    </addressGroup>
    <addressGroup>
      <primaryAddress>192.168.4.1</primaryAddress>
      <secondaryAddresses>
        <ipAddress>192.168.4.2</ipAddress>
        <ipAddress>192.168.4.3</ipAddress>
      </secondaryAddresses>
    </addressGroup>
  </addressGroups>
</vnic>

```

```

        </secondaryAddresses>
        <subnetMask>255.255.255.0</subnetMask>  <!-- GET will always have subnetMask
            field for ipv4 and subnetPrefixLength for ipv6 -->
    </addressGroup>
    <addressGroup>
        <primaryAddress>ffff::1</primaryAddress>
        <secondaryAddresses>
            <ipAddress>ffff::2</ipAddress>
        </secondaryAddresses>
        <subnetPrefixLength>64</subnetPrefixLength>
    </addressGroup>
</addressGroups>
<mtu>1500</mtu>
<enableProxyArp>>false</enableProxyArp>
<enableSendRedirects>>true</enableSendRedirects>
<isConnected>>true</isConnected>
</vnic>

```

---

## Modify an Interface

Modifies the specified interface.

### Example 9-224. Modify interface

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/vnics/index>

Request Body:

```

<vnic>
  <index>0</index>  <!-- optional. System has default Names. format vnic0 ... vnic7 -->
  <name>uplink-vnic-network-2581</name>  <!-- optional. Default is internal>
  <type>uplink</type>
  <portgroupId>network-2581</portgroupId>  <!-- Possible values are portgroupIds or
            virtualwireId. portgroupId needs to be defined if isConnected=true -->
  <addressGroups>
    <addressGroup>  <!-- vnic can be configured to have more than one
                    addressGroup/subnets -->
      <primaryAddress>10.112.2.40</primaryAddress>  <!-- This is mandatory for an
                    addressGroup -->
      <secondaryAddresses>  <!-- Optional. Should be used to add/defined other IPs
                    used for NAT, LB, VPN, etc -->
        <ipAddress>10.112.2.42</ipAddress>
      </secondaryAddresses>
      <subnetMask>255.255.254.0</subnetMask>
    </addressGroup>
  </addressGroups>
  <macAddress>  <!-- optional. When not specified, macAddresses will be managed
                by VC -->
    <edgeVmHaIndex>0</edgeVmHaIndex>
    <value>00:50:56:01:03:23</value>
  </macAddress>
  <fenceParameter>  <!-- optional -->
    <key>ethernet0.filter1.param1</key>
    <value>1</value>
  </fenceParameter>
  <mtu>1500</mtu>  <!-- Default is 1500.-->
  <enableProxyArp>>false</enableProxyArp>  <!--Default is false.-->
  <enableSendRedirects>>true</enableSendRedirects>  <!--Default is true.-->
  <isConnected>true</isConnected>  <!--Default is false.-->
  <inShapingPolicy>  <!-- optional -->
    <averageBandwidth>200000000</averageBandwidth>
    <peakBandwidth>200000000</peakBandwidth>
    <burstSize>0</burstSize>
    <enabled>true</enabled>
    <inherited>>false</inherited>

```

```

</inShapingPolicy>
<outShapingPolicy>  <!-- optional -->
  <averageBandwidth>400000000</averageBandwidth>
  <peakBandwidth>400000000</peakBandwidth>
  <burstSize>0</burstSize>
  <enabled>true</enabled>
  <inherited>>false</inherited>
</outShapingPolicy>
</vnic>

```

---

### Delete Interface Configuration

Deletes the interface configuration and resets it to the factory default.

**Example 9-225.** Delete interface configuration

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/vnics/index
```

---

## Query Interface Statistics

### Query Statistics for all Interfaces

Retrieves statistics for all configured interfaces between the specified start and end times. When start and end time are not specified, all statistics since the Edge deployed are displayed. When no end time is specified, the current Edge Manager time is set as endTime. Each record has the stats of 5 minutes granularity.

**Example 9-226.** Get interface statistics

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/statistics/interfaces
```

Response Body:

```

<statistics>
  <meta>
    <startTime>1336068000</startTime>  <!-- in seconds -->
    <endTime>1336100700</endTime>  <!-- in seconds -->
    <interval>300</interval>  <!-- 5 mins interval -->
  </meta>
  <data>
    <statistic>
      <vnic>0</vnic>
      <timestamp>1336068000</timestamp>
      <in>9.1914285714e+02</in>  <!-- Rx rate ( Kilobits per second - kbps ) -->
      <out>5.1402857143e+02</out>  <!-- Tx rate ( Kilobits per second - kbps ) -->
    </statistic>
    ...
    ...
    <statistic>
      <vnic>1</vnic>
      <timestamp>1336100700</timestamp>
      <in>9.2914285714e+02</in>
      <out>5.2402857143e+02</out>
    </statistic>
  </data>
</statistics>

```

---

## Query Statistics for Uplink Interfaces

Retrieves statistics for all uplink interfaces between the specified start and end times. When start and end time are not specified, all statistics since the Edge deployed are displayed. When no end time is specified, the current Edge Manager time is set as endTime. Each record has the stats of 5 minutes granularity.

### Example 9-227. Get uplink interface statistics

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/statistics/interfaces/uplink>

Response Body:

```
<statistics>
  <meta>
    <startTime>1336068000</startTime>  <!-- in seconds -->
    <endTime>1336100700</endTime>  <!-- in seconds -->
    <interval>300</interval>  <!-- 5 mins interval -->
  </meta>
  <data>
    <statistic>
      <vnic>0</vnic>
      <timestamp>1336068000</timestamp>
      <in>9.1914285714e+02</in>  <!-- Rx rate ( Kilobits per second - kbps ) -->
      <out>5.1402857143e+02</out>  <!-- Tx rate ( Kilobits per second - kbps ) -->
    </statistic>
    ...
    <statistic>
      <vnic>1</vnic>
      <timestamp>1336100700</timestamp>
      <in>9.2914285714e+02</in>
      <out>5.2402857143e+02</out>
    </statistic>
  </data>
</statistics>
```

---

## Query Statistics for Internal Interfaces

Retrieves statistics for all internal interfaces between the specified start and end times. When start and end time are not specified, all statistics since the Edge deployed are displayed. When no end time is specified, the current Edge Manager time is set as endTime. Each record has the stats of 5 minutes granularity.

### Example 9-228. Get internal interface statistics

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/statistics/interfaces/internal>

Response Body:

```
<statistics>
  <meta>
    <startTime>1336068000</startTime>  <!-- in seconds -->
    <endTime>1336100700</endTime>  <!-- in seconds -->
    <interval>300</interval>  <!-- 5 mins interval -->
  </meta>
  <data>
    <statistic>
      <vnic>0</vnic>
      <timestamp>1336068000</timestamp>
      <in>9.1914285714e+02</in>  <!-- Rx rate ( Kilobits per second - kbps ) -->
      <out>5.1402857143e+02</out>  <!-- Tx rate ( Kilobits per second - kbps ) -->
    </statistic>
    ...
```

```

...
<statistic>
  <vnic>1</vnic>
  <timestamp>1336100700</timestamp>
  <in>9.2914285714e+02</in>
  <out>5.2402857143e+02</out>
</statistic>
</data>
</statistics>

```

---

## Query Dashboard Statistics

Retrieves dashboard statistics between the specified start and end times. When start and end time are not specified, all statistics since the Edge deployed are displayed. When no end time is specified, the current Edge Manager time is set as endTime. Each record has the stats of 5 minutes granularity.

### Example 9-229. Get interface statistics

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId/statistics/dashboard
/interface?interval=range
```

Response Body:

```

<dashboardstatistics>
  <meta>
    <startTime>1336068000</startTime>  <!-- in seconds -->
    <endTime>1336100700</endTime>  <!-- in seconds -->
    <interval>300</interval>  <!-- 5 mins interval -->
  </meta>
  <data>
    <interfaces>
      <vnic_0_in_pkt>
        <dashboardStatistic>
          <timestamp></timestamp>
          <value></value>
        </dashboardStatistic>
        <dashboardStatistic>
          <timestamp></timestamp>
          <value></value>
        </dashboardStatistic>
        ...
      </vnic_0_in_pkt>
      ...
    </interfaces>
  </data>
</data>
</dashboardstatistics>

```

---



# Firewall Management

Distributed Firewall is a hypervisor kernel-embedded firewall that provides visibility and control for virtualized workloads and networks. You can create access control policies based on VMware vCenter objects like datacenters and clusters, virtual machine names and tags, network constructs like IP/VLAN/VXLAN addresses, as well as user group identity from Active Directory. Firewall rules are enforced at the vNIC level of each virtual machine to provide consistent access control even when the virtual machine gets vMotioned. The hypervisor-embedded nature of the firewall delivers close to line rate throughput to enable higher workload consolidation on physical servers. The distributed nature of the firewall provides a scale-out architecture that automatically extends firewall capacity when additional hosts are added to a prepared cluster.

Distributed Firewall offers multiple sets of configurable rules: Layer 3 (L3) rules (General tab), Layer 2 (L2) rules (Ethernet tab), and Layer 3 Redirect (Partner security services tab). Layer 2 firewall rules are processed before Layer 3 rules.

The default firewall rule allows all L3 and L2 traffic to pass through all clusters in your infrastructure. The default rule is always at the bottom of the rules table and cannot be deleted or added to. However, you can change the Action element of the rule from Allow to Block, add comments for the rule, and indicate whether traffic for that rule should be logged.

If you have a partner service deployed in your environment, you can redirect traffic to the partner service by adding rules in the Layer 3 Redirect section.

Edge Firewall provides perimeter security functionality including firewall, Network Address Translation (NAT) as well as Site to site IPSec and SSL VPN functionality. This solution is available in the virtual machine form factor and can be deployed in a High Availability mode.

Distributed Firewall rules and Edge Firewall rules can be managed in a centralized manner. You create rules at the global level and then narrow the scope at which you want to apply the rule by using the AppliedTo keyword. This is particularly useful in a multi-tenant environment where providers can define high-level traffic flow rules (also referred to as pre rules). Tenants can then add rules at an individual NSX Edge level, which are referred to as local rules.

The following table lists the elements that can be used in firewall rules.

**Table 10-1.** Firewall rule elements

Element	Keyword for API	Used In
All Edges	ALL_EDGES	appliedTo
application	Application	service
application group	ApplicationGroup	service
cluster compute resource	ClusterComputeResource	appliedTo
datacenter	Datacenter	source/destination appliedTo
distributed firewall	DISTRIBUTED_FIREWALL	appliedTo

**Table 10-1.** Firewall rule elements

Element	Keyword for API	Used In
distributed virtual port group	DistributedVirtualPortgroup	source/destination appliedTo
Edge ID	Edge	appliedTo
global root	GlobalRoot	source/destination
host	HostSystem	appliedTo
IP set	IPSet	source/destination
IPv4 addresses	Ipv4Address	source/destination
IPv6 addresses	Ipv6Address	source/destination
logical switch	VirtualWire	source/destination appliedTo
MAC set	MACSet	source/destination
network	Network	for legacy portgroups, network can be used in source or destination instead of appliedTo
profile	ALL_PROFILE_BINDINGS	
resource pool	ResourcePool	source/destination
security group	SecurityGroup	source/destination
virtual app	VirtualApp	source/destination
virtual machine	VirtualMachine	source/destination appliedTo
vNIC	Vnic	source/destination appliedTo

For information on creating an IPSet, see [“Working with IPsets”](#) on page 100. For information on creating a security group, see [“Working with Security Groups”](#) on page 93.

Distributed firewall can help in creating identity-based rules as well. Administrators can enforce access control based on the user's group membership as defined in the enterprise Active Directory. Here are some scenarios where identity-based firewall rules can be used:

- User accessing virtual applications using a laptop/mobile device where AD is used for user authentication
- User accessing virtual applications using VDI infrastructure where the virtual machines are Microsoft Windows based

For information about Edge firewall, see [“Working with NSX Edge Firewall”](#) on page 172.

This chapter includes the following topics:

- [“Configuring Firewall”](#) on page 297
- [“Working with Firewall Sections”](#) on page 301
- [“Working with Firewall Rules”](#) on page 306
- [“Working with Layer3 Redirect Sections and Rules”](#) on page 310
- [“Query Status”](#) on page 322
- [“Working with Memory and CPU Thresholds”](#) on page 324
- [“Tuning Firewall Performance”](#) on page 325
- [“Synchronizing and Enabling Firewall”](#) on page 326

- [“Importing and Exporting Firewall Configurations”](#) on page 327
- [“Working with SpoofGuard”](#) on page 331
- [“Getting Flow Statistic Details”](#) on page 335
- [“Flow Exclusion”](#) on page 339
- [“Working with IPFix”](#) on page 341
- [“Excluding Virtual Machines from Firewall Protection”](#) on page 342
- [“Where memberId is the vc-moref-id of a virtual machine.”](#) on page 343

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Configuring Firewall

The firewall table includes one section by default that contains the default rule. You can add additional sections to segregate firewall rules.

Each traffic session is checked against the top rule in the Firewall table before moving down the subsequent rules in the table. The first rule in the table that matches the traffic parameters is enforced.

## Query Firewall Configuration

You can retrieve the full firewall configuration consisting of all rules that has been defined on the NSX Manager. All L2, L3, and L3 redirect rule types are returned.

When Distributed Firewall is used with Service Composer, firewall sections and rules created by Service Composer contain an additional attribute in the XML called managedBy.

### Example 10-1. Get firewall configuration for NSX Manager

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
```

Response Body:

```
<firewallConfiguration timestamp="1360144793284">
  <contextId>globalroot-0</contextId>
  <layer3Sections>
    <section id="2" name="defaultSectionLayer3" generationNumber="1360144793284"
      timestamp="1360144793284">
      <rule id="2" disabled="false" logged="false">
        <name>Default Rule</name>
        <action>DENY</action>
        <appliedToList>
          <appliedTo>
            <name>DISTRIBUTED_FIREWALL</name>
            <value>DISTRIBUTED_FIREWALL</value>
            <type>DISTRIBUTED_FIREWALL</type>
            <isValid>true</isValid>
          </appliedTo>
        </appliedToList>
        <sectionId>2</sectionId>
      </rule>
    </section>
  </layer3Sections>
  <layer2Sections>
    <section id="1" name="defaultSectionLayer2" generationNumber="1360144793284"
      timestamp="1360144793284">
      <rule id="1" disabled="false" logged="false">
        <name>Default Rule</name>
        <action>ALLOW</action>
```

```

        <appliedToList>
          <appliedTo>
            <name>DISTRIBUTED_FIREWALL</name>
            <value>DISTRIBUTED_FIREWALL</value>
            <type>DISTRIBUTED_FIREWALL</type>
            <isValid>true</isValid>
          </appliedTo>
        </appliedToList>
      <sectionId>1</sectionId>
    </rule>
  </section>
</layer2Sections>
</firewallConfiguration>

```

---

## Filter Firewall Configuration

You can use a wide number of criteria to filter your ruleset, which allows for easy rule modification. Rules can be filtered by source or destination virtual machines, rule action, logging, rule name, comments, and rule ID.

### Example 10-2. Filter firewall configuration

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config?ruleType=LAYER3&source=10.24.106.52&destination=vm-61&enable=true&logging=any&action=allow&ruleId=1013&comment=cluster&name=default
```

---

where the filtering criteria can be one or more of the following.

- ruleType can be LAYER3, LAYER2, L3REDIRECT. Currently filtering is not supported for L2 rules, so, specifying request params for that rule type will return all rule types. This parameter is mandatory if any of the other parameters is specified.
- source/destination can contain IPv4/v6 address or vm-id.
- ruleId is the rule ID.
- comment can contain any portion of the comment entered for the rules. Search is case insensitive.
- name can contain any portion of the rule name entered for the rules. Search is case insensitive.
- siProfile can contain any portion of the Service Profile name associated with L3 redirect rule. Search is case insensitive.
- edgeId is the Edge ID, that displays rules applicable to the specified Edge.

### Example 10-3. Filter rules for an Edge

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config?edgeId=edgeId
```

---

## Modify Firewall Configuration

Follow the procedure below to modify the firewall configuration.

- 1 Run a GET call for the firewall configuration.
- 2 Extract the XML from the response body of the GET call and modify it as required.
- 3 From the Response Header in Step 1, copy the Etag header value.

```

1. Status Code      : 200 OK
2. Cache-Control   : private, no-cache
3. Content-Type    : application/xhtml+xml
4. Date            : Tue, 01 Apr 2014 20:04:58 GMT
5. Etag            : 1396034461743
6. Expires         : Thu, 01 Jan 1970 00:00:00 GMT
7. Server          : NSX Manager

```

- 4 Add the number as the If-Match header in the PUT call.



- 5 Pass the modified XML as the Request Body in a PUT call.

- Not all fields are required while sending the request. Refer to the optional field in the schema definition of various objects. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP Set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.
- IDs for new objects (rule/section) should be removed or set to zero.
- If new entities (sections/rules) have been sent in the request, the response will contain the system-generated ids, which are assigned to these new entities. These ID identifies the resource and can be used in the urls if you want to operate on these entities using those URLs.

#### Example 10-4. Modify firewall configuration

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
--header 'Content-Type:text/xml' --header 'if-match:"1396034461743"
```

Request Body:

```
<firewallConfiguration timestamp="1359979620727">
  <contextId>globalroot-0</contextId>
  <layer3Sections>
    <section id="2" name="defaultSectionLayer3" generationNumber="1359979620727"
      timestamp="1359979620727">
      <rule disabled="false" logged="true">
        <name>okn-1</name>
        <action>ALLOW</action>
        <sources excluded="false">
          <source>
            <value>datacenter-57</value>
            <type>Datacenter</type>
          </source>
          <source>
            <value>domain-c62</value>
            <type>ClusterComputerResource</type>
          </source>
          <source>
            <value>10.112.1.1</value>
            <type>Ipv4Address</type>
          </source>
        </sources>
        <services>
          <service>
            <destinationPort>80</destinationPort>
            <protocol>6</protocol>
            <subProtocol>6</subProtocol>
          </service>
        </services>
      </rule>
    </section>
  </layer3Sections>
</firewallConfiguration>
```

```

    </service>
    <service>
      <value>application-161</value>
      <type>Application</type>
    </service>
  </services>
  <appliedToList>
    <appliedTo>
      <value>5013bcd8-c666-1e28-c7a9-600da945954f.000</value>
      <type>Vnic</type>
    </appliedTo>
    <appliedTo>
      <value>vm-126</value>
      <type>VirtualMachine</type>
    </appliedTo>
  </appliedToList>
</rule>
<rule disabled="true" logged="true">
  <name>Matru-1</name>
  <action>ALLOW</action>
  <sectionId>2</sectionId>
</rule>
<rule disabled="true" logged="true">
  <name>Matru-2</name>
  <action>ALLOW</action>
  <sectionId>2</sectionId>
</rule>
<rule disabled="true" logged="true">
  <name>Matru-3</name>
  <action>ALLOW</action>
  <sectionId>2</sectionId>
</rule>
<rule id="2" disabled="true" logged="false">
  <name>Default Rule</name>
  <action>DENY</action>
  <sectionId>2</sectionId>
</rule>
</section>
</layer3Sections>
<layer2Sections>
  <section id="1" name="defaultSectionLayer2" generationNumber="1359979620727"
    timestamp="1359979620727">
    <rule id="1" disabled="false" logged="false">
      <name>Default Rule</name>
      <action>ALLOW</action>
      <sectionId>1</sectionId>
    </rule>
  </section>
</layer2Sections>
</firewallConfiguration>

```

---

where:

- appliedTo can a valid object from [“Firewall rule elements”](#) on page 295.
- action can be ALLOW, BLOCK, or REJECT. REJECT sends reject message for unaccepted packets; RST packets are sent for TCP connections and ICMP unreachable code packets are sent for UDP, ICMP, and other IP connections
- source and destination can have an exclude flag. For example, if you add an exclude tag for 1.1.1.1 in the source parameter, the rule looks for traffic originating from all IPs other than 1.1.1.1.

## Delete Firewall Configuration

Restores default configuration, which means one defaultLayer3 section with default allow rule and one defaultLayer2Section with default allow rule.

**Example 10-5.** Delete firewall configuration

Request:

DELETE <https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config>

## Working with Firewall Sections

You can use sections in the firewall table to group logical rules based on AppliedTo or for a tenant use case. A firewall section is the smallest unit of configuration which can be updated independently. Section types are as follows:

- Layer3Section contains layer3 rules
- Layer2Section contains layer2 rules
- Layer3RedirectSection contains traffic redirect rules. For more information, see [“Working with Layer3 Redirect Sections and Rules”](#) on page 310.

When Distributed Firewall is used with Service Composer, firewall sections created by Service Composer contain an additional attribute in the XML called managedBy. You should not modify Service Composer firewall sections using Distributed Firewall REST APIs. If you do, you must synchronize firewall rules from Service Composer. For more information, see [“Synchronizing Service Composer Rules with Distributed Firewall”](#) on page 394.

## Query Firewall Sections

Retrieves section configuration either by section ID or section name.

**Example 10-6.** Get section configuration by section name or section ID

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections>  
\layer2sections/sectionId

OR

GET <https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections>  
\layer2sections?name=sectionName

Response Body:

```
<section id="4" name="TestSection" generationNumber="1360149234572"
  timestamp="1360149234572">
  <rule id="16" disabled="false" logged="true">
    <name>okn-2</name>
    <action>ALLOW</action>
    <appliedToList>
      <appliedTo>
        <name>vm1 - Network adapter 1</name>
        <value>5013bcd8-c666-1e28-c7a9-600da945954f.000</value>
        <type>vnic</type>
        <isValid>true</isValid>
      </appliedTo>
      <appliedTo>
        <name>Small XP-2</name>
        <value>vm-126</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>4</sectionId>
    <sources excluded="false">
      <source>
        <name>5.1 ESX</name>
```

```

        <value>datacenter-57</value>
        <type>Datacenter</type>
        <isValid>true</isValid>
    </source>
    <source>
        <name>5.1</name>
        <value>domain-c62</value>
        <type>ClusterComputeResource</type>
        <isValid>true</isValid>
    </source>
    <source>
        <value>10.112.1.1</value>
        <type>Ipv4Address</type>
        <isValid>true</isValid>
    </source>
</sources>
<services>
    <service>
        <destinationPort>80</destinationPort>
        <protocol>6</protocol>
        <subProtocol>6</subProtocol>
    </service>
    <service>
        <name>VMware-VDM2.x-Ephemeral</name>
        <value>application-161</value>
        <isValid>true</isValid>
    </service>
</services>
<appliedToList>
    <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
    </appliedTo>
</appliedToList>
</rule>
<rule id="15" disabled="true" logged="true">
    <name>Matru-3</name>
    <action>ALLOW</action>
    <appliedToList>
        <appliedTo>
            <name>DISTRIBUTED_FIREWALL</name>
            <value>DISTRIBUTED_FIREWALL</value>
            <type>DISTRIBUTED_FIREWALL</type>
            <isValid>true</isValid>
        </appliedTo>
    </appliedToList>
    <sectionId>4</sectionId>
</rule>
<rule id="14" disabled="true" logged="true">
    <name>test-3</name>
    <action>ALLOW</action>
    <appliedToList>
        <appliedTo>
            <name>DISTRIBUTED_FIREWALL</name>
            <value>DISTRIBUTED_FIREWALL</value>
            <type>DISTRIBUTED_FIREWALL</type>
            <isValid>true</isValid>
        </appliedTo>
    </appliedToList>
    <sectionId>4</sectionId>
</rule>
<rule id="13" disabled="true" logged="true">
    <name>test-2</name>
    <action>ALLOW</action>
    <appliedToList>
        <appliedTo>

```

```

        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
    </appliedTo>
</appliedToList>
<sectionId>4</sectionId>
</rule>
<rule id="12" disabled="true" logged="false">
    <name>test-1</name>
    <action>DENY</action>
    <appliedToList>
        <appliedTo>
            <name>DISTRIBUTED_FIREWALL</name>
            <value>DISTRIBUTED_FIREWALL</value>
            <type>DISTRIBUTED_FIREWALL</type>
            <isValid>true</isValid>
        </appliedTo>
    </appliedToList>
    <sectionId>4</sectionId>
</rule>
</section>

```

---

## Add Firewall Section

Adds a section at the top of the firewall table.

### Example 10-7. Add section

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections
\layer2sections
```

Request Body:

```

<section name="TestSection">
  <rule disabled="false" logged="true">
    <name>okn-2</name>
    <action>ALLOW</action>
    <appliedToList>
      <appliedTo>
        <name>vm1 - Network adapter 1</name>
        <value>5013bcd8-c666-1e28-c7a9-600da945954f.000</value>
        <type>vnic</type>
        <isValid>true</isValid>
      </appliedTo>
      <appliedTo>
        <name>Small XP-2</name>
        <value>vm-126</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sources excluded="false">
      <source>
        <name>5.1 ESX</name>
        <value>datacenter-57</value>
        <type>Datacenter</type>
        <isValid>true</isValid>
      </source>
      <source>
        <name>5.1</name>
        <value>domain-c62</value>
        <type>ClusterComputerResource</type>
        <isValid>true</isValid>
      </source>
    </sources>
  </rule>
</section>

```

```

    <source>
      <value>10.112.1.1</value>
      <type>Ipv4Address</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <services>
    <service>
      <destinationPort>80</destinationPort>
      <protocol>6</protocol>
      <subProtocol>6</subProtocol>
    </service>
    <service>
      <name>VMware-VDM2.x-Ephemeral</name>
      <value>application-161</value>
      <isValid>true</isValid>
    </service>
  </services>
</rule>
<rule disabled="true" logged="true">
  <name>Matru-3</name>
  <action>ALLOW</action>
</rule>
<rule disabled="true" logged="true">
  <name>test-3</name>
  <action>ALLOW</action>
</rule>
<rule disabled="true" logged="true">
  <name>test-2</name>
  <action>ALLOW</action>
</rule>
<rule disabled="true" logged="false">
  <name>test-1</name>
  <action>DENY</action>
</rule>
</section>

```

Location Header in the response body contains the resource url for the newly created rule resource. This URL can be used to identify this resource.

## Modify Firewall Section

Follow the procedure below to modify a firewall section.

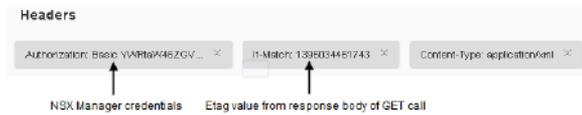
- 1 Run a GET call for the firewall section.
- 2 Extract the XML from the response body of the GET call and modify it as required.
- 3 From the Response Header in Step 1, copy the Etag header value.

```

1.  Status Code      : 200 OK
2.  Cache-Control   : private, no-cache
3.  Content-Type    : application/xhtml+xml
4.  Date            : Tue, 01 Apr 2014 20:04:58 GMT
5.  Etag            : 1395034461743
6.  Expires         : Thu, 01 Jan 1970 00:00:00 GMT
7.  Server          : NSX Manager

```

- 4 Add the number as the If-Match header in the PUT call.



5 Pass the modified XML as the Request Body in a PUT call.

- Not all fields are required while sending the request. Refer to the optional field in the schema definition of various objects. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP Set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.
- ID for new section should be removed or set to zero.
- If new sections have been sent in the request, the response will contain the system-generated ids, which are assigned to these new sections. These ID identifies the resource and can be used in the URLs if you want to operate on these entities using those URLs.

#### Example 10-8. Modify section

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
    /layer3sections|layer2sections/sectionId/sectionName
    --header 'Content-Type:text/xml' --header 'if-match:"1396034461743"
```

Request Body:

```
<section id="4" name="TestSectionRenamed" generationNumber="1336034461743"
    timestamp="1360149234572">
  <rule id="16" disabled="false" logged="false">
    <name>okn-2</name>
    <action>ALLOW</action>
    <appliedToList>
      <appliedTo>
        <name>vm1 - Network adapter 1</name>
        <value>5013bcd8-c666-1e28-c7a9-600da945954f.000</value>
        <type>Vnic</type>
        <isValid>true</isValid>
      </appliedTo>
      <appliedTo>
        <name>Small XP-2</name>
        <value>vm-126</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>4</sectionId>
    <sources excluded="false">
      <source>
        <name>5.1 ESX</name>
        <value>datacenter-57</value>
        <type>Datacenter</type>
        <isValid>true</isValid>
      </source>
      <source>
        <name>5.1</name>
        <value>domain-c62</value>
        <type>ClusterComputeResource</type>
        <isValid>true</isValid>
      </source>
      <source>
        <value>10.112.1.1</value>
        <type>Ipv4Address</type>
        <isValid>true</isValid>
      </source>
    </sources>
  </services>
```

```

    <service>
      <destinationPort>80</destinationPort>
      <protocol>6</protocol>
      <subProtocol>6</subProtocol>
    </service>
  </services>
</rule>
<rule id="15" disabled="true" logged="true">
  <name>Matru-3</name>
  <action>DENY</action>
  <sectionId>4</sectionId>
</rule>
<rule id="14" disabled="true" logged="true">
  <name>test-3</name>
  <action>ALLOW</action>
  <sectionId>4</sectionId>
</rule>
<rule id="13" disabled="true" logged="true">
  <name>test-2</name>
  <action>ALLOW</action>
  <sectionId>4</sectionId>
</rule>
<rule id="12" disabled="true" logged="false">
  <name>test-1</name>
  <action>DENY</action>
  <sectionId>4</sectionId>
</rule>
</section>

```

---

## Delete Firewall Section

Deletes the specified section. If the section contains a default rule, the section is not deleted but all rules except for the default rule are removed from that section.

If the section does not contain a default rule, the section and all its rules are deleted.

### Example 10-9. Delete section

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections
\layer2sections/sectionId
```

---

## Working with Firewall Rules

You add firewall rules at the global scope. You can then narrow down the scope (datacenter, cluster, distributed virtual port group, network, virtual machine, vNIC, or logical switch) at which you want to apply the rule. Firewall allows you to add multiple objects at the source and destination levels for each rule, which helps reduce the total number of firewall rules to be added.

To add a identity based firewall rule, first create a security group based on Directory Group objects. Then create a firewall rule with the security group as the source or destination.

Rules that direct traffic to a third part service are referred to as layer3 redirect rules, and are displayed in the layer3 redirect tab.

When Distributed Firewall is used with Service Composer, firewall rules created by Service Composer contain an additional attribute in the XML called managedBy.

## Query Firewall Rule

Retrieves rule details from either a Layer3 or Layer2 section.

### Example 10-10. Get firewall rule

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections
\layer2sections/sectionId/rules/ruleId
```

Response Body:

```
<rule id="1807" disabled="false" logged="true">
  <name>Section-2-Rule-1</name>
  <action>allow</action>
  <notes>Example with multile sources and any appliedTo with source containing vnics and
    raw-ips</notes>
  <sources excluded="false">
    <source>
      <value>10.112.1.0-10.112.1.10</value>
      <type>Ipv4Address</type>
      <isvalid>true</isvalid>
    </source>
    <source>
      <name>2-rhel53-srv-32-local-129-fa110b77-c303-4113-ab66-88c5ed9a5177 - Network
        adapter 1</name>
      <value>fa110b77-c303-4113-ab66-88c5ed9a5177.000</value>
      <type>Vnic</type>
      <isvalid>true</isvalid>
    </source>
    <source>
      <value>192.168.1.1</value>
      <type>Ipv4Address</type>
      <isvalid>true</isvalid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>1-datacenter-129</name>
      <value>datacenter-237</value>
      <type>Datacenter</type>
      <isvalid>true</isvalid>
    </destination>
  </destinations>
  <services>
    <service>
      <name>AD Server</name>
      <value>application-256</value>
      <type>Application</type>
      <isvalid>true</isvalid>
    </service>
  </services>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isvalid>true</isvalid>
    </appliedTo>
  </appliedToList>
</rule>
```

---

## Add Firewall Rule

Adds a rule at the top of the existing configuration in a Layer2 or Layer3 section.

Follow the procedure below to add a rule. In this procedure, you will retrieve another rule and use its contents as a template to create a new rule.

- 1 Run a GET call for the firewall section to add a rule to. To add a rule to the top most section, run a GET call for the complete firewall configuration.
- 2 Extract the XML from the response body of the GET call and modify it as required.
- 3 From the Response Header in Step 1, copy the Etag header value. Each section contains its own Etag, GenerationNumber, and timestamp. When adding a new rule, *you must use the Etag value of the firewall section to which you wish to add the rule.*

```

1. Status Code      : 200 OK
2. Cache-Control   : private, no-cache
3. Content-Type    : application/xhtml+xml
4. Date           : Tue, 01 Apr 2014 20:04:58 GMT
5. Etag           : 1395034461743
6. Expires        : Thu, 01 Jan 1970 00:00:00 GMT
7. Server         : NSX Manager
  
```

- 4 Add the Etag number as the If-Match header in the POST call.



- 5 Pass the modified XML as the Request Body in a POST call.
  - Not all fields are required while sending the request. Refer to the optional field in the schema definition of various objects. All the optional fields are safe to be ignored while sending the rules configuration to server. For example, if an IP Set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.
  - IDs for new rule should be removed or set to zero.

If new rules have been sent in the request, the response will contain the system-generated ids, which are assigned to these new rules. These ID identifies the resource and can be used in the URLs if you want to operate on these entities using those URLs.

#### Example 10-11. Add firewall rule

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections
\layer2sections/sectionId/rules
--header 'Content-Type:text/xml' --header 'if-match:"1380747467905"'
```

Request Body:

```
<rule disabled="false" logged="false">
  <name>AddRuleTest</name>
  <action>allow</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <value>datacenter-26</value>
      <type>Datacenter</type>
    </appliedTo>
  </appliedToList>
  <sectionId>2</sectionId>
  <sources excluded="true">
    <source>
      <value>datacenter-26</value>
    </source>
  </sources>
</rule>
```

```

        <type>Datacenter</type>
    </source>
</sources>
<services>
    <service>
        <value>application-216</value>
    </service>
</services>
</rule>

```

Location Header in the response body contains the resource url for the newly created rule resource. This URL can be used to identify this resource.

## Modify Firewall Rule

Modifies a rule in the Layer2 or Layer3 section. Follow the procedure below to modify a rule.

- 1 Run a GET call for the firewall rules.
- 2 Extract the XML from the response body of the GET call and modify it as required.
- 3 From the Response Header in Step 1, copy the Etag header value. Important: *This is the Etag value of the firewall section to which you wish to add the rule.* If you are keeping this rule in the same section, you must keep the same Etag number.

```

1.  Status Code      : 200 OK
2.  Cache-Control   : private, no-cache
3.  Content-Type    : application/xhtml+xml
4.  Date            : Tue, 01 Apr 2014 20:04:58 GMT
5.  Etag            : 1395034461743
6.  Expires         : Thu, 01 Jan 1970 00:00:00 GMT
7.  Server          : NSX Manager

```

- 4 Add the number as the If-Match header in the PUT call.



- 5 Pass the modified XML as the Request Body in a PUT call.
  - Not all fields are required while sending the request. Refer to the optional field in the schema definition of various objects. All the optional fields are safe to be ignored while sending the rules configuration to server. For example, if an IP Set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.
  - IDs for new rules should be removed or set to zero.
  - If new rules have been sent in the request, the response will contain the system-generated ids, which are assigned to these new rules. These ID identifies the resource and can be used in the URLs if you want to operate on these entities using those URLs.

### Example 10-12. Modify firewall rule

Request:

```

PUT https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections\layer2sections/sectionId/rules/ruleId
--header 'Content-Type:text/xml' --header 'if-match:"1380747467905"'

```

Request Body:

```

<rule id="23" disabled="enabled" logged="true">
  <name>AddRuleTestUpdated</name>
  <action>allow</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <value>datacenter-26</value>
      <type>Datacenter</type>
    </appliedTo>
  </appliedToList>
  <sectionId>2</sectionId>
  <sources excluded="true">
    <source>
      <value>datacenter-26</value>
      <type>Datacenter</type>
    </source>
  </sources>
  <services>
    <service>
      <value>application-216</value>
    </service>
  </services>
</rule>

```

---

Location Header in the response body contains the resource url for the newly created rule resource. This URL can be used to identify this resource.

## Delete Firewall Rule

Deletes the specified rule.

**Example 10-13.** Delete firewall rule

---

Request:

```

DELETE https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections
\layer2sections/sectionId
/rules/ruleId

```

---

## Working with Layer3 Redirect Sections and Rules

Firewall rules redirecting traffic to a partner service (such as Palo Alto Networks Firewall) are displayed in the layer3 redirect section. On the UI, layer3 redirect sections and rules are in the Partner Security Services tab.

### Query Layer3 Redirect Rules (All)

In order to see all layer3 redirect rules for all sections, use the following command:

**Example 10-14.** Get all layer3 redirect rules

---

```

GET https://10.114.223.26/api/4.0/firewall/globalroot-0/config

```

### Query Layer3 Redirect Section

Retrieves layer3 redirect section configuration either by *section ID* or *section name*.

**Example 10-15.** Get layer3 redirect section configuration

---

Request:

```

GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
/layer3redirectsections/sectionId|sectionName

```

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<section id="1002" name="Default Section" generationNumber="1464286473045"
      timestamp="1464286473045" type="L3REDIRECT">
<rule id="1006" disabled="false" logged="true">
  <name>Copy Packets between VM1 and VM2 to Service</name>
  <action>redirect</action>
  <appliedToList>
    <appliedTo>
      <name>ALL_PROFILE_BINDINGS</name>
      <value>ALL_PROFILE_BINDINGS</value>
      <type>ALL_PROFILE_BINDINGS</type>
      <isvalid>true</isvalid>
    </appliedTo>
  </appliedToList>
  <sectionId>1002</sectionId>
  <sources excluded="false">
    <source>
      <name>2-vm_RHEL-srv</name>
      <value>vm-28</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>2-vm_RHEL-srv</name>
      <value>vm-28</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </destination>
  </destinations>
  <siProfile>
    <objectId>serviceprofile-2</objectId>
    <revision>0</revision>
    <name>ABC Company Service Profile Name</name>
    <isuniversal>false</isuniversal>
    <universalRevision>0</universalRevision>
  </siProfile>
  <services>
    <service>
      <name>service_abc</name>
      <value>application-122</value>
      <type>Application</type>
      <isvalid>true</isvalid>
    </service>
  </services>
  <siRuleIdList>
    <siRuleId>375</siRuleId>
  </siRuleIdList>
  <direction>inout</direction>
  <packetType>any</packetType>
  <stateless>false</stateless>
</rule>
</section>
```

---

## Add Layer3 Redirect Section

Adds layer3 redirect section configuration.

**Example 10-16.** Add layer3 redirect section configuration

---

Request:

POST `https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3redirectsections`

Request Body:

```
<section name="layer3Redirect_user_section">
  <rule disabled="false" logged="true">
    <name>Copy packets between VM3 and VM4 to Service</name>
    <action>redirect</action>
    <appliedToList>
      <appliedTo>
        <name>ALL_PROFILE_BINDINGS</name>
        <value>ALL_PROFILE_BINDINGS</value>
        <type>ALL_PROFILE_BINDINGS</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sources excluded="false">
      <source>
        <name>3-vm_RHEL-srv</name>
        <value>vm-26</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
      </source>
    </sources>
    <destinations excluded="false">
      <destination>
        <name>3-vm_RHEL-srv</name>
        <value>vm-26</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
      </destination>
    </destinations>
    <siProfile>
      <objectId>serviceprofile-2</objectId>
      <name>ABC Company Service Profile Name</name>
    </siProfile>
    <services>
      <service>
        <name>HTTP</name>
        <value>application-278</value>
        <type>Application</type>
        <isValid>true</isValid>
      </service>
    </services>
    <direction>inout</direction>
    <packetType>any</packetType>
    <stateless>false</stateless>
  </rule>
  <rule>.....</rule>
</section>
```

---

## Modify Layer3 Redirect Section

Follow the procedure below to modify a firewall section.

- 1 Run a GET call for the layer3 redirect section.
- 2 Extract the XML from the response body of the GET call and modify it as required.
- 3 From the Response Header in Step 1, copy the Etag header value.

```

1. Status Code      : 200 OK
2. Cache-Control   : private, no-cache
3. Content-Type    : application/xhtml+xml
4. Date            : Tue, 01 Apr 2014 20:04:58 GMT
5. Etag            : 1396034461743
6. Expires         : Thu, 01 Jan 1970 00:00:00 GMT
7. Server          : NSX Manager

```

- 4 Add the number as the If-Match header in the PUT call.



- 5 Pass the modified XML as the Request Body in a PUT call.
- Not all fields are required while sending the request. Refer to the optional field in the schema definition of various objects. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP Set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.
  - IDs for new objects ( rule/section) should be removed or set to zero.
  - If new entities (sections/rules) have been sent in the request, the response will contain the system-generated ids, which are assigned to these new entities. These ID identifies the resource and can be used in the urls if you want to operate on these entities using those URLs.

#### Example 10-17. Modify layer3 redirect section

Request:

```

PUT https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
/layer3redirectsections/sectionId\sectionName
--header 'Content-Type:text/xml' --header 'if-match:"1396034461743"

```

Request Body:

```

<section id="1002" name="Default Section" generationNumber="1464286473045"
timestamp="1464286473045" type="L3REDIRECT">
  <rule id="1006" disabled="false" logged="true">
    <name>Copy Packets between VM1 and VM2 to Service_updated_Rule</name>
    <action>redirect</action>
    <appliedToList>
      <appliedTo>
        <name>ALL_PROFILE_BINDINGS</name>
        <value>ALL_PROFILE_BINDINGS</value>
        <type>ALL_PROFILE_BINDINGS</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>1002</sectionId>
    <sources excluded="false">
      <source>
        <name>2-vm_RHEL-srv</name>
        <value>vm-28</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
      </source>
    </sources>
    <destinations excluded="false">
      <destination>
        <name>2-vm_RHEL-srv</name>
        <value>vm-28</value>

```

```

        <type>VirtualMachine</type>
        <isValid>true</isValid>
    </destination>
</destinations>
<siProfile>
    <objectId>serviceprofile-2</objectId>
    <revision>0</revision>
    <name>ABC Company Service Profile Name</name>
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
</siProfile>
<services>
    <service>
        <name>HTTPS</name>
        <value>application-279</value>
        <type>Application</type>
        <isValid>true</isValid>
    </service>
</services>
<direction>inout</direction>
<packetType>any</packetType>
<stateless>false</stateless>
</rule>
</section>

```

---

## Delete Layer3 Redirect Section

Deletes the specified section.

**Example 10-18.** Delete layer3 redirect section

---

Request:

```

DELETE https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
        /layer3redirectsections/sectionId/sectionName
        --header 'Content-Type:text/xml' --header 'if-match:"1396034461743"

```

---

## Query Layer3 Redirect Rules

Retrieves layer3 redirect rules for the specified section.

**Example 10-19.** Get layer3 redirect rules

---

Request:

```

GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
        /layer3redirectsections/sectionId/rules/ruleId

```

Response Body:

```

<rule id="1006" disabled="false" logged="true">
    <name>Copy Packets between VM1 and VM2 to Service</name>
    <action>redirect</action>
    <appliedToList>
        <appliedTo>
            <name>ALL_PROFILE_BINDINGS</name>
            <value>ALL_PROFILE_BINDINGS</value>
            <type>ALL_PROFILE_BINDINGS</type>
            <isValid>true</isValid>
        </appliedTo>
    </appliedToList>
    <sources excluded="false">
        <source>
            <name>2-vm_RHEL-srv</name>
            <value>vm-28</value>
        </source>
    </sources>
</rule>

```

```

        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
    </source>
    <source>
        <name>1-vm_RHEL-srv</name>
        <value>vm-22</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
    </source>
</sources>
<destinations excluded="false">
    <destination>
        <name>2-vm_RHEL-srv</name>
        <value>vm-28</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
    </destination>
    <destination>
        <name>1-vm_RHEL-srv</name>
        <value>vm-22</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
    </destination>
</destinations>
<services>
    <service>
        <name>HTTP</name>
        <value>application-278</value>
        <type>Application</type>
        <isvalid>true</isvalid>
    </service>
</services>
<siProfile>
    <objectId>serviceprofile-2</objectId>
    <revision>0</revision>
    <name>ABC Company Service Profile Name</name>
    <clientHandle />
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
</siProfile>
<siRuleIdList>
    <siRuleId>375</siRuleId>
</siRuleIdList>
<direction>inout</direction>
<packetType>any</packetType>
<stateless>false</stateless>
</rule>

```

---

## Add Layer3 Redirect Rule

Adds layer3 redirect rule.

**Example 10-20.** Add layer3 redirect rule

---

Request:

```

POST https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
     /layer3redirectsections/sectionId/rules
     --header 'Content-Type:text/xml' --header 'if-match:"1396034461743"

```

Request Body:

```

<rule disabled="false" logged="true">
  <name>add_rule_to_section</name>
  <action>redirect</action>
  <sources excluded="false">
    <source>

```

```

    <name>2-vm_RHEL-srv</name>
    <value>vm-28</value>
    <type>VirtualMachine</type>
    <isValid>true</isValid>
  </source>
</sources>
<destinations excluded="false">
  <destination>
    <name>2-vm_RHEL-srv</name>
    <value>vm-28</value>
    <type>VirtualMachine</type>
    <isValid>true</isValid>
  </destination>
</destinations>
<siProfile>
  <objectId>serviceprofile-2</objectId>
  <name>ABC Company Service Profile Name</name>
</siProfile>
<direction>inout</direction>
<packetType>any</packetType>
<stateless>>false</stateless>
</rule>

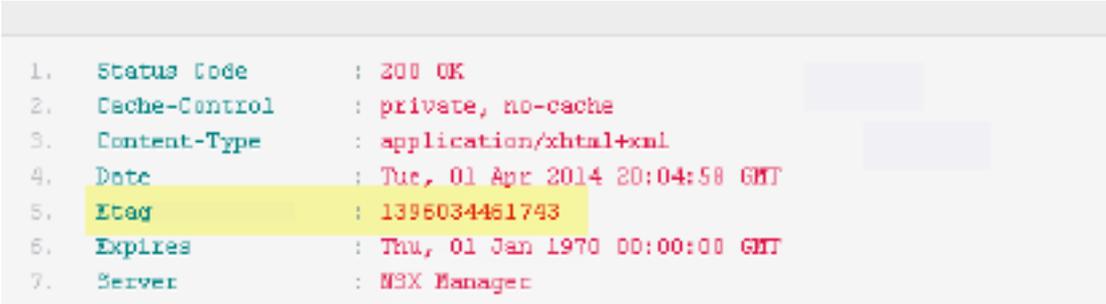
```

## Modify Layer3 Redirect Rule

Modifies a layer3 redirect rule.

Follow the procedure below to modify a rule.

- 1 Run a GET call for the firewall rules.
- 2 Extract the XML from the response body of the GET call and modify it as required.
- 3 From the Response Header in Step 1, copy the Etag header value.



```

1. Status Code      : 200 OK
2. Cache-Control   : private, no-cache
3. Content-Type    : application/xhtml+xml
4. Date            : Tue, 01 Apr 2014 20:04:58 GMT
5. Etag            : 1395034461743
6. Expires         : Thu, 01 Jan 1970 00:00:00 GMT
7. Server          : NSX Manager

```

- 4 Add the number as the If-Match header in the PUT call.



- 5 Pass the modified XML as the Request Body in a PUT call.
  - Not all fields are required while sending the request. Refer to the optional field in the schema definition of various objects. All the optional fields are safe to be ignored while sending the configuration to server. For example, if an IP Set is referenced in the rule only IPSet and Type is needed in the Source/Destination objects and not Name and isValid tags.
  - IDs for new objects (rule/section) should be removed or set to zero.
  - If new entities (sections/rules) have been sent in the request, the response will contain the system-generated ids, which are assigned to these new entities. These ID identifies the resource and can be used in the URLs if you want to operate on these entities using those URLs.

**Example 10-21.** Modify layer3 redirect rule

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
    /layer3redirectsections/sectionId/rules/ruleId
    --header 'Content-Type:text/xml' --header 'if-match:"1396034461743"
```

Request Body:

```
<rule id="1006" disabled="false" logged="false">
  <name>update_rule_into_section</name>
  <action>redirect</action>
  <appliedToList>
    <appliedTo>
      <name>ALL_PROFILE_BINDINGS</name>
      <value>ALL_PROFILE_BINDINGS</value>
      <type>ALL_PROFILE_BINDINGS</type>
      <isvalid>true</isvalid>
    </appliedTo>
  </appliedToList>
  <sources excluded="false">
    <source>
      <name>2-vm_RHEL-srv</name>
      <value>vm-28</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>2-vm_RHEL-srv</name>
      <value>vm-28</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </destination>
  </destinations>
  <siProfile>
    <objectId>serviceprofile-2</objectId>
    <name>ABC Company Service Profile Name</name>
  </siProfile>
  <direction>inout</direction>
  <packetType>any</packetType>
  <stateless>false</stateless>
</rule>
```

**Delete Layer3 Redirect Rule**

Deletes the specified layer3 redirect rule.

**Example 10-22.** Delete layer3 redirect rule

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
    /layer3redirectsections/sectionId/rules/ruleId
    --header 'Content-Type:text/xml' --header 'if-match:"1396034461743"
```

**Query Service Insertion Profiles**

Retrieves the Service Insertion profiles that can be applied to layer3 redirect rules.

**Example 10-23.** Query Service Insertion profiles

Request:

GET https://NSX-Manager-IP-Address/api/4.0/firewall/layer3redirect/profiles

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<serviceProfiles>
  <serviceProfile>
    <objectId>serviceprofile-2</objectId>
    <objectTypeName>ServiceProfile</objectTypeName>
    <vsmUuid>422F028E-C460-944D-7A25-0BEED51D982D</vsmUuid>
    <nodeId>84c344df-1ae1-4694-8688-e07e337b2d63</nodeId>
    <revision>1</revision>
    <type>
      <typeName>ServiceProfile</typeName>
    </type>
    <name>ABC Company Service Profile Name</name>
    <description>ABC Company Service Profile Name Description</description>
    <clientHandle />
    <extendedAttributes />
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
    <profileAttributes>
      <id>358</id>
      <revision>1</revision>
      <attribute>
        <id>361</id>
        <revision>0</revision>
        <key>tenantID</key>
        <name>Tenant</name>
        <value>tenant</value>
      </attribute>
      <attribute>
        <id>362</id>
        <revision>0</revision>
        <key>ssl_encryption_questions__offload_ssl</key>
        <name>SSL encryption offload</name>
        <value>No</value>
      </attribute>
      <attribute>
        <id>359</id>
        <revision>0</revision>
        <key>basic__addr</key>
        <name>Virtual server address</name>
        <value>80</value>
      </attribute>
      <attribute>
        <id>360</id>
        <revision>0</revision>
        <key>failOpen</key>
        <name>Failure Policy</name>
        <value>true</value>
      </attribute>
    </profileAttributes>
    <service>
      <objectId>service-9</objectId>
      <objectTypeName>Service</objectTypeName>
      <vsmUuid>422F028E-C460-944D-7A25-0BEED51D982D</vsmUuid>
      <nodeId>84c344df-1ae1-4694-8688-e07e337b2d63</nodeId>
      <revision>3</revision>
      <type>
        <typeName>Service</typeName>
      </type>
      <name>ABC Company Service</name>
      <clientHandle />
      <extendedAttributes />
    </service>
  </serviceProfile>
</serviceProfiles>

```

```

    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
  </service>
  <serviceInstance>
    <objectId>serviceinstance-3</objectId>
    <objectTypeName>ServiceInstance</objectTypeName>
    <vsmUuid>422F028E-C460-944D-7A25-0BEED51D982D</vsmUuid>
    <nodeId>84c344df-1ae1-4694-8688-e07e337b2d63</nodeId>
    <revision>1</revision>
    <type>
      <typeName>ServiceInstance</typeName>
    </type>
    <name>ABC Company Service-GlobalInstance</name>
    <clientHandle />
    <extendedAttributes />
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
  </serviceInstance>
  <vendorTemplate>
    <id>341</id>
    <revision>0</revision>
    <name>ABC Company Vendor Template</name>
    <description>ABC Company Vendor Template Description</description>
    <idFromVendor>ABC Company Vendor Id</idFromVendor>
    <vendorAttributes>
      <id>342</id>
      <revision>0</revision>
      <attribute>
        <id>344</id>
        <revision>0</revision>
        <key>Key 2</key>
        <name>Value 2</name>
        <value>Name 2</value>
      </attribute>
      <attribute>
        <id>343</id>
        <revision>0</revision>
        <key>Key 1</key>
        <name>Value 1</name>
        <value>Name 1</value>
      </attribute>
    </vendorAttributes>
    <functionalities>
      <functionality>
        <type>IDS_IPS</type>
        <revision>0</revision>
      </functionality>
    </functionalities>
  </vendorTemplate>
  <status>IN_SERVICE</status>
  <vendorAttributes>
    <id>364</id>
    <revision>1</revision>
    <attribute>
      <id>368</id>
      <revision>0</revision>
      <key>Key 2</key>
      <name>Value 2</name>
      <value>Name 2</value>
    </attribute>
    <attribute>
      <id>366</id>
      <revision>0</revision>
      <key>Key 1</key>
      <name>Value 1</name>
      <value>Name 1</value>
    </attribute>
  </vendorAttributes>

```

```

        <id>365</id>
        <revision>0</revision>
        <key>server_pools__create_new_pool</key>
        <name>Server pool</name>
        <value>Create New Pool</value>
    </attribute>
    <attribute>
        <id>367</id>
        <revision>0</revision>
        <key>optimizations__lan_or_wan</key>
        <name>Network Optimization</name>
        <value>Lan</value>
    </attribute>
</vendorAttributes>
<runtime>
    <nonCompliantDvpg />
    <nonCompliantVwire />
</runtime>
<serviceProfileBinding>
    <distributedVirtualPortGroups>
        <string>dvportgroup-20</string>
    </distributedVirtualPortGroups>
    <virtualWires />
    <excludedVnics />
    <virtualServers />
    <securityGroups />
</serviceProfileBinding>
<vendorTypedAttributes>
    <id>369</id>
    <revision>0</revision>
</vendorTypedAttributes>
<vendorTables />
<vendorSections />
<priority>1</priority>
</serviceProfile>
<serviceProfile>
    <objectId>serviceprofile-4</objectId>
    <objectTypeName>ServiceProfile</objectTypeName>
    <vsmUuid>422F028E-C460-944D-7A25-0BEED51D982D</vsmUuid>
    <nodeId>84c344df-1ae1-4694-8688-e07e337b2d63</nodeId>
    <revision>3</revision>
    <type>
        <typeName>ServiceProfile</typeName>
    </type>
    <name>DEF Service_DEF</name>
    <description>AutoCreated Default ServiceProfile</description>
    <clientHandle />
    <extendedAttributes />
    <isUniversal>false</isUniversal>
    <universalRevision>0</universalRevision>
    <profileAttributes>
        <id>406</id>
        <revision>1</revision>
        <attribute>
            <id>407</id>
            <revision>0</revision>
            <key>failOpen</key>
            <name>Failure Policy</name>
            <value>true</value>
        </attribute>
    </profileAttributes>
    <service>
        <objectId>service-10</objectId>
        <objectTypeName>Service</objectTypeName>
        <vsmUuid>422F028E-C460-944D-7A25-0BEED51D982D</vsmUuid>
        <nodeId>84c344df-1ae1-4694-8688-e07e337b2d63</nodeId>
        <revision>3</revision>
    </service>
    <type>

```

```

        <typeName>Service</typeName>
    </type>
    <name>DEF Service</name>
    <clientHandle />
    <extendedAttributes />
    <isUniversal>>false</isUniversal>
    <universalRevision>0</universalRevision>
</service>
<serviceInstance>
    <objectId>serviceinstance-4</objectId>
    <objectTypeName>ServiceInstance</objectTypeName>
    <vsmUuid>422F028E-C460-944D-7A25-0BEED51D982D</vsmUuid>
    <nodeId>84c344df-1ae1-4694-8688-e07e337b2d63</nodeId>
    <revision>1</revision>
    <type>
        <typeName>ServiceInstance</typeName>
    </type>
    <name>DEF Service-GlobalInstance</name>
    <clientHandle />
    <extendedAttributes />
    <isUniversal>>false</isUniversal>
    <universalRevision>0</universalRevision>
</serviceInstance>
<vendorTemplate>
    <id>403</id>
    <revision>0</revision>
    <name>DEF</name>
    <description />
    <idFromVendor>DEF</idFromVendor>
    <vendorAttributes>
        <id>405</id>
        <revision>0</revision>
    </vendorAttributes>
    <typedAttributes>
        <id>404</id>
        <revision>0</revision>
    </typedAttributes>
    <functionalities>
        <functionality>
            <type>FIREWALL</type>
            <revision>0</revision>
        </functionality>
    </functionalities>
</vendorTemplate>
<status>IN_SERVICE</status>
<vendorAttributes>
    <id>409</id>
    <revision>0</revision>
</vendorAttributes>
<runtime>
    <nonCompliantDvpg />
    <nonCompliantVwire />
</runtime>
<serviceProfileBinding>
    <distributedVirtualPortGroups />
    <virtualWires />
    <excludedVnics />
    <virtualServers />
    <securityGroups />
</serviceProfileBinding>
<vendorTypedAttributes>
    <id>410</id>
    <revision>0</revision>
</vendorTypedAttributes>
<vendorTables />
<vendorSections />
    <priority>1</priority>
</serviceProfile>

```

```
<serviceProfile>.....</serviceProfile>
</serviceProfiles>
```

---

## Query Status

Retrieves status of last publish action for each cluster in the NSX environment.

### Query Firewall Configuration Status

**Example 10-24.** Get firewall configuration status

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/status
```

Response Body:

```
<firewallStatus>
  <startTime>1380747467905</startTime>
  <status>published</status>
  <generationNumber>1380747467905</generationNumber>
  <clusterList>
    <clusterStatus>
      <clusterId>domain-c256</clusterId>
      <status>published</status>
      <generationNumber>1380747467905</generationNumber>
      <hostStatusList>
        <hostStatus>
          <hostId>host-244</hostId>
          <hostName>10.24.227.43</hostName>
          <status>published</status>
          <errorCode>0</errorCode>
          <startTime>1380725776946</startTime>
          <endTime>1380747469986</endTime>
          <generationNumber>1380747467905</generationNumber>
          <clusterId>domain-c256</clusterId>
        </hostStatus>
      </hostStatusList>
    </clusterStatus>
    <clusterStatus>
      <clusterId>domain-c322</clusterId>
      <status>published</status>
      <generationNumber>1380747467905</generationNumber>
      <hostStatusList>
        <hostStatus>
          <hostId>host-310</hostId>
          <hostName>10.24.227.75</hostName>
          <status>published</status>
          <errorCode>0</errorCode>
          <startTime>1380746933333</startTime>
          <endTime>1380747470292</endTime>
          <generationNumber>1380747467905</generationNumber>
          <clusterId>domain-c322</clusterId>
        </hostStatus>
      </hostStatusList>
    </clusterStatus>
  </clusterList>
</firewallStatus>
```

---

### Query Layer3 Section Status

Retrieves status of last publish action for specified Layer 3 section.

**Example 10-25.** Get Layer3 status

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/status/layer3sections
/sectionId
```

Response Body:

```
<firewallStatus>
  <startTime>1380747467905</startTime>
  <status>published</status>
  <generationNumber>1380747467905</generationNumber>
  <clusterList>
    <clusterStatus>
      <clusterId>domain-c256</clusterId>
      <status>published</status>
      <generationNumber>1380747467905</generationNumber>
      <hostStatusList>
        <hostStatus>
          <hostId>host-244</hostId>
          <hostName>10.24.227.43</hostName>
          <status>published</status>
          <errorCode>0</errorCode>
          <startTime>1380725776946</startTime>
          <endTime>1380747469986</endTime>
          <generationNumber>1380747467905</generationNumber>
          <clusterId>domain-c256</clusterId>
        </hostStatus>
      </hostStatusList>
    </clusterStatus>
    <clusterStatus>
      <clusterId>domain-c322</clusterId>
      <status>published</status>
      <generationNumber>1380747467905</generationNumber>
      <hostStatusList>
        <hostStatus>
          <hostId>host-310</hostId>
          <hostName>10.24.227.75</hostName>
          <status>published</status>
          <errorCode>0</errorCode>
          <startTime>1380746933333</startTime>
          <endTime>1380747470292</endTime>
          <generationNumber>1380747467905</generationNumber>
          <clusterId>domain-c322</clusterId>
        </hostStatus>
      </hostStatusList>
    </clusterStatus>
  </clusterList>
</firewallStatus>
```

**Query Layer2 Section Status**

Retrieves status of last publish action for specified Layer 3 section.

**Example 10-26.** Get layer2 status

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/status/layer2sections
/sectionId
```

Response Body:

```
<firewallStatus>
  <startTime>1380747467905</startTime>
  <status>published</status>
```

```

<generationNumber>1380747467905</generationNumber>
<clusterList>
  <clusterStatus>
    <clusterId>domain-c256</clusterId>
    <status>published</status>
    <generationNumber>1380747467905</generationNumber>
    <hostStatusList>
      <hostStatus>
        <hostId>host-244</hostId>
        <hostName>10.24.227.43</hostName>
        <status>published</status>
        <errorCode>0</errorCode>
        <startTime>1380725776946</startTime>
        <endTime>1380747469986</endTime>
        <generationNumber>1380747467905</generationNumber>
        <clusterId>domain-c256</clusterId>
      </hostStatus>
    </hostStatusList>
  </clusterStatus>
  <clusterStatus>
    <clusterId>domain-c322</clusterId>
    <status>published</status>
    <generationNumber>1380747467905</generationNumber>
    <hostStatusList>
      <hostStatus>
        <hostId>host-310</hostId>
        <hostName>10.24.227.75</hostName>
        <status>published</status>
        <errorCode>0</errorCode>
        <startTime>1380746933333</startTime>
        <endTime>1380747470292</endTime>
        <generationNumber>1380747467905</generationNumber>
        <clusterId>domain-c322</clusterId>
      </hostStatus>
    </hostStatusList>
  </clusterStatus>
</clusterList>
</firewallStatus>

```

---

## Working with Memory and CPU Thresholds

Knowing the host resource utilization at any given point of time can help you in better organizing your server utilization and network designs.

You can configure memory, CPU, and Connections Per Second (CPS) thresholds through REST API calls. The Firewall module generates system events when the memory and CPU usage crosses these thresholds.

### Configure Thresholds

Configures memory, CPU, and CPS thresholds for Firewall.

#### Example 10-27. Configure thresholds

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/firewall/stats/eventthresholds>

Request Body:

```

<eventThresholds>
  <cpu>
    <percentValue>80</percentValue>
  </cpu>
  <memory>
    <percentValue>90</percentValue>
  </memory>

```

```

    <connectionsPerSecond>
      <value>250000</value>
    </connectionsPerSecond>
  </eventThresholds>

```

---

## Query Thresholds

Retrieves memory, CPU, and CPS thresholds for Firewall.

### Example 10-28. Query thresholds

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/stats/eventthresholds
```

Response Body:

```

<eventThresholds>
  <cpu>
    <percentValue>80</percentValue>
  </cpu>
  <memory>
    <percentValue>90</percentValue>
  </memory>
  <connectionsPerSecond>
    <value>250000</value>
  </connectionsPerSecond>
</eventThresholds>

```

---

## Tuning Firewall Performance

You can use the following flags to improve Firewall performance:

- RuleOptimize has layer3RuleOptimize and layer2RuleOptimize to turn on/off rule optimization
- TCPStrict option helps in tighter access and forwarding control

### Example 10-29. Set RuleOptimize and TCPStrict flags

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/firewall/config/globalconfiguration
```

Request Body:

```

<globalConfiguration>
  <layer3RuleOptimize>true</layer3RuleOptimize>
  <layer2RuleOptimize>false</layer2RuleOptimize>
  <tcpStrictOption>true</tcpStrictOption>
</globalConfiguration>

```

---

### Example 10-30. Query RuleOptimize and TCPStrict flags

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/config/globalconfiguration
```

Response Body:

```

<globalConfiguration>
  <layer3RuleOptimize>true</layer3RuleOptimize>
  <layer2RuleOptimize>false</layer2RuleOptimize>
  <tcpStrictOption>true</tcpStrictOption>

```

---

```
</globalConfiguration>
```

---

## Synchronizing and Enabling Firewall

You can force hosts and clusters to synchronize with the last good configuration in the NSX Manager database.

### Force Sync Host

Forces the host to sync with the last good configuration

#### Example 10-31. Force sync host

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/firewall/forceSync/hostId
```

Response Body:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Set-Cookie: JSESSIONID=EADEDB6AC7323C3FE42E43B8739FBB1F; Path=/
Location: /api/2.0/services/taskservice/job/jobdata-658
Date: wed, 02 Oct 2013 21:08:52 GMT
Server: vShield Manager
Content-Length: 0
```

---

The location header contains the task URL, which can be used to monitor the overall task status.

### Force Sync Cluster

#### Example 10-32. Force sync cluster

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/firewall/forceSync/clusterId
```

Response Body:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Set-Cookie: JSESSIONID=EADEDB6AC7323C3FE42E43B8739FBB1F; Path=/
Location: /api/2.0/services/taskservice/job/jobdata-659
Date: wed, 02 Oct 2013 21:08:52 GMT
Server: vShield Manager
Content-Length: 0
```

---

The location header contains the task URL, which can be used to monitor the overall task status.

### Enable or Disable APIs for a Cluster

You can disable firewall components on a cluster. If firewall is disabled on a cluster, all network traffic passes through the hosts in that cluster without any validation.

#### Example 10-33. Enable or disable API

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/4.0/firewall/domainId/enable/true\false
```

---

## Importing and Exporting Firewall Configurations

You may make changes to a firewall configuration and save a draft copy for future use. A copy of every published configuration is also saved as a draft. A maximum of 100 configurations can be saved at a time. 90 out of these 100 can be auto saved configurations from a publish operation. When the limit is reached, the oldest configuration that is not marked for preserve is purged to make way for a new one.

You can also import and export firewall configurations in XML format.

### Save a Configuration

**Example 10-34.** Save a firewall configuration

---

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/drafts>

Request Body:

```
<firewallDraft name="TestDraft">
  <description>Test draft</description>  <!-- optional -->
  <preserve>true</preserve>  <!-- optional, default = true -->
  <mode>userdefined</mode>
  <config>
    <contextId>globalroot-0</contextId>
    <layer3Sections>
      <section name="Default Section Layer3" >
        <rule id="1001" disabled="false" logged="false">
          <name>Default Rule</name>
          <action>allow</action>
          <precedence>default</precedence>
        </rule>
      </section>
    </layer3Sections>
    <layer2Sections>
      <section name="Default Section Layer2">
        <rule id="1003" disabled="false" logged="false">
          <name>Default Rule</name>
          <action>allow</action>
          <precedence>default</precedence>
        </rule>
      </section>
    </layer2Sections>
  </config>
</firewallDraft>
```

Response Body:

```
HTTP/1.1 200 OK
<firewallDraft id="23" name="TestDraft" timestamp="1377631752553">
  <description>Test draft</description>
  <preserve>true</preserve>
  <user>localadmin</user>
  <mode>userdefined</mode>
</firewallDraft>
```

---

### Query all Saved Configurations

Displays the draft ID of all saved configurations. The draft ID is required for other operations.

**Example 10-35.** Get all saved firewall configurations

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/drafts/>

Response Body:

```
<firewallDrafts>
  <firewallDraft id="3" name="AutoSaved_2013-Aug-22 17:13:08" timestamp="1377191588887">
    <description>Auto saved draft</description>
    <preserve>false</preserve>
    <user>root</user>
    <mode>autosaved</mode>
  </firewallDraft>
  <firewallDraft id="2" name="AutoSaved_2013-Aug-22 15:46:40" timestamp="1377186400472">
    <description>Auto saved draft</description>
    <preserve>false</preserve>
    <user>root</user>
    <mode>autosaved</mode>
  </firewallDraft>
  <firewallDraft id="1" name="AutoSaved_2013-Aug-22 15:42:36" timestamp="1377186156947">
    <description>Auto saved draft</description>
    <preserve>false</preserve>
    <user>root</user>
    <mode>autosaved</mode>
  </firewallDraft>
</firewallDrafts>
```

---

## Query a Saved Configuration

Retrieve the *draftID* of the configuration. See [“Get all saved firewall configurations”](#) on page 327.

**Example 10-36.** Get a saved firewall configuration

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/drafts/draftId
```

Response Body:

```
<firewallDraft id="1" name="AutoSaved_2013-Aug-22 15:42:36" timestamp="1377186156947">
  <description>Auto saved draft</description>
  <preserve>false</preserve>
  <user>root</user>
  <mode>autosaved</mode>
  <config timestamp="1377186104244">
    <contextId>globalroot-0</contextId>
    <layer3Sections>
      <section id="1002" name="Default Section Layer3"
        generationNumber="1377186104244" timestamp="1377186104244">
        <rule disabled="false" logged="false">
          <name>Default Rule NDP - Edit</name>
          <action>allow</action>
          <sectionId>1002</sectionId>
          <services>
            <service>
              <name>IPv6-ICMP Neighbor Solicitation</name>
              <value>application-182</value>
              <type>Application</type>
              <isvalid>true</isvalid>
            </service>
          </services>
        </rule>
        <rule id="1002" disabled="false" logged="false">
          <name>Default Rule</name>
          <action>allow</action>
          <sectionId>1002</sectionId>
          <precedence>default</precedence>
        </rule>
      </section>
    </layer3Sections>
  </layer2Sections>
```

```

    <section id="1001" name="Default Section Layer2"
      generationNumber="1377186104244" timestamp="1377186104244">
      <rule id="1001" disabled="false" logged="false">
        <name>Default Rule</name>
        <action>allow</action>
        <sectionId>1001</sectionId>
        <precedence>default</precedence>
      </rule>
    </section>
  </layer2Sections>
  <generationNumber>1377285109371</generationNumber>
</config>
</firewallDraft>

```

---

## Modify a Saved Configuration

Retrieve the *draftID* of the configuration. See [“Get all saved firewall configurations”](#) on page 327.

### Example 10-37. Update a saved firewall configuration

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/drafts/draftId>

Request Body:

```

<firewallDraft name="TestDraft">
  <description>Test draft</description>  <!-- optional -->
  <preserve>true</preserve>  <!-- optional, default = true -->
  <mode>userdefined</mode>
  <config>
    <contextId>globalroot-0</contextId>
    <layer3Sections>
      <section name="Default Section Layer3" >
        <rule id="1001" disabled="false" logged="false">
          <name>Default Rule</name>
          <action>allow</action>
          <precedence>default</precedence>
        </rule>
      </section>
    </layer3Sections>
    <layer2Sections>
      <section name="Default Section Layer2">
        <rule id="1003" disabled="false" logged="false">
          <name>Default Rule</name>
          <action>allow</action>
          <precedence>default</precedence>
        </rule>
      </section>
    </layer2Sections>
  </config>
</firewallDraft>

```

Response Body:

```

HTTP/1.1 200 OK
<firewallDraft id="23" name="TestDraft" timestamp="1377631752553">
  <description>Test draft</description>
  <preserve>true</preserve>
  <user>localadmin</user>
  <mode>userdefined</mode>
</firewallDraft>

```

---

## Delete a Saved Configuration

Retrieve the *draftID* of the configuration. See [“Get all saved firewall configurations”](#) on page 327.

**Example 10-38.** Delete a saved firewall configuration

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/drafts/draftId
```

---

## Export a Saved Configuration

Retrieve the *draftID* of the configuration. See [“Get all saved firewall configurations”](#) on page 327.

**Example 10-39.** Export a saved firewall configuration

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/drafts/draftId/action/export
```

Response Body:

```
<firewallDraft id="23" name="TestDraft" timestamp="1377631752553">
  <description>Test draft Edit</description>
  <preserve>false</preserve>
  <user>localadmin</user>
  <mode>userdefined</mode>
  <config timestamp="0">
    <contextId>globalroot-0</contextId>
    <layer3Sections>
      <section name="Default Section Layer3" timestamp="0">
        <rule id="1002" disabled="false" logged="false">
          <name>Default Rule</name>
          <action>allow</action>
          <precedence>default</precedence>
        </rule>
      </section>
    </layer3Sections>
    <layer2Sections>
      <section name="Default Section Layer2" timestamp="0">
        <rule id="1001" disabled="false" logged="false">
          <name>Default Rule</name>
          <action>allow</action>
          <precedence>default</precedence>
        </rule>
      </section>
    </layer2Sections>
    <generationNumber>1377285109371</generationNumber>
  </config>
</firewallDraft>
```

---

## Import a Saved Configuration

Retrieve the *draftID* of the configuration. See [“Get all saved firewall configurations”](#) on page 327.

Use the response body of the export command as the request body in this command. See [“Export a saved firewall configuration”](#) on page 330.

**Example 10-40.** Import a saved firewall configuration

---

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/drafts/action/import
```

Request Body:

```
<firewallDraft id="23" name="TestDraft" timestamp="1377631752553">
  <description>Test draft Edit</description>
  <preserve>>false</preserve>
  <user>localadmin</user>
  <mode>userdefined</mode>
  <config timestamp="0">
    <contextId>globalroot-0</contextId>
    <layer3Sections>
      <section name="Default Section Layer3" timestamp="0">
        <rule id="1002" disabled="false" logged="false">
          <name>Default Rule</name>
          <action>allow</action>
          <precedence>default</precedence>
        </rule>
      </section>
    </layer3Sections>
    <layer2Sections>
      <section name="Default Section Layer2" timestamp="0">
        <rule id="1001" disabled="false" logged="false">
          <name>Default Rule</name>
          <action>allow</action>
          <precedence>default</precedence>
        </rule>
      </section>
    </layer2Sections>
    <generationNumber>1377285109371</generationNumber>
  </config>
</firewallDraft>
```

Response Body:

```
HTTP/1.1 200 OK
<firewallDraft id="24" name="TestDraft" timestamp="1377632629140">
  <description>Test draft Edit</description>
  <preserve>>false</preserve>
  <user>localadmin</user>
  <mode>imported</mode>
</firewallDraft>
```

## Working with SpoofGuard

After synchronizing with the vCenter Server, NSX Manager collects the IP addresses of all vCenter guest virtual machines from VMware Tools on each virtual machine. If a virtual machine has been compromised, the IP address can be spoofed and malicious transmissions can bypass firewall policies.

You create a SpoofGuard policy for specific networks that allows you to authorize the IP addresses reported by VMware Tools and alter them if necessary to prevent spoofing. SpoofGuard inherently trusts the MAC addresses of virtual machines collected from the VMX files and vSphere SDK. Operating separately from Firewall rules, you can use SpoofGuard to block traffic determined to be spoofed.

### Create SpoofGuard Policy

You can create a SpoofGuard policy to specify the operation mode for specific networks. The system generated policy applies to port groups and logical switches not covered by existing SpoofGuard policies.

**Example 10-41.** Create SpoofGuard policy

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/services/spoofguard/policies/
```

Request Body:

```
<spoofguardPolicy>
```

```

<name>rest-spoofguard-policy-1</name>
<description>Test description</description>
<operationMode>TOFU</operationMode>
<enforcementPoint>
  <id>dvportgroup-28</id>
  <name>network 1</name>
  <type>dvportgroup</type>
</enforcementPoint>
<enforcementPoint>
  <id>dvportgroup-12</id>
  <name>network 2</name>
  <type>dvportgroup</type>
</enforcementPoint>
<allowLocalIPs>true</allowLocalIPs>
</spoofguardPolicy>

```

Response Body:

HTTP/1.1 201 Created  
 Location: /api/4.0/services/spoofguard/policy/spoofguardpolicy-2

---

## Modify SpoofGuard Policy

Updates a SpoofGuard policy.

### Example 10-42. Modify SpoofGuard policy

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/services/spoofguard/policies/policyId>

Request Body:

```

<spoofguardPolicy>
  <policyId>spoofguardpolicy-2</policyId>
  <name>rest-spoofguard-policy-1</name>
  <description>Test description changed</description>
  <operationMode>TOFU</operationMode>
  <enforcementPoint>
    <id>dvportgroup-28</id>
    <name>network 1</name>
    <type>dvportgroup</type>
  </enforcementPoint>
  <enforcementPoint>
    <id>dvportgroup-12</id>
    <name>network 2</name>
    <type>dvportgroup</type>
  </enforcementPoint>
  <allowLocalIPs>true</allowLocalIPs>
</spoofguardPolicy>

```

---

## Query SpoofGuard Policy

Retrieves a SpoofGuard policy.

### Example 10-43. Query SpoofGuard policy

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/services/spoofguard/policies/policyId>

Request Body:

```

<spoofguardPolicy>
  <policyId>spoofguardpolicy-2</policyId>
  <name>rest-spoofguard-policy-1</name>

```

```

<description>Test description changed</description>
<operationMode>TOFU</operationMode>
<enforcementPoint>
  <id>dvportgroup-28</id>
  <name>network 1</name>
  <type>dvportgroup</type>
</enforcementPoint>
<enforcementPoint>
  <id>dvportgroup-12</id>
  <name>network 2</name>
  <type>dvportgroup</type>
</enforcementPoint>
<publishedOn>2011-10-28 16:12:20.0</publishedOn>
<publishedBy>system_user</publishedBy>
<allowLocalIPs>true</allowLocalIPs>
<publishedPending>>false</publishedPending>
<defaultPolicy>>false</defaultPolicy>
<publishPending>>false</publishPending>
<statistics>
  <inSync>true</inSync>
  <activeCount>0</activeCount>
  <inactiveCount>0</inactiveCount>
  <activeSinceLastPublishedCount>0</activeSinceLastPublishedCount>
  <requireReviewCount>0</requireReviewCount>
  <duplicateCount>0</duplicateCount>
  <unpublishedCount>0</unpublishedCount>
</statistics>
</spooftguardPolicy>

```

---

## Query all SpoofGuard Policies

Retrieves all SpoofGuard policies.

### Example 10-44. Query SpoofGuard policies

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/services/spooftguard/policies/
```

Response Body:

```

<spooftguardPolicies>
  <spooftguardPolicy>
    <policyId>spooftguardpolicy-1</policyId>
    <name>system-spoofguard-policy-1</name>
    <description>Test description</description>
    <operationMode>TOFU</operationMode>
    <allowLocalIPs>true</allowLocalIPs>
    <defaultPolicy>true</defaultPolicy>
    <publishedOn>2011-10-28 16:12:20.0</publishedOn>
  </spooftguardPolicy>
  <spooftguardPolicy>
    <policyId>spooftguardpolicy-2</policyId>
    <name>rest-spoofguard-policy-1</name>
    <description>Test description changed</description>
    <operationMode>TOFU</operationMode>
    <enforcementPoint>
      <id>dvportgroup-28</id>
      <name>network 1</name>
      <type>dvportgroup</type>
    </enforcementPoint>
    <enforcementPoint>
      <id>dvportgroup-12</id>
      <name>network 2</name>
      <type>dvportgroup</type>
    </enforcementPoint>
    <publishedOn>2011-10-28 16:12:20.0</publishedOn>
  </spooftguardPolicy>
</spooftguardPolicies>

```

```

    <publishedBy>system_user</publishedBy>
    <allowLocalIPs>true</allowLocalIPs>
    <publishedPending>false</publishedPending>
    <defaultPolicy>false</defaultPolicy>
  </spoofoGuardPolicy>
</spoofoGuardPolicies>

```

---

## Delete SpoofoGuard Policy

Deletes a SpoofoGuard policy.

**Example 10-45.** Delete SpoofoGuard policy

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/services/spoofoGuard/policies/policyId
```

---

## SpoofoGuard Operations

This section describes SpoofoGuard operations.

### Get IP details

Retrieves IP addresses for specified state.

**Example 10-46.** Get IP details

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/services/spoofoGuard/policyId?list
=ACTIVE\INACTIVE\PUBLISHED\UNPUBLISHED\REVIEW_PENDING\DUPLICATE
```

Response Body:

```

<spoofoGuardList>
  <spoofoGuard>
    <id>5009aa18-ab89-ab9d-9386-c7f0da8773aa.000</id>
    <vnicUuid>50204903-f1c9-0e97-e222-4b96f87ec7fe.000</vnicUuid>
    <approvedIpAddress>
      <ipAddress>10.24.123.129</ipAddress>
      <ipAddress>10.24.123.130</ipAddress>
      <ipAddress>10::129</ipAddress>
    </approvedIpAddress>
    <approvedMacAddress>00:50:56:be:00:06</approvedMacAddress>
    <approvedBy>system_user</approvedBy>
    <approvedOn>2011-10-28 16:12:20.0</approvedOn>
    <publishedIpAddress>
      <ipAddress>10.24.123.129</ipAddress>
      <ipAddress>10::129</ipAddress>
    </publishedIpAddress>
    <publishedMacAddress>00:50:56:be:00:06</publishedMacAddress>
    <publishedBy>system_user</publishedBy>
    <publishedOn>2011-10-28 16:12:20.0</publishedOn>
  </spoofoGuard>
  <spoofoGuard>
  </spoofoGuard>
</spoofoGuardList>

```

---

### Approve IP Addresses

Approves specified IP addresses.

**Example 10-47.** Approve IP addresses

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/services/spoofguard/policyId?action=approve>

Request Body:

```
<spoofguardList>
  <spoofguard>
    <id>5009aa18-ab89-ab9d-9386-c7f0da8773aa.000</id>
    <vnicUuid>50204903-f1c9-0e97-e222-4b96f87ec7fe.000</vnicUuid>
    <approvedIpAddress>
      <ipAddress>10.24.123.129</ipAddress>
      <ipAddress>10.24.123.130</ipAddress>
      <ipAddress>10::129</ipAddress>
    </approvedIpAddress>
    <approvedMacAddress>00:50:56:be:00:06</approvedMacAddress>
    <approvedBy>system_user</approvedBy>
    <approvedOn>2011-10-28 16:12:20.0</approvedOn>
    <publishedIpAddress>
      <ipAddress>10.24.123.129</ipAddress>
      <ipAddress>10::129</ipAddress>
    </publishedIpAddress>
    <publishedMacAddress>00:50:56:be:00:06</publishedMacAddress>
    <publishedBy>system_user</publishedBy>
    <publishedOn>2011-10-28 16:12:20.0</publishedOn>
  </spoofguard>
  <spoofguard>
    </spoofguard>
</spoofguardList>
```

**Publish Approved IP Addresses**

Publishes the approved IP addresses.

**Example 10-48.** Publish IP addresses

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/services/spoofguard/policyId?action=publish>**Publish Approved IP Addresses for a Specific vNIC**

Publishes the approved IP addresses for a specific vNIC by providing the optional vnicId argument to the API call where vnicId is the ID of the vNIC.

**Example 10-49.** Publish IP addresses

Request:

```
POST
      https://NSX-Manager-IP-Address/api/4.0/services/spoofguard/policyId?action=
      publish&vnicId=vnicId
```

## Getting Flow Statistic Details

You can retrieve a detailed view of the traffic on your virtual network that passed through Distributed Firewall.

## Get Flow Statistics

You can retrieve flow statistics for a datacenter, port group, virtual machine, or vNIC.

### Example 10-50. Retrieve flow statistics

Request:

```
GET https://NSX-Manager-IP-Address/api/2.1/flow/flowstats?contextId=datacenter-21
&flowType=TCP_UDP&startTime=0&endTime=1320917094000&startIndex=0&pageSize=2
```

Response Body:

```
<FlowStatsPage>
  <pagingInfo>
    <contextId>datacenter-2538</contextId>
    <flowType>TCP_UDP</flowType>
    <startTime>1327405883000</startTime>
    <endTime>1327482600000</endTime>
    <totalCount>817</totalCount>
    <startIndex>0</startIndex>
    <pageSize>2</pageSize>
  </pagingInfo>
  <flowStatsTcpudp>
    <startTime>1327405883000</startTime>
    <endTime>1327446000000</endTime>
    <ruleId>1001</ruleId>
    <blocked>0</blocked>
    <protocol>5</protocol>
    <direction>1</direction>
    <sessions>1449</sessions>
    <sourcePackets>1449</sourcePackets>
    <destinationPackets>0</destinationPackets>
    <sourceBytes>227493</sourceBytes>
    <destinationBytes>0</destinationBytes>
    <networkId>network-2553</networkId>
    <sourceIp>10.112.199.174</sourceIp>
    <destinationIp>255.255.255.255</destinationIp>
    <destinationPort>17500</destinationPort>
    <controlProtocol></controlProtocol>
    <controlSourceIp>0.0.0.0</controlSourceIp>
    <controlDestinationIp>0.0.0.0</controlDestinationIp>
    <controlDestinationPort>0</controlDestinationPort>
    <controlDirection>0</controlDirection>
  </flowStatsTcpudp>
  <flowStatsTcpudp>
    <startTime>1327405883000</startTime>
    <endTime>1327446000000</endTime>
    <ruleId>1001</ruleId>
    <blocked>0</blocked>
    <protocol>5</protocol>
    <direction>1</direction>
    <sessions>69</sessions>
    <sourcePackets>69</sourcePackets>
    <destinationPackets>0</destinationPackets>
    <sourceBytes>17832</sourceBytes>
    <destinationBytes>0</destinationBytes>
    <networkId>network-2553</networkId>
    <sourceIp>10.112.199.13</sourceIp>
    <destinationIp>10.112.199.255</destinationIp>
    <destinationPort>138</destinationPort>
    <controlProtocol></controlProtocol>
    <controlSourceIp>0.0.0.0</controlSourceIp>
    <controlDestinationIp>0.0.0.0</controlDestinationIp>
    <controlDestinationPort>0</controlDestinationPort>
    <controlDirection>0</controlDirection>
  </flowStatsTcpudp>
```

</FlowStatsPage>

Query parameters are described in the table below.

**Table 10-2.** Query parameters for retrieving flow statistics call

Parameter	Description
flowStats	Type of the flow to be retrieved. Possible values are TCP_UDP, LAYER2, and LAYER3
contextId	vc-moref-id of the datacenter, port group, virtual machine, or UUID of the vNIC for which traffic flow is to be retrieved.
startTime	Flows with start time greater than the specified time are to be retrieved.
endTime	Flows with start time lower than the specified time are to be retrieved.
startIndex	Optional parameter that specifies the starting point for retrieving the flows. If this parameter is not specified, flows are retrieved from the beginning.
pageSize	Optional parameter that limits the maximum number of entries returned by the API. The default value for this parameter is 256 and the valid range is 1-1024.

**Table 10-3.** Response values for retrieving flow statistics call

Value	Description
startTime	Start time for current flow.
endTime	End time for current flow.
ruleId	rule Id for current flow.
blocked	Indicates whether traffic is blocked – 0:Flow allowed, 1:Flow blocked, 2:Flow blocked by Spoofguard.
protocol	protocol in flow – 0:TCP, 1:UDP, 2:ICMP.
direction	Direction of flow – 0:To virtual machine, 1:From virtual machine.
sessions	Number of sessions in current flow.
sourcePackets	Count of Packets from Source to Destination in current flow.
destinationPackets	Count of Packets from Destination to Source in current flow.
sourceBytes	Count of Bytes transferred from Source to Destination in current flow.
destinationBytes	Count of Bytes transferred from Destination to Source in current flow.
sourceIp	Source IP of current flow.
destinationIp	Destination IP of current flow.
sourceMac	Source Mac of current flow.
destinationMac	Destination Mac of current flow.
subtype	Identifies the sub type of current flow.
destinationPort	Port number of Destination for TCP/UDP traffic.
controlProtocol	Control protocol for dynamic TCP traffic.
controlSourceIp	Control source IP for dynamic TCP traffic.
controlDestinationIp	Control destination IP for dynamic TCP traffic.
controlDestinationPort	Control destination port for dynamic TCP traffic.
controlDirection	Control direction for dynamic TCP traffic – 0: Source->Destination, 1:Destination->Source.

## Get Flow Meta-Data

You can retrieve the following information for each flow type:

- minimum stats time
- maximum end time
- total flow count

### Example 10-51. Get flow meta-data for flow type

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.1/flow/flowstats?contextId=datacenter-2538\
    &flowType=TCP_UDP&startTime=1327405883000&endTime=1327482600000&startInd
    ex=0&pageSize=2
```

Response Body:

```
<FlowStatsPage>
  <pagingInfo>
    <contextId>datacenter-2538</contextId>
    <flowType>TCP_UDP</flowType>
    <startTime>1327405883000</startTime>
    <endTime>1327482600000</endTime>
    <totalCount>817</totalCount>
    <startIndex>0</startIndex>
    <pageSize>2</pageSize>
  </pagingInfo>
  <flowStatsTcpudp>
    <startTime>1327405883000</startTime>
    <endTime>1327446000000</endTime>
    <ruleId>1001</ruleId>
    <blocked>0</blocked>
    <protocol>5</protocol>
    <direction>1</direction>
    <sessions>1449</sessions>
    <sourcePackets>1449</sourcePackets>
    <destinationPackets>0</destinationPackets>
    <sourceBytes>227493</sourceBytes>
    <destinationBytes>0</destinationBytes>
    <networkId>network-2553</networkId>
    <sourceIp>10.112.199.174</sourceIp>
    <destinationIp>255.255.255.255</destinationIp>
    <destinationPort>17500</destinationPort>
    <controlProtocol></controlProtocol>
    <controlSourceIp>0.0.0.0</controlSourceIp>
    <controlDestinationIp>0.0.0.0</controlDestinationIp>
    <controlDestinationPort>0</controlDestinationPort>
    <controlDirection>0</controlDirection>
  </flowStatsTcpudp>
  <flowStatsTcpudp>
    <startTime>1327405883000</startTime>
    <endTime>1327446000000</endTime>
    <ruleId>1001</ruleId>
    <blocked>0</blocked>
    <protocol>5</protocol>
    <direction>1</direction>
    <sessions>69</sessions>
    <sourcePackets>69</sourcePackets>
    <destinationPackets>0</destinationPackets>
    <sourceBytes>17832</sourceBytes>
    <destinationBytes>0</destinationBytes>
    <networkId>network-2553</networkId>
    <sourceIp>10.112.199.13</sourceIp>
    <destinationIp>10.112.199.255</destinationIp>
    <destinationPort>138</destinationPort>
    <controlProtocol></controlProtocol>
```

```

    <controlSourceIp>0.0.0.0</controlSourceIp>
    <controlDestinationIp>0.0.0.0</controlDestinationIp>
    <controlDestinationPort>0</controlDestinationPort>
    <controlDirection>0</controlDirection>
  </flowStatsTcpUdp>
</FlowStatsPage>

```

---



---



---

## Flow Exclusion

Firewalling is done by a kernel module present on each host. This kernel module on each host generates flow records for network activity happening on protected on VMs. These flow records generated on each host are sent to NSX Manager, which consumes the records from all hosts and displays aggregated meaningful information. Due to the vast amount of flow records which can be generated on a host, capability has been provided to exclude generation of flow records by the kernel module as per criteria chosen by administrator. Following knobs are provided to control flow exclusion. All exclusion parameters are applied globally on all hosts.

- Disable Flows completely at a global level
- Ignore allowed flows
- Ignore blocked flows
- Ignore layer 2 flows
- Source IPs to ignore. Ex: 10.112.3.14, 10.112.3.15-10.112.3.18,192.168.1.1\24
- Source containers to ignore. Container can contain Vm, vNic, IP Set, MAC Set
- Destination IPs to ignore.
- Destination containers to ignore. Container can contain Vm, vNic, IP Set, MAC Set
- Destination ports
- Service containers to ignore. Container can contain Application or Application group

Flow exclusion happens at the source of generation of flow records i.e. host itself. The following flows are discarded by default:

- Broadcast IP (255.255.255.255)
- Local multicast group (224.0.0.0/24)
- Broadcast MAC address (FF:FF:FF:FF:FF:FF)

## Exclude Flows

Excludes specified flows.

### Example 10-52. Exclude flows

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.1/flow/config
```

Request Body:

```

<FlowConfiguration>
  <collectFlows>true</collectFlows>
  <ignoreBlockedFlows>>false</ignoreBlockedFlows>
  <ignoreLayer2Flows>>false</ignoreLayer2Flows>
  <sourceIPs>10.112.3.14, 10.112.3.15-10.112.3.18,192.168.1.1\24</sourceIPs>
  <sourceContainer>

```

```

    <name>vm1 - Network adapter 1</name>
    <id>5013bcd8-c666-1e28-c7a9-600da945954f.000</id>
    <type>Vnic</type>
  </sourceContainer>
</sourceContainer>
  <name>Large XP-1</name>
  <id>vm-126</id>
  <type>VirtualMachine</type>
</sourceContainer>
<destinationIPs>10.112.3.14, 10.112.3.15-10.112.3.18,192.168.1.1\24</destinationIPs>
<destinationContainer>
  <name>vm2 - Network adapter 2</name>
  <id>5013bcd8-c666-1e28-c7a9-600da945954f.000</id>
  <type>Vnic</type>
</destinationContainer>
<destinationContainer>
  <name>Small XP-2</name>
  <id>vm-226</id>
  <type>VirtualMachine</type>
</destinationContainer>
<destinationPorts>22, 40-50, 60</destinationPorts>
<service>
  <name>VMware-VDM2.x-Ephemeral</name>
  <id>application-161</id>
</service>
</FlowConfiguration>

```

---

## Query Excluded Flows

Retrieves excluded flow details.

### Example 10-53. Get excluded flows

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.1/flow/config>

Response Body:

```

<FlowConfiguration>
  <collectFlows>true</collectFlows>
  <ignoreBlockedFlows>>false</ignoreBlockedFlows>
  <ignoreLayer2Flows>>false</ignoreLayer2Flows>
  <sourceIPs>10.112.3.14, 10.112.3.15-10.112.3.18,192.168.1.1\24</sourceIPs>
  <sourceContainer>
    <name>vm1 - Network adapter 1</name>
    <id>5013bcd8-c666-1e28-c7a9-600da945954f.000</id>
    <type>Vnic</type>
  </sourceContainer>
  <sourceContainer>
    <name>Large XP-1</name>
    <id>vm-126</id>
    <type>VirtualMachine</type>
  </sourceContainer>
  <destinationIPs>10.112.3.14, 10.112.3.15-10.112.3.18,192.168.1.1\24</destinationIPs>
  <destinationContainer>
    <name>vm2 - Network adapter 2</name>
    <id>5013bcd8-c666-1e28-c7a9-600da945954f.000</id>
    <type>Vnic</type>
  </destinationContainer>
  <destinationContainer>
    <name>Small XP-2</name>
    <id>vm-226</id>
    <type>VirtualMachine</type>
  </destinationContainer>
  <destinationPorts>22, 40-50, 60</destinationPorts>
  <service>

```

```

    <name>VMware-VDM2.x-Ephemeral</name>
    <id>application-161</id>
  </service>
</FlowConfiguration>

```

---

## Working with IPFix

Configuring IPFix exports specific flows directly from Firewall to a flow collector.

### Configure IPFix

**Example 10-54.** Configure IPFix

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/firewall/contextId/config/ipfix>

Request Body:

```

<ipfixConfiguration>
  <contextId>globalroot-0</contextId>
  <ipfixEnabled>true</ipfixEnabled>
  <observationDomainId>1234</observationDomainId>
  <flowTimeout>50</flowTimeout>
  <collector>
    <ip>FE80:0000:0000:0000:0202:B3FF:FE1E:8329</ip>
    <port>8080</port>
  </collector>
  <collector>
    <ip>11.11.12.13</ip>
    <port>8086</port>
  </collector>
</ipfixConfiguration>

```

---

### Query IPFix Configuration

**Example 10-55.** Query IPFix Configuration

---

Request:

GET <https://NSX-Manager-IP-Address/api/4.0/firewall/contextId/config/ipfix>

Response Body:

```

<ipfixConfiguration>
  <contextId>globalroot-0</contextId>
  <ipfixEnabled>true</ipfixEnabled>
  <observationDomainId>1234</observationDomainId>
  <flowTimeout>50</flowTimeout>
  <collector>
    <ip>11.11.12.14</ip>
    <port>8087</port>
  </collector>
  <collector>
    <ip>FE80:0000:0000:0000:0202:B3FF:FE1E:8329</ip>
    <port>8086</port>
  </collector>
</ipfixConfiguration>

```

---

### Delete IPFix Configuration

Deleting the IPFix configuration resets the configuration to the default values.

**Example 10-56.** Delete IPFix Configuration

Request:

DELETE <https://NSX-Manager-IP-Address/api/4.0/firewall/contextId/config/ipfix>

## Excluding Virtual Machines from Firewall Protection

You can exclude a set of virtual machines from being protected. This exclusion list is applied across Firewall rules within the specified NSX Manager. If a virtual machine has multiple vNICs, all of them are excluded from protection.

VMware recommends that you place the following service virtual machines in the Exclusion List

- vCenter Server. It can be moved into a cluster that is protected by Firewall, but it must already exist in the exclusion list to avoid connectivity issues.
- Partner service virtual machines.
- Virtual machines that require promiscuous mode. If these virtual machines are protected by Firewall, their performance may be adversely affected.

### Add a Virtual Machine to the Exclusion List

You can add a virtual machine to the exclusion list.

**Example 10-57.** Add a virtual machine to exclusion list

Request:

PUT <https://NSX-Manager-IP-Address/api/2.1/app/excludelist/memberId>

Where memberId is the vc-moref-id of a virtual machine.

### Get Virtual Machine Exclusion List

You can retrieve the set of virtual machines in the exclusion list.

**Example 10-58.** Get exclusion list

Request:

GET <https://NSX-Manager-IP-Address/api/2.1/app/excludelist/>

Response Body:

```
<vshieldAppConfiguration>
  <excludeListConfiguration>
    <objectId>excludeList-1</objectId>
    <type>
      <typeName>ExcludeList</typeName>
    </type>
    <revision>1</revision>
    <objectTypeName>ExcludeList</objectTypeName>
    <excludeMember>
      <member>
        <objectId>vm-2371</objectId>
        <type>
          <typeName>VirtualMachine</typeName>
        </type>
        <name>VC-win2k3</name>
        <revision>2</revision>
        <objectTypeName>VirtualMachine</objectTypeName>
        <scope>
```

```
        <id>domain-c731</id>
        <objectTypeName>ClusterComputeResource</objectTypeName>
        <name>Database-CL</name>
    </scope>
</member>
</excludeMember>
</excludeListConfiguration>
</vshieldAppConfiguration>
```

---

## Delete a Virtual Machine from Exclusion List

You can delete a virtual machines from the exclusion list.

**Example 10-59.** Delete virtual machine from exclusion list

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.1/app/excludelist/memberId
```

---

Where *memberId* is the vc-moref-id of a virtual machine.



# Distributed Firewall Examples

---

VMware NSX Distributed Firewall (DFW) Security Policy Rule Configuration using REST API

## Introduction

VMware NSX Distributed Firewall (DFW) provides the capability to enforce firewall functionality directly at the Virtual Machines (VM) vNIC layer. It is a core component of the micro-segmentation security model where east-west traffic can now be inspected at near line rate processing, preventing any lateral move type of attack.

DFW can be configured using vCenter web client or REST API calls directed to NSX manager.

This technical paper gives information about DFW policy rule configuration using REST API interface. Extensive use of examples will help reader to assimilate required method, URL and body construct.

Because grouping objects like Security Groups, IP Sets or MAC sets are commonly used in DFW security policy rules (in source or destination field), a full section of this document is dedicated to describe REST API calls for these structures. Reader will have a global view of supported functions to manage these grouping objects (again including examples).

We assume reader has already some knowledge about DFW and Security Groups functions. Please refer to the appropriate collateral if you need more information on these NSX components.

This paper is not intended to provide exhaustive list of REST API calls related to DFW.

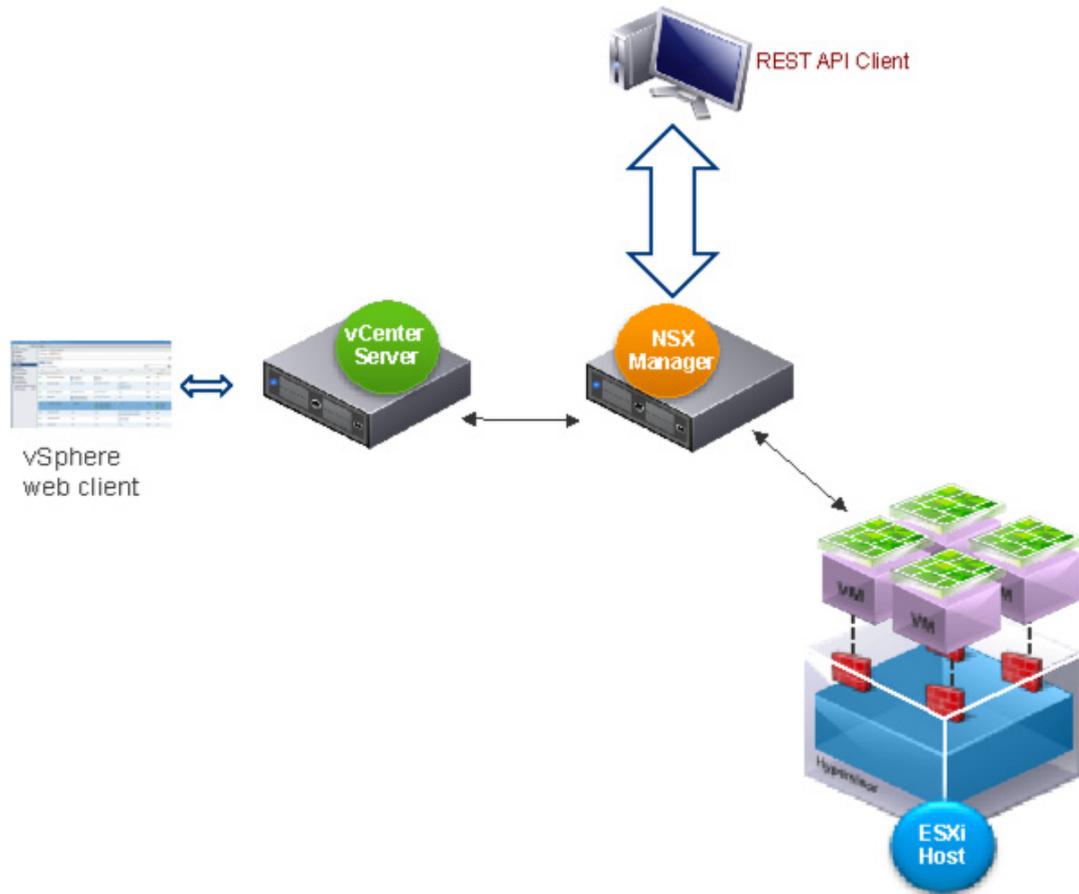
NSX API guide is the reference guide for all REST API calls. Please refer to this guide for any questions related to NSX API in general.

## NSX DFW REST API Functionalities

DFW REST API calls are serviced by NSX Manager. REST API client sends API calls to one unique entity, the NSX manager. All back end actions (like pushing rules to ESXi hosts) will be managed directly by NSX manager itself.

Access to NSX manager for API calls is protected by user authentication (username and password are Base64 encoded and sent as Request Header information).

Diagram below shows the overall architecture:



The following capabilities are supported when using DFW REST API for policy rules configuration:

- 1) List all policy rules configuration (output displays sections + rules).
- 2) List specific section configuration (all rules in the section are displayed).
- 3) List specific policy rule configuration (any rule in the rule table).
- 4) Add new section at the top of policy rule table. New section contains new policy rules.
- 5) Add new policy rule at the top of a specific section.
- 6) Modify specific section (i.e modify rules(s) inside the section).
- 7) Modify specific rule in a section.
- 8) Modify complete policy rules configuration.
- 9) Delete specific section.
- 10) Delete specific policy rule.
- 11) Delete all policy rules.

The following actions are **not** supported:

- a) Rename a section.
- b) Create a new section at a particular place (for instance between section-x and section-y).
- c) Create a new rule at a particular place (for instance between rule-x and rule-y).

- d) Create a new rule at the top of policy rule table out of a user defined section.
- e) When modifying a section, it is not possible to create new rules (or delete existing rules) within the section. Only modification of the existing rules in the section are allowed.
- f) When modifying complete policy configuration, it is not possible to create new rules (or delete existing rules). Only modification of the existing rules in the policy configuration are allowed.
- g) Move existing rule to a new place in the policy rule table.
- h) Modify rule which is not in a specific section.
- i) Add a brand new configuration (use rather export and import DFW configuration capability for this purpose).

Beside policy rules configuration, DFW REST API provides calls to:

- i) List VM defined inside exclusion list.
- ii) Add VM into exclusion list.
- iii) Remove VM from exclusion list.
- iv) List CPU/Memory/CPS configured threshold values.
- v) Set CPU/Memory/CPS threshold values.

#### NSX DFW REST API Call Structure

DFW REST API call structure is based on the following construct:

<METHOD> <URL> <HEADERS> <BODY>

<METHOD> can take the following value:

1. GET for configuration retrieval.
2. POST for new section (with rules) or new rules creation.
3. PUT for existing section (with rules) or existing rule update.
4. DELETE for section or rule removal.

<URL> specifies which part of DFW to focus on. Some examples are:

1. `https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config` <!--deals with global config.-->
2. `https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections/1022` <!--deals with particular section.-->
3. `https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections/1022/rules/1048` <!--deals with particular policy rule.-->

<HEADER> consist of the following parameters:

1. Content-Type: application/xml
2. Authorization: Basic <username/password Base64 encrypted>
3. If-Match: <Etag value>. If-Match is needed only in case of DFW modification (i.e when using POST or PUT method).

<BODY> specifies XML form that is pushed to NSX Manager for section creation/modification or rule creation/modification. An example of BODY structure is displayed below:

```
<rule id="1057" disabled="false" logged="false">
  <name>web VM to db VM</name>
  <action>deny</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <name>CUST2-web-vm-01</name>
      <value>vm-343</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </appliedTo>
    <appliedTo>
      <name>CUST2-db-vm-01</name>
      <value>vm-348</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </appliedTo>
  </appliedToList>
  <sources excluded="false">
    <source>
      <name>CUST2-web-vm-01</name>
      <value>vm-343</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>CUST2-db-vm-01</name>
      <value>vm-348</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </destination>
  </destinations>
</rule>
```

#### Invoking DFW REST API Call

REST API calls can be invoked using web browser REST CLIENT plug-in (RESTClient for FIREFOX for instance) or using curl utility.

When using curl utility, invoke DFW REST API call using the following syntax:

```
curl -k -H 'Content-type: application/xml' -H 'Authorization: Basic YWRtaW46Vk13YXJlMSE='
-X GET https://<nsx-mgr>/api/4.0/firewall/globalroot-0/config
```

-k option turns off curl's verification of SSL certificate.

-H specifies HEADER value.

-X specifies request command (here GET method on DFW configuration URL).

Note: in order to populate correctly the authorization header, use base64 encode utility (online utility available here: <https://www.base64encode.org/>) and enter username:password in the clear field. Click on encode and base64 encrypted value should appear (in our example: YWRtaW46Vk13YXJlMSE=).

## NSX DFW Rule Structure

DFW rule structure is based on the following construct:

```
<rule id="1057" disabled="false" logged="false">
  <name>web VM to db VM</name>
  <action>deny</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <name>CUST2-web-vm-01</name>
      <value>vm-343</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </appliedTo>
    <appliedTo>
      <name>CUST2-db-vm-01</name>
      <value>vm-348</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>1026</sectionId>
  <sources excluded="false">
    <source>
      <name>CUST2-web-vm-01</name>
      <value>vm-343</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>CUST2-db-vm-01</name>
      <value>vm-348</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </destination>
  </destinations>
</rule>
```

\* <rule id> is a unique ID set to identify a particular rule. It is automatically generated by NSX manager once instantiated. When creating a new rule, <rule id> must be set to 0.

\* <name> defines the content of rule name field.

\* <action> defines policy rule action field: allow / deny / reject.

\* <appliedTo> defines the scope of publishing for the rule: VM name, Logical Switch, Cluster, É.

\* <source> defines policy rule source field.

\* <destination> defines policy rule destination field.

<appliedTo>, <source> and <destination> sections require the following fields block:

<name>, <value>, <type> and <isValid>.

\* In case VM name is used, the block looks like the following:

```
<source>
  <name>CUST2-web-vm-01</name>  <!-- name displayed on source field -->
  <value>vm-343</value>  <!-- vm-id for VM CUST2-web-vm-01 -->
  <type>VirtualMachine</type>  <!-- type of the object -->
  <isValid>true</isValid>
</source>
```

vm-id can be retrieved using vCenter Managed Object Base (MOB):

Go to `https://<vCenter server>/mob` and select content -> rootFolder -> childEntity -> vmFolder.

The UI will show all VMs managed by vCenter.

vm-id will appear in association with VM name.

To forge a REST API request to vCenter and get vm-id (with VM name) directly from the body response, use the following METHOD and URL:

```
GET https://vCenter-server-IP-Address/mob/?moid=group-v3
```

\* In case Logical switch is used, the block looks like the following:

```
<source>
  <name>WEB-LS</name>  <!--name displayed on source field-->
  <value>virtualwire-9</value>  <!--identifier for logical switch WEB-LS. Must be
    retrieved from NSX MGR DB-->
  <type>VirtualWire</type>
  <isvalid>true</isvalid>
</source>
```

Please refer to API guide to get Logical Switch information (name and value retrieval).

\* In case Security Group is used, the block looks like the following:

```
<source>
  <name>SG-WEB-1</name>
  <value>securitygroup-21</value>
  <type>SecurityGroup</type>
  <isvalid>true</isvalid>
</source>
```

Please refer to API guide to get Security Group information (name and value retrieval).

## Distributed Firewall (DFW) REST API Call Examples

Let's use the following initial DFW policy rules configuration:

There are 5 pre-created sections:

- 1) IP Section: contains rule using IPv4 subnets as source and destination field value.
- 2) VM Section: contains rule using VM name as source and destination field value.
- 3) LS Section: contains rule using Logical Switch as source and destination field value.
- 4) Security-Group Section: contains rule using Security-Group as source and destination field value.
- 5) Default Section Layer 3: contains default rules.

Note that Applied To field has been used in different ways based on section rule content.

## Distributed Firewall Configuration

**Example 11-1.** List all DFW Policy Rules Configuration.

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
```

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<firewallConfiguration timestamp="1415915640317">
  <contextId>globalroot-0</contextId>
  <layer3Sections>
```

```

<section id="1027" name="IP Section" generationNumber="1415915640317"
  timestamp="1415915640317">
  <rule id="1058" disabled="false" logged="false">
    <name>subnet 1 to subnet 2</name>
    <action>reject</action>
    <notes></notes>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>1027</sectionId>
    <sources excluded="false">
      <source>
        <value>10.1.1.0/24</value>
        <type>Ipv4Address</type>
        <isValid>>true</isValid>
      </source>
    </sources>
    <destinations excluded="false">
      <destination>
        <value>10.2.2.0/24</value>
        <type>Ipv4Address</type>
        <isValid>>true</isValid>
      </destination>
    </destinations>
    <services>
      <service>
        <name>ICMP Echo Reply</name>
        <value>application-337</value>
        <type>Application</type>
        <isValid>>true</isValid>
      </service>
      <service>
        <name>ICMP Echo</name>
        <value>application-70</value>
        <type>Application</type>
        <isValid>>true</isValid>
      </service>
    </services>
  </rule>
</section>
<section id="1026" name="VM Section" generationNumber="1415915640317"
  timestamp="1415915640317">
  <rule id="1057" disabled="false" logged="false">
    <name>web VM to db VM</name>
    <action>deny</action>
    <notes></notes>
    <appliedToList>
      <appliedTo>
        <name>CUST2-web-vm-01</name>
        <value>vm-343</value>
        <type>VirtualMachine</type>
        <isValid>>true</isValid>
      </appliedTo>
      <appliedTo>
        <name>CUST2-db-vm-01</name>
        <value>vm-348</value>
        <type>VirtualMachine</type>
        <isValid>>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>1026</sectionId>
    <sources excluded="false">
      <source>

```

```

        <name>CUST2-web-vm-01</name>
        <value>vm-343</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
    </source>
</sources>
<destinations excluded="false">
    <destination>
        <name>CUST2-db-vm-01</name>
        <value>vm-348</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
    </destination>
</destinations>
</rule>
</section>
<section id="1025" name="LS Section" generationNumber="1415915640317"
    timestamp="1415915640317">
    <rule id="1056" disabled="false" logged="false">
        <name>WEB LS to APP LS</name>
        <action>allow</action>
        <notes></notes>
        <appliedToList>
            <appliedTo>
                <name>APP-LS</name>
                <value>virtualwire-10</value>
                <type>VirtualWire</type>
                <isvalid>true</isvalid>
            </appliedTo>
            <appliedTo>
                <name>WEB-LS</name>
                <value>virtualwire-9</value>
                <type>VirtualWire</type>
                <isvalid>true</isvalid>
            </appliedTo>
        </appliedToList>
        <sectionId>1025</sectionId>
        <sources excluded="false">
            <source>
                <name>WEB-LS</name>
                <value>virtualwire-9</value>
                <type>VirtualWire</type>
                <isvalid>true</isvalid>
            </source>
        </sources>
        <destinations excluded="false">
            <destination>
                <name>APP-LS</name>
                <value>virtualwire-10</value>
                <type>VirtualWire</type>
                <isvalid>true</isvalid>
            </destination>
        </destinations>
        <services>
            <service>
                <name>SSH</name>
                <value>application-223</value>
                <type>Application</type>
                <isvalid>true</isvalid>
            </service>
        </services>
    </rule>
</section>
<section id="1024" name="Security-Group Section" generationNumber="1415915640317"
    timestamp="1415915640317">
    <rule id="1055" disabled="false" logged="false">
        <name>SG-WEB1 to SG-WEB2</name>
        <action>allow</action>

```

```

<notes></notes>
<appliedToList>
  <appliedTo>
    <name>SG-WEB2</name>
    <value>securitygroup-22</value>
    <type>SecurityGroup</type>
    <isValid>true</isValid>
  </appliedTo>
  <appliedTo>
    <name>SG-WEB-1</name>
    <value>securitygroup-21</value>
    <type>SecurityGroup</type>
    <isValid>true</isValid>
  </appliedTo>
</appliedToList>
<sectionId>1024</sectionId>
<sources excluded="false">
  <source>
    <name>SG-WEB-1</name>
    <value>securitygroup-21</value>
    <type>SecurityGroup</type>
    <isValid>true</isValid>
  </source>
</sources>
<destinations excluded="false">
  <destination>
    <name>SG-WEB2</name>
    <value>securitygroup-22</value>
    <type>SecurityGroup</type>
    <isValid>true</isValid>
  </destination>
</destinations>
<services>
  <service>
    <name>HTTPS</name>
    <value>application-315</value>
    <type>Application</type>
    <isValid>true</isValid>
  </service>
  <service>
    <name>HTTP</name>
    <value>application-278</value>
    <type>Application</type>
    <isValid>true</isValid>
  </service>
</services>
</rule>
</section>
<section id="1003" name="Default Section Layer3" generationNumber="1415915640317"
  timestamp="1415915640317">
  <rule id="1004" disabled="false" logged="false">
    <name>Default Rule NDP</name>
    <action>allow</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>1003</sectionId>
    <services>
      <service>
        <name>IPv6-ICMP Neighbor Solicitation</name>
        <value>application-182</value>
        <type>Application</type>
        <isValid>true</isValid>
      </service>
    </services>
  </rule>
</section>

```

```

        </service>
        <service>
          <name>IPv6-ICMP Neighbor Advertisement</name>
          <value>application-128</value>
          <type>Application</type>
          <isValid>true</isValid>
        </service>
      </services>
    </rule>
    <rule id="1003" disabled="false" logged="false">
      <name>Default Rule DHCP</name>
      <action>allow</action>
      <appliedToList>
        <appliedTo>
          <name>DISTRIBUTED_FIREWALL</name>
          <value>DISTRIBUTED_FIREWALL</value>
          <type>DISTRIBUTED_FIREWALL</type>
          <isValid>true</isValid>
        </appliedTo>
      </appliedToList>
      <sectionId>1003</sectionId>
      <services>
        <service>
          <name>DHCP-Server</name>
          <value>application-261</value>
          <type>Application</type>
          <isValid>true</isValid>
        </service>
        <service>
          <name>DHCP-Client</name>
          <value>application-355</value>
          <type>Application</type>
          <isValid>true</isValid>
        </service>
      </services>
    </rule>
    <rule id="1002" disabled="false" logged="true">
      <name>Default Rule</name>
      <action>allow</action>
      <notes></notes>
      <appliedToList>
        <appliedTo>
          <name>DISTRIBUTED_FIREWALL</name>
          <value>DISTRIBUTED_FIREWALL</value>
          <type>DISTRIBUTED_FIREWALL</type>
          <isValid>true</isValid>
        </appliedTo>
      </appliedToList>
      <sectionId>1003</sectionId>
      <precedence>default</precedence>
    </rule>
  </section>
</layer3Sections>
<SKIP L2 default section and traffic redirection section>
<generationNumber>1415915640317</generationNumber>
</firewallConfiguration>

```

---

**Example 11-2.** List Specific Section Configuration (All Rules in the Section are Displayed).

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections
/sectionId
```

Response Body:

```

<section id="1026" name="VM Section" generationNumber="1415915640317"
  timestamp="1415915640317">
  <rule id="1057" disabled="false" logged="false">
    <name>web VM to db VM</name>
    <action>deny</action>
    <notes></notes>
    <appliedToList>
      <appliedTo>
        <name>CUST2-web-vm-01</name>
        <value>vm-343</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
      </appliedTo>
      <appliedTo>
        <name>CUST2-db-vm-01</name>
        <value>vm-348</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
      </appliedTo>
    </appliedToList>
    <sectionId>1026</sectionId>
    <sources excluded="false">
      <source>
        <name>CUST2-web-vm-01</name>
        <value>vm-343</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
      </source>
    </sources>
    <destinations excluded="false">
      <destination>
        <name>CUST2-db-vm-01</name>
        <value>vm-348</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
      </destination>
    </destinations>
  </rule>
</section>

```

---

**Example 11-3.** List Specific Policy Rule Configuration (Any Rule in the Policy Table).

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections
/sectionId/rules/ruleId
```

Response Body:

```

<rule id="1057" disabled="false" logged="false">
  <name>web VM to db VM</name>
  <action>deny</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <name>CUST2-web-vm-01</name>
      <value>vm-343</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </appliedTo>
    <appliedTo>
      <name>CUST2-db-vm-01</name>
      <value>vm-348</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </appliedTo>
  </appliedToList>

```

```

<sources excluded="false">
  <source>
    <name>CUST2-web-vm-01</name>
    <value>vm-343</value>
    <type>VirtualMachine</type>
    <isvalid>true</isvalid>
  </source>
</sources>
<destinations excluded="false">
  <destination>
    <name>CUST2-db-vm-01</name>
    <value>vm-348</value>
    <type>VirtualMachine</type>
    <isvalid>true</isvalid>
  </destination>
</destinations>
</rule>

```

---

**Example 11-4.** Add New Section on Top of Policy Rules Table (New Section Must Contains New Policy Rules).

---

Request:

POST <https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections>

Request Body:

```

<section name="New VM Section">
  <rule disabled="false" logged="false">
    <name>web VM to db VM</name>
    <action>deny</action>
    <notes></notes>
    <appliedToList>
      <appliedTo>
        <name>CUST2-web-vm-01</name>
        <value>vm-343</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
      </appliedTo>
      <appliedTo>
        <name>CUST2-db-vm-01</name>
        <value>vm-348</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
      </appliedTo>
    </appliedToList>
    <sectionId>1026</sectionId>
    <sources excluded="false">
      <source>
        <name>CUST2-web-vm-01</name>
        <value>vm-343</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
      </source>
    </sources>
    <destinations excluded="false">
      <destination>
        <name>CUST2-db-vm-01</name>
        <value>vm-348</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
      </destination>
    </destinations>
  </rule>
</section>

```

---

**Example 11-5.** Add New Policy Rule on Top of a Specific Section.

Request:

POST `https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections/sectionId/rules`

Request Body (set rule-id value to zero):

```
<rule id="0" disabled="false" logged="false">
  <name>new rule-web to db VM</name>
  <action>deny</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <name>web-vm-01</name>
      <value>vm-72</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </appliedTo>
    <appliedTo>
      <name>db-vm-01</name>
      <value>vm-84</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </appliedTo>
  </appliedToList>
  <sectionId>1028</sectionId>
  <sources excluded="false">
    <source>
      <name>web-vm-01</name>
      <value>vm-72</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>db-vm-01</name>
      <value>vm-84</value>
      <type>VirtualMachine</type>
      <isvalid>true</isvalid>
    </destination>
  </destinations>
</rule>
```

**Example 11-6.** Modify a Specific Section (i.e Modify Rules inside the Section).

Request:

PUT `https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections/sectionId`

Request Body:

```
<section id="1028" name="New VM Section" generationNumber="1415925354365"
  timestamp="1415925354365">
  <rule id="1061" disabled="false" logged="false">
    <name>new rule-web to db VM 2</name>
    <action>allow</action>
    <notes></notes>
    <appliedToList>
      <appliedTo>
        <name>web-vm-01</name>
        <value>vm-72</value>
        <type>VirtualMachine</type>
        <isvalid>true</isvalid>
      </appliedTo>
```

```

    <appliedTo>
      <name>db-vm-01</name>
      <value>vm-84</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
</sectionId>1028</sectionId>
<sources excluded="false">
  <source>
    <name>web-vm-01</name>
    <value>vm-72</value>
    <type>VirtualMachine</type>
    <isValid>true</isValid>
  </source>
</sources>
<destinations excluded="false">
  <destination>
    <name>db-vm-01</name>
    <value>vm-84</value>
    <type>VirtualMachine</type>
    <isValid>true</isValid>
  </destination>
</destinations>
</rule>
<rule id="1059" disabled="false" logged="false">
  <name>web VM to db VM 2</name>
  <action>allow</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <name>CUST2-web-vm-01</name>
      <value>vm-343</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </appliedTo>
    <appliedTo>
      <name>CUST2-db-vm-01</name>
      <value>vm-348</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>1028</sectionId>
  <sources excluded="false">
    <source>
      <name>CUST2-web-vm-01</name>
      <value>vm-343</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>CUST2-db-vm-01</name>
      <value>vm-348</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </destination>
  </destinations>
</rule>
</section>

```

---

**Example 11-7.** Modify a Specific Rule in a Section.

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections/sectionId/rules/ruleId>

Request Body:

```
<rule id="1061" disabled="false" logged="false">
  <name>new rule-web to db VM 3</name>
  <action>reject</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <name>web-vm-01</name>
      <value>vm-72</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </appliedTo>
    <appliedTo>
      <name>db-vm-01</name>
      <value>vm-84</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>1028</sectionId>
  <sources excluded="false">
    <source>
      <name>web-vm-01</name>
      <value>vm-72</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>db-vm-01</name>
      <value>vm-84</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </destination>
  </destinations>
</rule>
```

---

### Example 11-8. Modify Complete Policy Rules Configuration

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config>

Request Body:

```
<firewallConfiguration timestamp="1415925061030">
  <contextId>globalroot-0</contextId>
  <layer3Sections>
    <section id="1028" name="New VM Section" generationNumber="1415928741501"
      timestamp="1415928741501">
      <rule id="1061" disabled="false" logged="false">
        <name>new rule-web to db VM 3</name>
        <action>reject</action>
        <notes></notes>
        <appliedToList>
          <appliedTo>
            <name>web-vm-01</name>
            <value>vm-72</value>
            <type>VirtualMachine</type>
            <isValid>true</isValid>
          </appliedTo>
          <appliedTo>
```

```

        <name>db-vm-01</name>
        <value>vm-84</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
    </appliedTo>
</appliedToList>
<sectionId>1028</sectionId>
<sources excluded="false">
    <source>
        <name>web-vm-01</name>
        <value>vm-72</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
    </source>
</sources>
<destinations excluded="false">
    <destination>
        <name>db-vm-01</name>
        <value>vm-84</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
    </destination>
</destinations>
</rule>
<rule id="1059" disabled="false" logged="false">
    <name>web VM to db VM 2</name>
    <action>reject</action>
    <notes></notes>
    <appliedToList>
        <appliedTo>
            <name>CUST2-web-vm-01</name>
            <value>vm-343</value>
            <type>VirtualMachine</type>
            <isValid>true</isValid>
        </appliedTo>
        <appliedTo>
            <name>CUST2-db-vm-01</name>
            <value>vm-348</value>
            <type>VirtualMachine</type>
            <isValid>true</isValid>
        </appliedTo>
    </appliedToList>
    <sectionId>1028</sectionId>
    <sources excluded="false">
        <source>
            <name>CUST2-web-vm-01</name>
            <value>vm-343</value>
            <type>VirtualMachine</type>
            <isValid>true</isValid>
        </source>
    </sources>
    <destinations excluded="false">
        <destination>
            <name>CUST2-db-vm-01</name>
            <value>vm-348</value>
            <type>VirtualMachine</type>
            <isValid>true</isValid>
        </destination>
    </destinations>
</rule>
</section>
<section id="1027" name="IP Section" generationNumber="1415925061030"
    timestamp="1415925061030">
    <rule id="1058" disabled="false" logged="false">
        <name>subnet 1 to subnet 2</name>
        <action>reject</action>
        <notes></notes>
        <appliedToList>

```

```

    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
</sectionId>1027</sectionId>
<sources excluded="false">
  <source>
    <value>10.1.1.0/24</value>
    <type>Ipv4Address</type>
    <isValid>true</isValid>
  </source>
</sources>
<destinations excluded="false">
  <destination>
    <value>10.2.2.0/24</value>
    <type>Ipv4Address</type>
    <isValid>true</isValid>
  </destination>
</destinations>
<services>
  <service>
    <name>ICMP Echo Reply</name>
    <value>application-337</value>
    <type>Application</type>
    <isValid>true</isValid>
  </service>
  <service>
    <name>ICMP Echo</name>
    <value>application-70</value>
    <type>Application</type>
    <isValid>true</isValid>
  </service>
</services>
</rule>
</section>
<section id="1026" name="VM Section" generationNumber="1415925061030"
  timestamp="1415925061030">
  <rule id="1057" disabled="false" logged="false">
    <name>web VM to db VM</name>
    <action>reject</action>
    <notes></notes>
    <appliedToList>
      <appliedTo>
        <name>CUST2-web-vm-01</name>
        <value>vm-343</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
      </appliedTo>
      <appliedTo>
        <name>CUST2-db-vm-01</name>
        <value>vm-348</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>1026</sectionId>
    <sources excluded="false">
      <source>
        <name>CUST2-web-vm-01</name>
        <value>vm-343</value>
        <type>VirtualMachine</type>
        <isValid>true</isValid>
      </source>
    </sources>
    <destinations excluded="false">

```

```

        <destination>
          <name>CUST2-db-vm-01</name>
          <value>vm-348</value>
          <type>VirtualMachine</type>
          <isValid>true</isValid>
        </destination>
      </destinations>
    </rule>
  </section>
  <section id="1025" name="LS Section" generationNumber="1415925061030"
    timestamp="1415925061030">
    <rule id="1056" disabled="false" logged="false">
      <name>WEB LS to APP LS</name>
      <action>reject</action>
      <notes></notes>
      <appliedToList>
        <appliedTo>
          <name>APP-LS</name>
          <value>virtualwire-10</value>
          <type>VirtualWire</type>
          <isValid>true</isValid>
        </appliedTo>
        <appliedTo>
          <name>WEB-LS</name>
          <value>virtualwire-9</value>
          <type>VirtualWire</type>
          <isValid>true</isValid>
        </appliedTo>
      </appliedToList>
      <sectionId>1025</sectionId>
      <sources excluded="false">
        <source>
          <name>WEB-LS</name>
          <value>virtualwire-9</value>
          <type>VirtualWire</type>
          <isValid>true</isValid>
        </source>
      </sources>
      <destinations excluded="false">
        <destination>
          <name>APP-LS</name>
          <value>virtualwire-10</value>
          <type>VirtualWire</type>
          <isValid>true</isValid>
        </destination>
      </destinations>
      <services>
        <service>
          <name>SSH</name>
          <value>application-223</value>
          <type>Application</type>
          <isValid>true</isValid>
        </service>
      </services>
    </rule>
  </section>
  <section id="1024" name="Security-Group Section" generationNumber="1415925061030"
    timestamp="1415925061030">
    <rule id="1055" disabled="false" logged="false">
      <name>SG-WEB1 to SG-WEB2</name>
      <action>reject</action>
      <notes></notes>
      <appliedToList>
        <appliedTo>
          <name>SG-WEB2</name>
          <value>securitygroup-22</value>
          <type>SecurityGroup</type>
          <isValid>true</isValid>

```

```

    </appliedTo>
    <appliedTo>
      <name>SG-WEB-1</name>
      <value>securitygroup-21</value>
      <type>SecurityGroup</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>1024</sectionId>
  <sources excluded="false">
    <source>
      <name>SG-WEB-1</name>
      <value>securitygroup-21</value>
      <type>SecurityGroup</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>SG-WEB2</name>
      <value>securitygroup-22</value>
      <type>SecurityGroup</type>
      <isValid>true</isValid>
    </destination>
  </destinations>
  <services>
    <service>
      <name>HTTPS</name>
      <value>application-315</value>
      <type>Application</type>
      <isValid>true</isValid>
    </service>
    <service>
      <name>HTTP</name>
      <value>application-278</value>
      <type>Application</type>
      <isValid>true</isValid>
    </service>
  </services>
</rule>
</section>
<section id="1003" name="Default Section Layer3" generationNumber="1415925061030"
  timestamp="1415925061030">
  <rule id="1004" disabled="false" logged="false">
    <name>Default Rule NDP</name>
    <action>reject</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>1003</sectionId>
    <services>
      <service>
        <name>IPv6-ICMP Neighbor Solicitation</name>
        <value>application-182</value>
        <type>Application</type>
        <isValid>true</isValid>
      </service>
      <service>
        <name>IPv6-ICMP Neighbor Advertisement</name>
        <value>application-128</value>
        <type>Application</type>
        <isValid>true</isValid>
      </service>
    </services>
  </rule>
</section>

```

```

    </services>
  </rule>
  <rule id="1003" disabled="false" logged="false">
    <name>Default Rule DHCP</name>
    <action>reject</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>1003</sectionId>
    <services>
      <service>
        <name>DHCP-Server</name>
        <value>application-261</value>
        <type>Application</type>
        <isValid>true</isValid>
      </service>
      <service>
        <name>DHCP-Client</name>
        <value>application-355</value>
        <type>Application</type>
        <isValid>true</isValid>
      </service>
    </services>
  </rule>
  <rule id="1002" disabled="false" logged="true">
    <name>Default Rule</name>
    <action>reject</action>
    <notes></notes>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>1003</sectionId>
    <precedence>default</precedence>
  </rule>
</section>
</layer3Sections>
<layer2Sections>
  <section id="1001" name="Default Section Layer2" generationNumber="1415925061030"
    timestamp="1415925061030">
    <rule id="1001" disabled="false" logged="false">
      <name>Default Rule</name>
      <action>allow</action>
      <appliedToList>
        <appliedTo>
          <name>DISTRIBUTED_FIREWALL</name>
          <value>DISTRIBUTED_FIREWALL</value>
          <type>DISTRIBUTED_FIREWALL</type>
          <isValid>true</isValid>
        </appliedTo>
      </appliedToList>
      <sectionId>1001</sectionId>
      <precedence>default</precedence>
    </rule>
  </section>
</layer2Sections>
<layer3RedirectSections>
  <section id="1014" name="WEB traffic" generationNumber="1415925061030"
    timestamp="1415925061030">

```

```

<rule id="1023" disabled="false" logged="false">
  <name>app to db - redirect to PAN</name>
  <action>redirect</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <name>ALL_PROFILE_BINDINGS</name>
      <value>ALL_PROFILE_BINDINGS</value>
      <type>ALL_PROFILE_BINDINGS</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>1014</sectionId>
  <sources excluded="false">
    <source>
      <name>PAN-app-vm2-01</name>
      <value>vm-98</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>PAN-db-vm2-02</name>
      <value>vm-103</value>
      <type>VirtualMachine</type>
      <isValid>true</isValid>
    </destination>
  </destinations>
  <siProfile>
    <objectId>serviceprofile-1</objectId>
    <revision>0</revision>
    <name>Palo Alto Networks profile 1</name>
    <description>ServiceProfile</description>
    <clientHandle></clientHandle>
  </siProfile>
  <siRuleIdList>
    <siRuleId>3629</siRuleId>
  </siRuleIdList>
</rule>
<rule id="1024" disabled="false" logged="false">
  <name>app to db - vNIC - redirect</name>
  <action>redirect</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <name>ALL_PROFILE_BINDINGS</name>
      <value>ALL_PROFILE_BINDINGS</value>
      <type>ALL_PROFILE_BINDINGS</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>1014</sectionId>
  <sources excluded="false">
    <source>
      <name>PAN-app-vm2-01 - Network adapter 1</name>
      <value>50031300-ad53-cc80-f9cb-a97254336c01.000</value>
      <type>Vnic</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>PAN-db-vm2-01 - Network adapter 1</name>
      <value>50032775-5507-ce3d-38e9-e0a63f9059fd.000</value>
      <type>Vnic</type>
      <isValid>true</isValid>
    </destination>
  </destinations>

```

```

</destinations>
<siProfile>
  <objectId>serviceprofile-1</objectId>
  <revision>0</revision>
  <name>Palo Alto Networks profile 1</name>
  <description>ServiceProfile</description>
  <clientHandle></clientHandle>
</siProfile>
<siRuleIdList>
  <siRuleId>3625</siRuleId>
</siRuleIdList>
</rule>
<rule id="1025" disabled="false" logged="false">
  <name>app to db - vNIC - red2</name>
  <action>redirect</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <name>ALL_PROFILE_BINDINGS</name>
      <value>ALL_PROFILE_BINDINGS</value>
      <type>ALL_PROFILE_BINDINGS</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>1014</sectionId>
  <sources excluded="false">
    <source>
      <name>PAN-app-vm2-02 - Network adapter 1</name>
      <value>50030837-6f65-1ca5-f281-1427423c0dbd.000</value>
      <type>vnic</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>PAN-db-vm2-02 - Network adapter 1</name>
      <value>5003a674-7edf-1502-98bd-d64b93338e4b.000</value>
      <type>vnic</type>
      <isValid>true</isValid>
    </destination>
  </destinations>
  <siProfile>
    <objectId>serviceprofile-1</objectId>
    <revision>0</revision>
    <name>Palo Alto Networks profile 1</name>
    <description>ServiceProfile</description>
    <clientHandle></clientHandle>
  </siProfile>
  <siRuleIdList>
    <siRuleId>3621</siRuleId>
  </siRuleIdList>
</rule>
<rule id="1019" disabled="false" logged="false">
  <name>rediret WEB traffic to PAN</name>
  <action>redirect</action>
  <notes></notes>
  <appliedToList>
    <appliedTo>
      <name>ALL_PROFILE_BINDINGS</name>
      <value>ALL_PROFILE_BINDINGS</value>
      <type>ALL_PROFILE_BINDINGS</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>1014</sectionId>
  <destinations excluded="false">
    <destination>
      <name>SG-PAN-WEB</name>

```

```

        <value>securitygroup-17</value>
        <type>SecurityGroup</type>
        <isValid>true</isValid>
    </destination>
</destinations>
<services>
    <service>
        <name>SSH</name>
        <value>application-223</value>
        <type>Application</type>
        <isValid>true</isValid>
    </service>
    <service>
        <name>ICMP Echo Reply</name>
        <value>application-337</value>
        <type>Application</type>
        <isValid>true</isValid>
    </service>
    <service>
        <name>HTTPS</name>
        <value>application-315</value>
        <type>Application</type>
        <isValid>true</isValid>
    </service>
    <service>
        <name>HTTP</name>
        <value>application-278</value>
        <type>Application</type>
        <isValid>true</isValid>
    </service>
    <service>
        <name>ICMP Echo</name>
        <value>application-70</value>
        <type>Application</type>
        <isValid>true</isValid>
    </service>
    <service>
        <name>HTTP-8080</name>
        <value>application-371</value>
        <type>Application</type>
        <isValid>true</isValid>
    </service>
</services>
<siProfile>
    <objectId>serviceprofile-1</objectId>
    <revision>0</revision>
    <name>Palo Alto Networks profile 1</name>
    <description>ServiceProfile</description>
    <clientHandle></clientHandle>
</siProfile>
<siRuleIdList>
    <siRuleId>3519</siRuleId>
    <siRuleId>3523</siRuleId>
    <siRuleId>3526</siRuleId>
    <siRuleId>3530</siRuleId>
    <siRuleId>3533</siRuleId>
    <siRuleId>3537</siRuleId>
</siRuleIdList>
</rule>
</section>
<section id="1002" name="Default section" generationNumber="1415925061030"
    timestamp="1415925061030" />
</layer3RedirectSections>
<generationNumber>1415928741501</generationNumber>
</firewallConfiguration>

```

**Example 11-9.** Delete Specific Section.

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections
/sectionId
```

**Example 11-10.** Delete Specific Policy Rule.

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config/layer3sections
/sectionId/rules/ruleId
```

**Example 11-11.** Delete All Policy Rules.

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/config
```

## Distributed Firewall Exclusions

**Example 11-12.** List VM Defined inside DFW Exclusion List.

Request:

```
GET https://NSX-Manager-IP-Address/api/2.1/app/excludelist
```

Response Body:

```
<vshieldAppConfiguration>
  <excludelistConfiguration>
    <objectId>excludelist-1</objectId>
    <objectTypeName>Excludelist</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>5</revision>
    <type>
      <typeName>Excludelist</typeName>
    </type>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <excludemember>
      <member>
        <objectId>vm-400</objectId>
        <objectTypeName>VirtualMachine</objectTypeName>
        <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
        <revision>11</revision>
        <type>
          <typeName>VirtualMachine</typeName>
        </type>
        <name>B03-vm-01</name>
        <scope>
          <id>resgroup-412</id>
          <objectTypeName>ResourcePool</objectTypeName>
          <name>VM</name>
        </scope>
        <clientHandle></clientHandle>
        <extendedAttributes></extendedAttributes>
      </member>
    </excludemember>
    <excludemember>
      <member>
        <objectId>vm-396</objectId>
```

```

    <objectTypeName>VirtualMachine</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>11</revision>
    <type>
      <typeName>VirtualMachine</typeName>
    </type>
    <name>B02-vm-01</name>
    <scope>
      <id>resgroup-410</id>
      <objectTypeName>ResourcePool</objectTypeName>
      <name>VM</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </member>
</excludeMember>
<excludeMember>
  <member>
    <objectId>vm-398</objectId>
    <objectTypeName>VirtualMachine</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>10</revision>
    <type>
      <typeName>VirtualMachine</typeName>
    </type>
    <name>B02-vm-02</name>
    <scope>
      <id>resgroup-410</id>
      <objectTypeName>ResourcePool</objectTypeName>
      <name>VM</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </member>
</excludeMember>
</excludeListConfiguration>
</VshieldAppConfiguration>

```

---

#### Example 11-13. Add VM into Exclusion List.

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.1/app/excludelist/VM-Id
```

---

#### Example 11-14. Remove VM from Exclusion List.

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.1/app/excludelist/VM-Id
```

---

## CPU/Memory/CPS Configuration

#### Example 11-15. List CPU/Memory/CPS Configured Threshold Values.

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/stats/eventthresholds
```

Response Body:

```

<eventThresholds>
  <cpu>
    <percentValue>2</percentValue>
  </cpu>

```

```

    <memory>
      <percentValue>2</percentValue>
    </memory>
    <connectionsPerSecond>
      <value>2</value>
    </connectionsPerSecond>
  </eventThresholds>

```

---

#### Example 11-16. Set CPU/Memory/CPS Threshold Values.

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/firewall/stats/eventthresholds>

Request Body:

```

<eventThresholds>
  <cpu>
    <percentValue>10</percentValue>
  </cpu>
  <memory>
    <percentValue>20</percentValue>
  </memory>
  <connectionsPerSecond>
    <value>300000</value>
  </connectionsPerSecond>
</eventThresholds>

```

---

## Security Groups

#### Example 11-17. List All Security Groups.

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/scope/globalroot-0>

Response Body:

```

<list>
  <securitygroup>
    <objectId>securitygroup-22</objectId>
    <objectTypeName>SecurityGroup</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>2</revision>
    <type>
      <typeName>SecurityGroup</typeName>
    </type>
    <name>SG-WEB2</name>
    <description></description>
    <scope>
      <id>globalroot-0</id>
      <objectTypeName>GlobalRoot</objectTypeName>
      <name>Global</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <inheritanceAllowed>>false</inheritanceAllowed>
    <member>
      <objectId>vm-77</objectId>
      <objectTypeName>VirtualMachine</objectTypeName>
      <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
      <revision>42</revision>
      <type>
        <typeName>VirtualMachine</typeName>
      </type>
    </member>
  </securitygroup>
</list>

```

```

    <name>web-vm-02</name>
    <scope>
      <id>domain-c26</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>VXLAN-COMPUTE-1</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </member>
</securitygroup>
<securitygroup>
  <objectId>securitygroup-21</objectId>
  <objectTypeName>SecurityGroup</objectTypeName>
  <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
  <revision>2</revision>
  <type>
    <typeName>SecurityGroup</typeName>
  </type>
  <name>SG-WEB-1</name>
  <description></description>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>>false</inheritanceAllowed>
  <member>
    <objectId>vm-72</objectId>
    <objectTypeName>VirtualMachine</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>18</revision>
    <type>
      <typeName>VirtualMachine</typeName>
    </type>
    <name>web-vm-01</name>
    <scope>
      <id>domain-c26</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>VXLAN-COMPUTE-1</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </member>
</securitygroup>
<securitygroup>
  <objectId>securitygroup-1</objectId>
  <objectTypeName>SecurityGroup</objectTypeName>
  <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
  <revision>5</revision>
  <type>
    <typeName>SecurityGroup</typeName>
  </type>
  <name>Activity Monitoring Data Collection</name>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>>false</inheritanceAllowed>
</securitygroup>
</list>

```

**Example 11-18.** List All Members of a Security Group.

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/securitygroupId`

Response Body:

```

<securitygroup>
  <objectId>securitygroup-22</objectId>
  <objectTypeName>SecurityGroup</objectTypeName>
  <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
  <revision>2</revision>
  <type>
    <typeName>SecurityGroup</typeName>
  </type>
  <name>SG-WEB2</name>
  <description></description>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>false</inheritanceAllowed>
  <member>
    <objectId>vm-77</objectId>
    <objectTypeName>VirtualMachine</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>42</revision>
    <type>
      <typeName>VirtualMachine</typeName>
    </type>
    <name>web-vm-02</name>
    <scope>
      <id>domain-c26</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>VXLAN-COMPUTE-1</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </member>
</securitygroup>

```

**Example 11-19.** List only VM Members of a Security Group.

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/securitygroupId/translation/virtualmachines`

Response Body:

```

<vmnodes>
  <vmnode>
    <vmId>vm-77</vmId>
    <vmName>web-vm-02</vmName>
  </vmnode>
</vmnodes>

```

Note: to retrieve IP addresses, MAC addresses and vNIC information known by a Security Group, use the following REST API calls:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/securitygroupId
    /translation/ipaddresses

GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/securitygroupId
    /translation/macaddresses

GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/securitygroupId
    /translation/vnics
```

---

**Example 11-20.** Add New Member into Security Group.

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/securitygroupId/members
    /member-moref
```

(member-moref is obtained using vCenter MOB)

---

**Example 11-21.** Delete Member from Security Group.

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/securitygroupId
    /members/member-moref
```

---

**Example 11-22.** Add Security Group.

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/securitygroup//bulk/globalroot-0
```

Request Body:

```
<securitygroup>
  <objectId />
  <objectTypeName>SecurityGroup</objectTypeName>
  <type>
    <typeName>SecurityGroup</typeName>
  </type>
  <name>SG-WEB-24</name>
  <description></description>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>false</inheritanceAllowed>
  <member>
    <objectId>vm-84</objectId>
    <objectTypeName>VirtualMachine</objectTypeName>
    <type>
      <typeName>VirtualMachine</typeName>
    </type>
    <name>db-vm-01</name>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
  </member>
</securitygroup>
```

---

**Example 11-23.** Delete Security Group.

Request:

DELETE <https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/securitygroupId>**Grouping Objects using IPsets****Example 11-24.** List All IPsets.

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/ipset/scope/globalroot-0>

Response Body:

```

<list>
  <ipset>
    <objectId>ipset-2</objectId>
    <objectTypeName>IPSet</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>1</revision>
    <type>
      <typeName>IPSet</typeName>
    </type>
    <name>dmz_app1_web</name>
    <description></description>
    <scope>
      <id>globalroot-0</id>
      <objectTypeName>GlobalRoot</objectTypeName>
      <name>Global</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <inheritanceAllowed>>false</inheritanceAllowed>
    <value>192.168.201.100/24</value>
  </ipset>
  <ipset>
    <objectId>ipset-3</objectId>
    <objectTypeName>IPSet</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>1</revision>
    <type>
      <typeName>IPSet</typeName>
    </type>
    <name>dmz_app1_db</name>
    <description></description>
    <scope>
      <id>globalroot-0</id>
      <objectTypeName>GlobalRoot</objectTypeName>
      <name>Global</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <inheritanceAllowed>>false</inheritanceAllowed>
    <value>192.168.200.50/24</value>
  </ipset>
  <ipset>
    <objectId>ipset-1</objectId>
    <objectTypeName>IPSet</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>2</revision>
    <type>
      <typeName>IPSet</typeName>
    </type>
    <name>sys-gen-empty-ipset-edge-fw</name>

```

```

<scope>
  <id>globalroot-0</id>
  <objectTypeName>GlobalRoot</objectTypeName>
  <name>Global</name>
</scope>
<clientHandle></clientHandle>
<extendedAttributes>
  <extendedAttribute>
    <name>isReadOnly</name>
    <value>true</value>
  </extendedAttribute>
</extendedAttributes>
<inheritanceAllowed>true</inheritanceAllowed>
</ipset>
</list>

```

---

#### Example 11-25. List Specific IP Sets.

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/ipset/ipsetId>

Response Body:

```

<ipset>
  <objectId>ipset-2</objectId>
  <objectTypeName>IPSet</objectTypeName>
  <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
  <revision>1</revision>
  <type>
    <typeName>IPSet</typeName>
  </type>
  <name>dmz_app1_web</name>
  <description></description>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>false</inheritanceAllowed>
  <value>192.168.201.100/24</value>
</ipset>

```

---

#### Example 11-26. Modify IP Sets.

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/services/ipset/ipsetId>

Request Body:

```

<ipset>
  <objectId>ipset-2</objectId>
  <objectTypeName>IPSet</objectTypeName>
  <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
  <revision>6</revision>
  <type>
    <typeName>IPSet</typeName>
  </type>
  <name>dmz_app1_web</name>
  <description></description>
  <scope>
    <id>globalroot-0</id>

```

```

    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>false</inheritanceAllowed>
  <value>192.168.201.100/24,192.168.202.100/24,192.168.203.100/24</value>
</ipset>

```

---

#### Example 11-27. Add IP Sets.

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/services/ipset/globalroot-0>

Request Body:

```

<ipset>
  <name>dmz_app2_web</name>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>false</inheritanceAllowed>
  <value>192.168.23.1/24</value>
</ipset>

```

---

#### Example 11-28. Delete IP Sets.

---

Request:

DELETE <https://NSX-Manager-IP-Address/api/2.0/services/ipset/ipsetId>

---

## Grouping Objects using MAC Sets

#### Example 11-29. List All MAC Sets.

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/macset/scope/globalroot-0>

Response Body:

```

<list>
  <macset>
    <objectId>macset-2</objectId>
    <objectTypeName>MACSet</objectTypeName>
    <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
    <revision>1</revision>
    <type>
      <typeName>MACSet</typeName>
    </type>
    <name>macsets-1</name>
    <description></description>
    <scope>
      <id>globalroot-0</id>
      <objectTypeName>GlobalRoot</objectTypeName>
      <name>Global</name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <inheritanceAllowed>false</inheritanceAllowed>
    <value>11:22:33:44:55:66</value>
  </macset>
  <macset>
    <objectId>macset-1</objectId>
    <objectTypeName>MACSet</objectTypeName>

```

```

<vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
<revision>4</revision>
<type>
  <typeName>MACSet</typeName>
</type>
<name>system-generated-broadcast-macset</name>
<scope>
  <id>globalroot-0</id>
  <objectTypeName>GlobalRoot</objectTypeName>
  <name>Global</name>
</scope>
<clientHandle></clientHandle>
<extendedAttributes>
  <extendedAttribute>
    <name>isReadOnly</name>
    <value>true</value>
  </extendedAttribute>
  <extendedAttribute>
    <name>isHidden</name>
    <value>true</value>
  </extendedAttribute>
  <extendedAttribute>
    <name>facadeHidden</name>
    <value>true</value>
  </extendedAttribute>
</extendedAttributes>
<inheritanceAllowed>>false</inheritanceAllowed>
<value>FF:FF:FF:FF:FF:FF</value>
</macset>
<macset>
  <objectId>macset-3</objectId>
  <objectTypeName>MACSet</objectTypeName>
  <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
  <revision>1</revision>
  <type>
    <typeName>MACSet</typeName>
  </type>
  <name>macsets-2</name>
  <description></description>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>>false</inheritanceAllowed>
  <value>aa:bb:cc:dd:ee:ff</value>
</macset>
</list>

```

---

### Example 11-30. List Specific MACSets.

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/macset/macsetId>

Response Body:

```

<macset>
  <objectId>macset-2</objectId>
  <objectTypeName>MACSet</objectTypeName>
  <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
  <revision>1</revision>
  <type>
    <typeName>MACSet</typeName>
  </type>

```

```

<name>macsets-1</name>
<description></description>
<scope>
  <id>globalroot-0</id>
  <objectTypeName>GlobalRoot</objectTypeName>
  <name>Global</name>
</scope>
<clientHandle></clientHandle>
<extendedAttributes></extendedAttributes>
<inheritanceAllowed>false</inheritanceAllowed>
<value>11:22:33:44:55:66</value>
</macset>

```

---

### Example 11-31. Modify MACSets.

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/services/macset/macsetId>

Request Body:

```

<macset>
  <objectId>macset-2</objectId>
  <objectTypeName>MACSet</objectTypeName>
  <vsmUuid>420315E0-3430-03EC-8CFF-3C3425CA17EB</vsmUuid>
  <revision>2</revision>
  <type>
    <typeName>MACSet</typeName>
  </type>
  <name>macsets-1</name>
  <description></description>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>false</inheritanceAllowed>
  <value>11:22:33:44:55:66,11:22:33:44:55:77</value>
</macset>

```

---

### Example 11-32. Add MACSets.

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/services/macset/globalroot-0>

Request Body:

```

<macset>
  <name>macsets-3</name>
  <extendedAttributes></extendedAttributes>
  <inheritanceAllowed>false</inheritanceAllowed>
  <value>33:33:33:33:33:33</value>
</macset>

```

---

### Example 11-33. Delete MACSets.

---

Request:

DELETE <https://NSX-Manager-IP-Address/api/2.0/services/macset/macsetId>

---

# Service Composer Management

Service Composer helps you provision and assign network and security services to applications in a virtual infrastructure. You map these services to a security group, and the services are applied to the virtual machines in the security group.

## Security Group

You begin by creating a security group to define assets that you want to protect. Security groups may be static (including specific virtual machines) or dynamic where membership may be defined in one or more of the following ways:

- vCenter containers (clusters, port groups, or datacenters)
- Security tags, IPset, MACset, or even other security groups. For example, you may include a criteria to add all members tagged with the specified security tag (such as AntiVirus.virusFound) to the security group.
- Directory Groups (if NSX Manager is registered with Active Directory)
- Regular expressions such as virtual machines with name *VM1*

Note that security group membership changes constantly. For example, a virtual machine tagged with the AntiVirus.virusFound tag is moved into the Quarantine security group. When the virus is cleaned and this tag is removed from the virtual machine, it again moves out of the Quarantine security group.

## Security Policy

A security policy is a collection of the following service configurations.

**Table 12-1.** Security services contained in a security policy

Service	Description	Applies to
Firewall rules	Rules that define the traffic to be allowed to, from, or within the security group.	vNIC
Endpoint service	Data Security or third party solution provider services such as anti-virus or vulnerability management services.	virtual machines
Network introspection services	Services that monitor your network such as IPS.	virtual machines

## Applying Security Policy to Security Group

You apply a security policy (say SP1) to a security group (say SG1). The services configured for SP1 are applied to all virtual machines that are members of SG1.

If a virtual machine belongs to more than one security group, the services that are applied to the virtual machine depends on the precedence of the security policy mapped to the security groups.

Service Composer profiles can be exported and imported as backups or for use in other environments. This approach to managing network and security services helps you with actionable and repeatable security policy management.

This chapter includes the following topics:

- [“Working with Security Policies”](#) on page 380
- [“Default Applied To Value for Firewall Rules”](#) on page 388
- [“Working with Security Actions”](#) on page 389
- [“Synchronizing Service Composer Rules with Distributed Firewall”](#) on page 394
- [“Query Security Policies Mapped to a Security Group”](#) on page 395
- [“Query Service Provider Data”](#) on page 396
- [“Query Security Group Effective Membership”](#) on page 396
- [“Query Security Groups to which a VM Belongs”](#) on page 396

---

**IMPORTANT** All NSX vSphere REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authentication.

---

## Working with Security Policies

A security policy is a set of Endpoint, firewall, and network introspection services that can be applied to a security group.

For information on creating a security group, see [“Working with Security Groups”](#) on page 93.

### Creating a Security Policy

When creating a security policy, a parent security policy can be specified if required. The security policy inherits services from the parent security policy. Security group bindings and actions can also be specified while creating the policy. Note that execution order of actions in a category is implied by their order in the list. The response of the call has Location header populated with the URI using which the created object can be fetched.

#### Prerequisites

Ensure that:

- the required VMware built in services (such as Distributed Firewall, Data Security, and Endpoint) are installed. See *NSX Installation and Upgrade Guide*.
- the required partner services have been registered with NSX Manager.
- the required security groups have been created.

#### Example 12-1. Create security policy

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy
```

Request Body:

```
<securityPolicy>
  <name>name</name>
  <description>description</description>
  <precedence></precedence>
  <parent>
    <objectId></objectId>
  </parent>
  <securityGroupBinding>
    <objectId></objectId>
  </securityGroupBinding>
  <securityGroupBinding>
    ...
  </securityGroupBinding>
  ...
</securityPolicy>
```

```

...
<securityGroupBinding>
  ...
</securityGroupBinding>
<actionsByCategory>
  <category>firewall</category>
  <action class="firewallSecurityAction">
    <name>name</name>
    <description>description</description>
    <category></category>
    <actionType></actionType>
    <isActionEnforced></isActionEnforced>
    <isActive></isActive>
    <isEnabled></isEnabled>
    <secondarySecurityGroup>
      <objectId></objectId>
    </secondarySecurityGroup>
    <secondarySecurityGroup>
      ...
    </secondarySecurityGroup>
    ...
    <secondarySecurityGroup>
      ...
    </secondarySecurityGroup>
    <applications>
      <application>
        <objectId></objectId>
      </application>
      <applicationGroup>
        <objectId></objectId>
      </applicationGroup>
      ...
    </applications>
    <logged></logged>
    <action></action>
    <direction></direction>
    <outsideSecondaryContainer></outsideSecondaryContainer>
  </action>
  <action>
    ...
  </action>
  ...
  <action>
    ...
  </action>
</actionsByCategory>
<actionsByCategory>
  <category>endpoint</category>
  <action class="endpointSecurityAction">
    <name>name</name>
    <description>description</description>
    <category></category>
    <actionType></actionType>
    <isActionEnforced></isActionEnforced>
    <isActive></isActive>
    <isEnabled></isEnabled>
    <serviceId></serviceId>
    <serviceProfile>
      <objectId>serviceprofile-1</objectId>
      ...
    </serviceProfile>
    <invalidServiceProfile>>false</invalidServiceProfile>
  </action>
</actionsByCategory>
<actionsByCategory>

```

```

    <category>traffic_steering</category>
    <action class="trafficSteeringSecurityAction">
      <name>name</name>
      <description>description</description>
      <category></category>
      <actionType></actionType>
      <isActionEnforced></isActionEnforced>
      <isActive></isActive>
      <isEnabled></isEnabled>
      <logged></logged>
      <redirect></redirect>
      <serviceProfile>
        <objectId></objectId>
      </serviceProfile>
    </action>
  </actionsByCategory>
</securityPolicy>

```

---

## Description of Tags

This section describes the tags specific to Service Composer management.

### Common Tags

- `actionType` - Defines the type of action belonging to a given `executionOrderCategory`
- `executionOrderCategory` - Category to which the action belongs to (endpoint, firewall or `traffic_steering`)
- `isActive` - In a security policy hierarchy, an action within a policy may or may not be active based on the precedence of the policy or usage of `isActionEnforced` flag in that hierarchy
- `isActionEnforced` - Enforces an action of a parent policy on its child policies for a given `actionType` and `executionOrderCategory`. Note that in a policy hierarchy, for a given `actionType` and `executionOrderCategory`, there can be only one action which can be marked as enforced.
- `isEnabled` - Indicates whether an action is enabled
- `secondarySecurityGroup` - Applicable for actions which need secondary security groups, say a source-destination firewall rule
- `securityPolicy` - Parent policy in an action

### Output only Tags

- `executionOrder` - Defines the sequence in which actions belonging to an `executionOrderCategory` are executed. Note that this is not an input parameter and its value is implied by the index in the list.

### Firewall Category Tags

- `action` - Allow or block the traffic
- `applications` - Applications / application groups on which the rules are to be applied
- `direction` - Direction of traffic towards primary security group. Possible values: inbound, outbound, intra
- `logged` - Flag to enable logging of the traffic that is hit by this rule
- `outsideSecondaryContainer` - Flag to specify outside i.e. outside securitygroup-3

### Endpoint Category Tags

- `serviceId` - ID of the service (as registered with the service insertion module). If this tag is null, the functionality type (as defined in `actionType` tag) is not applied which will also result in blocking the actions (of given functionality type) that are inherited from the parent security policy. This is true if there is no action of enforce type.

- `invalidServiceId` - Flag to indicate that the service that was referenced in this rule is deleted, which make the rule ineffective (or deviate from the original intent that existed while configuring the rule). You must either modify this rule by adding correct Service or delete this rule.
- `serviceName` -Name of the service
- `serviceProfile` - Profile to be referenced in Endpoint rule.
- `invalidServiceProfile` - Flag to indicate that the service profile that was referenced in this rule is deleted, which makes the rule ineffective (or deviate from the original intent that existed while configuring the rule). You must either modify this rule by adding correct Service Profile or delete this rule.

The following tags are deprecated:

- `vendorTemplateId`
- `invalidVendorTemplateId`
- `vendorTemplateName`

### Traffic Steering/NetX Category Tags

- `redirect` - Flag to indicate whether to redirect the traffic or not
- `serviceProfile` - Service profile for which redirection is being configured
- `logged` - Flag to enable logging of the traffic that is hit by this rule

## Querying Security Policies

You can retrieve a specific security policy by specifying its ID or all security policies.

### Example 12-2. Query security policies

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy/objectId|a11
```

Response Body:

```
<securityPolicy>
  <securityPolicy>
    <name>name</name>
    <description>description</description>
    <precedence></precedence>
    <parent>
      <objectId></objectId>
    </parent>
    <securityGroupBinding>
      <objectId></objectId>
    </securityGroupBinding>
    <securityGroupBinding>
      ...
    </securityGroupBinding>
    ...
    <securityGroupBinding>
      ...
    </securityGroupBinding>
    <actionsByCategory>
      <category>firewall</category>
      <action class="firewallSecurityAction">
        <name>name</name>
        <description>description</description>
        <category></category>
        <actionType></actionType>
        <isActionEnforced></isActionEnforced>
        <isActive></isActive>
        <isEnabled></isEnabled>
      </action>
    </actionsByCategory>
  </securityPolicy>
</securityPolicy>
```

```

    <secondarySecurityGroup>
      <objectId></objectId>
    </secondarySecurityGroup>
  <secondarySecurityGroup>
    ...
  </secondarySecurityGroup>
  ...
  ...
  <secondarySecurityGroup>
    ...
  </secondarySecurityGroup>
  <applications>
    <application>
      <objectId></objectId>
    </application>
    <applicationGroup>
      <objectId></objectId>
    </applicationGroup>
    ...
    ...
  </applications>
  <logged></logged>
  <action></action>
  <direction></direction>
  <outsideSecondaryContainer></outsideSecondaryContainer>
</action>
<action>
  ...
</action>
...
...
<action>
  ...
</action>
</actionsByCategory>
<actionsByCategory>
  <category>endpoint</category>
  <action class="endpointSecurityAction">
    <name>name</name>
    <description>description</description>
    <category></category>
    <actionType></actionType>
    <isActionEnforced></isActionEnforced>
    <isActive></isActive>
    <isEnabled></isEnabled>
    <serviceId></serviceId>
    <vendorTemplateId></vendorTemplateId>
  </action>
</actionsByCategory>
<actionsByCategory>
  <category>traffic_steering</category>
  <action class="trafficSteeringSecurityAction">
    <name>name</name>
    <description>description</description>
    <category></category>
    <actionType></actionType>
    <isActionEnforced></isActionEnforced>
    <isActive></isActive>
    <isEnabled></isEnabled>
    <logged></logged>
    <redirect></redirect>
    <serviceProfile>
      <objectId></objectId>
    </serviceProfile>
  </action>
</actionsByCategory>
</securityPolicy>
<name></name>

```

```

</description></description>
</precedence></precedence>
</parent>
  <objectId></objectId>
</parent>
</securityGroupBinding>
  <objectId></objectId>
</securityGroupBinding>
</securityGroupBinding>
  ...
</securityGroupBinding>
  ...
  ...
</securityGroupBinding>
  ...
</securityGroupBinding>
</actionsByCategory>
  <category>firewall</category>
  <action class="firewallSecurityAction">
    <name>name</name>
    <description>description</description>
    <category></category>
    <actionType></actionType>
    <isActionEnforced></isActionEnforced>
    <isActive></isActive>
    <isEnabled></isEnabled>
    <secondarySecurityGroup>
      <objectId></objectId>
    </secondarySecurityGroup>
    <secondarySecurityGroup>
      ...
    </secondarySecurityGroup>
    ...
    ...
    <secondarySecurityGroup>
      ...
    </secondarySecurityGroup>
    <applications>
      <application>
        <objectId></objectId>
      </application>
      <applicationGroup>
        <objectId></objectId>
      </applicationGroup>
      ...
      ...
    </applications>
    <logged></logged>
    <action></action>
    <direction></direction>
    <outsideSecondaryContainer></outsideSecondaryContainer>
  </action>
  <action>
    ...
  </action>
  ...
  ...
  <action>
    ...
  </action>
</actionsByCategory>
</actionsByCategory>
  <category>endpoint</category>
  <action class="endpointSecurityAction">
    <name>name</name>
    <description>description</description>
    <category></category>
    <actionType></actionType>

```

```

        <isActionEnforced></isActionEnforced>
        <isActive></isActive>
        <isEnabled></isEnabled>
        <serviceId></serviceId>
        <vendorTemplateId></vendorTemplateId>
    </action>
</actionsByCategory>
<actionsByCategory>
    <category>traffic_steering</category>
    <action class="trafficSteeringSecurityAction">
        <name>name</name>
        <description>description</description>
        <category></category>
        <actionType></actionType>
        <isActionEnforced></isActionEnforced>
        <isActive></isActive>
        <isEnabled></isEnabled>
        <logged></logged>
        <redirect></redirect>
        <serviceProfile>
            <objectId></objectId>
        </serviceProfile>
    </action>
</actionsByCategory>
</securityPolicy>

```

---

## Edit a Security Policy

To update a security policy, you must first fetch it. For more information, see [Querying Security Policies](#).

You then edit the received XML and pass it back as the input. The specified configuration replaces the current configuration.

Security group mappings provided in the PUT call replaces the security group mappings for the security policy. To remove all mappings, delete the *securityGroupBindings* parameter.

You can add or update actions for the security policy by editing the *actionsByCategory* parameter. To remove all actions (belonging to all categories), delete the *actionsByCategory* parameter. To remove actions belonging to a specific category, delete the block for that category.

### Example 12-3. Edit a security policy

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy/objectId
```

Response Body:

See [Example 12-2](#).

---

## Delete a Security Policy

When you delete a security policy, its child security policies and all the actions in it are deleted as well.

### Example 12-4. Delete a security policy

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy/objectId
?force=true/false
```

---

If you set the *force* parameter to true, the security policy is deleted even if it is being used somewhere.

## Export a Security Policy Configuration

You can export a Service Composer configuration (along with the security groups to which the security policies are mapped) and save it to your desktop. The saved configuration can be used as a backup for situations where you may accidentally delete a policy configuration, or it can be exported for use in another NSX Manager environment.

---

### Example 12-5. Export a security policy

Request for selective export:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy/hierarchy?policyIds=comma_separated_securitypolicy_ids&prefix=optional_some_prefix_before_names
```

Request for exporting all policies:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/hierarchy?prefix=optional_some_prefix_before_names
```

Response Body:

```
<securityPolicyHierarchy>
  <name>name</name>
  <description>description</description>
  <securityPolicy></securityPolicy>
  <securityPolicy></securityPolicy>
  ...
  ...
  <securityPolicy></securityPolicy>
  <securityGroup></securityGroup>
  <securityGroup></securityGroup>
  ...
  ...
  <securityGroup></securityGroup>
</securityPolicyHierarchy>
```

---

If a prefix is specified, it is added before the names of the security policy, security action, and security group objects in the exported XML. The prefix can thus be used to indicate the remote source from where the hierarchy was exported.

## Import a Security Policy Configuration

You can create multiple security policies and parent-child hierarchies using the data fetched through export. All objects including security policies, security groups and security actions are created on a global scope.

---

### Example 12-6. Import a security policy

Request for selective export:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy/hierarchy?suffix=optional_suffix_to_be_added_after_names
```

Request Body:

See [Example 12-5](#).

---

The policy that is being imported needs to be included as a payload (request body) with the request.

If a suffix is specified, it is added after the names of the security policy, security action, and security group objects in the exported XML. The suffix can thus be used to differentiate locally created objects from imported ones.

Location of the newly created security policy objects (multiple locations are separated by commas) is populated in the Location header of the response.

## Query Security Actions for a Security Policy

You can retrieve all security actions applicable on a security policy. This list includes security actions from associated parent security policies, if any. Security actions per Execution Order Category are sorted based on the weight of security actions in descending order.

### Example 12-7. Query security actions for a security policy

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy/
    /securitypolicyId/securityactions
```

Response Body:

```
<securityPolicies>
  <securityPolicy></securityPolicy>
  <securityPolicy></securityPolicy>
  ...
  ...
  <securityPolicy></securityPolicy>
</securityPolicies>
```

## Default Applied To Value for Firewall Rules

You can set the applied to setting for all firewall rules created through Service Composer to either Distributed Firewall or Policy's Security Groups. By default, the applied to is set to Distributed Firewall. When Service Composer firewall rules have an applied to setting of distributed firewall, the rules are applied to all clusters on which distributed firewall is installed. If the firewall rules are set to apply to the policy's security groups, you have more granular control over the firewall rules, but may need multiple security policies or firewall rules to get the desired result.

**Table 12-2.** Applied To Values for Service Composer Firewall Rules

Value	Description
dfw_only	Firewall rules are applied to all clusters on which Distributed Firewall is installed.
policy_security_group	Firewall rules are applied to security groups on which the security policy is applied.

## Query Default Applied To Value for Firewall Rules

Retrieves the current applied to value for firewall rules created through Service Composer.

### Example 12-8. Query Default Applied To Value

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy/
    serviceprovider/firewall/
```

Response Body:

```
<SecurityPolicyFirewallConfig>
  <appliedTo>dfw_only</appliedTo>
</SecurityPolicyFirewallConfig>
```

## Change Default Applied To Value for Firewall Rules

Changes the applied to value for firewall rules created through Service Composer. Valid values are dfw\_only and policy\_security\_group.

### Example 12-9. Change Default Applied To Value

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy/
    serviceprovider/firewall/
```

Request Body:

```
<SecurityPolicyFirewallConfig>
  <appliedTo>policy_security_group</appliedTo>
</SecurityPolicyFirewallConfig>
```

---

## Working with Security Actions

### Query Virtual Machines for a Security Action

You can fetch all VirtualMachine objects on which security action of a given category and attribute has been applied.

**Example 12-10.** Query virtual machines for security action

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/securityaction/category
    /virtualmachines?attributeKey=attribute_name&attributeValue=attribute_value
```

Response Body:

```
<vmnodes>
  <vmnode>
    <vmId></vmId>
    <vmName></vmName>
  </vmnode>
  <vmnode>
    <vmId></vmId>
    <vmName></vmName>
  </vmnode>
  ...
  ...
  <vmnode>
    <vmId></vmId>
    <vmName></vmName>
  </vmnode>
</vmnodes>
```

---

### Query Security Actions Applicable on a Security Group

You can fetch all security actions applicable on a security group for all ExecutionOrderCategories. The list is sorted based on the weight of security actions in descending order. The **isActive** tag indicates if a securityaction will be applied (by the enforcement engine) on the security group.

**Example 12-11.** Query security actions for security group

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/securitygroup/securitygroupId
    /securityactions
```

Response Body:

```
<securityActionsByCategoryMap>
  <actionsByCategory>
    <category>firewall</category>
    <action class="firewallSecurityAction">
      <objectId></objectId>
      <objectTypeName></objectTypeName>
      <vsmUuid></vsmUuid>
```

```

</revision></revision>
</type>
  <typeName></typeName>
</type>
<name>name</name>
<description>description</description>
<category></category>
<executionOrder></executionOrder>
<actionType></actionType>
<isActionEnforced></isActionEnforced>
<isActive></isActive>
<isEnabled></isEnabled>
<secondarySecurityGroup>
  <objectId></objectId>
  <objectTypeName></objectTypeName>
  <vsmUuid></vsmUuid>
  <revision></revision>
  <type>
    <typeName></typeName>
  </type>
  <name>name</name>
  <description>description</description>
  <scope>
    <id></id>
    <objectTypeName></objectTypeName>
    <name>name</name>
    <description>description</description>
  </scope>
  <extendedAttributes></extendedAttributes>
</secondarySecurityGroup>
<secondarySecurityGroup>
  ...
</secondarySecurityGroup>
...
...
<secondarySecurityGroup>
  ...
</secondarySecurityGroup>
<securityPolicy>
  <objectId></objectId>
  <objectTypeName></objectTypeName>
  <vsmUuid></vsmUuid>
  <revision></revision>
  <type>
    <typeName></typeName>
  </type>
  <name>name</name>
  <description>description</description>
  <scope>
    <id></id>
    <objectTypeName></objectTypeName>
    <name>name</name>
    <description>description</description>
  </scope>
</securityPolicy>
<invalidSecondaryContainers></invalidSecondaryContainers>
<applications>
  <application>
    <objectId></objectId>
    <objectTypeName></objectTypeName>
    <vsmUuid></vsmUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name></name>
    <scope>
      <id></id>

```

```

        <objectTypeName></objectTypeName>
        <name></name>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <inheritanceAllowed></inheritanceAllowed>
    <element>
        <applicationProtocol></applicationProtocol>
        <value></value>
    </element>
</application>
<application>
    ...
</application>
...
</applications>
<invalidApplications>false</invalidApplications>
<logged>false</logged>
<action>block</action>
<direction>inbound</direction>
<outsideSecondaryContainer>true</outsideSecondaryContainer>
</action>
<action>
</action>
...
...
<action>
    ...
</action>
</actionsByCategory>
<actionsByCategory>
    <category>endpoint</category>
    <action class="endpointSecurityAction">
        <objectId></objectId>
        <objectTypeName></objectTypeName>
        <vsmUuid></vsmUuid>
        <revision></revision>
        <type>
            <typeName></typeName>
        </type>
        <name>name</name>
        <description>description</description>
        <category></category>
        <executionOrder></executionOrder>
        <actionType></actionType>
        <isActionEnforced></isActionEnforced>
        <isActive></isActive>
        <isEnabled></isEnabled>
        <securityPolicy>
            <objectId></objectId>
            <objectTypeName></objectTypeName>
            <vsmUuid></vsmUuid>
            <revision></revision>
            <type>
                <typeName></typeName>
            </type>
            <name></name>
            <description></description>
            <scope>
                <id></id>
                <objectTypeName></objectTypeName>
                <name>name</name>
                <description>description</description>
            </scope>
        </securityPolicy>
    </serviceName></serviceName>
    <serviceId></serviceId>

```

```

    <invalidServiceId></invalidServiceId>
    <ServiceProfile>
      <objectId>serviceprofile-1</objectId>
      ...
    </ServiceProfile>
    <invalidServiceProfile>>false</invalidServiceProfile>
  </action>
</action>
</action>
...
...
<action>
  ...
</action>
</actionsByCategory>
<actionsByCategory>
  <category>traffic_steering</category>
  <action class="trafficSteeringSecurityAction">
    <objectId></objectId>
    <objectTypeName></objectTypeName>
    <vsmUuid></vsmUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name>name</name>
    <description>description</description>
    <category></category>
    <executionOrder></executionOrder>
    <actionType></actionType>
    <isActionEnforced></isActionEnforced>
    <isActive></isActive>
    <isEnabled></isEnabled>
    <securityPolicy>
      <objectId></objectId>
      <objectTypeName></objectTypeName>
      <vsmUuid></vsmUuid>
      <revision></revision>
      <type>
        <typeName></typeName>
      </type>
      <name>name</name>
      <description>description</description>
      <scope>
        <id></id>
        <objectTypeName></objectTypeName>
        <name>name</name>
        <description>description</description>
      </scope>
    </securityPolicy>
    <logged></logged>
  </serviceProfile>
    <objectId></objectId>
    <objectTypeName></objectTypeName>
    <vsmUuid></vsmUuid>
    <revision></revision>
    <type>
      <typeName></typeName>
    </type>
    <name>P</name>
    <clientHandle>
    </clientHandle>
    <extendedAttributes></extendedAttributes>
  </profileAttributes>
    <id></id>
    <revision></revision>
    <attribute>
      <id></id>

```

```

        <revision></revision>
        <key></key>
        <name></name>
        <value></value>
    </attribute>
    <attribute>
        ...
    </attribute>
</profileAttributes>
<service>
    <objectId></objectId>
    <objectTypeName></objectTypeName>
    <vsmUuid></vsmUuid>
    <revision></revision>
    <type>
        <typeName></typeName>
    </type>
    <name>name</name>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
</service>
<category></category>
<vendorTemplate>
    <id></id>
    <revision></revision>
    <name>name</name>
    <idFromVendor></idFromVendor>
    <vendorAttributes>
        <id></id>
        <revision></revision>
    </vendorAttributes>
</vendorTemplate>
<status></status>
<vendorAttributes>
    <id></id>
    <revision></revision>
</vendorAttributes>
<runtime>
    <nonCompliantDvpg/>
    <nonCompliantVwire></nonCompliantVwire>
</runtime>
<serviceProfileBinding>
    <distributedVirtualPortGroups></distributedVirtualPortGroups>
    <virtualWires></virtualWires>
    <excludedVnics></excludedVnics>
    <virtualServers></virtualServers>
</serviceProfileBinding>
</serviceProfile>
    <redirect></redirect>
</action>
<action>
</action>
...
...
<action>
    ...
</action>
</actionsByCategory>
</securityActionsByCategoryMap>

```

## Query Security Action Applicable on A Virtual Machine

You can fetch the security actions applicable on a virtual machine for all ExecutionOrderCategories. The list of SecurityActions per ExecutionOrderCategory is sorted based on the weight of security actions in descending order. The **isActive** tag indicates whether a security action will be applied (by the enforcement engine) on the virtual machine.

**Example 12-12.** Query security actions on a virtual machine

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/virtualmachine
/virtualMachineId/securityactions
```

Response Body:

```
<securityPolicies>
  <securityPolicy></securityPolicy>
  <securityPolicy></securityPolicy>
  ...
  ...
  <securityPolicy></securityPolicy>
</securityPolicies>
```

## Synchronizing Service Composer Rules with Distributed Firewall

If Service Composer goes out of sync with Distributed Firewall, you must re-synchronize Service Composer rules with firewall rules. If Service Composer stays out of sync, firewall configuration may not stay enforced as expected.

### Query Firewall Out-of-Sync Time Stamp

You can query the time since when Service Composer firewall is out of sync with Distributed Firewall.

**Example 12-13.** Query time stamp

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/serviceprovider/firewall
```

Response Body:

```
<keyValues>
  <keyValue>
    <key>getServiceComposerFirewallOutOfSyncTimestamp</key>
  </keyValue>
</keyValues>
```

The response body will contain the UNIX time stamp that represents the time since when Service Composer firewall is out of sync. If Service Composer firewall is not out-of-sync, the body will not contain any data.

### Synchronize Service Composer Firewall

You can synchronize Service Composer firewall with Distributed Firewall.

**Example 12-14.** Synchronize Service Composer firewall

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/serviceprovider/firewall
```

Response Body:

```
<keyValues>
  <keyValue>
    <key>forcesync</key>
  </keyValue>
</keyValues>
```

## Configuring Auto Save Draft for Service Composer

It is possible to configure the system to automatically create firewall drafts for Service Composer. This setting can be enabled or disabled. When disabled no draft is created in the Distributed Firewall for policy work flows. This limits the number of drafts that are automatically created in the system and provides for better performance.

### Query the Auto Save Draft Setting in Service Composer

You can query the state of the auto save draft property in Service Composer.

---

#### Example 12-15. Query auto save draft in Service Composer

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/serviceprovider/firewall
```

Request Body:

```
<keyValues>
  <keyValue>
    <key>getAutoSaveDraft</key>
  </keyValue>
</keyValues>
```

Response Body:

```
<boolean>true</boolean>
```

---

The response body will contain the boolean state of the auto save draft property.

### Change the Auto Save Draft Setting in Service Composer

You can change the state of the auto save draft property in Service Composer.

---

#### Example 12-16. Change auto save draft in Service Composer

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/policy/serviceprovider/firewall
```

Response Body:

```
<keyValues>
  <keyValue>
    <key>autoSaveDraft</key>
    <value>true</value> <!-- Required. Possible values are true and false. -->
  </keyValue>
</keyValues>
```

---

## Query Security Policies Mapped to a Security Group

You can retrieve the security policies mapped to a security group. The list is sorted based on the precedence of security policy precedence in descending order. The security policy with the highest precedence (highest numeric value) is the first entry (index = 0) in the list.

---

#### Example 12-17. Query security policies mapped to a security group

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/securitygroup/securitygroupId
  /securitypolicies
```

Response Body:

```

<securityPolicies>
  <securityPolicy></securityPolicy>
  <securityPolicy></securityPolicy>
  ...
  ...
  <securityPolicy></securityPolicy>
</securityPolicies>

```

---

## Query Service Provider Data

You can query the service provider of a given category to fetch an object containing provider specific data based on the requested property/value pairs.

**Example 12-18.** Query service provider data

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/serviceprovider/category
```

Response Body:

Request Body:

```

<keyValues>
  <keyValue>
    <key></key>
    <value></value>
  </keyValue>
  <keyValue>
    ..
  </keyValue>
  ..
  ..
  <keyValue>
    ..
  </keyValue>
</keyValues>

```

---

## Query Security Group Effective Membership

Retrieves effective membership of a security group in terms of virtual machines. The effective membership is calculated using all the three membership components of a security group - static include, static exclude, and dynamic using the following formula:

Effective membership virtual machines = [ (VMs resulting from static include component + VMs resulting from dynamic component) - (VMs resulting from static exclude component) ]

**Example 12-19.** Query virtual machines in a security group

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/securityGroupId/translation/virtualmachines
```

---

## Query Security Groups to which a VM Belongs

Retrieves the collection of security groups to which a virtual machine is a direct or indirect member. Indirect membership involves nesting of security groups.

**Example 12-20.** Query security groups to which a virtual machine belongs

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/lookup/virtualmachine
/virtualMachineId
```

---

## Status of Service Composer

You can use this API to get the consolidated status of Service Composer. The possible return of value for status are: `in_sync`, `in_progress`, `out_of_sync`, and `pending`.

**Example 12-21.** Consolidated status of Service Composer

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy/status/
```

Response Body:

```
<serviceComposerStatus>
  <status>in_sync</status>
</serviceComposerStatus>
```

---

## System Alarms on Service Composer

You can use this API to get all the system alarms that are raised at Service Composer level and policy level.

**Example 12-22.** All system alarms on Service Composer

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/policy/securitypolicy/alarms/all
```

---

Response Body:

```
<systemAlarms>
  <systemAlarm>
    <eventId></eventId>
    <timestamp></timestamp>
    <severity></severity>
    <eventSource></eventSource>
    <eventCode></eventCode>
    <message></message>
    <module></module>
    <objectId></objectId>
    <reporterName></reporterName>
    <reporterType></reporterType>
    <sourceType></sourceType>
    <displayName></displayName>
    <eventMetadata>
      <data>
        <key></key>
        <value></value>
      </data>
      <data>
        ...
      </data>
      ...
      <data>
        ...
      </data>
    </eventMetadata>
    <resolutionAttempted></resolutionAttempted>
    <resolvable></resolvable>
    <alarmId></alarmId>
    <alarmCode></alarmCode>
```

```
<alarmSource></alarmSource>
<alarmBeingResolved></alarmBeingResolved>
<alarmMetadata>
  <data>
    <key></key>
    <value></value>
  </data>
  <data>
    ...
  </data>
  ...
  <data>
    ...
  </data>
</alarmMetadata>
</systemAlarm>
</systemAlarms>
```

---

# Data Security Configuration

Data Security provides visibility into sensitive data stored within your organization's virtualized and cloud environments. Based on the violations reported by Data Security, you can ensure that sensitive data is adequately protected and assess compliance with regulations around the world.

This chapter includes the following topics:

- [“Data Security User Roles”](#) on page 399
- [“Defining a Data Security Policy”](#) on page 400
- [“Saving and Publishing Policies”](#) on page 405
- [“Data Security Scanning”](#) on page 407
- [“Querying Scan Results”](#) on page 408
- [“Querying Violation Details”](#) on page 411

To begin using Data Security, you create a policy that defines the regulations that apply to data security in your organization and specifies the areas of your environment and files to be scanned. When you start a Data Security scan, analyzes the data on the virtual machines in your vSphere inventory and reports the number of violations detected and the files that violated your policy.

After you analyze the results of the scan, you can edit your policy as required. When you edit a policy, you must enable it by publishing the changes.

Note that you cannot install Data Security using a REST API. For information on installing Data Security, see the *NSX Installation and Upgrade Guide*.

To deploy Data Security, you must install the latest version of VMware Tools on each virtual machine that you want to scan. This installs a Thin Agent, which allows the SVM to scan the virtual machines.

## Data Security User Roles

A user's role determines the actions that the user can perform. A user can only have one role. You cannot add a role to a user, or remove an assigned role from a user, but you can change the assigned role for a user.

**Table 13-1.** Data Security User Roles

Role	Actions Allowed
Enterprise administrator	All operations and security.
vShield administrator	NSX operations only: for example, install virtual appliances, and configure port groups.
Security administrator	Create and publish policies, view violation reports. Cannot start or stop data security scans.
Auditor	View configured policies and violation reports. Read-only.

## Defining a Data Security Policy

In order to detect sensitive data in your environment, you must create a data security policy. You must be a Security Administrator to create policies.

To define a policy, you must specify the following:

- Regulations

A regulation is a data privacy law for protecting PCI (Payment Card Industry), PHI (Protected Health Information) and PII (Personally Identifiable Information) information. You can select the regulations that your company needs to comply to. When you run a scan, Data Security identifies data that violates the regulations in your policy, and is hence sensitive for your organization.

- Participating areas

By default, your entire vCenter inventory is scanned. To scan a subset of your inventory, you can specify the security groups that you want to include or exclude.

- File filters

You can create filters to limit the data being scanned and exclude the file types unlikely to contain sensitive data from the scan.

In the data security APIs, dlp in the pathname stands for data loss prevention (DLP).

## Query Regulations

You can retrieve the list of available regulations for a policy. The output includes regulation IDs and the embedded classifications for each regulation.

### Example 13-1. Get all SDD policy regulations

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/regulation
```

Response Body:

```
<set>
  <Regulation>
    <id>66</id>  <!-- regulation ID -->
    <name>California AB-1298</name>
    <description>Identifies documents and transmissions that contain protected health
      information (ePHI) and personally identifiable information (PII) as
      regulated by California AB-1298 (Civil Code 56, 1785 and
      1798)</description>
    <classifications>
      <Classification>
        <id>10</id>  <!-- classification ID -->
        <name>Credit Card Track Data</name>
        <providerName>Credit Card Track Data</providerName>
        <description>Credit Card Track Data</description>
        <customizable>false</customizable>
      </Classification>
    </classifications>
  </Regulation>
</set>
```

---

## Enable a Regulation

You can enable one or more regulations by putting the regulation IDs into the policy. You can get the appropriate regulation IDs from the output of the retrieve regulations API (see [Example 13-1](#)). In the example request body, regulation 66 is California AB-1298, and regulations 67 and 68 originate elsewhere.

**Example 13-2.** Enable a regulation

Request:

PUT `https://NSX-Manager-IP-Address/api/2.0/dlp/policy/regulations`

Request Body:

```
<set>
  <long>66</long>
  <long>67</long>
  <long>68</long>
</set>
```

**Query Classification Value**

You can retrieve the classification values associated with regulations that monitor Group Insurance Numbers, Health Plan Beneficiary Numbers, Medical Record Numbers, or Patient Identification Numbers. The output includes the classification ID.

**Example 13-3.** Get all classification values associated with customizable classifications

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/dlp/classificationvalue`**Configure a Customized Regex as a Classification Value**

You can configure a ClassificationValue with a customized regex that must be matched during violation inspection. You must include the appropriate classification ID, which you can get from the output of the retrieve classification value API.

**Example 13-4.** Configure a customized regex as a classification value

Request:

PUT `https://NSX-Manager-IP-Address/api/2.0/dlp/policy/classificationvalues`

Authorization: Basic YWRtaW46ZGVmYXVsdA==

Request Body:

```
<set>
  <ClassificationValue>
    <id>3</id>
    <classification>
      <id>15</id>
      <name>Health Plan Beneficiary Numbers</name>
      <providerName>Health Plan Beneficiary Numbers</providerName>
      <description>Health Plan Beneficiary Numbers</description>
      <customizable>true</customizable>
    </classification>
    <value>PATNUM-[0-9]{10}</value>
  </ClassificationValue>
</set>
```

**View the List of Excludable Areas**

You can retrieve the list of datacenters, clusters, and resource pools in your inventory to help you determine the areas you might want to exclude from policy inspection.

**Example 13-5.** View the list of excludable areas

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/d1p/excludableareas>

Response Body:

```

<set>
  <EnhancedInfo>
    <objectId>datacenter-2</objectId>
    <name>jdoe</name>
    <revision>32</revision>
    <objectTypeName>Datacenter</objectTypeName>
    <ownerName>VMware</ownerName>
  </EnhancedInfo>
  <EnhancedInfo>
    <objectId>datacenter-94</objectId>
    <name>jdoe</name>
    <revision>32</revision>
    <objectTypeName>Datacenter</objectTypeName>
    <ownerName>VMware</ownerName>
  </EnhancedInfo>
  <EnhancedInfo>
    <objectId>resgroup-3725</objectId>
    <name>ResourcePool1</name>
    <revision>2</revision>
    <objectTypeName>ResourcePool</objectTypeName>
    <ownerName>jdoe</ownerName>
  </EnhancedInfo>
  <EnhancedInfo>
    <objectId>domain-c2720</objectId>
    <name>Cluster1</name>
    <revision>17</revision>
    <objectTypeName>ClusterComputeResource</objectTypeName>
    <ownerName>jdoe</ownerName>
  </EnhancedInfo>
  <EnhancedInfo>
    <objectId>resgroup-3726</objectId>
    <name>ResourcePool2</name>
    <revision>1</revision>
    <objectTypeName>ResourcePool</objectTypeName>
    <ownerName>jdoe</ownerName>
  </EnhancedInfo>
</set>

```

**Exclude Areas from Policy Inspection**

This API is deprecated as of 5.0.1. Instead, use the API for excluding security groups from a scan. For more information, see [Example 13-8, “Exclude a security group from the scan,”](#) on page 403.

You can exclude one or more datacenters, resource pools or clusters from policy inspection by including the object ID of each area to exclude. You can get the object ID from the output of the View the list of excludable areas API (see [Example 13-5](#)).

**Example 13-6.** Exclude areas from policy inspection

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/d1p/policy/excludedareas>

Authorization: Basic YWRtaW46ZGVmYXVsdA==

Request Body:

```

<set>
  <string>datacenter-3720</string>

```

```
</set>
```

---

## Specify Security Groups to be Scanned

To scan a subset of your inventory, you can specify the security groups that you want to include or exclude in the data security scan.

### Example 13-7. Include a security group in the scan

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/dlp/policy/includedsecuritygroups/
```

Request Body:

```
<set>
  <string>securitygroup-id-1</string>
  <string>securitygroup-id-1</string>
</set>
```

---

### Example 13-8. Exclude a security group from the scan

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/dlp/policy/excludedsecuritygroups/
```

Request Body:

```
<set>
  <string>securitygroup-id-1</string>
  <string>securitygroup-id-1</string>
</set>
```

---

## Query Security Groups Being Scanned

You can retrieve the security groups that have been included or excluded from data security scans.

### Example 13-9. Get included security groups

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/policy/includedsecuritygroups
```

Response:

```
<set>
  <basicinfo>
    <objectId>securitygroup-1</objectId>
    <type>
      <typeName>SecurityGroup</typeName>
    </type>
    <name>included</name>
    <revision>2</revision>
    <objectTypeName>SecurityGroup</objectTypeName>
    <scope>
      <id>datacenter-2</id>
      <objectTypeName>Datacenter</objectTypeName>
      <name>jkiryakoza</name>
    </scope>
  </basicinfo>
</set>
```

---

**Example 13-10.** Get excluded security groups

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/dlp/policy/excludedsecuritygroups/>

Response:

```
<set>
  <basicinfo>
    <objectId>securitygroup-1</objectId>
    <type>
      <typeName>SecurityGroup</typeName>
    </type>
    <name>included</name>
    <revision>2</revision>
    <objectTypeName>SecurityGroup</objectTypeName>
    <scope>
      <id>datacenter-2</id>
      <objectTypeName>Datacenter</objectTypeName>
      <name>jkiryakoza</name>
    </scope>
  </basicinfo>
</set>
```

## Configure File Filters

You can restrict the files you want to scan based on size, last modified date, or file extensions.

The following file filters are available:

- `sizeLessThanBytes` – scan only files with a byte size less than the specified number.
- `lastModifiedBefore` – scan only files modified before the specified date. The date must be specified in GMT format (YYYY-MM-DD HH:MM:SS).
- `lastModifiedAfter` – scan only files modified after the specified date. The date must be specified in GMT format (YYYY-MM-DD HH:MM:SS).
- `extensionsIncluded` – Boolean value as in [Table 13-1](#).

**Table 13-2.** Included extensions parameter

Value of the <code>extensionsIncluded</code> parameter	Result
true followed by the extensions parameter containing one or more extensions	Only files with the specified extensions are scanned
false followed by the extensions parameter containing one or more extensions	All files are scanned except those with the specified extensions.

The `scanAllFiles` parameter determines if all files should be inspected during a scan operation. This parameter overrides all other parameters, so set this parameter to false if you are configuring a filter.

**Example 13-11.** Scan only PDF and XLSX files modified after 10/19/2011

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/dlp/policy/FileFilters>

Request Body:

```
<FileFilters>
  <scanAllFiles>false</scanAllFiles>
  <lastModifiedAfter>2011-10-19 15:16:04.0 EST</lastModifiedAfter>
  <extensionsIncluded>true</extensionsIncluded>
  <extensions>pdf,xlsx</extensions>
```

```
</FileFilters>
```

---

**Example 13-12.** Scan all files except PDF and XLXS files
 

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/dlp/policy/FileFilters
```

Request Body:

```
<FileFilters>
  <scanAllFiles>false</scanAllFiles>
  <extensionsIncluded>false</extensionsIncluded>
  <extensions>pdf,xlsx</extensions>
</FileFilters>
```

---

**Example 13-13.** Scan PDF and XLXS files that are less than 100 MB in size
 

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/dlp/policy/FileFilters
```

Request Body:

```
<FileFilters>
  <scanAllFiles>false</scanAllFiles>
  <sizeLessThanBytes>100000000</sizeLessThanBytes>
  <extensionsIncluded>true</extensionsIncluded>
  <extensions>pdf,xlsx</extensions>
</FileFilters>
```

---

## Saving and Publishing Policies

After you have defined a data security policy, you can edit it by changing the regulations selected, areas excluded from the scan, or the file filters. To apply the edited policy, you must publish it.

### Query Saved Policy

As a best practice, you should retrieve and review the last saved policy before publishing it. Each policy contains a revision value that can be used to track version history.

---

**Example 13-14.** Get saved SDD policy
 

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/policy/saved
```

Authorization: Basic YWRtaW46ZGVmYXVsdA==

Response Body: the following response contains a policy with a single regulation, Indiana HB-1101.

```
<DlpPolicy>
  <objectId>DlpPolicy-1</objectId>
  <type>
    <typeName>DlpPolicy</typeName>
  </type>
  <name>DlpPolicy-One</name>
  <revision>6</revision>
  <objectTypeName>DlpPolicy</objectTypeName>
  <regulations>
    <Regulation>
      <id>37</id>
      <name>Indiana HB-1101</name>
      <description>Indiana HB-1101</description>
      <classifications>
        <Classification>
          <id>16</id>
```

```

        <name>US National Provider Identifier</name>
        <providerName>US National Provider Identifier</providerName>
        <description>US National Provider Identifier</description>
        <customizable>false</customizable>
    </Classification>
</classifications>
<regions>
    <string>North America</string>
    <string>USA</string>
</regions>
<categories>
    <string>PHI</string>
    <string>PCI</string>
    <string>PII</string>
</categories>
</Regulation>
</regulations>
<regulationsChanged>>false</regulationsChanged>
<excludedAreas/>
<excludedAreasChanged>>false</excludedAreasChanged>
<fileFilters>
    <scanAllFiles>>false</scanAllFiles>
    <sizeLessThanBytes>0</sizeLessThanBytes>
    <extensionsIncluded>>false</extensionsIncluded>
</fileFilters>
<fileFiltersChanged>>false</fileFiltersChanged>
<classificationValues>
    <ClassificationValue>
        <id>1</id>
        <classification>
            <id>19</id>
            <name>Patient Identification Numbers</name>
            <providerName>Patient Identification Numbers</providerName>
            <description>Patient Identification Numbers</description>
            <customizable>>true</customizable>
        </classification>
        <value>deg</value>
    </ClassificationValue>
</classificationValues>
<classificationValuesChanged>>false</classificationValuesChanged>
<lastUpdatedOn class="sql-timestamp">2012-01-04 21:25:08.0</lastUpdatedOn>
<lastUpdatedBy>admin</lastUpdatedBy>
</DlpPolicy>

```

---

## Query Published Policy

You can retrieve the currently published SDD policy that is active on all vShield Endpoint SVMs.

**Example 13-15.** Get published SDD policy

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/policy/published
```

```
Authorization: Basic YWRtaW46ZGVmYXVsdA==
```

---

## Publish the Updated Policy

After updating a policy with added regulations, excluded areas, or customized regex values publish the policy to enforce the new parameters.

**Example 13-16.** Publish the updated policy

---

Request:

---

```
PUT https://NSX-Manager-IP-Address/api/2.0/dlp/policy/publish
```

---

## Data Security Scanning

Running a data security scan identifies data in your virtual environment that violates your policy.

All virtual machines in your datacenter are scanned once during a scan. If the policy is edited and published while a scan is running, the scan restarts. This rescan ensures that all virtual machines comply with the edited policy. A rescan is triggered by publishing an edited policy, not by data updates on your virtual machines. After you start a scan, it continues to run until you pause or stop it.

If new virtual machines are added to your inventory while a scan is in progress, those machines will also be scanned. If a virtual machine is moved to an excluded cluster or resource pool while the data security scan is in progress, the files on that virtual machine are not scanned. In case a virtual machine is moved via vMotion to another host, the scan continues on the second host (files that were scanned while the virtual machine was on the previous host are not scanned again).

Data Security scans one virtual machine on a host at a time to minimize impact on performance. VMware recommends that you pause the scan during normal business hours to avoid any performance overhead.

### Start, Pause, Resume, or Stop a Scan Operation

You can start or stop a scan operation. The scan operation options are as follows:

- START: Start a new scan.
- PAUSE: Pause a started scan.
- RESUME: Resume a paused scan.
- STOP: Stop any scan.

**Example 13-17.** Start, pause, resume, or stop a scan operation

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/dlp/scanop
```

Request Body:

```
<ScanOp>STOP</ScanOp>
```

---

### Query Status for a Scan Operation

You can retrieve the status of the scan operation to determine if a scan is STARTED (that is, in progress), PAUSED, or STOPPED. The nextScanOps parameter indicates the scan operations possible from your current state. In the following example, the current scan state is Stopped and the only action you can perform is Start the scan.

**Example 13-18.** Get scan status

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/scanstatus
```

Response Body:

```
<DlpScanStatus>
  <currentScanState>STOPPED</currentScanState>
  <nextScanOps><ScanOp>START</ScanOp></nextScanOps>
  <vmsInProgress>0</vmsInProgress>
  <vmsCompleted>0</vmsCompleted>
</DlpScanStatus>
```

---

## Querying Scan Results

You can retrieve detailed results of the current data security scan as well as summary results for the previous five scans.

### Get List of Virtual Machines Being Scanned

You can retrieve information about the virtual machines being scanned by a scan.

**Example 13-19.** Get list of virtual machines being scanned

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/d1p/scan/current/vms/id?scanstatus=COMPLETED
&pagesize=10&startindex=1
```

Response Body:

```
<VmScanStatusDp>
  <dataPage>
    <pagingInfo>
      <pageSize>10</pageSize>
      <startIndex>1</startIndex>
      <totalCount>2</totalCount>
      <sortOrderAscending>false</sortOrderAscending>
    </pagingInfo>
    <VmScanStatus>
      <startTime>1320803585000</startTime>
      <endTime>1320803826000</endTime>
      <vmMoId>vm-25</vmMoId>
      <scanStatus>COMPLETED</scanStatus>
      <violationCount>8</violationCount>
      <vmName>jim-win2k8-32-mux</vmName>
      <dcName>jack</dcName>
    </VmScanStatus>
  </dataPage>
</VmScanStatusDp>
```

---

Where

- *id* is an optional parameter which limits the filter results by the VC MOID of a datacenter, cluster, or resource pool.
- *scanstatus* specifies the scan status of the virtual machines to be retrieved. Possible values are all, notstarted, started, and completed. This limits the results to virtual machines that have the specified scan state.
- *pagesize* limits the maximum number of entries returned by the API. The default value for this parameter is 256 and the valid range is 1-1024.
- *startindex* specifies the starting point for retrieving the logs. If this parameter is not specified, logs are retrieved from the beginning.

### Get Number of Virtual Machines Being Scanned

You can retrieve the number of virtual machines being scanned.

**Example 13-20.** Get number of virtual machines being scanned

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/d1p/scan/current/vms/count/id?scanstatus
=COMPLETED
```

---

Where

- `scanstatus` is an optional parameter that specifies the scan status of the virtual machines to be retrieved. Possible values are `all`, `notstarted`, `started`, and `completed`. This limits the results to virtual machines that have the specified scan state.
- `id` is an optional parameter which limits the filter results by the VC MOID of a datacenter, cluster, or resource pool.

## Get Summary Information about the Last Five Scans

You can retrieve the start and end time, total number of virtual machines scanned, and total number of violations for the last five completed data security scans.

**Example 13-21.** Get summary information about last five scans

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/completedscansummaries
```

Response Body:

```
<list>
  <CompletedScanSummary>
    <globalScanId>5</globalScanId>
    <startTime class="sql-timestamp">2011-11-09 17:02:48.0</startTime>
    <endTime class="sql-timestamp">2011-11-09 17:02:55.0</endTime>
    <totalVmsScannedCount>0</totalVmsScannedCount>
    <totalViolationCount>0</totalViolationCount>
  </CompletedScanSummary>
</list>
```

---

## Get Information for Virtual Machines Scanned During Previous Scan

You can retrieve the following information about the virtual machines scanned during the previous data security scan:

- ID
- Name
- Scan status
- Violation count

**Example 13-22.** Get Information for virtual machines scanned during last scan

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/scan/scanId/detailsascsv
```

---

## Retrieve Information About Previous Scan Results

You can retrieve a detailed report about the results of the previous scan in a CSV format.

**Example 13-23.** Retrieves Information for virtual machines scanned during last scan

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/scan/scanId/violatingfilesascsv
```

---

## Get XML Representation of Policy Used for Previous Scan

You can retrieve the XML representation of the policy used in the previous scan.

**Example 13-24.** Get XML representation of policy used in previous scan

Request:

GET https://NSX-Manager-IP-Address/api/2.0/dlp/scan/scanId/policyasxml

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<DlpPolicy>
  <objectId>dlppolicy-2</objectId>
  <type>
    <typeName>DlpPolicy</typeName>
  </type>
  <name>Published Policy</name>
  <revision>2</revision>
  <objectTypeName>DlpPolicy</objectTypeName>
  <regulations />
  <regulationsChanged>>false</regulationsChanged>
  <excludedAreas />
  <excludedAreasChanged>>false</excludedAreasChanged>
  <excludedSecurityGroups>
    <basicinfo>
      <objectId>securitygroup-1</objectId>
      <type>
        <typeName>SecurityGroup</typeName>
      </type>
      <name>included</name>
      <revision>2</revision>
      <objectTypeName>SecurityGroup</objectTypeName>
      <scope>
        <id>datacenter-2</id>
        <objectTypeName>Datacenter</objectTypeName>
        <name>jkiryakoza</name>
      </scope>
    </basicinfo>
  </excludedSecurityGroups>
  <excludedSecurityGroupsChanged>>false</excludedSecurityGroupsChanged>
  <includedSecurityGroups>
    <basicinfo>
      <objectId>securitygroup-1</objectId>
      <type reference="../../../../excludedSecurityGroups/basicinfo/type"></type>
      <name>included</name>
      <revision>2</revision>
      <objectTypeName>SecurityGroup</objectTypeName>
      <scope>
        <id>datacenter-2</id>
        <objectTypeName>Datacenter</objectTypeName>
        <name>jkiryakoza</name>
      </scope>
    </basicinfo>
  </includedSecurityGroups>
  <includedSecurityGroupsChanged>>false</includedSecurityGroupsChanged>
  <fileFilters>
    <scanAllFiles>>false</scanAllFiles>
    <sizeLessThanBytes>0</sizeLessThanBytes>
    <extensionsIncluded>>true</extensionsIncluded>
    <extensions>doc, docm, docx, dot, dotx, dotm, wri, xla, xlam, xls, xlt, xltx, xltm, xlsx, xlsb, xlsx, ppt, pptx, pptm, pot, potx, potm, ppsx, ppsm, mdb, mpp, pdf, txt, log, csv, htm, html, xml, text, rtf, svg, ps, gs, vis, msg, rfc822, pm, swf, dgn, jpg, CATAnalysis, CATDrawing, CATFCT, CATMaterial, CATPart, CATProcess, CATProduct, CATShape, CATSWL, CATSystem, 3DXML, 7z, cab, emx, gz, hqx, jar, lha, lzh, rar, tar, uue, z, zip, eml, mail, cal, cont, task, note, jrn, pst</extensions>
  </fileFilters>
  <fileFiltersChanged>>false</fileFiltersChanged>
  <classificationValues>
    <classificationValue>
      <id>33</id>
      <classification>

```

```

        <id>90</id>
        <name>Custom Accounts</name>
        <providerName>Custom Accounts</providerName>
        <description>Custom Accounts</description>
        <customizable>true</customizable>
    </classification>
</ClassificationValue>
<ClassificationValue>
    ...
    <classificationValuesChanged>>false</classificationValuesChanged>
    <lastUpdatedOn class="sql-timestamp">2011-11-09 16:59:01.0</lastUpdatedOn>
    <lastUpdatedBy>dlp</lastUpdatedBy>
</ClassificationValue>
</classificationValues>
</DlpPolicy>

```

---

## Querying Violation Details

Once you start a data security scan, NSX reports the regulations that are being violated by the files in your inventory, and the violating files. If you fix a violating file (by deleting the sensitive information from the file, deleting or encrypting the file, or editing the policy), the file will continue to be displayed in the Violating files section until the current scan completes, and a new scan starts and completes.

You must be a Security Administrator or Auditor to view reports.

## Get List of Violation Counts

You can view a report that displays the violated regulations with the number of violations for each regulation. The violating files report requires filtering by node ID.

**Example 13-25.** Get violation count for entire inventory

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/violations/
```

---

**Example 13-26.** Get violation count for specific resource

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/violations/contextId
```

Response Body:

```

<list>
  <violations>
    <scope>
      <objectId>group-d1</objectId>
      <type>
        <typeName>Folder</typeName>
      </type>
      <name>Datacenters</name>
      <revision>1</revision>
      <objectTypeName>Folder</objectTypeName>
    </scope>
    <regulation>
      <id>100</id>
      <name>California AB-1298</name>
      <description>Identifies documents and transmissions that contain protected
        health information (ePHI) and personally identifiable
        information (PII) as regulated by California AB-1298 (Civil Code
        56, 1785 and 1798). California residents medical and health
        insurance information, when combined with personally
        identifiable information must be protected from unauthorized

```

access, destruction, use, modification, or disclosure. Any business that operates in California and owns or licenses computerized ePHI and PII data for California residents, regardless of the physical location of the business, is required to comply with this law. This policy detects US Social Security Numbers, credit card numbers, California drivers license numbers, US National Provider Numbers, group insurance numbers, health plan beneficiary numbers, medical record numbers, patient identifiers, birth and death certificates and Healthcare Dictionaries.</description>

```
<classifications>
  <Classification>
    <id>76</id>
    <name>Health Plan Beneficiary Numbers</name>
    <providerName>Health Plan Beneficiary Numbers</providerName>
    <description>Health Plan Beneficiary Numbers</description>
    <customizable>true</customizable>
  </Classification>
</classifications>
<regions>
  <string>NA</string>
</regions>
<categories>
  <string>PHI</string>
  <string>PCI</string>
  <string>PII</string>
</categories>
</regulation>
<violationCount>1</violationCount>
</Violations>
<violations>
</list>
```

Where *contextId* is the MOID of a datacenter, cluster, folder, resource pool, or virtual machine.

## Get List of Violating Files

You can view a report that displays the violating files and the regulations each file violated. This API requires filtering by context node ID, and returns a formatted XML report showing violating files.

### Example 13-27. Get violating files for entire inventory

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/d1p/violatingfiles?pagesize=xx&startindex=yy
```

Where:

- *pagesize* is the number of results to view.
- *startindex* is the page number from which the results should be displayed.

### Example 13-28. Get violating files for a resource

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/d1p/violatingfiles/contextId?pagesize=xx
&startindex=yy
```

Response Body:

```
<ViolatingFiles>
  <dataPage>
    <pagingInfo>
      <pageSize>10</pageSize>
      <startIndex>0</startIndex>
```

```

    <totalCount>1</totalCount>
    <sortOrderAscending>>false</sortOrderAscending>
  </pagingInfo>
  <ViolatingFile>
    <identifier>59</identifier>
    <revision>0</revision>
    <fileName>C:\TruePositives\SocialSecurityNumbersTP1.05.txt</fileName>
    <fileExtension />
    <fileLastModifiedTime class="sql-timestamp">2011-02-01
      15:02:00.0</fileLastModifiedTime>

    <vm>
      <name>jim-xp32-dlp1</name>
      <revision>0</revision>
    </vm>
    <cluster>
      <name>JimCluster</name>
      <revision>0</revision>
    </cluster> \
    <dataCenter>
      <name>jkiryakoza</name>
      <revision>0</revision>
    </dataCenter>
  </ViolatingFile>
  <violations>
    <ViolationInfo>
      <identifier>99</identifier>
      <revision>0</revision>
      <regulation>
        <objectId>152</objectId>
        <name>California SB-1386</name>
        <description>Identifies documents and transmissions that contain
          personally identifiable information (PII) as
          regulated by California SB-1386 (Civil Code 1798).
          Businesses that own or license computerized PII about
          California residents are required to maintain
          security procedures and practices to protect it from
          unauthorized access, destruction, use, modification,
          or disclosure. Any business that operates in
          California and owns or licenses computerized PII data
          for California residents, regardless of the physical
          location of the business, is required to comply with
          this law. This policy detects US Social Security
          numbers, credit card numbers and California drivers
          license numbers. This regulation has been amended to
          protect health and medical information that can be
          found in California AB-1298. </description>
        <revision>0</revision> </regulation>
        <firstViolationReportedTime class="sql-timestamp">2012-01-26
          12:56:42.0</firstViolationReportedTime>
        <lastViolationReportedTime class="sql-timestamp">2012-01-26
          12:56:42.0</lastViolationReportedTime>
        <cumulativeViolationCount>1</cumulativeViolationCount>
        <violationCount>0</violationCount>
      </ViolationInfo>
    </violations>
  </ViolatingFile>
</dataPage>
</ViolatingFiles>

```

Where:

- *contextId* is the MOID of a datacenter, cluster, folder, resource pool, or virtual machine..
- *pagesize* is the number of results to view.
- *startIndex* is the page number from which the results should be displayed.

## Get List of Violating Files in CSV Format

You can view a report that displays the violating files and the regulations each file violated in a CSV format.

**Example 13-29.** Get list of violating files in CSV format

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/violatingfilesascsv
```

---

## Get Violations in Entire Inventory

You can view a report of the violated regulations and the violating files for the entire inventory in CSV (comma separated variable) format.

**Example 13-30.** Get list of violated regulations

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/dlp/violatingfilescsv/contextId
```

---

Where *contextId* is the MOID of a datacenter, cluster, folder, resource pool, or virtual machine.

# Activity Monitoring

---

Activity Monitoring provides visibility into your virtual network to ensure that security policies at your organization are being enforced correctly.

A Security policy may mandate who is allowed access to what applications. The Cloud administrator can generate Activity Monitoring reports to see if the IP based firewall rule that they set is doing the intended work. By providing user and application level detail, Activity Monitoring translates high level security policies to low level IP address and network based implementation.

Once you enable data collection for Activity Monitoring, you can run reports to view inbound traffic (such as virtual machines being accessed by users) as well as outbound traffic (resource utilization, interaction between inventory containers, and AD groups that accessed a server).

The chapter includes the following topics:

- [“Data Collection”](#) on page 415
- [“Query Resources”](#) on page 418
- [“Query User Details”](#) on page 421
- [“Query Discovered User Details”](#) on page 425
- [“Working with Domains”](#) on page 426
- [“Working with Activity Monitoring Syslog Support”](#) on page 429

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Data Collection

You must enable data collection for one or more virtual machines on a vCenter Server before running an Activity Monitoring report. Before running a report, ensure that the enabled virtual machines are active and are generating network traffic.

You should also register NSX Manager with the AD Domain Controller. See [“Working with Domains”](#) on page 426.

Note that only active connections are tracked by Activity Monitoring. Virtual machine traffic blocked by firewall rules at the vNIC level is not reflected in reports.

In case of an emergency such as a network overload, you can turn off data collection at a global level. This overrides all other data collection settings.

Some API calls may require the VMID, which is the MOID of the guest virtual machine. You can retrieve this by queuing the vCenter mob structure (<https://VC-IP-Address/mob>). The VMID is listed under host structure.

## Enable Data Collection on a Single Virtual Machine

You must enable data collection at least five minutes before running an Activity Monitoring report.

### Example 14-1. Enable data collection on a virtual machine

---

Request:

POST `https://NSX-Manager-IP-Address/api/1.0/eventcontrol/vm/virtualMachineId/request`

Request Body:

```
<perVmConfig>
  <actions>
    <action>
      <type>per_vm_config</type>
      <value>enabled</value>
    </action>
  </actions>
</perVmConfig>
```

---

## Disable Data Collection on a Single Virtual Machine

### Example 14-2. Disable data collection on a virtual machine

---

Request:

POST `https://NSX-Manager-IP-Address/api/1.0/eventcontrol/vm/virtualMachineId/request`

Request Body:

```
<perVmConfig>
  <actions>
    <action>
      <type>per_vm_config</type>
      <value>disabled</value>
    </action>
  </actions>
</perVmConfig>
```

---

## Override Data Collection

In case of an emergency such as a network overload, you can turn off data collection at a global level (jill switch). This overrides all other data collection settings.

### Turn On Kill Switch

#### Example 14-3. Turn on kill switch

---

Request:

POST `https://NSX-Manager-IP-Address/api/1.0/eventcontrol/eventcontrol-root/request`

Request Body:

```
<request>
  <actions>
    <action>
      <type>global_switch</type>
      <value>disabled</value>
    </action>
  </actions>
</request>
```

---

## Turn Off Kill Switch

### Example 14-4. Turn off kill switch

---

Request:

POST <https://NSX-Manager-IP-Address/api/1.0/eventcontrol/eventcontrol-root/request>

Request Body:

```
<request>
  <actions>
    <action>
      <type>global_switch</type>
      <value>enabled</value>
    </action>
  </actions>
</request>
```

---

## Query Per Virtual Machine Data Collection

When reporting per virtual machine configuration, current kill switch status is also reported too. The effective configuration of a virtual machine is determined by both kill switch config and per virtual machine configuration. If kill switch is on, event collection is effectively disabled regardless of what its per virtual machine configuration is; if kill switch is off, per virtual machine configuration determines whether event collection should be performed for this virtual machine.

### Example 14-5. Retrieve per virtual machine configuration when kill switch is on and when per virtual machine configuration is enabled for specified virtual machine

---

Request:

GET <https://NSX-Manager-IP-Address/api/1.0/eventcontrol/eventcontrol/config/vm/virtualMachineId>

Response Body:

```
<perVmConfig>
  <actions>
    <action>
      <type>global_switch</type>
      <value>disabled</value>
    </action>
    <action>
      <type>per_vm_config</type>
      <value>enabled</value>
    </action>
  </actions>
</perVmConfig>
```

---

### Example 14-6. Retrieve per virtual machine configuration when kill switch is off and when per virtual machine configuration is enabled for specified virtual machine

---

Request:

GET <https://NSX-Manager-IP-Address/api/1.0/eventcontrol/eventcontrol/config/vm/virtualMachineId>

Response Body:

```
<perVmConfig>
  <actions>
    <action>
      <type>global_switch</type>
      <value>enabled</value>
```

```

    </action>
  <action>
    <type>per_vm_config</type>
    <value>enabled</value>
  </action>
</actions>
</perVmConfig>

```

## Query Resources

This method allow you to get the aggregated user activity (action records) for the given set of parameters. The same API is used for all reports.

### Prerequisites

- vShield Endpoint must be installed in your environment. See *NSX Installation and Upgrade Guide*.
- NSX Manager must be registered with Active Directory.
- Data collection must be enabled on one or more virtual machines.

**Table 14-1.** Parameters for GET `https://NSX-Manager-IP-Address/api/3.0/ai/records`

Parameter Name	Description	Mandatory?	Valid Values	Default Value	Example
query	Name of report	Yes	resource,adg,containers,sam,vma	query=resource	None
interval	Relative time to current time	Yes	number followed by either of m,h,d, or s	interval=60m, interval=1h	60m
stime	Start time for query	No. Interval is used if stime and etime are not specified.	yyyy-mm-ddTh24:mi:ss	stime=2012-02-28T21:00:00	None
etime	End time for query	No. Interval is used if stime and etime are not specified.	yyyy-mm-ddTh24:mi:ss	etime=2012-02-29T21:00:00	None
param	Parameter to be applied to query	Depends on query	format, <param-name>:<param-type>:<comma-separated-values>:<operator>	param:src:SECURITY_GROUP:1:INCLUDE	None
pagesize	Number of records to be retrieved	No	Any number (recommended is between 100-2000)	pagesize=1000	1024
startindex	Start record number (used for pagination)	No	number for the next page you want to retrieve	startindex=100	0

## View Outbound Activity

You can view what applications are being run by a security group or desktop pool and then drill down into the report to find out which client applications are making outbound connections by a particular group of users. You can also discover all user groups and users who are accessing a particular application, which can help you determine if you need to adjust identity firewall in your environment.

### Parameter Values

- query = resource
- param = <param-name>:<param-type>:<comma-separated-values>:<operator>

- possible values for "resource" query type,
- <param-name>
  - src
  - dest
  - app
- required parameters = src, dest
- <param-type>
  - for src - SECURITY\_GROUP, DIRECTORY\_GROUP, DESKTOP\_POOL
  - for dest - VIRTUAL\_MACHINE
  - for app - SRC\_APP
- Parameter Values - comma-separated numbers (optional). If none specified then no filter is applied.
- <operator> - INCLUDE, EXCLUDE (default is INCLUDE)

**Example 14-7.** View user activities to VM id 1 originating from application id 1

---

Request:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/records?query=resource&interval=60m&param
=src:DIRECTORY_GROUP&param=dest:VIRTUAL_MACHINE:1&param=app:SRC_APP:1
```

---

## View Inbound Activity

You can view all inbound activity to a server by desktop pool, security group, or AD group.

### Parameter Values

- query = sam
- param = <param-name>:<param-type>:<comma-separated-values>:<operator>
- <param-name>
  - src
  - dest
  - app
- required parameters = src, dest
- <param-type>
  - for src - SECURITY\_GROUP, DIRECTORY\_GROUP, DESKTOP\_POOL
  - for dest - VIRTUAL\_MACHINE
  - for app - DEST\_APP
- Parameter Values - comma-separated numbers (optional). If none specified then no filter is applied.
- <operator> - INCLUDE, EXCLUDE, NOT (default is INCLUDE)

**Example 14-8.** View user activities to VM id 1 originating from application id 1

---

Request:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/records?query=containers&interval=60m&param
=dest:SECURITY_GROUP:1:EXCLUDE&param=src:SECURITY_GROUP:1
```

---

## View Interaction between Inventory Containers

You can view the traffic passing between defined containers such as AD groups, security groups and/or desktop pools. This can help you identify and configure access to shared services and to resolve misconfigured relationships between Inventory container definitions, desktop pools and AD groups.

### Parameter Values

- query = containers
- param = <param-name>:<param-type>:<comma-separated-values>:<operator>
- <param-name>
  - src
  - dest
- required parameters = src, dest
- <param-type>
  - for src - SECURITY\_GROUP, DIRECTORY\_GROUP, DESKTOP\_POOL
  - for dest - SECURITY\_GROUP, DESKTOP\_POOL
- Parameter Values - comma-separated numbers (optional). If none specified then no filter is applied.
- <operator> - INCLUDE, EXCLUDE, NOT (default is INCLUDE)

#### Example 14-9. View interaction between inventory containers

---

Request:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/records?query=containers&interval=60m&param=dest:SECURITY_GROUP:1:EXCLUDE&param=src:SECURITY_GROUP:1
```

---

## View Outbound AD Group Activity

You can view the traffic between members of defined Active Directory groups and can use this data to fine tune your firewall rules.

### Parameter Values

- query = adg
- param = <param-name>:<param-type>:<comma-separated-values>:<operator>
- <param-name>
  - src
  - adg
- required parameters = src
- <param-type>
  - src - SECURITY\_GROUP, DESKTOP\_POOL
  - adg- USER
- Parameter Values - comma-separated numbers (optional). If none specified then no filter is applied.
- <operator> - INCLUDE, EXCLUDE (default is INCLUDE)

#### Example 14-10. View interaction between inventory containers

---

Request:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/records?query=adg&interval=24h&param=adg:USER:1:INCLUDE&param=src:SECURITY_GROUP:1:EXCLUDE
```

## Query User Details

This method allows you to retrieve user detail records for the given set of parameters.

**Table 14-2.** Parameters for GET `https://NSX-Manager-IP-Address/api/3.0/ai/userdetails`

Parameter Name	Description	Mandatory?	Valid Values	Default Value	Example
query	Name of report	Yes	resource,adg,containers,sam,vma	query=resource	None
interval	Relative time to current time	Yes	number followed by either of m,h,d, or s	interval=60m, interval=1h	60m
stime	Start time for query	No. Interval is used if stime and etime are not specified.	yyyy-mm-ddTh24:mi:ss	stime=2012-02-28T21:00:00	None
etime	End time for query	No. Interval is used if stime and etime are not specified.	yyyy-mm-ddTh24:mi:ss	etime=2012-02-29T21:00:00	None
param	Parameter to be applied to query	Depends on query	format, <param-name>:<param-type>:<comma-separated-values>:<operator>	param:src:SECURITY_GROUP:1:INCLUDE	None
pagesize	Number of records to be retrieved	No	Any number (recommended is between 100-2000)	pagesize=1000	1024
startindex	Start record number (used for pagination)	No	number for the next page you want to retrieve	startindex=100	0

## View Outbound Activity

You can view what applications are being run by a security group or desktop pool and then drill down into the report to find out which client applications are making outbound connections by a particular group of users. You can also discover all user groups and users who are accessing a particular application, which can help you determine if you need to adjust identity firewall in your environment.

### Parameter Values

- query = resource
- param = <param-name>:<param-type>:<comma-separated-values>:<operator>
- possible values for "resource" query type,
- <param-name>
  - src
  - dest
- required parameters = src, dest
- <param-type>
  - for src - SECURITY\_GROUP, DIRECTORY\_GROUP, DESKTOP\_POOL
  - for dest - IP (this has to be a valid IP address in the dot notation, xx.xx.xx.xx)

- for app - SRC\_APP
- <operator> - INCLUDE, EXCLUDE (default is INCLUDE)

**Example 14-11.** View user activities to VM id1 originating from application id1

---

Request:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/userdetails?query=resource&stime
=2012-10-15T00:00:00&etime=2012-10-20T00:00:00&param=src:DIRECTORY_GROUP
:2&param=app:SRC_APP:16&param=dest:IP:172.16.4.52
```

---

## View Inbound Activity

You can view all inbound activity to a server by desktop pool, security group, or AD group.

### Parameter Values

- query = sam
- param = <param-name>:<param-type>:<comma-separated-values>:<operator>
- <param-name>
  - src
  - dest
  - app
- required parameters = src, dest, app
- <param-type>
  - for src - SECURITY\_GROUP, DIRECTORY\_GROUP, DESKTOP\_POOL
  - for dest - VIRTUAL\_MACHINE
  - for app - DEST\_APP
- Parameter Values - comma-separated numbers (optional). If none specified then no filter is applied.
- <operator> - INCLUDE, EXCLUDE, NOT (default is INCLUDE)

**Example 14-12.** View user activities to VM id 1 originating from application id 1

---

Request:

```
GET https://NSX-Manager-IP-Address/api/3.0/userdetails?query=sam&interval=60m&param=app
:DEST_APP:1:EXCLUDE&param=dest:IP:1:EXCLUDE&param=src:SECURITY_GROUP:1
:EXCLUDE
```

---

## View Interaction between Inventory Containers

You can view the traffic passing between defined containers such as AD groups, security groups and/or desktop pools. This can help you identify and configure access to shared services and to resolve misconfigured relationships between Inventory container definitions, desktop pools and AD groups.

### Parameter Values

- query = containers
- param = <param-name>:<param-type>:<comma-separated-values>:<operator>
- <param-name>

- src
- dest
- required parameters = src, dest
- <param-type>
  - for src - SECURITY\_GROUP, DIRECTORY\_GROUP, DESKTOP\_POOL
  - for dest - SECURITY\_GROUP, DESKTOP\_POOL
- Parameter Values - comma-separated numbers (optional). If none specified then no filter is applied.
- <operator> - INCLUDE, EXCLUDE, NOT (default is INCLUDE)

**Example 14-13.** View interaction between inventory containers

---

Request:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/userdetails?query=containers&interval=60m&param=dest:SECURITY_GROUP:1:EXCLUDE&param=src:SECURITY_GROUP:1
```

---

## View Outbound AD Group Activity

You can view the traffic between members of defined Active Directory groups and can use this data to fine tune your firewall rules.

### Parameter Values

- query = adg
- param = <param-name>:<param-type>:<comma-separated-values>:<operator>
- <param-name>
  - src
  - adg
- required parameters = src
- <param-type>
  - src - SECURITY\_GROUP, DESKTOP\_POOL
  - adg- USER
- Parameter Values - comma-separated numbers (optional). If none specified then no filter is applied.
- <operator> - INCLUDE, EXCLUDE (default is INCLUDE)

**Example 14-14.** View interaction between inventory containers

---

Request:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/userdetails?query=adg&interval=24h&param=adg:USER:1:INCLUDE&param=src:SECURITY_GROUP:1:EXCLUDE
```

---

## View Virtual Machine Activity Report

You can view traffic to or from a virtual machine or a set of virtual machines in your environment.

### Parameter Values

- query = vma

- param = <param-name>:<param-type>:<comma-separated-values>:<operator>
- <param-name>
  - src (for outbound traffic)
  - dest(for inbound traffic)
  - app - SRC\_APP, DEST\_APP
- required parameters = none (if no parameter passed then this would show all SAM activities)
- <param-type>
  - src - SECURITY\_GROUP, DESKTOP\_POOL
  - dest - VIRTUAL\_MACHINE,VM\_UUID
  - adg- USER
- Parameter Values - comma-separated numbers (optional). If none specified then no filter is applied.
- <operator> - INCLUDE, EXCLUDE (default is INCLUDE)

**Example 14-15.** View inbound vm activities to a VM id1 for a specific service used (app=16)

---

Request:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/userdetails?query=vma&interval=60m&param
=dest:VIRTUAL_MACHINE:1&param=app:DEST_APP:16
```

Response Body:

```
<DataPage>
  <pagingInfo>
    <pageSize>1024</pageSize>
    <startIndex>0</startIndex>
    <totalCount>5</totalCount>
    <sortOrderAscending>>false</sortOrderAscending>
  </pagingInfo>
  <aiActionRecord>
    <application>JABBER</application>
    <connectionCount>3</connectionCount>
    <destHost>PMI-BL-X61$</destHost>
    <destIP>172.16.4.21</destIP>
    <id>0</id>
    <srcContainer>HOKUIFLVPC</srcContainer>
  </aiActionRecord>
  <aiActionRecord>
    <application>SLP</application>
    <connectionCount>2</connectionCount>
    <destHost>ENGG-LAPTOP-002$</destHost>
    <destIP>172.16.4.48</destIP>
    <id>0</id>
    <srcContainer>HOKUIFLVPC</srcContainer>
  </aiActionRecord>
  <aiActionRecord>
    <application>KEYSERV</application>
    <connectionCount>1</connectionCount>
    <destHost>PMI00ELTON03$</destHost>
    <destIP>172.16.1.12</destIP>
    <id>0</id>
    <srcContainer>HOKUIFLVPC</srcContainer>
  </aiActionRecord>
  <aiActionRecord>
    <application>ACCOUNT_MGMT</application>
    <connectionCount>1</connectionCount>
    <destHost>PMIFEEXCH01$</destHost>
    <destIP>172.16.4.70</destIP>
    <id>0</id>
    <srcContainer>HOKUIFLVPC</srcContainer>
  </aiActionRecord>
```

```

</aiActionRecord>
<aiActionRecord>
  <application>PNA</application>
  <connectionCount>3</connectionCount>
  <destHost>IDC-DEV-1$</destHost>
  <destIP>10.0.200.92</destIP>
  <id>0</id>
  <srcContainer>HOKUIFLVPC</srcContainer>
</aiActionRecord>
</DataPage>

```

---

## Query Discovered User Details

This method retrieves the list of all discovered users (both by agent introspection and LDAP Sync) and their detail.

### Example 14-16. Retrieve user details

---

Retrieve user details for a specific user:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/user/userId
```

Retrieve app details:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/app
```

Retrieve application details for a specific application:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/app/appId
```

Retrieve list of all discovered hosts (both by agent introspection and LDAP Sync) and their detail:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/host
```

Retrieve host details:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/host/hostId
```

Retrieve list of all discovered desktop pools by agent introspection:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/desktoppool
```

Retrieve details specific desktop pool:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/desktoppool/desktoppoolId
```

Retrieve list of all discovered virtual machines:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/vm
```

Retrieve details about a specific virtual machine:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/vm/virtualMachineId
```

Retrieve list of all the discovered (and configured) LDAP directory groups:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/directorygroup
```

Retrieve details about a specific directory groups:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/directorygroup/directorygroupId
```

Retrieve list of AD groups a user belongs to:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/directorygroup/user/userId
```

Retrieve list of all the observed security groups. Observed entities are the ones that are reported by the agents. For ex, if a host activity is reported by an agent and if that host belongs to a security group then that security group would reported as observed in SAM database:

```
GET https://NSX-Manager-IP-Address/api/3.0/ai/securitygroup
```

Retrieve details about specific security group:

GET <https://NSX-Manager-IP-Address/api/3.0/ai/securitygroup/securitygroupId>

---

## Working with Domains

After you create a domain, you can apply a security policy to it and run queries to view the applications and virtual machines being accessed by the users of a domain.

### Register a Domain with NSX Manager

You can register one or more Windows domains with an NSX Manager and associated vCenter server.

NSX Manager gets group and user information as well as the relationship between them from each domain that it is registered with. NSX Manager also retrieves Active Directory credentials.

You can apply security policies on an Active Directory domain and run queries to get information on virtual machines and applications accessed by users within an Active Directory domain.

#### Example 14-17. Register or update domain

---

Request:

POST <https://NSX-Manager-IP-Address/api/3.0/directory/updateDomain>

Request Body:

```
<DirectoryDomain>
  <name>vs4.net</name>
  <type>ActiveDirectory</type>
  <netbiosName>VS4</netbiosName>
  <username>Administrator</username>
  <password>xxx</password>
</DirectoryDomain>
```

Response Body:

```
<DirectoryDomain>
  <id>2</id>
  <name>vs4.net</name>
  <type>ActiveDirectory</type>
  <netbiosName>VS4</netbiosName>
  <username>Administrator</username>
  <baseDn>DC=vs4,DC=net</baseDn>
</DirectoryDomain>
```

---

### Parameter Values for Register/Update Domain

Parameter Name	Description	Mandatory?
ID	Domain id. If you want to create a new domain, do not provide this value. Otherwise, system will find an existing domain object by this ID and update it.	true if update existing domain
name	Domain name. This should be domain's full qualified name. In case agent discovered, this will be NetBIOS name, so you need to update it to FQN in order to support LDAP sync and event log reader.	true if creating a new domain
description	Domain description	false
type	Domain type. Valid value include: AGENT_DISCOVERED, ActiveDirectory, SPECIAL Do NOT modify SPECIAL domain (we will put guard later). For LDAP sync and event log reader work, this need to be sent to ActiveDirectory.	true if creating a new domain

Parameter Name	Description	Mandatory?
netbiosName	NetBIOS name of domain. This is Domain's NetBIOS name. Check windows domain setting, for value of it. Normally Agent report domain name is NetBIOS name. But confirm from Windows domain setting.	false
baseDn	Domain's Base DN (for LDAP sync). Base DN is REQUIRED for LDAP Sync. If you have a domain like: w2k3.vshield.vmware.com, the base DN is very likely to be: DC=w2k3,DC=vshield,DC=vmware,DC=com. Another example is: domain name is: vs4.net, the base DN should be: DC=vs4,DC=net. If you don't know what is this, use a LDAP client and connect to domain controller, that will give you domain's base DN.	false
rootDn	LDAP Sync root DN. Specify where should LDAP sync start from LDAP tree. This could be absolute path, for example: OU=Engineer,DC=vs4,DC=net, or relative path (relate to Base DN), for example: OU=Engineer. Don't use this column in most cases.	false
securityId	Domain's Security ID (SID). This should be filled by LDAP sync process, just don't use this column unless you know what you are doing.	false
username	Domain's User name (Used for LDAP Sync and/or Event Log reader)	false
password	User password	false
eventLogUsername	Domain's event log reader username (will use above username if this is NULL)	false
eventLogPassword	Domain's event log reader password	false

## Query Domains

Retrieves all agent discovered (or configured) LDAP domains.

### Example 14-18. Query domains

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/directory/listDomains
```

Response Body:

```
<DirectoryDomains>
  <DirectoryDomain>
    <id>2</id>
    <name>vs4.net</name>
    <type>ActiveDirectory</type>
    <netbiosName>VS4</netbiosName>
    <username>Administrator</username>
    <baseDn>DC=vs4,DC=net</baseDn>
  </DirectoryDomain>
</DirectoryDomains>
```

## Delete Domain

Deletes domain.

### Example 14-19. Delete domain

Request:

```
DELETE https://NSX-Manager-IP-Address/api/1.0/directory/deleteDomain/domainId
```

## Working with LDAP Servers

### Example 14-20. Create LDAP server

---

Request:

POST *https://NSX-Manager-IP-Address/api/1.0/directory/updateLdapServer*

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<LDAPServer>
  <domainId>4</domainId>
  <hostName>10.142.72.70</hostName>
  <enabled>true</enabled>
</LDAPServer>
```

---

If the Response Body is not 200 for OK, log in to your NSX Manager and try to ping the hostname.

### Example 14-21. LDAP server calls

---

Query LDAP servers for a domain:

GET *https://NSX-Manager-IP-Address/api/1.0/directory/listLdapServersForDomain/domainId*

Start LDAP full sync:

PUT *https://NSX-Manager-IP-Address/api/1.0/directory/fullSync/domainId*

Start LDAP delta sync:

PUT *https://NSX-Manager-IP-Address/api/1.0/directory/deltaSync/domainId*

Delete LDAP server:

DELETE *https://NSX-Manager-IP-Address/api/1.0/directory/deleteLdapServer/LdapServerId*

---

## Working with EventLog Servers

### Example 14-22. Create EventLog server

---

Request:

POST *https://NSX-Manager-IP-Address/api/1.0/directory/updateEventLogServer*

Request Body:

```
<EventLogServer>
  <id>1</id>
  <domainId>4</domainId>
  <hostName>10.142.72.70</hostName>
  <enabled>false</enabled>
</EventLogServer>
```

---

### Example 14-23. EventLog server calls

---

Query EventLog servers for a domain:

GET *https://NSX-Manager-IP-Address/api/1.0/directory/listEventLogServersForDomain/EventLogServerId*

Delete EventLog server:

DELETE *https://NSX-Manager-IP-Address/api/1.0/directory/deleteEventLogServer/EventLogServerId*

---

## Working with Mapping Lists

### Example 14-24. Query mapping lists

---

Query user-to-ip mapping list from database:

```
GET https://NSX-Manager-IP-Address/api/1.0/identity/userIpMapping
```

Query host-to-ip mapping list from database:

```
GET https://NSX-Manager-IP-Address/api/1.0/identity/hostIpMapping
```

Query set of users associated with a given set of IP addresses during a specified time period. Since more than one user can be associated with a single IP address during the specified time period, each IP address can be associated with zero or more (i.e a SET of) users:

```
GET https://NSX-Manager-IP-Address/api/1.0/identity/ipToUserMapping
```

Query set of Windows Domain Groups (AD Groups) to which the specified user belongs:

```
GET https://NSX-Manager-IP-Address/api/1.0/identity/directoryGroupsForUser
```

Create static user IP mapping:

```
POST https://NSX-Manager-IP-Address/api/1.0/identity/staticUserMapping/userId/IP
```

Query static user IP mapping list:

```
GET https://NSX-Manager-IP-Address/api/1.0/identity/staticUserMappings
```

Query static user IP mapping for specified user:

```
GET https://NSX-Manager-IP-Address/api/1.0/identity/staticUserMappingsbyUser/userId
```

Query static user IP mapping for specified IP:

```
GET https://NSX-Manager-IP-Address/api/1.0/identity/staticUserMappingsbyIP/userId
```

Delete static user IP mapping for specified user:

```
DELETE https://NSX-Manager-IP-Address/api/1.0/identity/staticUserMappingsbyUser/userId
```

Delete static user IP mapping for specified IP:

```
DELETE https://NSX-Manager-IP-Address/api/1.0/identity/staticUserMappingsbyIP/userId
```

---

## Working with Activity Monitoring Syslog Support

### Example 14-25. Enable Activity Monitoring syslog support

---

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/sam/syslog/enable
```

---

### Example 14-26. Disable Activity Monitoring syslog support

---

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/sam/syslog/disable
```

---



## Communication Channel Health

This feature allows the user to check the connection status between the NSX Manager and the host(s). A Hash Map is used to hold all Hosts' connection status. It will remember the latest heartbeat from each Host. When querying a Host's connection status, NSX Manager will get the latest heartbeat information to compare the last heartbeat time and current time. If the duration is longer than a threshold, it returns "DOWN", otherwise it returns "UP". If no last heartbeat information is found and this host has not been prepared or the netcpa version on this host is lower than 6.2.0, it will return "NOT\_AVAILABLE". But if no last heartbeat information is found and the host has been prepared with netcpa version no less than 6.2.0, it will return "DOWN". When a Host has been unprepared, its heartbeat information will be removed from the NSX Manager memory.

### Checking the Connection Status of a Single Host

**Example 15-1.** Query connection status of single host

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/inventory/host/hostId/connection/status
```

Response Body:

```
<hostConnStatus>
  <hostId>host-xx</hostId>
  <nsxMgrToFirewallAgentConn>UP</nsxMgrToFirewallAgentConn>
  <nsxMgrToControlPlaneAgentConn>UP</nsxMgrToControlPlaneAgentConn>
  <hostToControllerConn>UP</hostToControllerConn>
  <fullSyncCount>13</fullSyncCount>
</hostConnStatus>
```

### Checking the Connection Status of a List of Hosts

**Example 15-2.** Query connection status of multiple hosts

Request:

```
GET https://<ip>/api/2.0/vdn/inventory/hosts/connection/status?hostId=hostId1
&hostId=hostId2...
```

Response Body:

```
<hostConnStatusList>
  <hostConnStatuses>
    <hostConnStatus>
      <hostId>host-31</hostId>
      <nsxMgrToFirewallAgentConn>UP</nsxMgrToFirewallAgentConn>
      <nsxMgrToControlPlaneAgentConn>UP</nsxMgrToControlPlaneAgentConn>
      <hostToControllerConn>UP</hostToControllerConn>
```

```

        <fullSyncCount>3</fullSyncCount>
    </hostConnStatus>
    <hostConnStatus>
        <hostId>host-32</hostId>
        <nsxMgrToFirewallAgentConn>UP</nsxMgrToFirewallAgentConn>
        <nsxMgrToControlPlaneAgentConn>UP</nsxMgrToControlPlaneAgentConn>
        <hostToControllerConn>DOWN</hostToControllerConn>
        <fullSyncCount>0</fullSyncCount>
    </hostConnStatus>
        ...
    </hostConnStatuses>
</hostConnStatusList>

```

---

## Central CLI Methods

The Central Command Line Interface (Central CLI) commands are run from the NSX Manager, and retrieve information from the NSX Manager and other devices. These commands can also be executed in the API. Given here is the general structure for making a Central CLI command call in the API, as well as a specific sample of one such command. For a complete list of the Central CLI commands executable through the API, please see the Central CLI chapter of the *NSX Command Line Interface Reference* available on the VMware documentation website.

### General Central CLI use in the API

**Example 15-3.** General Central CLI command for use in the API

---

Request:

```
POST https://NSX-Manager-IP-Address/api/1.0/nsx/cli?action=execute
```

Request Body:

```

<nsxcli>
  <command>CLI Command</command>
</nsxcli>

```

---

Note: *CLI Command* can be any Central CLI command.

### Sample Central CLI command in the API

**Example 15-4.** Example of CLI command use in the API

---

Request:

```
curl -k -u admin:VMware1VMware! -H 'Content-Type: application/xml' -X
  POST https://10.156.222.74/api/1.0/nsx/cli?action=execute
```

Request Body:

```

<nsxcli>
  <command>show logical-switch list host host-21 vni</command>
</nsxcli>

```

---

## Traceflow

For Traceflow to work as expected, make sure that the controller cluster is connected and in healthy state. The Traceflow operation requires active communication between vCenter, NSX Manager, controller cluster, and netcpa User World Agents (UWA) on the host. Traceflow observes marked packet as it traverses overlay network. Each packet is delivered to host VM and monitored as it crosses overlay network until it reaches the destination VM. The packet is never delivered to the destination guest VM. This means that Traceflow packet

delivery is successful even when the guest VM is powered down. Unknown L2 Packets are always be sent to the bridge. Typically, the bridge forwards these packets to a VLAN and reports the Traceflow packet as delivered. The packet which is reported as delivered need not necessarily mean that the trace packet was delivered to the destination specified. You should conclude only after validating the observations. vdl2 serves ARP proxy for ARP packets coming from VMs. However, traceflow bypasses this process, hence vdl2 may broadcast the traceflow packet out.

## Creating Traceflows

### Example 15-5. Create Traceflow (ICMP implicit)

---

Request Body:

POST <https://NSX-Manager-IP-Address/api/2.0/vdn/traceflow>

Request Body:

```
<traceflowRequest>
  <vnicId>74eb1145-d40b-4061-8e64-1caddf2dbf81.001</vnicId>
  <timeout>10000</timeout>
  <routed>>true</routed>
  <packet class="fieldsPacketData">
    <resourceType>FieldsPacketData</resourceType>
    <ethHeader>
      <srcMac>00:50:56:83:7e:87</srcMac>
      <dstMac>00:50:56:83:fa:6c</dstMac>
      <ethType>2048</ethType>
    </ethHeader>
    <ipHeader>
      <ttl>64</ttl>
      <srcIp>172.32.1.5</srcIp>
      <dstIp>172.34.1.5</dstIp>
    </ipHeader>
  </packet>
</traceflowRequest>
```

---

### Example 15-6. Create Traceflow (ICMP explicit)

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/vdn/traceflow>

Request Body:

```
<traceflowRequest>
  <vnicId>74eb1145-d40b-4061-8e64-1caddf2dbf81.001</vnicId>
  <timeout>10000</timeout>
  <routed>>false</routed>
  <packet class="fieldsPacketData">
    <resourceType>FieldsPacketData</resourceType>
    <ethHeader>
      <srcMac>00:50:56:83:7e:87</srcMac>
      <dstMac>00:50:56:83:fa:6c</dstMac>
      <ethType>2048</ethType>
    </ethHeader>
    <ipHeader>
      <srcIp>172.32.1.5</srcIp>
      <dstIp>172.34.1.5</dstIp>
    </ipHeader>
    <transportHeader>
      <icmpEchoRequestHeader>
        <sequence>1</sequence>
        <id>12</id>
      </icmpEchoRequestHeader>
    </transportHeader>
  </packet>
```

```
</traceflowRequest>
```

---

### Example 15-7. Create Traceflow (TCP)

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/vdn/traceflow>

Request Body:

```
<traceflowRequest>
  <vnicId>74eb1145-d40b-4061-8e64-1caddf2dbf81.001</vnicId>
  <timeout>10000</timeout>
  <routed>false</routed>
  <packet class="fieldsPacketData">
    <resourceType>FieldsPacketData</resourceType>
    <ethHeader>
      <srcMac>00:50:56:83:7e:87</srcMac>
      <dstMac>00:50:56:83:fa:6c</dstMac>
      <ethType>2048</ethType>
    </ethHeader>
    <ipHeader>
      <srcIp>172.32.1.5</srcIp>
      <dstIp>172.34.1.5</dstIp>
    </ipHeader>
    <transportHeader>
      <tcpHeader>
        <srcPort>80</srcPort>
        <dstPort>80</dstPort>
        <tcpFlags>2</tcpFlags>
      </tcpHeader>
    </transportHeader>
  </packet>
</traceflowRequest>
```

---

### Example 15-8. Create Traceflow UDP

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/vdn/traceflow>

Request Body:

```
<traceflowRequest>
  <vnicId>50079744-72f3-37ae-f8a0-4f3aca672500.000</vnicId>
  <timeout>10000</timeout>
  <routed>>true</routed>
  <packet class="fieldsPacketData">
    <resourceType>FieldsPacketData</resourceType>
    <ethHeader>
      <srcMac>00:50:56:87:15:c2</srcMac>
      <dstMac>00:50:56:94:52:49</dstMac>
      <ethType>2048</ethType>
    </ethHeader>
    <ipHeader>
      <srcIp>192.168.20.2</srcIp>
      <dstIp>192.168.10.2</dstIp>
    </ipHeader>
    <transportHeader>
      <udpHeader>
        <srcPort>999</srcPort>
        <dstPort>333</dstPort>
      </udpHeader>
    </transportHeader>
  </packet>
```

```
</traceflowRequest>
```

---

## Querying Traceflows

### Example 15-9. Query Traceflow

---

Request Body:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/traceflow/traceflowId
```

Response Body:

```
<traceflowDto>
  <operState>COMPLETE</operState>
  <vnicId>74eb1145-d40b-4061-8e64-1caddf2dbf81.001</vnicId>
  <id>00000000-0000-0000-0000-000056b5dec3</id>
  <receivedCount>2</receivedCount>
  <forwardedCount>1</forwardedCount>
  <deliveredCount>1</deliveredCount>
  <logicalReceivedCount>4</logicalReceivedCount>
  <logicalDroppedCount>0</logicalDroppedCount>
  <logicalForwardedCount>4</logicalForwardedCount>
  <timeout>10000</timeout>
  <completeAvailable>true</completeAvailable>
  <result>SUCCESS</result>
  <resultSummary>Traceflow delivered observation(s) reported</resultSummary>
  <srcIp>172.32.1.5</srcIp>
  <srcMac>00:50:56:83:7e:87</srcMac>
  <dstMac>172.34.1.5</dstMac>
  <lifMac>00:50:56:83:fa:6c</lifMac>
</traceflowDto>
```

---

Note: *traceflowId* is the value returned after executing the *Create Traceflow* API call.

### Example 15-10. Query Traceflow Observations

---

Request Body:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/traceflow/traceflowId/observations
```

Response Body:

```
<traceflowObservations>
  <traceflowObservationsDataPage>
    <pagingInfo>
      <pageSize>100</pageSize>
      <startIndex>0</startIndex>
      <totalCount>12</totalCount>
      <sortOrderAscending>true</sortOrderAscending>
      <sortBy />
    </pagingInfo>
    <traceflowObservationReceived>
      <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
      <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
      <hostName>10.146.104.42</hostName>
      <hostId>host-22</hostId>
      <component>PHYS</component>
      <compDisplayName>vNIC</compDisplayName>
      <hopCount>0</hopCount>
    </traceflowObservationReceived>
    <traceflowObservationLogicalReceived>
      <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
      <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
      <hostName>10.146.104.42</hostName>
      <hostId>host-22</hostId>
      <component>FW</component>
    </traceflowObservationLogicalReceived>
  </traceflowObservationsDataPage>
</traceflowObservations>
```

```

    <compDisplayName>Firewall</compDisplayName>
    <hopCount>1</hopCount>
</traceflowObservationLogicalReceived>
<traceflowObservationLogicalForwarded>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>FW</component>
    <compDisplayName>Firewall</compDisplayName>
    <hopCount>2</hopCount>
    <ruleId>1001</ruleId>
</traceflowObservationLogicalForwarded>
<traceflowObservationLogicalForwarded>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>LS</component>
    <compDisplayName>1-switch-3</compDisplayName>
    <hopCount>3</hopCount>
    <vni>10000</vni>
    <logicalCompId>universalwire-1</logicalCompId>
    <logicalCompName>1-switch-3</logicalCompName>
</traceflowObservationLogicalForwarded>
<traceflowObservationLogicalReceived>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>LR</component>
    <compDisplayName>1-vm-3</compDisplayName>
    <hopCount>4</hopCount>
    <vni>10000</vni>
    <lifName>27100000000a</lifName>
    <compId>10000</compId>
    <srcNsxManager>4204ad55-71ec-927b-ca1b-aabfa36863ad</srcNsxManager>
    <srcGlobal>true</srcGlobal>
    <compName>default+edge-bbe379a7-e7b9-4ece-b97c-466cf746c93e</compName>
    <logicalCompId>edge-bbe379a7-e7b9-4ece-b97c-466cf746c93e</logicalCompId>
    <logicalCompName>1-vm-3</logicalCompName>
    <otherLogicalCompId>universalwire-1</otherLogicalCompId>
    <otherLogicalCompName>1-switch-3</otherLogicalCompName>
</traceflowObservationLogicalReceived>
<traceflowObservationLogicalForwarded>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>LR</component>
    <compDisplayName>1-vm-3</compDisplayName>
    <hopCount>5</hopCount>
    <vni>10002</vni>
    <lifName>27100000000c</lifName>
    <compId>10000</compId>
    <compName>default+edge-bbe379a7-e7b9-4ece-b97c-466cf746c93e</compName>
    <srcNsxManager>4204ad55-71ec-927b-ca1b-aabfa36863ad</srcNsxManager>
    <srcGlobal>true</srcGlobal>
    <logicalCompId>edge-bbe379a7-e7b9-4ece-b97c-466cf746c93e</logicalCompId>
    <logicalCompName>1-vm-3</logicalCompName>
    <otherLogicalCompId>universalwire-3</otherLogicalCompId>
    <otherLogicalCompName>3-switch-98</otherLogicalCompName>
</traceflowObservationLogicalForwarded>
<traceflowObservationLogicalReceived>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a02fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>

```

```

    <component>LS</component>
    <compDisplayName>3-switch-98</compDisplayName>
    <hopCount>6</hopCount>
    <vni>10002</vni>
    <logicalCompId>universalwire-3</logicalCompId>
    <logicalCompName>3-switch-98</logicalCompName>
  </traceflowObservationLogicalReceived>
  <traceflowObservationForwarded>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>a2fe610-5358-4a3d-8fef-7be60b7d4ea5</transportNodeId>
    <hostName>10.146.104.42</hostName>
    <hostId>host-22</hostId>
    <component>PHYS</component>
    <compDisplayName>10.146.104.42</compDisplayName>
    <hopCount>7</hopCount>
    <remoteIpAddress>172.19.172.142</remoteIpAddress>
    <context>5109430534275084</context>
  </traceflowObservationForwarded>
  <traceflowObservationReceived>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>d2fd4b26-a664-423f-b0aa-8ba760cd967f</transportNodeId>
    <hostName>10.146.103.3</hostName>
    <hostId>host-20</hostId>
    <component>PHYS</component>
    <compDisplayName>10.146.103.3</compDisplayName>
    <hopCount>8</hopCount>
  </traceflowObservationReceived>
  <traceflowObservationLogicalReceived>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>d2fd4b26-a664-423f-b0aa-8ba760cd967f</transportNodeId>
    <hostName>10.146.103.3</hostName>
    <hostId>host-20</hostId>
    <component>FW</component>
    <compDisplayName>Firewall</compDisplayName>
    <hopCount>9</hopCount>
  </traceflowObservationLogicalReceived>
  <traceflowObservationLogicalForwarded>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>d2fd4b26-a664-423f-b0aa-8ba760cd967f</transportNodeId>
    <hostName>10.146.103.3</hostName>
    <hostId>host-20</hostId>
    <component>FW</component>
    <compDisplayName>Firewall</compDisplayName>
    <hopCount>10</hopCount>
    <ruleId>1001</ruleId>
  </traceflowObservationLogicalForwarded>
  <traceflowObservationDelivered>
    <roundId>00000000-0000-0000-0000-000056b5dec3</roundId>
    <transportNodeId>d2fd4b26-a664-423f-b0aa-8ba760cd967f</transportNodeId>
    <hostName>10.146.103.3</hostName>
    <hostId>host-20</hostId>
    <component>PHYS</component>
    <compDisplayName>vNIC</compDisplayName>
    <hopCount>11</hopCount>
    <vlanId>0</vlanId>
  </traceflowObservationDelivered>
</traceflowObservationsDataPage>
</traceflowObservations>

```

---

Note: *traceflowId* is the value returned after executing the *Create Traceflow* API call.



# Managing Hardware Gateways

---

VMware partners provide hardware gateway products that you can integrate into your NSX deployment. This chapter includes the following topics:

- [“About the Hardware Gateway APIs”](#) on page 439
- [“Managing Hardware Gateways”](#) on page 439
- [“Managing Replication Clusters”](#) on page 442
- [“Getting Hardware Gateway Inventory Information”](#) on page 444
- [“Managing Hardware Gateway Bindings”](#) on page 445
- [“Connecting/Disconnecting a Hardware Gateway with a Virtual Wire”](#) on page 448
- [“Managing Bidirectional Forwarding Detection \(BFD\)”](#) on page 450
- [“Managing Hardware Gateways”](#) on page 439

## About the Hardware Gateway APIs

Use these RESTful APIs used to administer a hardware gateway deployment.

Note: REST API requests include `https://<nsx-ip>`. For example, if the request is shown in this document as:

```
POST: /api/2.0/vdn/hardwaregateways
```

You would use:

```
POST: https://<nsx-ip>/api/2.0/vdn/hardwaregateways
```

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Managing Hardware Gateways

Use these REST APIs to manage hardware gateways.

### Install a Hardware Gateway

**Example 16-1.** Install a Hardware Gateway

---

Request:

```
POST: /api/2.0/vdn/hardwaregateways
```

Request Body:

```
<hardwareGatewaySpec>  
  <name> name of the hardware gateway </name>
```

```

    <description>desc</description>
    <certificate> certificate of the hardware gateway </certificate>
    <bfdEnabled> False </bfdEnabled>
  </hardwareGatewaySpec>

```

The default value of the bfdEnabled flag is true.

Response Body:

Hardware Gateway Object

```

mylogin@launcher-virtual-machine:/root/git/nsx-qe-main-01/vdnet/automation/pylib$ curl -k
-u admin:default -H 'ContentType:application/json' -X GET
https://10.144.139.50/api/2.0/vdn/hardwaregateways
<?xml version="1.0" encoding="UTF-8"?>
<list>
  <hardwareGateway>
    <objectId>torgateway-1</objectId>
    <revision>0</revision>
    <name>torgateway1</name>
    <description>this is tor instance 1</description>
    <clientHandle />
    <isUniversal>>false</isUniversal>
    <universalRevision>0</universalRevision>
    <uuid>3e5ffd66-448d-4e54-82ec-92fffd46d4af</uuid>
    <status>UP</status>
    <thumbprint>80:7F:39:FC:7D:1D:C4:32:8A:67:DE:6D:23:0B:64:52:AB:24:6B:25</thumbprint
    >
    <bfdEnabled>>true</bfdEnabled>
    <managementIp>10.144.137.91</managementIp>
    <bindingCount>2</bindingCount>
  </hardwareGateway>
  <hardwareGateway>
    <objectId>torgateway-2</objectId>
    <revision>0</revision>
    <name>torgateway2</name>
    <description>this is tor instance 2</description>
    <clientHandle />
    <isUniversal>>false</isUniversal>
    <universalRevision>0</universalRevision>
    <uuid>6c43af48-d742-43b4-9416-10c508edbdcf</uuid>
    <status>UP</status>
    <thumbprint>80:E1:15:4B:7F:15:23:F4:A1:81:F7:1D:DE:04:18:10:D0:64:FF:A9</thumbprint
    >
    <bfdEnabled>>true</bfdEnabled>
    <managementIp>10.144.138.116</managementIp>
    <bindingCount>2</bindingCount>
  </hardwareGateway>
</list>

```

## List all Hardware Gateways

**Example 16-2.** List all Hardware Gateways

---

Request:

```
GET : /api/2.0/vdn/hardwaregateways
```

Response Body:

List of all hardwareGateway objects

```

mylogin@launcher-virtual-machine:~$ curl -k -u admin:default -H
'ContentType:application/json' -X GET
https://10.116.254.110/api/2.0/vdn/hardwaregateways
<?xml version="1.0" encoding="UTF-8"?>
<list>
  <hardwareGateway>
    <objectId>torgateway-1</objectId>

```

```

    <revision>0</revision>
    <name>torgateway1</name>
    <description>this is tor instance 1</description>
    <clientHandle />
    <isUniversal>>false</isUniversal>
    <universalRevision>0</universalRevision>
    <uuid>6536bcf5-2f55-47f6-8b26-9fa632061d8c</uuid>
    <status>UP</status>
    <thumbprint>B9:0E:E9:6C:AA:7B:AD:11:64:4C:33:92:4E:0C:D8:16:10:95:02:A7</thumbprint
    >
    <bfdEnabled>>true</bfdEnabled>
    <managementIp>10.116.255.160</managementIp>
    <bindingCount>2</bindingCount>
  </hardwareGateway>
  <hardwareGateway>
    <objectId>torgateway-2</objectId>
    <revision>0</revision>
    <name>torgateway2</name>
    <description>this is tor instance 2</description>
    <clientHandle />
    <isUniversal>>false</isUniversal>
    <universalRevision>0</universalRevision>
    <uuid>f1e9b733-c0c3-4905-b00d-4bd6d8649f48</uuid>
    <status>UP</status>
  <thumbprint>3C:9D:C0:9B:F7:57:AF:EA:6A:9F:49:27:7B:23:25:D3:5E:0D:53:ED</thumbprint>
    <bfdEnabled>>true</bfdEnabled>
    <managementIp>10.116.251.149</managementIp>
    <bindingCount>2</bindingCount>
  </hardwareGateway>
</list>

```

## Get a Hardware Gateway

### Example 16-3. Get a Hardware Gateway

---

Request:

```
GET : /api/2.0/vdn/hardwaregateways/<id>
```

Response Body:

hardwareGateway object

```

mylogin@launcher-virtual-machine:~$ curl -k -u admin:default -H
      'ContentType:application/json' -X GET
      https://10.116.254.110/api/2.0/vdn/hardwaregateways/torgateway-1
<?xml version="1.0" encoding="UTF-8"?>
<hardwareGateway>
  <objectId>torgateway-1</objectId>
  <revision>0</revision>
  <name>torgateway1</name>
  <description>this is tor instance 1</description>
  <clientHandle />
  <isUniversal>>false</isUniversal>
  <universalRevision>0</universalRevision>
  <uuid>6536bcf5-2f55-47f6-8b26-9fa632061d8c</uuid>
  <status>UP</status>
  <thumbprint>B9:0E:E9:6C:AA:7B:AD:11:64:4C:33:92:4E:0C:D8:16:10:95:02:A7</thumbprint>
  <bfdEnabled>>true</bfdEnabled>
  <managementIp>10.116.255.160</managementIp>
  <bindingCount>2</bindingCount>
</hardwareGateway>

```

## Update a Hardware Gateway

### Example 16-4. Update a Hardware Gateway

---

Request:

PUT : /api/2.0/vdn/hardwaregateways/<id>

Request Body:

hardwareGatewaySpec

Response Body:

Updated hardwareGateway Object

## Delete a Hardware Gateway Instance

### Example 16-5. Delete a Hardware Gateway Instance

---

Request:

DELETE: /api/2.0/vdn/hardwaregateways/<id>

## Managing Replication Clusters

Use these REST APIs to manage replication clusters.

### Add or Delete Hosts on a Replication Cluster

#### Example 16-6. Add or Delete Hosts on a Replication Cluster

---

Request:

PUT : /api/2.0/vdn/hardwaregateways/replicationcluster

Request Body:

```
<replicationCluster>
  <hosts>
    <basicinfo>
      <objectId>host-20</objectId>
    </basicinfo>
    <basicinfo>
      <objectId>host-21</objectId>
    </basicinfo>
    <basicinfo>
      <objectId>host-26</objectId>
    </basicinfo>
  </hosts>
</replicationCluster>
```

Response Body:

replicationCluster DTO

### Get a Replication Cluster

#### Example 16-7. Get a Replication Cluster

---

Request:

GET : /api/2.0/vdn/hardwaregateways/replicationcluster

Response Body:

replicationCluster DTO

```

mylogin@launcher-virtual-machine:~$ curl -k -u admin:default -H
      'ContentType:application/json' -X GET
      https://10.116.254.110/api/2.0/vdn/hardwaregateways/replicationcluster
<?xml version="1.0" encoding="UTF-8"?>
<replicationCluster>
  <hosts>
    <basicinfo>
      <objectId>host-26</objectId>
      <objectTypeName>HostSystem</objectTypeName>
      <vsmUuid>422874E3-6873-972C-DE9E-67D5B846042E</vsmUuid>
      <nodeId>e5a97efd-89e1-44b1-bfe8-9d07a8d92f08</nodeId>
      <revision>32</revision>
      <type>
        <typeName>HostSystem</typeName>
      </type>
      <name>10.116.254.9</name>
      <scope>
        <id>domain-c24</id>
        <objectTypeName>ClusterComputerResource</objectTypeName>
        <name>ComputeCluster2-$$</name>
      </scope>
      <clientHandle />
      <extendedAttributes />
      <isUniversal>false</isUniversal>
      <universalRevision>0</universalRevision>
    </basicinfo>
    <basicinfo>
      <objectId>host-21</objectId>
      <objectTypeName>HostSystem</objectTypeName>
      <vsmUuid>422874E3-6873-972C-DE9E-67D5B846042E</vsmUuid>
      <nodeId>e5a97efd-89e1-44b1-bfe8-9d07a8d92f08</nodeId>
      <revision>31</revision>
      <type>
        <typeName>HostSystem</typeName>
      </type>
      <name>10.116.247.220</name>
      <scope>
        <id>domain-c18</id>
        <objectTypeName>ClusterComputerResource</objectTypeName>
        <name>ComputeCluster1-$$</name>
      </scope>
      <clientHandle />
      <extendedAttributes />
      <isUniversal>false</isUniversal>
      <universalRevision>0</universalRevision>
    </basicinfo>
    <basicinfo>
      <objectId>host-20</objectId>
      <objectTypeName>HostSystem</objectTypeName>
      <vsmUuid>422874E3-6873-972C-DE9E-67D5B846042E</vsmUuid>
      <nodeId>e5a97efd-89e1-44b1-bfe8-9d07a8d92f08</nodeId>
      <revision>33</revision>
      <type>
        <typeName>HostSystem</typeName>
      </type>
      <name>10.116.254.157</name>
      <scope>
        <id>domain-c18</id>
        <objectTypeName>ClusterComputerResource</objectTypeName>
        <name>ComputeCluster1-$$</name>
      </scope>
      <clientHandle />
      <extendedAttributes />
      <isUniversal>false</isUniversal>
      <universalRevision>0</universalRevision>
    </basicinfo>
  </hosts>
</replicationCluster>

```

## Getting Hardware Gateway Inventory Information

Use these REST APIs to get hardware gateway inventory information.

### Get Hardware Gateway Switches

#### Example 16-8. Get Hardware Gateway Switches

---

Request:

```
GET : /api/2.0/vdn/hardwaregateways/<id>/switches
```

Response Body:

```
<hardwareGatewaySwitches>
  <hardwareGatewaySwitch>
    <switchame> hardwaregateway switch name </switchname>
    <description>description</description>
    <faults></faults>
  </hardwareGatewaySwitch>
</hardwareGatewaySwitches>
```

```
mylogin@launcher-virtual-machine:~$ curl -k -u admin:default -H
      'ContentType:application/json' -X GET
      https://10.116.254.110/api/2.0/vdn/hardwaregateways/torgateway-1/switches
<?xml version="1.0" encoding="UTF-8"?>
<hardwareGatewaySwitches>
  <hardwareGatewaySwitch>
    <switchname>1-switch-579</switchname>
    <description />
    <faults />
  </hardwareGatewaySwitch>
  <hardwareGatewayId>torgateway-1</hardwareGatewayId>
</hardwareGatewaySwitches>
```

### Get Hardware Gateway Port Names for a Switch

#### Example 16-9. Get Hardware Gateway Port Names for a Switch

---

Request:

```
GET : /api/2.0/vdn/hardwaregateway/<id>/switches/<switchname>/switchports
```

Response Body:

```
<hardwareGatewaySwitchPorts>
  <hardwareGatewaySwitchPort>
    <portname> hardware gateway portname </portname>
    <description>description</description>
    <faults></faults>
  </hardwareGatewaySwitchPort>
</hardwareGatewaySwitchPorts>
```

```
mylogin@launcher-virtual-machine:~$ curl -k -u admin:default -H
      'ContentType:application/json' -X GET
      https://10.116.254.110/api/2.0/vdn/hardwaregateways/torgateway-1/switches/1
      -switch-579/switchports
<?xml version="1.0" encoding="UTF-8"?>
<hardwareGatewaySwitchPorts>
  <hardwareGatewaySwitchPort>
    <portname>p4</portname>
    <description />
    <faults />
  </hardwareGatewaySwitchPort>
  <hardwareGatewaySwitchPort>
    <portname>p3</portname>
    <description />
```

```

    <faults />
  </hardwareGatewaySwitchPort>
</hardwareGatewaySwitchPort>
  <portname>p2</portname>
  <description />
  <faults />
</hardwareGatewaySwitchPort>
</hardwareGatewaySwitchPort>
  <portname>p1</portname>
  <description />
  <faults />
</hardwareGatewaySwitchPort>
</hardwareGatewaySwitch>
  <switchname>1-switch-579</switchname>
</hardwareGatewaySwitch>
  <hardwareGatewayId>torgateway-1</hardwareGatewayId>
</hardwareGatewaySwitchPorts>

```

## Managing Hardware Gateway Bindings

Use these REST APIs to manage hardware gateway bindings.

### Get Hardware Gateway Bindings per Virtual Wire

**Example 16-10.** Get Hardware Gateway Bindings per Virtual Wire

Request:

```
GET : /api/2.0/vdn/virtualwires/<virtualwire id>/hardwaregateways
```

Response Body:

List of hardwareGatewayBinding objects.

```

<hardwareGatewayBinding>
  <id>hw_gateway_binding_id</id>
  <virtualWire>virtualwire id</virtualWire>
  <hardwareGatewayId> hw_gateway Id </hardwareGatewayId>
  <switchName> switchname </switchName>
  <portname> portname </portname>
  <vlan> vlan </vlan>
  <vni>vni</vni>
</hardwareGatewayBinding>

```

...

```

mylogin@launcher-virtual-machine:~$ curl -k -u admin:default -H
  'ContentType:application/json' -X GET
  https://10.116.254.110/api/2.0/vdn/virtualwires/virtualwire-1/hardwaregatew
  ays

```

```

<?xml version="1.0" encoding="UTF-8"?>
<list>
  <hardwareGatewayBinding>
    <id>torbinding-2</id>
    <hardwareGatewayId>torgateway-1</hardwareGatewayId>
    <switchName>1-switch-579</switchName>
    <portname>p1</portname>
    <vlan>0</vlan>
    <virtualWire>virtualwire-1</virtualWire>
    <vni>5342</vni>
  </hardwareGatewayBinding>
  <hardwareGatewayBinding>
    <id>torbinding-1</id>
    <hardwareGatewayId>torgateway-2</hardwareGatewayId>
    <switchName>1-switch-104</switchName>
    <portname>p1</portname>
    <vlan>0</vlan>

```

```

    <virtualWire>virtualwire-1</virtualWire>
    <vni>5342</vni>
  </hardwareGatewayBinding>
</list>

```

## Create a Hardware Gateway Binding

**Example 16-11.** Create a Hardware Gateway Binding

---

Request:

POST : /api/2.0/vdn/hardwaregateway/bindings

Request Body:

```

<hardwareGatewayBinding>
  <hardwareGatewayId>hardwaregateway1</hardwareGatewayId>
  <vlan>v1</vlan>
  <switchName>s1</switchName>
  <portName>s1</portName>
</hardwareGatewayBinding>

```

Response Body:

hardwareGatewayBinding object

## Get a List of Hardware Gateway Bindings

**Example 16-12.** Get a List of Hardware Gateway Bindings

---

Request:

GET :  
       /api/2.0/vdn/hardwaregateway/bindings?hardwareGatewayId=<hardwareGatewayId>  
       &vni=<vni>

Request Body:

optional String hardwareGatewayId  
 optional Integer vni

Response Body:

hardwareGatewayBinding objects

```

<hardwareGatewayBinding>
  <id>hardware gateway binding id</id>
  <hardwareGatewayId>hwgateway1</hardwareGatewayId>
  <vlan>201</vlan>
  <switchName>s1</switchName>
  <portname>s1</portname>
  <virtualWire>virtualwire-1</virtualWire>
  <vni>5000</vni>
</hardwareGatewayBinding>

```

## Get a Hardware Gateway Binding Object

**Example 16-13.** Get a Hardware Gateway Binding Object

---

Request:

GET : /api/2.0/vdn/hardwaregateway/bindings/<binding id>

Response Body:

hardwareGatewayBinding object

## Update a Hardware Gateway Binding Object

You can update the binding parameters. This API will fail if:

- the specified hardwareGatewayId does not exist
- the specified virtualWire is not present or there is a software gateway on the binding
- the new binding value is a duplicate of an existing binding

### Example 16-14. Update a Hardware Gateway Binding Object

---

Request:

```
PUT : /api/2.0/vdn/hardwaregateway/bindings/<binding id>
```

Request Body:

An updated binding object. Any one of the fields should be different from the existing object. You can also update the virtualWire.

```
<hardwareGatewayBinding>
  <hardwareGatewayId>hardwaregateway1</hardwareGatewayId>
  <vlan>201</vlan>
  <switchName>s1</switchName>
  <portname>s1</portname>
  <virtualWire>virtualwire-1</virtualWire>
</hardwareGatewayBinding>
```

Response Body:

```
hardwareGatewayBinding object
```

## Delete a Hardware Gateway Binding

### Example 16-15. Delete a Hardware Gateway Binding

---

Request:

```
DELETE : /api/2.0/vdn/hardwaregateway/bindings/<binding id>
```

## Manage Hardware Gateway Binding Objects

Use this API to attach, detach, and update multiple bindings in a single API call.

This API accepts three lists for add, update, and delete. Each list accepts a hardwareGatewayManageBindingsItem with a full description of the new binding with its objectID. This API handles a maximum of 100 HardwareGatewayManageBindingsItem objects for each of the Add/Update/Delete lists.

### Example 16-16. Manage Hardware Gateway Binding Objects

---

Request:

```
POST: /api/2.0/vdn/hardwaregateway/bindings/manage
```

Request Body:

```
<hardwareGatewayManageBindings>
  <addItems>
    <hardwareGatewayManageBindingItem>
      <hardwareGatewayId> </hardwareGatewayId>
      <virtualWireId> </virtualWireId>
      <switchName> </switchName>
      <portname> </portname>
      <vlan> (int) </vlan>
      <virtualWire> (int) </virtualWire>
    </ hardwareGatewayManageBindingItem>
```

```

</addItems>
  <updateItems>
    <hardwareGatewayManageBindingItem>
      <objectId> </objectId>
      <hardwareGatewayId> </hardwareGatewayId>
      <virtualWireId> </virtualWireId>
      <switchName> </switchName>
      <portname> </portname>
      <vlan> (int) </vlan>
      <virtualWire> (int) </virtualWire>
    </ hardwareGatewayManageBindingItem>
  </updateItems>
  <deleteItems>
    <hardwareGatewayManageBindingItem>
      <objectId> </objectId>
    </ hardwareGatewayManageBindingItem>
  </deleteItems>
</hardwareGatewayManageBindings>

```

Response Body:

```

<hardwareGatewayManageBindings>
  <addItems>
    <hardwareGatewayManageBindingItem>
      <hardwareGatewayId> </hardwareGatewayId>
      <virtualWireId> </virtualWireId>
      <switchName> </switchName>
      <portname> </portname>
      <vlan> (int) </vlan>
      <vni> (int) </vni>
      <objectId> </objectId>
      <errorString> <errorString>
      <responseStatus> <responseStatus>
    </ hardwareGatewayManageBindingItem>
  </addItems>
  <updateItems>
</updateItems>
  <deleteItems>
</deleteItems>
</hardwareGatewayManageBindings>

```

## Get Statistic Information per Hardware Gateway Binding

**Example 16-17.** Get Statistic Information per Hardware Gateway Binding

---

Request:

```
GET : /api/2.0/vdn/hardwaregateway/bindings/{bindingId}/statistic
```

Response Body:

Hardware Gateway Binding Statistic Info DTO

```

<hardwareGatewayStats>
  <bindingId>hwgwbinding-5</bindingId>
  <timestamp>long type timestamp for this query response</timestamp>
  <packetsFromLocal>23431</packetsFromLocal>
  <bytesFromLocal>734754</bytesFromLocal>
  <packetsToLocal>2343</packetsToLocal>
  <bytesToLocal>74364</bytesToLocal>
</hardwareGatewayStats>

```

## Connecting/Disconnecting a Hardware Gateway with a Virtual Wire

Use these REST APIs to manage the connection between a hardware gateway and a virtual wire.

## Attach a Hardware Gateway to a Virtual Wire

There are two ways in which to attach a hardwareGateway to a virtualwire.

### Option #1

User can create a binding and input the binding Id.

#### Example 16-18. Attach a Hardware Gateway to a Virtual Wire

---

Request:

```
POST: /api/2.0/vdn/virtualwires/<virtualwireId>/hardwaregateways/<bindingId>?action=attach
```

Request Body:

```
bindingId
```

Response Body:

Virtual wire object

```
<virtualwire>
  .....
  <hardwareGatewayBindings>
    <hardwareGatewayBinding>
      <id> binding id </id>
    </hardwareGatewayBinding>
  </hardwareGatewayBindings>
</virtualwire>
```

### Option #2

User can also give the binding entries and the API would in turn create the binding object and attach to the virtualwire.

#### Example 16-19. Attach a Hardware Gateway to a Virtual Wire

---

Request:

```
POST: /api/2.0/vdn/virtualwires/<virtualwire id>/hardwaregateways
```

Request Body:

```
<hardwareGatewayBinding>
  <hardwareGatewayId>hardwaregateway1</hardwareGatewayId>
  <vlan>v1</vlan>
  <switchName>s1</switchName>
  <portName>s1</portName>
</hardwareGatewayBinding>
```

Response Body:

virtualWire Object

## ?Detach a Hardware Gateway from a Virtual Wire

Detaching a binding from a hardware gateway will also delete the binding.

#### Example 16-20. ?Detach a Hardware Gateway from a Virtual Wire

---

Request:

```
POST:/api/2.0/vdn/virtualwires/<virtualwireId>/hardwaregateways/<bindingId>?action=detach
```

Response Body:

Updated virtualWire Object

## Managing Bidirectional Forwarding Detection (BFD)

Use these REST APIs to manage bidirectional forwarding detection (BFD).

### Set Global BFD Parameter Values

#### Example 16-21. Set Global BFD Parameter Values

---

Request:

PUT: /api/2.0/vdn/hardwaregateway/bfd/config

Request Body:

```
<hardwareGatewayBfdParams>
  <bfdEnabled>true</bfdEnabled>
  <probeInterval>100</probeInterval>
</hardwareGatewayBfdParams>
```

Response Body:

```
<hardwareGatewayBfdParams>
  <bfdEnabled>true</bfdEnabled>
  <probeInterval>100</probeInterval>
</hardwareGatewayBfdParams>
```

### Get Global BFD Parameter Values

#### Example 16-22. Get Global BFD Parameter Values

---

Request:

GET: /api/2.0/vdn/hardwaregateway/bfd/config

Response Body:

```
<hardwareGatewayBfdParams>
  <bfdEnabled>true</bfdEnabled>
  <probeInterval>100</probeInterval>
</hardwareGatewayBfdParams>
```

### Get the Tunnel BFD Status

#### Example 16-23. Get the Tunnel BFD Status

---

Request:

GET: /api/2.0/vdn/hardwaregateway/bfd/status

Response Body:

HardwareGatewayBfdStatus for all tunnel endpoints (including hosts and hardwareGateways)

```
mylogin@launcher-virtual-machine:/root/git/nsx-qe-main-01/vdnet/automation/TDS/NSX$ curl
  -k -u admin:default -H 'ContentType:application/json' -X GET
  https://192.161.255.248/api/2.0/vdn/hardwaregateway/bfd/status
<?xml version="1.0" encoding="UTF-8"?>
<hardwareGatewayBfdStatusList>
  <statuses>
    <hardwareGatewayBfdStatus>
      <probeSourceId>torgateway-2</probeSourceId>
      <bfdTunnelList>
        <bfdTunnelStatus>
          <diagnostic>Neighbor Signaled Session Down</diagnostic>
          <enabled>true</enabled>
          <forwarding>true</forwarding>
          <info></info>
          <localVtepIp>172.21.145.84</localVtepIp>
```

```

    <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
    <remoteState>UP</remoteState>
    <remoteVtepIp>172.19.152.226</remoteVtepIp>
    <state>UP</state>
  </bfdTunnelStatus>
  <bfdTunnelStatus>
    <diagnostic>Neighbor Signaled Session Down</diagnostic>
    <enabled>true</enabled>
    <forwarding>true</forwarding>
    <info></info>
    <localVtepIp>172.21.145.84</localVtepIp>
    <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
    <remoteState>UP</remoteState>
    <remoteVtepIp>172.18.171.169</remoteVtepIp>
    <state>UP</state>
  </bfdTunnelStatus>
  <bfdTunnelStatus>
    <diagnostic>Neighbor Signaled Session Down</diagnostic>
    <enabled>true</enabled>
    <forwarding>true</forwarding>
    <info></info>
    <localVtepIp>172.21.145.84</localVtepIp>
    <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
    <remoteState>UP</remoteState>
    <remoteVtepIp>172.18.171.168</remoteVtepIp>
    <state>UP</state>
  </bfdTunnelStatus>
</bfdTunnelList>
</hardwareGatewayBfdStatus>
<hardwareGatewayBfdStatus>
  <probeSourceId>torgateway-1</probeSourceId>
  <bfdTunnelList>
    <bfdTunnelStatus>
      <diagnostic>Control Detection Time Expired</diagnostic>
      <enabled>true</enabled>
      <forwarding>true</forwarding>
      <info></info>
      <localVtepIp>172.21.145.85</localVtepIp>
      <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
      <remoteState>UP</remoteState>
      <remoteVtepIp>172.19.152.226</remoteVtepIp>
      <state>UP</state>
    </bfdTunnelStatus>
    <bfdTunnelStatus>
      <diagnostic>Neighbor Signaled Session Down</diagnostic>
      <enabled>true</enabled>
      <forwarding>true</forwarding>
      <info></info>
      <localVtepIp>172.21.145.85</localVtepIp>
      <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
      <remoteState>UP</remoteState>
      <remoteVtepIp>172.18.171.168</remoteVtepIp>
      <state>UP</state>
    </bfdTunnelStatus>
    <bfdTunnelStatus>
      <diagnostic>Neighbor Signaled Session Down</diagnostic>
      <enabled>true</enabled>
      <forwarding>true</forwarding>
      <info></info>
      <localVtepIp>172.21.145.85</localVtepIp>
      <remoteDiagnostic>Control Detection Time Expired</remoteDiagnostic>
      <remoteState>UP</remoteState>
      <remoteVtepIp>172.18.171.169</remoteVtepIp>
      <state>UP</state>
    </bfdTunnelStatus>
  </bfdTunnelList>
</hardwareGatewayBfdStatus>
</statuses>

```

```
</hardwareGatewayBfdStatusList>
```

# Managing NSX in a Cross-vCenter Environment

---

# 17

## Cross-vCenter Distributed Routing

### Universal Distributed Logical Router

If you are creating, modifying, or deleting universal distributed logical routers, you must run the API request on the primary NSX Manager. Universal distributed logical routers are read-only from secondary NSX Managers.

---

**Example 17-1.** Create a UDLR (Universal Distributed Logical Router)

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/edges/?isuniversal=true
```

Request Body:

```
<edge>
  <type>DISTRIBUTED_ROUTER</type>
  <localEgressEnabled>true</localEgressEnabled>  <!-- Optional. Default is true -->
</edge>
```

---

---

**Example 17-2.** Query a UDLR

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/edges/edgeId?isuniversal=true
```

Response Body:

```
<edge>
  <globalRevision></globalRevision>
  <isuniversal>true</isuniversal>
  <localEgressEnabled></localEgressEnabled>
</edge>
```

---

### Cluster Level Locale ID

---

**Example 17-3.** Query locale ID on cluster

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/nwFabric/clusters/clusterId
```

---

**Example 17-4.** Update locale ID on cluster

---

Request:

PUT `https://NSX-Manager-IP-Address/api/2.0/nwFabric/clusters/clusterId`

Request Body:

```
<nwFabricClusterConfig>
  <localeId>uuid1</localeId>
</nwFabricClusterConfig>
```

---

**Example 17-5.** Delete locale ID on cluster

---

Request:

DELETE `https://NSX-Manager-IP-Address/api/2.0/nwFabric/clusters/clusterId`

---

## Host Level Locale ID

**Example 17-6.** Query locale ID on host

---

Request:

GET `https://NSX-Manager-IP-Address/api/2.0/nwFabric/hosts/hostId`

---

**Example 17-7.** Update locale ID on host

---

Request:

PUT `https://NSX-Manager-IP-Address/api/2.0/nwFabric/hosts/hostId`

Request Body:

```
<nwFabricHostConfig>
  <localeId>uuid1</localeId>
</nwFabricHostConfig>
```

---

**Example 17-8.** Delete locale ID on host

---

Request:

DELETE `https://NSX-Manager-IP-Address/api/2.0/nwFabric/hosts/hostId`

---

## NSX Manager Roles

**Example 17-9.** You can set the role of an NSX Manager to primary, the secondary, or standalone. If you set an NSX Manager's role to primary, then use it to create universal objects, and then set the role to standalone, the role will be set as transit. In the transit role, the universal objects will still exist, but cannot be modified, other than being deleted. Marking NSX Manager as primary

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/universalsync/configuration
      /role?action=set-as-primary
```

---

**Example 17-10.** Marking NSX Manager as standalone

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/universalsync/configuration
      /role?action=set-as-standalone
```

**Example 17-11.** Query current role on NSX manager

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/universalsync/configuration/role
```

**Example 17-12.** Query certificate thumbprint from the secondary VSM

Request:

```
GET https://NSX-Manager-IP-Address/api/1.0/appliance-management/certificatemanager
      /certificates/nsx
```

Read the field “sha1Hash” for the thumbprint.

**Example 17-13.** Create a Secondary NSX Manager

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/universalsync/configuration/nsxmanagers
```

Request Body:

```
<nsxManagerInfo>
  <nsxManagerIp>10.110.9.131</nsxManagerIp>
  <nsxManagerUsername>admin</nsxManagerUsername>
  <nsxManagerPassword>default</nsxManagerPassword>
  <certificateThumbprint>EA:63:7C:C8:61:80:D9:C8:D4:E7:CB:AA:85:BC:C1:7D:94:8E:6E:14
    </certificateThumbprint>
  <isPrimary>>false</isPrimary>
</nsxManagerInfo>
```

**Example 17-14.** Query Secondary NSX Manager configuration

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/universalsync/configuration/nsxmanagers
```

**Example 17-15.** Update NSX Manager Configuration (IP/Thumbprint)

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/universalsync/configuration/nsxmanagers
     /thumbprint
```

Request Body:

```
<nsxManagerInfo>
  <uuid>bff9b907-829c-4180-a7a1-8dfb04f5a958</uuid>
  <nsxManagerIp>10.112.10.228</nsxManagerIp>
  <certificateThumbprint>9D:45:16:93:68:21:6B:C9:C1:1A:60:AF:08:28:EE:31:76:A5:B0:30
    </certificateThumbprint>
  <isPrimary>>false</isPrimary>
</nsxManagerInfo>
```

---

```
</nsxManagerInfo>
```

---



---

**Example 17-16.** Query Secondary NSX Manager UUID
 

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/universalsync/configuration/nsxmanagers/uuid
```

---



---

**Example 17-17.** Delete Secondary NSX Manager UUID
 

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/universalsync/configuration/nsxmanagers/uuid?force=true\false
```

---



---

**Example 17-18.** Delete Secondary NSX Manager configuration
 

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/universalsync/configuration/nsxmanagers
```

---



---

**Example 17-19.** Sync all objects on NSX Manager
 

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/universalsync/sync?action=invoke
```

---



---

**Example 17-20.** Query sync status per entity
 

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/universalsync/entitystatus?objectType=objectType&objectId=objectId
```

```
e.g.: GET https://NSX-Manager-IP-Address/api/2.0/universalsync/entitystatus?objectType=VdnScope&objectId=globalVdnScope
```

Response Body:

```
<entitySyncStatus>
  <objectId>globalVdnScope</objectId>
  <objectType>VdnScope</objectType>
  <isInSync>false</isInSync>
  <elements>
    <entitySyncElement>
      <vsmId>PRIMARY</vsmId>
      <objectExists>true</objectExists>
      <revision>2</revision>
    </entitySyncElement>
    <entitySyncElement>
      <vsmId>42039A7B-F72C-B0C5-C5FA-2226EB02CE7A</vsmId>
      <objectExists>true</objectExists>
      <revision>1</revision>
    </entitySyncElement>
  </elements>
</entitySyncStatus>
```

---



---

**Example 17-21.** Query config sync status (replicator)
 

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/universalsync/status
```

Response Body:

```
<configstatus>
  <status>successful</status>
  <timestamp>123344567</timestamp>
</configstatus>
```

---

## Universal Transport Zones

### Example 17-22. Create Universal Transport Zone (replicator)

---

You can have only one universal transport zone. You must create the universal transport zone on the primary NSX Manager. Clusters specified in the request must be found in the vCenter Server that is linked to the primary NSX Manager. To add clusters managed by other vCenter Servers in the cross-vCenter environment to the universal transport zones, you must run the expand a cluster API on the associated secondary NSX Manager. See [Appendix 17, “Edit a Transport Zone,”](#) on page 457

On the master:

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/scopes?isUniversal=true
```

Request Body:

```
<vdnScope>
  <name>gtz-post1</name>
  <clusters>
    <cluster>
      <cluster>
        <objectId>domain-c29</objectId>
      </cluster>
    </cluster>
  </clusters>
</vdnScope>
```

---

### Example 17-23. Edit a Transport Zone

---

You can add a cluster (expand scope) to or delete a cluster (shrink scope) from network scope by using following APIs.

Following are supported actions:

expand => Add specified clusters to the existing Universal Transport Zone

shrink => Remove specified clusters from the existing Universal Transport Zone

On the master:

Request :

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/universalvdnscope?action
      =expand
```

Request Body:

```
<vdnScope>
  <name>gtz-post1</name>
  <objectId>globalvdnscope</objectId>
  <clusters>
```

```

    <cluster>
      <cluster>
        <objectId>domain-c35</objectId>
      </cluster>
    </cluster>
  </clusters>
</vdnScope>

```

---

#### Example 17-24. Update Attributes on a Transport Zone

---

On the master:

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/universalvdsnscope
/attributes
```

Request Body :

```

<vdnScope>
  <objectId>universalvdsnscope</objectId>
  <name>gtz-updated</name>
  <description>gtz-description</description>
  <controlPlaneMode>UNICAST_MODE</controlPlaneMode>
  <clusters>
    <cluster>
      <cluster>
        <objectId>domain-c29</objectId>
      </cluster>
    </cluster>
  </clusters>
</vdnScope>

```

---

#### Example 17-25. Query Universal Transport Zone

---

The flag “isUniversal” indicates whether the queried Transport Zone is universal, or local to the NSX Manager.

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/utznId
```

Response Body:

```

<vdnScope>
  <objectId>vdsnscope-2</objectId>
  <type>
    <typeName>VdnScope</typeName>
  </type>
  <name>My Name</name>
  <description>My description</description>
  <revision>0</revision>
  <objectTypeName>VdnScope</objectTypeName>
  <extendedAttributes></extendedAttributes>
  <id>vdsnscope-2</id>
  <clusters>
    <cluster>
      <objectId>domain-c124</objectId>
      <type>
        <typeName>ClusterComputeResource</typeName>
      </type>
      <name>vxlان-cluster</name>
      <scope>
        <id>datacenter-2</id>
      </scope>
    </cluster>
  </clusters>

```

```

        <objectTypeName>Datacenter</objectTypeName>
        <name>dc1</name>
    </scope>
    <extendedAttributes></extendedAttributes>
</cluster>
<cluster>...</cluster>
</clusters>
<virtualWireCount>10</virtualWireCount>
<isUniversal>TRUE</isUniversal>
</vdnScope>

```

---

### Example 17-26. Query all Transport Zones

---

You can retrieve all transport zones on a given NSX Manager, and the `isUniversal` flag will indicate if the transport zone is universal, or local to that NSX Manager.

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/scopes
```

Response Body:

```

<vdnScopes>
  <vdnScope>
    <objectId>global-vdnscope</objectId>
    <type>
      <typeName>vdnScope</typeName>
    </type>
    <name>My Name</name>
    <description>My description</description>
    <revision>0</revision>
    <objectTypeName>vdnScope</objectTypeName>
    <extendedAttributes></extendedAttributes>
    <id>global-vdnscope</id>
    <clusters>
      <cluster>
        <objectId>domain-c124</objectId>
        <type>
          <typeName>ClusterComputeResource</typeName>
        </type>
        <name>vxlان-cluster</name>
        <scope>
          <id>datacenter-2</id>
          <objectTypeName>Datacenter</objectTypeName>
          <name>dc1</name>
        </scope>
        <extendedAttributes></extendedAttributes>
      </cluster>
      <cluster>...</cluster>
    </clusters>
    <virtualWireCount>10</virtualWireCount>
    <isUniversal>TRUE</isUniversal>
  </vdnScope>
  <vdnScope>...</vdnScope>
</vdnScopes>

```

---

### Example 17-27. Delete Universal Transport Zone

---

On the master:

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/universalvdsnscope
```

## Universal Logical Switches

If you are creating, modifying, or deleting universal logical switches, you must run the API request on the primary NSX Manager. Universal logical switches are read-only from secondary NSX Managers. The `isUniversal` flag will indicate if the logical switch is universal, or local to that NSX Manager.

---

### Example 17-28. Create Universal Logical Switch

This request will create Universal Logical Switch on the Universal Transport Zone.

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/scopes/universalvdnscope
/virtualwires
```

Request Body:

```
<virtualWireCreateSpec>
  <name>Universal Logical Switch</name>
  <description>ULS description</description>
  <tenantId>ULS-tenant</tenantId>
</virtualWireCreateSpec>
```

---

### Example 17-29. Update Universal Logical Switch

You can modify the control plane mode of a ULS. The possible options are: Unicast & Hybrid.

On the master:

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/vdn/virtualwires/ulsId
```

Request Body:

```
<virtualWire>
  <name>Universal Logical Switch</name>
  <description>ULS description - update1</description>
  <tenantId>gls-tenant</tenantId>
  <vdsScopeId>universalvdnscope</vdsScopeId>
  <controlPlaneMode>UNICAST_MODE</controlPlaneMode>
</virtualWire>
```

---

### Example 17-30. Query Universal Logical Switch

The `isUniversal` flag will indicate if a logical switch is universal, or local to that NSX Manager.

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/virtualwires/ulsId
```

Response Body:

```
<virtualWire>
  <name>Test logical switch</name>
  <description>Test logical switch Description</description>
  <objectId>virtualwire-4</objectId>
  <vdsScopeId>vdscope-3</vdsScopeId>
  <revision>1</revision>
  <vdsContextWithBacking>
    <teaming>ETHER_CHANNEL</teaming>
    <switchId>dvs-162</switchId>
```

```

    <backingType>PortGroup</backingType>
    <backingValue>pg-moid</backingValue>
  </vdsContextWithBacking>
  <vdnId>5002</vdnId>
  <multicastAddr>239.0.0.3</multicastAddr>
  <isUniversal>TRUE</isUniversal>
</virtualWire>

```

---

### Example 17-31. Query all Universal Logical Switches

---

You can retrieve all logical switches on a given NSX Manager, and the `isUniversal` flag will indicate if the logical switch is universal, or local to that NSX Manager.

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/virtualwires
```

Response Body:

```

<virtualwires>
  <sortedDataPage>
    <datapart class="virtualWire">
      <objectId>virtualwire-1</objectId>
      <name>vwire1</name>
      <description>logical switch 1</description>
      <tenantId>logical switch tenant</tenantId>
      <revision>0</revision>
      <vdnScopeId>vdnscope-7</vdnScopeId>
      <vdsContextWithBacking>
        <teaming>ETHER_CHANNEL</teaming>
        <switchId>dvs-81</switchId>
        <backingType>portgroup</backingType>
        <backingValue>dvportgroup-88</backingValue>
      </vdsContextWithBacking>
      <vdnId>5002</vdnId>
      <multicastAddr>239.0.0.3</multicastAddr>
      <isUniversal>TRUE</isUniversal>
    </datapart>
    <datapart class="virtualWire">...</datapart>
  <pagingInfo>
    <pageSize>20</pageSize>
    <startIndex>0</startIndex>
    <totalCount>3</totalCount>
    <sortOrderAscending>false</sortOrderAscending>
  </pagingInfo>
</sortedDataPage>
</virtualwires>

```

---

### Example 17-32. Delete Universal Logical Switch

---

On the master:

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/virtualwires/uuid
```

---

## Universal Segment ID Pool (VNI Pool)

You can retrieve all segment ID pool (VNI pools) on a given NSX Manager, and the `isUniversal` flag will indicate if the segment ID pool is universal, or local to that NSX Manager.

**Example 17-33.** Create new Universal VNI Pool

The segment range is inclusive - the beginning and ending IDs are included.

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments?isUniversal=true>

Request Body:

```
<segmentRange>
  <id>1</id>
  <name>name</name>
  <desc>desc</desc>
  <begin>1000</begin>
  <end>1500</end>
</segmentRange>
```

**Example 17-34.** Query All Segment Ranges

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments>

Response Body:

```
<segmentRanges>
  <segmentRange>
    <id>1</id>
    <name>name</name>
    <desc>desc</desc>
    <begin>1000</begin>
    <end>1500</end>
    <isUniversal>TRUE</isUniversal>
  </segmentRange>
  <segmentRange>
    <id>2</id>
    <name>name</name>
    <desc>desc</desc>
    <begin>1501</begin>
    <end>1510</end>
    <isUniversal>FALSE</isUniversal>
  </segmentRange>
</segmentRanges>
```

**Example 17-35.** Query a specific VNI Pool Segment

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments/segmentId>

Response Body:

```
<segmentRange>
  <id>1</id>
  <name>name</name>
  <desc>desc</desc>
  <begin>1000</begin>
  <end>1500</end>
  <isUniversal>TRUE</isUniversal>
</segmentRange>
```

**Example 17-36.** Update a Universal Segment VNI Pool

You can update the name, description, or end of a segment ID range. If you are creating, modifying, or deleting universal segment ID pool, you must run the API request on the primary NSX Manager. Universal segment ID pools are read-only from secondary NSX Managers.

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments/segmentId
```

Request Body:

```
<segmentRange>
  <id>1</id>
  <name>name</name>
  <desc>desc</desc>
  <end>1500</end>
</segmentRange>
```

**Example 17-37.** Delete a VNI Segment Range

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/config/segments/segmentId
```

**Universal Multicast Address Range**

API for CRUD on Universal Multicast address Pool will use the existing Local Multicast address Pool API. In order to differentiate it from Local Multicast address Pool API, we will add a new boolean field 'isUniversal' to MulticastRangeDto. MulticastRangeDto is the payload to CREATE/UPDATE API on Multicast address Pools. Also the GET calls will remain same, and API user will get all the Multicast address Pools (Local + Universal). API user can easily differentiate between Local and Universal Multicast address Pools returned by checking the 'global' boolean field in MulticastRangeDto.

If you are creating, modifying, or deleting universal multicast address ranges, you must run the API request on the primary NSX Manager. Universal multicast address ranges are read-only from secondary NSX Managers. The `isUniversal` flag will indicate if the logical switch is universal, or local to that NSX Manager.

**Example 17-38.** Create new Universal Multicast Address Pool

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/vdn/config/multicasts?isUniversal=true
```

Request Body:

```
<multicastRange>
  <name>Global multi</name>
  <desc>desc</desc>
  <begin>239.1.1.1</begin>
  <end>239.3.3.3</end>
</multicastRange>
```

**Example 17-39.** Query All Multicast Address Ranges

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/config/multicasts
```

**Example 17-40.** Query a specific Multicast Address Range

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/vdn/config/multicasts/poolId
```

**Example 17-41.** Update a Universal Multicast Address Range

You can update the name, description, or end of a multicastaddress range.

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/vdn/config/multicasts/segmentId
```

Request Body:

```
<multicastRange>
  <name>Global multi updated</name>
  <desc>desc updated</desc>
  <begin>239.1.1.1</begin>
  <end>239.3.3.4</end>
</multicastRange>
```

**Example 17-42.** Delete a Multicast Address pool

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/vdn/config/multicasts/poolId
```

## Distributed Firewall for Cross-vCenter NSX Environments

You can create a universal distributed firewall rule section, and any rules inside that section will be synchronized from the primary NSX Manager to the secondary NSX Managers. You can have one universal section for layer 2 firewall rules, and one universal section for layer 3 firewall rules.

If you are creating, modifying, or deleting universal distributed firewall sections, you must run the API request on the primary NSX Manager. Universal distributed firewall sections are read-only from secondary NSX Managers.

**Example 17-43.** Query All universal sections

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/config/sections?managedBy=universalroot-0
```

Response Body:

```
<sections>
  <section class="section" id="500a3923-2c97-4816-97b1-20e3d62c0322" name="gsection-1"
    generationNumber="1432835107627" timestamp="1432835107627"
    managedBy="universalroot-0" type="LAYER3" />
</sections>
```

**Example 17-44.** Add new Universal Section

Request:

```
POST https://NSX-Manager-IP-Address/api/4.0/firewall/config/sections
```

Request Body:

```

<section id="112dd42a-38be-48ff-b3a5-839fefec68ca" name="UniversalSection"
  generationNumber="1432849937014" timestamp="1432849937014"
  managedBy="universalroot-0" type="LAYER3">
  <rule id="2147483653" disabled="false" logged="false" managedBy="universalroot-0">
    <action>allow</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>112dd42a-38be-48ff-b3a5-839fefec68ca</sectionId>
    <sources excluded="false">
      <source>
        <name>GlobalIPSet</name>
        <value>ipset-7d4ea07c-c28b-4324-a439-a76999c80b3e</value>
        <type>IPSet</type>
        <isValid>true</isValid>
      </source>
    </sources>
    <services>
      <service>
        <name>FTP</name>
        <value>application-5293f54f-4669-4ff8-a0fd-7d0f6c57d6a5</value>
        <type>Application</type>
        <isValid>true</isValid></service>
    </services>
    <direction>inout</direction>
    <packetType>any</packetType>
  </rule>
  <rule id="2147483652" disabled="false" logged="false" managedBy="universalroot-0">
    <action>allow</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>112dd42a-38be-48ff-b3a5-839fefec68ca</sectionId>
    <sources excluded="false">
      <source>
        <value>192.168.1.1</value>
        <type>Ipv4Address</type>
        <isValid>true</isValid>
      </source>
    </sources>
    <destinations excluded="false">
      <destination>
        <name>GlobalSG</name>
        <value>securitygroup-382a5eac-2191-49e5-8979-18de016a166b</value>
        <type>SecurityGroup</type>
        <isValid>true</isValid>
      </destination>
    </destinations>
    <direction>in/out</direction>
    <packetType>any</packetType>
  </rule>
</section>

```

**Example 17-45.** Query a Universal Section

Request:

GET https://NSX-Manager-IP-Address/api/4.0/firewall/config/sections/*sectionId*

Response Body:

```

<section id="112dd42a-38be-48ff-b3a5-839fefec68ca" name="UniversalSection"
  generationNumber="1432849937014" timestamp="1432849937014"
  managedBy="universalroot-0" type="LAYER3">
  <rule id="2147483653" disabled="false" logged="false" managedBy="universalroot-0">
    <action>allow</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>112dd42a-38be-48ff-b3a5-839fefec68ca</sectionId>
    <sources excluded="false">
      <source>
        <name>GlobalIPSet</name>
        <value>ipset-7d4ea07c-c28b-4324-a439-a76999c80b3e</value>
        <type>IPSet</type>
        <isValid>true</isValid>
      </source>
    </sources>
    <services>
      <service>
        <name>FTP</name>
        <value>application-5293f54f-4669-4ff8-a0fd-7d0f6c57d6a5</value>
        <type>Application</type>
        <isValid>true</isValid></service>
    </services>
    <direction>inout</direction>
    <packetType>any</packetType>
  </rule>
  <rule id="2147483652" disabled="false" logged="false" managedBy="universalroot-0">
    <action>allow</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>112dd42a-38be-48ff-b3a5-839fefec68ca</sectionId>
    <sources excluded="false">
      <source>
        <value>192.168.1.1</value>
        <type>Ipv4Address</type>
        <isValid>true</isValid>
      </source>
    </sources>
    <destinations excluded="false">
      <destination>
        <name>GlobalSG</name>
        <value>securitygroup-382a5eac-2191-49e5-8979-18de016a166b</value>
        <type>SecurityGroup</type>
        <isValid>true</isValid>
      </destination>
    </destinations>
    <direction>in/out</direction>
    <packetType>any</packetType>
  </rule>
</section>

```

```

</rule>
</section>

```

---

**Example 17-46.** Update an existing Universal Section
 

---

Request:

PUT <https://NSX-Manager-IP-Address/api/4.0/firewall/config/sections/sectionId>

Request Body:

```

<section id="112dd42a-38be-48ff-b3a5-839fefec68ca" name="UniversalSection"
  generationNumber="1432851508109" timestamp="1432851508109"
  managedBy="universalroot-0" type="LAYER3">
  <rule id="2147483653" disabled="false" logged="false" managedBy="universalroot-0">
    <action>allow</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>112dd42a-38be-48ff-b3a5-839fefec68ca</sectionId>
    <sources excluded="false">
      <source>
        <name>GlobalIPSet</name>
        <value>ipset-7d4ea07c-c28b-4324-a439-a76999c80b3e</value>
        <type>IPSet</type>
        <isValid>true</isValid>
      </source>
    </sources>
    <services>
      <service>
        <name>FTP</name>
        <value>application-5293f54f-4669-4ff8-a0fd-7d0f6c57d6a5</value>
        <type>Application</type>
        <isValid>true</isValid>
      </service>
    </services>
    <direction>inout</direction>
    <packetType>any</packetType>
  </rule>
  <rule id="2147483652" disabled="false" logged="false" managedBy="universalroot-0">
    <action>allow</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>112dd42a-38be-48ff-b3a5-839fefec68ca</sectionId>
    <sources excluded="false">
      <source>
        <value>192.168.1.1</value>
        <type>Ipv4Address</type>
        <isValid>true</isValid>
      </source>
    </sources>
    <destinations excluded="false">
      <destination>
        <name>GlobalSG</name>
        <value>securitygroup-382a5eac-2191-49e5-8979-18de016a166b</value>
        <type>SecurityGroup</type>

```

```

        <isValid>true</isValid>
      </destination>
    </destinations>
    <direction>inout</direction>
    <packetType>any</packetType>
  </rule>
</section>

```

---

#### Example 17-47. Delete a Universal Section

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/4.0/firewall/config/sections/sectionId
```

---

#### Example 17-48. Query status (realized state locally) for universal section

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/status/sections/sectionId
```

Response Body:

```

<firewallStatus>
  <startTime>1432853158515</startTime>
  <status>published</status>
  <generationNumber>1432853158515</generationNumber>
  <clusterList>
    <clusterStatus>
      <clusterId>domain-c18</clusterId>
      <status>published</status>
      <generationNumber>1432853158515</generationNumber>
      <hostStatusList>
        <hostStatus>
          <hostId>host-20</hostId>
          <hostName>10.24.227.86</hostName>
          <status>published</status>
          <errorCode>0</errorCode>
          <startTime>1432818086424</startTime>
          <endTime>1432853159533</endTime>
          <generationNumber>1432853158515</generationNumber>
          <clusterId>domain-c18</clusterId>
        </hostStatus>
        <hostStatus>
          <hostId>host-21</hostId>
          <hostName>10.24.227.124</hostName>
          <status>published</status>
          <errorCode>0</errorCode>
          <startTime>1432818084422</startTime>
          <endTime>1432853159537</endTime>
          <generationNumber>1432853158515</generationNumber>
          <clusterId>domain-c18</clusterId>
        </hostStatus>
      </hostStatusList>
    </clusterStatus>
  </clusterList>
</firewallStatus>

```

---

#### Example 17-49. Query draft with latest global (Secondary Only)

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/drafts/draftId
```

Response Body:

```

<firewallDraft id="23" name="AutoSaved_Thursday, May 28, 2015 10:46:00 PM GMT"
  timestamp="1432853160790">
  <description>Auto saved configuration</description>
  <preserve>>false</preserve>
  <user>replicator-cb6da5bf-cf89-439b-b24d-f4e2a8dfe697</user>
  <mode>autosaved</mode>
  <config timestamp="1432817726773">
    <contextId>globalroot-0</contextId>
    <layer3Sections>
      <section id="cda3ed25-222b-4236-9310-9c3bce192392" name="UniversalSection"
        generationNumber="1432853158515" timestamp="1432853160753"
        managedBy="universalroot-0" type="LAYER3">
        <rule id="2147483655" disabled="false" logged="false"
          managedBy="universalroot-0">
          <action>allow</action>
          <appliedToList>
            <appliedTo>
              <name>DISTRIBUTED_FIREWALL</name>
              <value>DISTRIBUTED_FIREWALL</value>
              <type>DISTRIBUTED_FIREWALL</type>
              <isValid>>true</isValid>
            </appliedTo>
          </appliedToList>
          <sectionId>cda3ed25-222b-4236-9310-9c3bce192392</sectionId>
          <sources excluded="false">
            <source>
              <name>GlobalIPSet</name>
              <value>ipset-7d4ea07c-c28b-4324-a439-a76999c80b3e</value>
              <type>IPSet</type>
              <isValid>>true</isValid>
            </source>
          </sources>
          <services>
            <service>
              <name>FTP</name>
              <value>application-5293f54f-4669-4ff8-a0fd-7d0f6c57d6a5</value>
              <type>Application</type>
              <isValid>>true</isValid>
            </service>
          </services>
          <direction>inout</direction>
          <packetType>any</packetType>
        </rule>
        <rule id="2147483654" disabled="false" logged="false"
          managedBy="universalroot-0">
          <action>allow</action>
          <appliedToList>
            <appliedTo>
              <name>DISTRIBUTED_FIREWALL</name>
              <value>DISTRIBUTED_FIREWALL</value>
              <type>DISTRIBUTED_FIREWALL</type>
              <isValid>>true</isValid>
            </appliedTo>
          </appliedToList>
          <sectionId>cda3ed25-222b-4236-9310-9c3bce192392</sectionId>
          <sources excluded="false">
            <source>
              <value>192.168.1.1</value>
              <type>Ipv4Address</type>
              <isValid>>true</isValid>
            </source>
          </sources>
          <destinations excluded="false">
            <destination>
              <name>GlobalSG</name>
              <value>securitygroup-382a5eac-2191-49e5-8979-18de016a166b</value>
              <type>SecurityGroup</type>
              <isValid>>true</isValid>

```

```

        </destination>
      </destinations>
      <direction>inout</direction>
      <packetType>any</packetType>
    </rule>
  </section>
  <section id="1003" name="Default Section Layer3"
    generationNumber="1432817726773" timestamp="1432817726773" type="LAYER3">
    <rule id="1003" disabled="false" logged="false">
      <name>Default Rule NDP</name>
      <action>allow</action>
      <appliedToList>
        <appliedTo>
          <name>DISTRIBUTED_FIREWALL</name>
          <value>DISTRIBUTED_FIREWALL</value>
          <type>DISTRIBUTED_FIREWALL</type>
          <isValid>true</isValid>
        </appliedTo>
      </appliedToList>
      <sectionId>1003</sectionId>
      <services>
        <service>
          <name>IPv6-ICMP Neighbor Advertisement</name>
          <value>application-26</value>
          <type>Application</type>
          <isValid>true</isValid>
        </service>
        <service>
          <name>IPv6-ICMP Neighbor Solicitation</name>
          <value>application-287</value>
          <type>Application</type>
          <isValid>true</isValid>
        </service>
      </services>
      <direction>inout</direction>
      <packetType>any</packetType>
    </rule>
    <rule id="1002" disabled="false" logged="false">
      <name>Default Rule DHCP</name>
      <action>allow</action>
      <appliedToList>
        <appliedTo>
          <name>DISTRIBUTED_FIREWALL</name>
          <value>DISTRIBUTED_FIREWALL</value>
          <type>DISTRIBUTED_FIREWALL</type>
          <isValid>true</isValid>
        </appliedTo>
      </appliedToList>
      <sectionId>1003</sectionId>
      <services>
        <service>
          <name>DHCP-Server</name>
          <value>application-262</value>
          <type>Application</type>
          <isValid>true</isValid>
        </service>
        <service>
          <name>DHCP-Client</name>
          <value>application-18</value>
          <type>Application</type>
          <isValid>true</isValid>
        </service>
      </services>
      <direction>inout</direction>
      <packetType>any</packetType>
    </rule>
    <rule id="1001" disabled="false" logged="false">
      <name>Default Rule</name>

```

```

    <action>allow</action>
    <appliedToList>
      <appliedTo>
        <name>DISTRIBUTED_FIREWALL</name>
        <value>DISTRIBUTED_FIREWALL</value>
        <type>DISTRIBUTED_FIREWALL</type>
        <isValid>true</isValid>
      </appliedTo>
    </appliedToList>
    <sectionId>1003</sectionId>
    <precedence>default</precedence>
    <direction>inout</direction>
    <packetType>any</packetType>
  </rule>
</section>
</layer3Sections>
<layer2Sections>
  <section id="1001" name="Default Section Layer2"
    generationNumber="1432817726773" timestamp="1432817726773" type="LAYER2">
    <rule id="1004" disabled="false" logged="false">
      <name>Default Rule</name>
      <action>allow</action>
      <appliedToList>
        <appliedTo>
          <name>DISTRIBUTED_FIREWALL</name>
          <value>DISTRIBUTED_FIREWALL</value>
          <type>DISTRIBUTED_FIREWALL</type>
          <isValid>true</isValid>
        </appliedTo>
      </appliedToList>
      <sectionId>1001</sectionId>
      <precedence>default</precedence>
      <direction>inout</direction>
      <packetType>any</packetType>
    </rule>
  </section>
</layer2Sections>
<layer3RedirectSections>
  <section id="1002" name="Default Section" generationNumber="1432817726773"
    timestamp="1432817726773" type="L3REDIRECT" />
</layer3RedirectSections>
<generationNumber>1432853158515</generationNumber>
</config>
</firewallDraft>

```

---

**Example 17-50.** Export Draft with latest global (Secondary Only)
 

---

Request:

```
GET https://NSX-Manager-IP-Address/api/4.0/firewall/globalroot-0/drafts/draftId/action
/export?getLatestForUniversal=true
```

Response Body:

```

<firewallDraft name="AutoSaved_Thursday, May 28, 2015 10:46:00 PM GMT"
  timestamp="1432853160790">
  <description>Auto saved configuration</description>
  <preserve>>false</preserve>
  <user>replicator-cb6da5bf-cf89-439b-b24d-f4e2a8dfe697</user>
  <mode>autosaved</mode>
  <config timestamp="1432817726773">
    <contextId>globalroot-0</contextId>
    <layer3Sections>
      <section id="cda3ed25-222b-4236-9310-9c3bce192392" name="UniversalSection"
        generationNumber="1432853158515" timestamp="1432853160753"
        managedBy="universalroot-0" type="LAYER3">

```

```

<rule id="2147483655" disabled="false" logged="false"
  managedBy="universalroot-0">
  <action>allow</action>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>cda3ed25-222b-4236-9310-9c3bce192392</sectionId>
  <sources excluded="false">
    <source>
      <name>GlobalIPSet</name>
      <value>ipset-7d4ea07c-c28b-4324-a439-a76999c80b3e</value>
      <type>IPSet</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <services>
    <service>
      <name>FTP</name>
      <value>application-5293f54f-4669-4ff8-a0fd-7d0f6c57d6a5</value>
      <type>Application</type>
      <isValid>true</isValid>
    </service>
  </services>
  <direction>inout</direction>
  <packetType>any</packetType>
</rule>
<rule id="2147483654" disabled="false" logged="false"
  managedBy="universalroot-0">
  <action>allow</action>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <isValid>true</isValid>
    </appliedTo>
  </appliedToList>
  <sectionId>cda3ed25-222b-4236-9310-9c3bce192392</sectionId>
  <sources excluded="false">
    <source>
      <value>192.168.1.1</value>
      <type>Ipv4Address</type>
      <isValid>true</isValid>
    </source>
  </sources>
  <destinations excluded="false">
    <destination>
      <name>GlobalSG</name>
      <value>securitygroup-382a5eac-2191-49e5-8979-18de016a166b</value>
      <type>SecurityGroup</type>
      <isValid>true</isValid>
    </destination>
  </destinations>
  <direction>inout</direction>
  <packetType>any</packetType>
</rule>
</section>
<section id="1003" name="Default Section Layer3"
  generationNumber="1432817726773" timestamp="1432817726773" type="LAYER3">
  <rule id="1003" disabled="false" logged="false">
    <name>Default Rule NDP</name>
    <action>allow</action>
    <appliedToList>

```

```

    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <invalid>true</invalid>
    </appliedTo>
  </appliedToList>
</sectionId>1003</sectionId>
<services>
  <service>
    <name>IPv6-ICMP Neighbor Advertisement</name>
    <value>application-26</value>
    <type>Application</type>
    <invalid>true</invalid>
  </service>
  <service>
    <name>IPv6-ICMP Neighbor Solicitation</name>
    <value>application-287</value>
    <type>Application</type>
    <invalid>true</invalid>
  </service>
</services>
<direction>inout</direction>
<packetType>any</packetType>
</rule>
<rule id="1002" disabled="false" logged="false">
  <name>Default Rule DHCP</name>
  <action>allow</action>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <invalid>true</invalid>
    </appliedTo>
  </appliedToList>
</sectionId>1003</sectionId>
<services>
  <service>
    <name>DHCP-Server</name>
    <value>application-262</value>
    <type>Application</type>
    <invalid>true</invalid>
  </service>
  <service>
    <name>DHCP-Client</name>
    <value>application-18</value>
    <type>Application</type>
    <invalid>true</invalid>
  </service>
</services>
<direction>inout</direction>
<packetType>any</packetType>
</rule>
<rule id="1001" disabled="false" logged="false">
  <name>Default Rule</name>
  <action>allow</action>
  <appliedToList>
    <appliedTo>
      <name>DISTRIBUTED_FIREWALL</name>
      <value>DISTRIBUTED_FIREWALL</value>
      <type>DISTRIBUTED_FIREWALL</type>
      <invalid>true</invalid>
    </appliedTo>
  </appliedToList>
</sectionId>1003</sectionId>
<precedence>default</precedence>
<direction>inout</direction>

```

```

        <packetType>any</packetType>
    </rule>
</section>
</layer3Sections>
<layer2Sections>
    <section id="1001" name="Default Section Layer2"
        generationNumber="1432817726773" timestamp="1432817726773" type="LAYER2">
        <rule id="1004" disabled="false" logged="false">
            <name>Default Rule</name>
            <action>allow</action>
            <appliedToList>
                <appliedTo>
                    <name>DISTRIBUTED_FIREWALL</name>
                    <value>DISTRIBUTED_FIREWALL</value>
                    <type>DISTRIBUTED_FIREWALL</type>
                    <isValid>true</isValid>
                </appliedTo>
            </appliedToList>
            <sectionId>1001</sectionId>
            <precedence>default</precedence>
            <direction>inout</direction>
            <packetType>any</packetType>
        </rule>
    </section>
</layer2Sections>
<layer3RedirectSections>
    <section id="1002" name="Default Section" generationNumber="1432817726773"
        timestamp="1432817726773" type="L3REDIRECT" />
</layer3RedirectSections>
<generationNumber>1432853158515</generationNumber>
</config>
</firewallDraft>

```

---

## Universal Grouping Object Universal IP Sets (IP Address Groups)

If you are creating, modifying, or deleting universal IP Sets, you must run the API request on the primary NSX Manager. Universal IP Sets are read-only from secondary NSX Managers.

### Example 17-51. Create Universal IP Set

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/services/ipset/universalroot-0>

Request Body:

```

<ipset>
    <description> New Description </description>
    <name>TestIPSet2</name>
    <value>10.112.201.8-10.112.201.14</value>
</ipset>

```

---

### Example 17-52.

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/ipset/objectId>

---

### Example 17-53.

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/ipset/scope/universalroot-0>

---

#### Example 17-54.

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/common/query/universal/IPSet>

---

#### Example 17-55.

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/services/ipset/objectId>

Request Body:

```
<ipset>
  <objectId>ipset-ae40752f-3b9b-4885-b63c-551fbaa459ab</objectId>
  <type>
    <typeName>IPSet</typeName>
  </type>
  <description>Updated Description</description>
  <name>TestIPSetUpdated</name>
  <revision>2</revision>
  <objectTypeName />
  <value>10.112.200.1,10.112.200.4-10.112.200.10</value>
</ipset>
```

---

#### Example 17-56.

---

Request:

DELETE <https://NSX-Manager-IP-Address/api/2.0/services/ipset/objectId?force=false>

---

#### Example 17-57.

---

Request:

POST <https://NSX-Manager-IP-Address/api/2.0/services/ipset/universalroot-0?objectId=objectId>

---

#### Example 17-58.

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/ipset/objectId>

---

#### Example 17-59.

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/common/query/universal/IPSet>

---

#### Example 17-60.

---

Request:

---

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/ipset/objectId
```

---

### Example 17-61.

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/ipset/objectId?force=true
```

---

## Universal MAC Sets

**Example 17-62.** If you are creating, modifying, or deleting universal MAC Sets, you must run the API request on the primary NSX Manager. Universal MAC Sets are read-only from secondary NSX Managers. **Create Universal MAC Sets**

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/macset/universalroot-0
```

Request Body:

```
<macset>
  <description>description</description>
  <name>TestMACSet1</name>
  <value>22:33:44:55:66:77,00:11:22:33:44:55,aa:bb:cc:dd:ee:ff</value>
</macset>
```

---

### Example 17-63. Query MACSet

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/macset/objectId
```

---

### Example 17-64. Query MACSet in a scope

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/macset/scope/universalroot-0
```

---

### Example 17-65. Query all MACSets

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/common/query/universal/MACSet
```

---

### Example 17-66. Update MACSet

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/macset/objectId
```

Request Body:

```
<macset>
  <objectId>macset-ae40752f-3b9b-4885-b63c-551fbaa459ab</objectId>
  <type>
    <typeName>MACSet</typeName>
  </type>
```

```

    <description>Updated Description</description>
    <name>TestMACSet1updated</name>
    <revision>2</revision>
    <objectTypeName />
    <value>22:33:44:55:66:77,00:11:22:33:44:55</value>
  </macset>

```

---

#### Example 17-67. Delete MACSet

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/macset/objectId?force=false
```

---

## Universal Services (Applications)

**Example 17-68.** If you are creating, modifying, or deleting universal services, you must run the API request on the primary NSX Manager. Universal services are read-only from secondary NSX Managers. **Create Application Service**

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/application/universalroot-0
```

Request Body:

```

<application>
  <description>description</description>
  <name> TestApplication</name>
  <element>
    <applicationProtocol>UDP</applicationProtocol>
    <value>10,20-30,45</value>
  </element>
</application>

```

---

#### Example 17-69. Query Application Service

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/application/objectId
```

---

#### Example 17-70. Query Application Services in a scope

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/application/scope/universalroot-0
```

---

#### Example 17-71. Query all Application Services

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/common//query/universal/Application
```

---

#### Example 17-72. Update Application Service

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/application/objectId
```

Request Body:

```
<application>
  <objectId>application-42f82c3a-2b1c-47ce-97af-6cee29415bd0</objectId>
  <type>
    <typeName>Application</typeName>
  </type>
  <description>Updated Description</description>
  <name>TestApplicationUpdated</name>
  <revision>2</revision>
  <objectTypeName>Application</objectTypeName>
  <element>
    <applicationProtocol>UDP</applicationProtocol>
    <value>10,20-30</value>
  </element>
</application>
```

---

#### Example 17-73. Delete Application Service

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/application/objectId?force=true
```

---

## Universal Service Groups (Application Groups)

If you are creating, modifying, or deleting universal service groups, you must run the API request on the primary NSX Manager. Universal service groups are read-only from secondary NSX Managers.

#### Example 17-74. Create Application Group

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/universalroot-0
```

Request Body:

```
<applicationGroup>
  <name>testglobalAG</name>
  <description>Updated with description</description>
</applicationGroup>
```

---

#### Example 17-75. Create bulk Application Group

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/bulk
      /universalroot-0
```

Request Body:

```
<applicationGroup>
  <name>testglobalAG</name>
  <description>Updated with description</description>
  <member>
    <objectId>application-42f82c3a-2b1c-47ce-97af-6cee29415bd0</objectId>
  </member>
</applicationGroup>
```

---

#### Example 17-76. Query Application Group

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/objectId>

---

**Example 17-77.** Query Application Groups in a scope

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/scope/universalroot-0>

---

**Example 17-78.** Query all Application Groups

---

Request:

GET <https://NSX-Manager-IP-Address/api/2.0/services/common/query/universal/ApplicationGroup>

---

**Example 17-79.** Update Application Group

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/services/common/query/universal/ApplicationGroup>

Request Body:

```
<applicationgroup>
  <objectId>applicationgroup-ae40752f-3b9b-4885-b63c-551fbaa459ab</objectId>
  <type>
    <typeName>ApplicationGroup</typeName>
  </type>
  <description> Updated Description </description>
  <name>TestApplicationGroup1updated</name>
  <member>
    <objectId>application-42f82c3a-2b1c-47ce-97af-6cee29415bd0</objectId>
  </member>
</applicationgroup>
```

---

**Example 17-80.** Update bulk Application Group

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/objectId>

Request Body:

```
<applicationgroup>
  <objectId>applicationgroup-ae40752f-3b9b-4885-b63c-551fbaa459ab</objectId>
  <type>
    <typeName>ApplicationGroup</typeName>
  </type>
  <description>updated Description </description>
  <name>TestApplicationGroup1updated</name>
  <revision>0</revision>
  <objectTypeName />
  <member>
    <objectId>application-42f82c3a-2b1c-47ce-97af-6cee29415bd0</objectId>
  </member>
</applicationgroup>
```

---

**Example 17-81.** Delete Application Group

---

Request:

---

```
DELETE https://NSX-Manager-IP-Address/api/2.0/services/applicationgroup/objectId?force=false
```

---

## Universal Security Group

**Example 17-82.** If you are creating, modifying, or deleting universal security groups, you must run the API request on the primary NSX Manager. Universal security groups are read-only from secondary NSX Managers.**Create Universal Security Group**

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/universalroot-0
```

Request Body:

```
<securitygroup>
  <name>SecurityGroup-1</name>
  <description>some desc</description>
</securitygroup>
```

---

### Example 17-83. Create bulk Security Group

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/bulk/universalroot-0
```

Request Body:

```
<securitygroup>
  <name>SecurityGroup-1</name>
  <member>
    <objectId>ipset-76c7550e-1453-4c98-94c3-ec6a408cddc0</objectId>
  </member>
</securitygroup>
```

---

### Example 17-84. Query Security Group

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId
```

---

### Example 17-85. Query Security Groups in a scope

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/scope/universalroot-0
```

---

### Example 17-86. Query all Security Groups

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/common/query/universal/SecurityGroup
```

---

### Example 17-87. Update Security Group

---

Request:

```
PUT https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId
```

Request Body:

```

<securitygroup>
  <objectId>securitygroup-7d649fae-ff27-4dbe-800d-793382826e4a</objectId>
  <objectTypeName>SecurityGroup</objectTypeName>
  <revision>2</revision>
  <type>
    <typeName>SecurityGroup updated</typeName>
  </type>
  <name>SecurityGroup-1 updated</name>
  <scope>
    <id>universalroot-0</id>
    <objectTypeName>UniversalRoot</objectTypeName>
    <name>Universal</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <isUniversal>true</isUniversal>
  <inheritanceAllowed>>false</inheritanceAllowed>
</securitygroup>

```

---

### Example 17-88. Update bulk Security Group

---

Request:

PUT <https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/bulk/objectId>

Request Body:

```

<securitygroup>
  <objectId>securitygroup-7d649fae-ff27-4dbe-800d-793382826e4a</objectId>
  <objectTypeName>SecurityGroup</objectTypeName>
  <revision>2</revision>
  <type>
    <typeName>SecurityGroup updated</typeName>
  </type>
  <name>SecurityGroup-1</name>
  <scope>
    <id>universalroot-0</id>
    <objectTypeName>UniversalRoot</objectTypeName>
    <name>Universal</name>
  </scope>
  <clientHandle></clientHandle>
  <extendedAttributes></extendedAttributes>
  <isUniversal>true</isUniversal>
  <inheritanceAllowed>>false</inheritanceAllowed>
  <member>
    <objectId>ipset-76c7550e-1453-4c98-94c3-ec6a408cddc0</objectId>
    </scope>
    <clientHandle></clientHandle>
    <extendedAttributes></extendedAttributes>
    <isUniversal>true</isUniversal>
  </member>
</securitygroup>

```

---

### Example 17-89. Delete Security Group

---

Request:

DELETE <https://NSX-Manager-IP-Address/api/2.0/services/securitygroup/objectId?force=false>

---



## Task Framework Management

---

The NSX Manager requires communication with your vCenter Server and services such as DNS and NTP to provide details on your VMware Infrastructure inventory.

The chapter includes the following topics:

- [“About Task Framework”](#) on page 483
- [“Query Job Instances for Job ID”](#) on page 484
- [“Query Latest Job Instances for Job ID”](#) on page 485
- [“Block REST Thread”](#) on page 485
- [“Query Job Instances by Criterion”](#) on page 485

---

**IMPORTANT** All REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

### About Task Framework

The task framework provides the abstraction needed to execute asynchronous tasks using a global thread pool.

A Job is identified by a Job ID. A job has a set of tasks within it. These tasks are executed either synchronously or in parallel based on their dependencies with other tasks in the Job. The Job is the primary interface to interact with the Task Framework to get the details of the job and the tasks within it. This could be the status of the job, the status of the tasks within it, etc.

When a Job is scheduled for execution, it is put into a queued state. This is true for a job that has to execute immediately or a job that is scheduled for later execution.

At the scheduled time when the task runs it is put into executing state. Once the task finishes its execution, it is considered as completed. The task framework then queries the task to check if the execution was successful or not. Based on this status, the task is marked as completed or failed. If the task is successful, the next task in the Job is executed. If the task fails, the appropriate fault policy action is taken.

The fault policy specifies the type of action to be taken as one of the following:

- **Retry:** Framework attempts to retry the task. Job data / data populated during the earlier run is supplied to the task before execution.
- **Rollback:** Framework rolls back the task.
- **Rollback Retry:** Framework rolls back the task and retries it.
- **Abort:** Framework aborts the task (and the Job).
- **Ignore:** Framework ignores the failure / timeout and proceeds with execution of subsequent tasks, if any, in the job.

Every task can define a timeout value which indicates the maximum estimated time for the task to complete. Beyond this time, the task is considered to have timed out and an appropriate fault policy action is taken on the task. The task framework monitors the executing tasks at periodic intervals of time to check whether they have timed out. If the fault policy indicates that a retry has to be done in case of a time out, the task framework retries the task.

## Query Job Instances for Job ID

Retrieves all job instances for the specified job ID. If a job is a one-time job, a single job instance is returned. If a job is a recurring job, all instances for the given job ID are returned.

### Example 18-1. Query job instances

Request Body:

GET <https://NSX-Manager-IP-Address/api/2.0/services/taskservice/job/jobId>

Response Body:

```
<jobInstances>
  <jobInstance>
    <id>jobinstance-1</id>
    <name>SVM Updater</name>
    <taskInstances>
      <taskInstance>
        <id>taskinstance-1</id>
        <name>SVM Updater</name>
        <startTimeMillis>1375867719752</startTimeMillis>
        <endTimeMillis>1375867720025</endTimeMillis>
        <taskStatus>COMPLETED</taskStatus>
        <timeoutRetryCount>0</timeoutRetryCount>
        <failureRetryCount>0</failureRetryCount>
        <taskOutput />
        <taskData />
      </taskInstance>
    </taskInstances>
    <startTimeMillis>1375867719663</startTimeMillis>
    <endTimeMillis>1375867720050</endTimeMillis>
    <status>COMPLETED</status>
    <timeoutRetryCount>0</timeoutRetryCount>
    <failureRetryCount>0</failureRetryCount>
  </job>
  <id>jobdata-1</id>
  <name>SVM Updater</name>
  <description>Updating all sdd SVMs at startup.</description>
  <creationTimeMillis>1375867718710</creationTimeMillis>
  <nextExecutionTimeMillis>0</nextExecutionTimeMillis>
  <taskList>
    <task>
      <id>task-1</id>
      <name>SVM Updater</name>
      <description>Updating all sdd SVMs at startup.</description>
      </description>
      <failurePolicy>
        <faultAction>RETRY</faultAction>
        <retryLimit>30</retryLimit>
        <retryInterval>60000</retryInterval>
      </failurePolicy>
      <timeoutPolicy>
        <faultAction>IGNORE</faultAction>
        <retryLimit>0</retryLimit>
        <retryInterval>-1</retryInterval>
      </timeoutPolicy>
      <priority>5</priority>
      <timeoutMillis>-1</timeoutMillis>
      <visible>false</visible>
      <systemTask>true</systemTask>
    </task>
  </taskList>
</jobInstances>
```

```

        <taskClass>com.vmware.vshield.dlp.service.impl.DlpServiceImpl$1
        </taskClass>
        <creationTimeMillis>1375867718729
        </creationTimeMillis>
        <jobId>jobdata-1</jobId>
        <nextExecutionTime>0</nextExecutionTime>
    </task>
</taskList>
<jobOwner>Unknown</jobOwner>
<scope>/globalroot-0</scope>
</job>
<jobOutput />
</jobInstance>
</jobInstances>

```

---

## Query Latest Job Instances for Job ID

In case of cron jobs or fixed-delay jobs, there can be multiple job instances for the same job depending upon the number of times the job was executed. This call fetches the latest job instance for a given job id.

### Example 18-2. Query job instances

---

Request Body:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/taskservice/job/jobId
```

Response Body:

See [Example 18-1](#)

---

## Block REST Thread

This is a blocking call where a service has scheduled a job and a REST thread needs to be blocked till the job gets completed. If the job was already completed, then the job instance is returned immediately. If the job is still executing then the REST thread is blocked and returns after the job completes.

### Example 18-3. Query job instances

---

Request Body:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/taskservice/job/jobId
```

Response Body:

See [Example 18-1](#).

---

## Query Job Instances by Criterion

You can specify filtering criteria and paging information and query the task framework.

### Example 18-4. Query job instances by criterion

---

Request Body:

```
GET https://NSX-Manager-IP-Address/api/2.0/services/taskservice/job/startIndex=<0>
    &pageSize=<10>&sortBy=startTime&sortOrderAscending=true/false
```

Response Body:

See [Example 18-1](#).

---

# vShield Endpoint Management

---



This chapter uses APIs for NSX 6.0 and earlier to describe vShield Endpoint solutions that were developed with ESXi Partner Program 5.0 or earlier – (for vShield 5.5 or earlier). These partner solutions are also supported on NSX 6.0 and need the APIs listed below. These APIs should not be used with partner solutions developed specifically for NSX 6.0 or later, as these newer solutions automate the registration and deployment process by using the new features introduced in NSX.

A vShield Endpoint appliance delivers an introspection-based antivirus solution that uses the hypervisor to scan guest virtual machines from the outside with only a thin agent on each guest virtual machine.

This chapter includes the following topics:

- [“Overview of Solution Registration”](#) on page 487
- [“Registering a Solution with vShield Endpoint Service”](#) on page 488
- [“Querying Registration Status of vShield Endpoint”](#) on page 489
- [“Querying Activated Security Virtual Machines for a Solution”](#) on page 490
- [“Unregistering a Solution with vShield Endpoint”](#) on page 491
- [“Status Codes and Error Schema”](#) on page 492

---

**IMPORTANT** All vShield REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

## Overview of Solution Registration

To register a third-party solution with vShield Endpoint, clients can use four REST calls to do the following:

- 1 Register the vendor.
- 2 Register one or more solutions.
- 3 Set the solution IP address and port (for all hosts).
- 4 Activate registered solutions per host.

**NOTE** Steps 1 through 3 need to be performed once per solution, while step 4 needs to be performed for each host.

To unregister a solution, clients essentially perform these steps in reverse:

- 1 Deactivate solutions per host.
- 2 Unset a solution’s IP address and port.
- 3 Unregister solutions.
- 4 Unregister the vendor.

To update registration information for a vendor or solution, clients must first unregister that entity and then reregister. The following sections detail the specific REST calls to perform registration and unregistration.

## Registering a Solution with vShield Endpoint Service

The APIs described in this section register a vendor, solutions, set network address, and activate solutions.

For a list of return status codes, see [“Return Status Codes”](#) on page 492.

### Register a Vendor

You can register the vendor of an antivirus solution.

#### Example 19-1. Register a vendor

---

Request:

POST `https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration`

Request Body:

```
<vendorInfo>
  <id>vendorId</id>
  <title>vendor title</title>
  <description>vendor description</description>
</VendorInfo>
```

---

In the request body, *vendorId* is the VMware-assigned ID for the vendor, while *vendor title* and *vendor description* are vendor provided strings.

### Register a Solution

You can register an antivirus solution.

#### Example 19-2. Register a solution

---

Request:

POST `https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration/vendorId`

Request Body:

```
<SolutionInfo>
  <altitude>solution altitude</altitude>
  <title>solution title</title>
  <description>solution description</description>
</SolutionInfo>
```

---

In the request, *vendorId* is the previously registered ID for the vendor.

In the request body, *solution altitude* is the VMware-assigned altitude for the solution, *solution title* and *solution description* are vendor provided strings. See [“Altitude of a Solution”](#) on page 488.

#### Altitude of a Solution

Altitude is a number that VMware assigns to uniquely identify the solution. The altitude describes the type of solution and the order in which the solution receives events relative to other solutions on the same host.

### IP Address and Port for a Solution

You can set a solution’s IP address and port on the vNIC host.

#### Example 19-3. Set IP address and port

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration/vendorId
/altitude/location
```

Request Body:

```
<LocationInfo>
  <ip>solution-ip-address</ip>
  <port>solution port</port>
</LocationInfo>
```

---

In the request, *vendorId* is the previously registered ID for the vendor, and *altitude* for the altitude.

In the request body, *solution-ip-address* is the solution's IPv4 address for the vNIC that is connected to the VMkernel port group (for example, 169.254.1.31). This address must be within the range of VMware-assigned IP addresses for the solution. The *solution port* is the port on which the solution accepts connections.

If you want to change the location of a solution, deactivate all security virtual machines, change the location, and then reactivate all security virtual machines.

## Activate a Solution

You can activate a solution that has been registered and located.

### Example 19-4. Activate solution

---

Request:

```
POST https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/activation/vendorId/altitude
```

Request Body:

```
<ActivationInfo>
  <moid>svm moid</moid>
</ActivationInfo>
```

---

In the request, *vendorId* is the previously registered ID for the vendor, and *altitude* for the altitude.

In the request body, *svm moid* is the managed object ID of the activated solution's virtual machine.

## Querying Registration Status of vShield Endpoint

You can use the same URLs shown in the previous section with the GET method to retrieve vendor registration information, solution registration information, location information, and solution activation status.

### Get Vendor Registration

You can retrieve vendor registration information.

#### Example 19-5. Get list of all registered vendors

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration/vendors
```

---

#### Example 19-6. Get vendor registration information

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration/vendorId
```

---

## Get Solution Registration

You can retrieve solution registration information.

**Example 19-7.** Get all registered solutions for a vendor

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration/vendorId/solutions
```

---

**Example 19-8.** Get solution registration information

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration/vendorId/altitude
```

---

## Get IP Address of a Solution

This call retrieves the IP address and port associated with a solution.

**Example 19-9.** Get IP address and port of a solution

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration/vendorId/altitude/location
```

---

## Get Activation Status of a Solution

This call retrieves solution activation status, given the managed object reference *moid* of its virtual machine.

**Example 19-10.** Get activation status of a solution

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/activation/vendorId/altitude/moid
```

---

Status can be false (not activated) or true (activated).

## Querying Activated Security Virtual Machines for a Solution

You can retrieve a list of activated security virtual machines for a solution, as well as the activation information for all activated security virtual machines on a host.

### Query Activated Security Virtual Machines

You can retrieve a list of activated security virtual machines for the specified solution.

**Example 19-11.** Get activated security virtual machines

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/activation/vendorId/solutionId
```

Response Body:

```

<ActivatedSVMs>
  <ActivationInfo>
    <moid>vm-819</moid>
    <hostMoid>host-9</hostMoid>
    <vmName>VMWARE-Data Security-10.24.130.174</vmName>
    <hostName>10.24.130.174</hostName>
    <clusterName>Dev</clusterName>
    <dcName>dev</dcName>
    <vendorId>VMWARE</vendorId>
    <solutionId>6341068275337723904</solutionId>
  </ActivationInfo>
  ...
</ActivatedSVMs>

```

---

In the request, *vendorId* is the VMware-assigned ID for the vendor, while *solutionId* is the solution ID.

## Query Activation Information

You can retrieve activation information for all activated security virtual machines on the specified host.

### Example 19-12. Get activation information

---

Request:

```
GET https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/activation?hostId=hostId
```

Response Body:

```

<ActivatedSVMs>
  <ActivationInfo>
    <moid>vm-819</moid>
    <hostMoid>host-9</hostMoid>
    <vmName>VMWARE-Data Security-10.24.130.174</vmName>
    <hostName>10.24.130.174</hostName>
    <clusterName>Dev</clusterName>
    <dcName>dev</dcName>
    <vendorId>VMWARE</vendorId>
    <solutionId>6341068275337723904</solutionId>
  </ActivationInfo>
  ...
</ActivatedSVMs>

```

---

## Unregistering a Solution with vShield Endpoint

You can use the same URIs shown in the first section with the DELETE method to unregister a vendor, unregister a solution, unset location information, or deactivate a solution.

### Unregister a Vendor

This call unregisters a vendor.

#### Example 19-13. Unregister a vendor

---

Request:

```
DELETE https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration/vendorId
```

---

### Unregister a Solution

This call unregisters a solution.

**Example 19-14.** Unregister a vendor

---

Request:

DELETE `https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration/vendorId/altitude`

---

**Unset IP Address**

This call unsets a solution's IP address and port.

**Example 19-15.** Unset IP address and port

---

Request:

DELETE `https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/registration/vendorId/altitude/location`

---

**Deactivate a Solution**

This call deactivates a solution on a host.

**Example 19-16.** Deactivate a solution

---

Request:

DELETE `https://NSX-Manager-IP-Address/api/2.0/endpointsecurity/activation/vendorId/altitude/moid`

---

**Status Codes and Error Schema**

This section lists various status codes returned from the REST API, and shows the error schema.

**Return Status Codes**

The 200 codes indicate success, the 400 codes indicate some failure, and the 600 codes are call specific.

- 200 OK operation successful
- 201 Created: Entity successfully altered.
- 400 Bad Request: Internal error codes. Please refer to the Error Schema for more details.
- 401 Unauthorized: Incorrect user name or password.
- 600 Unrecognized vendor ID.
- 601 Vendor is already registered.
- 602 Unrecognized altitude.
- 603 Solution is already registered.
- 604 Invalid IPv4 address.
- 605 Invalid port.
- 606 Port out of range.
- 607 Unrecognized moid.
- 608 Location information is already set.
- 609 Location not set.
- 612 Solutions still registered.
- 613 Solution location information still set.
- 614 Solution still activated.

- 615 Solution not activated.
- 616 Solution is already activated.
- 617 IP:Port already in use.
- 618 Bad solution ID.
- 619 vShield Endpoint is not licensed.
- 620 Internal error.

## Error Schema

Here is the XML schema for vShield Endpoint registration errors.

```
<error>  
  <details>Some error has occurred.</details>  
  <errorCode>601</errorCode>  
</error>
```



## vCenter Object IDs

---

This section describes how to retrieve the IDs for the objects in your virtual inventory.

The chapter includes the following topics:

- [“Query Datacenter MOID”](#) on page 495
- [“Query Datacenter ID”](#) on page 495
- [“Query Host ID”](#) on page 495 authentication
- [“Query Portgroup ID”](#) on page 496
- [“Query VMID”](#) on page 496

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 27 for details about basic authorization.

---

### Query Datacenter MOID

- 1 In a web browser, type the following:

`http://vCenter-IP-Address/mob`

- 2 Click content.
- 3 Click on the rootFolder value.
- 4 Click on the childEntity value.

The datacenter MOID is displayed on top of the window.

### Query Datacenter ID

- 1 In a web browser, type the following:

`http://vCenter-IP-Address/mob`

- 2 Click content.
- 3 Click on the rootFolder value.
- 4 Click on the childEntity value.

The datacenter value is the datacenter ID.

### Query Host ID

- 1 In a web browser, type the following:

`http://vCenter-IP-Address/mob`

- 2 Click content.
- 3 Click on the rootFolder value.
- 4 Click on the childEntity value.
- 1 Click on the datacenter value.

The host value is the host ID.

## Query Portgroup ID

- 1 In a web browser, type the following:

`http://vCenter-IP-Address/mob`

- 2 Click content.
- 3 Click on the rootFolder value.
- 4 Click on the childEntity value.
- 5 Click on the datacenter value.
- 6 Click on the host value.

The network property value is the portgroup ID.

## Query VMID

- In a web browser, type the following:

`http://vCenter-IP-Address/mob`

The VMID is listed under host structure.

## Deprecated APIs

The following APIs have been deprecated in the NSX 6.0 release.

**Table 21-1.** Deprecated APIs

Deprecated API	Alternate API(s)
Local user management	
/api/2.0/global/heartbeat	/api/1.0/appliance-management/global/info
/api/2.0/global/config	/api/2.0/services/vcconfig /api/2.0/services/ssoconfig /api/1.0/appliance-management/system/network/dns /api/1.0/appliance-management/system/timesettings
/api/2.0/global/vcInfo	/api/2.0/services/vcconfig
/api/2.0/global/ techsupportlogs	/api/1.0/appliance-management/techsupportlogs/NSX
/api/2.0/vdn/map/cluster/ clusterId	
/api/2.0/services/usermgmt/ securityprofile	



# Appendix A: Schemas

---

The REST API configuration of the vShield Edge and vShield App virtual machines supports schemas for installation and service management.

This appendix covers the following topics:

- [“Firewall Schemas”](#) on page 499
- [“Deprecated: vShield Manager Global Configuration Schema”](#) on page 501
- [“Deprecated: ESX Host Preparation and Uninstallation Schema”](#) on page 506
- [“Deprecated: vShield App Schemas”](#) on page 507
- [“Error Message Schema”](#) on page 513

## Firewall Schemas

### Firewall Configuration Schema

```
<xs:element name="firewallConfiguration" type="FirewallConfigurationDto">
</xs:element>

<xs:complexType name="FirewallConfigurationDto">
  <xs:sequence>
    <xs:element name="layer3Sections" type="FirewallLayer3SectionsDto"
      maxOccurs="1" minOccurs="1" />
    <xs:element name="layer2Sections" type="FirewallLayer2SectionsDto"
      maxOccurs="1" minOccurs="1" />
  </xs:sequence>
  <xs:attribute name="contextId" type="xs:string" use="required" />
  <xs:attribute name="timestamp" type="xs:long" use="optional" />
  <xs:attribute name="generationNumber" type="xs:long" use="optional" />
</xs:complexType>
```

## Firewall Section Schema

```
<xs:complexType name="FirewallLayer3SectionsDto">
  <xs:sequence>
    <xs:element name="section" type="FirewallSectionDto"
      maxOccurs="unbounded" minOccurs="1">
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="FirewallLayer2SectionsDto">
  <xs:sequence>
    <xs:element name="section" type="FirewallSectionDto"
      maxOccurs="unbounded" minOccurs="1">
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="FirewallSectionDto">
  <xs:sequence>
    <xs:element name="rule" type="FirewallRuleDto" maxOccurs="unbounded"
      minOccurs="0" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:long" use="optional" />
  <xs:attribute name="name" type="xs:string" use="required" />
  <xs:attribute name="timestamp" type="xs:long" use="optional" />
  <xs:attribute name="generationNumber" type="xs:string"
    use="optional" />
</xs:complexType>
```

## Firewall Sections Schema

```

<xs:complexType name="FirewallRuleDto">
  <xs:sequence>
    <xs:element name="appliedToList" type="AppliedToListDto" />
    <xs:element name="sources" type="FirewallSourcesDto" />
    <xs:element name="destination" type="FirewallDestinationsDto" />
    <xs:element name="services" type="FirewallServicesDto" />
    <xs:element name="action" type="xs:string" />
    <xs:element name="logged" type="xs:boolean" />
    <xs:element name="notes" type="xs:string" minOccurs="0" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:long" use="optional" />
  <xs:attribute name="disabled" type="xs:boolean" use="optional" />
  <xs:attribute name="precedence" type="xs:string" use="optional" />
</xs:complexType>

```

## Deprecated: vShield Manager Global Configuration Schema

The following schema shows vShield Manager REST configuration.

This replaces the 1.0 API schema items for vCenter synchronization, DNS service, virtual machine information, and security groups.

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="vmware.vshield.edge.2.0"
  xmlns:vse="vmware.vshield.edge.2.0"
  elementFormDefault="qualified">

  <xs:element name="nsxmgrGlobalConfig">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="vshieldEdgeReleaseInfo"
          type="vse:ReleaseInfoType"/> <!-- In response from server -->
        <xs:element minOccurs="0" name="vcInfo" type="vse:VCInfoType" />
        <xs:element minOccurs="0" name="hostInfo" type="vse:HostInfoType" />
        <xs:element minOccurs="0" name="techSupportLogsTarFilePath"
          type="xs:string"/>
        <xs:element minOccurs="0" name="auditLogs" type="vse:AuditLogsType" />
        <xs:element minOccurs="0" name="dnsInfo" type="vse:DnsInfoType" />
        <xs:element minOccurs="0" name="versionInfo" type="xs:string" /> <!--
          only in response -->
        <xs:element minOccurs="0" name="vpnLicensed" type="xs:boolean" /> <!--
          only in response -->
        <xs:element minOccurs="0" name="ipsecVpnTunnels" type="vse:IpsecVpnTunnels"
          /> <!-- only in response -->
        <xs:element minOccurs="0" maxOccurs="1" name="nsxmgrCapability"
          type="vse:nsxmgrCapabilityType"/>
        <!-- only in response -->
        <xs:element minOccurs="0" maxOccurs="1" name="timeInfo"
          type="vse:TimeInfoType"/>
      </xs:sequence>
    </xs:complexType>

```





```

    </xs:sequence>
  </xs:complexType >

  <xs:complexType name="SecurityGroupIdList">
    <xs:sequence>
      <xs:element name="securityGroupId" type="xs:string" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="IpList">
    <xs:sequence>
      <xs:element name="ip" type="xs:string" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="NodeList">
    <xs:sequence>
      <xs:element name="node" type="vse:SecurityGroupNode" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SecurityGroupNode">
    <xs:sequence>
      <xs:element name="id" type="xs:string" />
      <xs:element name="name" type="xs:string" minOccurs="0" />
      <xs:element name="ipList" type="vse:IpList" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Vnicstype">
    <xs:sequence>
      <xs:element name="vnic" type="vse:VnicType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="VnicType">
    <xs:sequence>
      <xs:element name="id" type="xs:string" />
      <xs:element name="name" type="xs:string" />
      <xs:element name="ipList" type="vse:IpList" minOccurs="0" maxOccurs="1"/>
      <!--will be good if we can also send this information
      <xs:element name="VLAN" type="xs:int" />
      <xs:element name="PortGroup" type="xs:string" />
      <xs:element name="Protected" type="xs:boolean"/> -->
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="AuditLogstype">
    <xs:sequence>
      <xs:element name="auditLog" type="vse:AuditLogType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="DnsInfoType">
    <xs:sequence>
      <xs:element name="primaryDns" type="xs:string"/>
      <xs:element minOccurs="0" name="secondaryDns" type="xs:string"/>
      <xs:element minOccurs="0" name="tertiaryDns" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="AuditLogType">
    <xs:sequence>
      <xs:element name="id" type="xs:string" />
      <xs:element name="userName" type="xs:string" />
      <xs:element name="accessInterface" type="xs:string" />
      <xs:element name="module" type="xs:string" />
      <xs:element name="operation" type="xs:string" />
      <xs:element name="status" type="xs:string" />
    </xs:sequence>
  </xs:complexType>

```

```

    <xs:element name="operationSpan" type="xs:string" />
    <xs:element name="resource" type="xs:string" />
    <xs:element name="timestamp" type="xs:string" />
    <xs:element name="notes" type="xs:string" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IpsecVpnTunnels">
  <xs:sequence>
    <xs:element name="lastEventId" type="xs:unsignedInt" />
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ipsecVpnTunnelStatusList"
      type="vse:IpsecVpnTunnelStatus" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IpsecVpnTunnelStatus">
  <xs:sequence>
    <xs:element name="networkId" type="xs:string" />
    <xs:element name="ipsecVpnTunnelConfig" type="vse:IpsecVpnTunnelConfigType" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IpsecVpnTunnelConfigType"> <!--only in response -->
  <xs:sequence>
    <xs:element name="peerName">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="1"/>
          <xs:maxLength value="256"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="peerId" type="xs:string" />
    <xs:element name="peerIpAddress" type="xs:string" />
    <xs:element maxOccurs="64" name="localSubnet" type="xs:string" /> <!--
      localSubnet * peerSubnet * noOfSites should not be more than 64 -->
    <xs:element maxOccurs="64" name="peerSubnet" type="xs:string" /> <!--
      localSubnet * peerSubnet * noOfSites should not be more than 64 -->
    <xs:element name="authenticationMode" >
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="((psk)|(x.509))"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element minOccurs="0" name="preSharedKey" type="xs:string" />
    <xs:element minOccurs="0" name="encryptionAlgorithm" type="xs:string" />
    <xs:element minOccurs="0" name="mtu" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="status" type="xs:string" />
    <xs:element minOccurs="0" name="stateChangeReason" type="xs:string" />
  </xs:sequence>
</xs:complexType>

  <xs:complexType name="nsxmgrCapabilityType">
    <xs:sequence>
      <xs:element name="ipsecVpnCapability" type="xs:boolean"/>
      <xs:element name="webLoadBalancerCapability" type="xs:boolean"/>
      <xs:element name="natCapability" type="xs:boolean"/>
      <xs:element name="firewallCapability" type="xs:boolean"/>
      <xs:element name="dhcpCapability" type="xs:boolean"/>
      <xs:element name="staticRoutingCapability" type="xs:boolean"/>
      <xs:element name="nsxmgrVersion" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="TimeInfoType">
    <xs:sequence>
      <xs:element minOccurs="0" name="clock" type="xs:string"/>

```

```

        <xs:element minOccurs="0" name="ntpServer" type="xs:string"/>
        <xs:element minOccurs="0" name="zone" type="xs:string"/>
    </xs:sequence>
</xs:complexType>

</xs:schema>

```

## Deprecated: ESX Host Preparation and Uninstallation Schema

This schema can be used to install or uninstall vShield App and vShield Endpoint services on an ESX host.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

    <xs:element name="vshieldConfiguration">
        <xs:complexType>
            <xs:all>
                <xs:element minOccurs="0" name="vszInstallParams"
                    type="vszInstallParams"/>
                <xs:element minOccurs="0" name="EpssecInstallParams" type="xs:boolean"/>
                <xs:element name="InstallAction" type="InstallAction"/> <!--
                    InstallAction to be taken on appliance - install/upgrade
                    -->
                <xs:element name="InstallStatus" type="InstallStatus"/> <!-- only in
                    response -->
            </xs:all>
        </xs:complexType>
    </xs:element>

    <xs:complexType name="InstallStatus">
        <xs:sequence>
            <xs:element minOccurs="0" name="ProgressState" type="xs:string"/>
            <xs:element minOccurs="0" name="ProgressSubState" type="xs:string"/>
            <xs:element minOccurs="0" name="InstalledServices" type="InstalledServices"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="InstalledServices">
        <xs:sequence>
            <xs:element name="vszInstalled" type="xs:boolean"/>
            <xs:element name="EpssecInstalled" type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>

    <!-- Install parameters -->
    <xs:complexType name="vszInstallParams">
        <xs:sequence>
            <xs:element name="DatastoreId" type="Moid"/>
            <xs:element name="ManagementPortSwitchId" type="xs:string"/> <!-- contains
                the networkId of the mgmt portgroup -->
            <xs:element name="MgmtInterface" type="MgmtInterfaceType"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="MgmtInterfaceType">
        <xs:sequence>
            <xs:element name="IpAddress" type="IP"/>
            <xs:element name="NetworkMask" type="IP"/>
            <xs:element name="DefaultGw" type="IP"/>
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="InstallAction">
        <xs:restriction base="xs:string">
            <xs:enumeration value="install"/>
            <xs:enumeration value="upgrade"/>
        </xs:restriction>
    </xs:simpleType>

```

```

</xs:simpleType>

<xs:simpleType name="IP">
  <xs:restriction base="xs:string">
    <xs:pattern value="
      ((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.){3}(25[0-5]|2[
0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="Moid">
  <xs:restriction base="xs:string">
    <xs:pattern value="[a-zA-Z0-9\-\+]"/>
  </xs:restriction>
</xs:simpleType>

</xs:schema>

```

## Deprecated: vShield App Schemas

The following schemas detail vShield App configuration via REST API.

### vShield App Configuration Schema

This schema configures a vShield App after installation.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <xs:element name="ZonesConfiguration">
    <xs:complexType>
      <xs:all>
        <xs:element name="VszInstallParams" type="VszInstallParams"
          minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>

  <!-- Install parameters -->
  <xs:complexType name="VszInstallParamsType">
    <xs:sequence>
      <xs:element name="NodeId" type="xs:string"/>
      <xs:element name="DatacenterId" type="xs:string"/>
      <xs:element name="DatastoreId" type="xs:string"/>
      <xs:element name="NameForZones" type="xs:string"/>
      <xs:element name="VswitchForMgmt" type="xs:string"/>
      <xs:element name="MgmtInterface" type="InterfaceType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="InterfaceType">
    <xs:sequence>
      <xs:element name="IpAddress" type="xs:NMTOKEN"/>
      <xs:element name="NetworkMask" type="xs:NMTOKEN"/>
      <xs:element name="DefaultGw" type="xs:NMTOKEN"/>
      <xs:element minOccurs="0" name="VlanTag" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>

```

### vShield App Firewall Schema

This schema configures the firewall rules enforced by a vShield App.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" >

```

```

<xs:element name="vshieldAppConfiguration">
  <xs:complexType>
    <xs:choice>
      <xs:element name="firewallConfiguration" type="FirewallConfigurationDto"
        />
      <xs:element name="firewallConfigurationHistoryList"
        type="FirewallConfigHistoryInfoListDto" />
      <xs:element name="consolidatedConfiguration"
        type="FirewallConfigurationDto" maxOccurs="unbounded" />
      <xs:element name="status" type="StatusDto" />
      <xs:element name="datacenterState" type="DatacenterStateDto" />
      <xs:element name="protocolsList" type="ProtocolListDto" />
      <xs:element name="protocolTypes" type="ProtocolTypeEnum" maxOccurs="4" />
    </xs:choice>
  </xs:complexType>
</xs:element>

<xs:complexType name="FirewallConfigHistoryInfoListDto">
  <xs:sequence>
    <xs:element name="contextId" type="xs:string" />
    <xs:element name="firewallConfigHistoryInfo"
      type="FirewallConfigHistoryInfoDto" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="FirewallConfigHistoryInfoDto">
  <xs:sequence>
    <xs:element name="configId" type="xs:long" />
    <xs:element name="userId" type="xs:string" />
    <xs:element name="timestamp" type="xs:long" />
    <xs:element name="status" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DatacenterStateDto">
  <xs:sequence>
    <xs:element name="datacenterId" type="xs:string" />
    <xs:element name="userId" type="xs:string" minOccurs="0" />
    <xs:element name="timestamp" type="xs:long" minOccurs="0" />
    <xs:element name="status" type="DatacenterStatusEnum" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="StatusDto">
  <xs:sequence>
    <xs:element name="currentState" type="ConfigStateEnum" />
    <xs:element name="failedPublishInfo" type="FailedPublishInfoDto"
      maxOccurs="unbounded" minOccurs="0" />
  </xs:sequence>
  <xs:attribute name="contextId" type="xs:string" use="required" />
  <xs:attribute name="generationNumber" type="xs:long" />
</xs:complexType>

<xs:complexType name="FailedPublishInfoDto">
  <xs:sequence>
    <xs:element name="applianceIp" type="xs:string" />
    <xs:element name="timestamp" type="xs:long" />
    <xs:element name="errorDescription" type="xs:string" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="FirewallConfigurationDto">
  <xs:sequence>
    <xs:element name="layer3FirewallRule" type="Layer3FirewallRuleDto"
      maxOccurs="unbounded" minOccurs="0" />
    <xs:element name="layer2FirewallRule" type="Layer2FirewallRuleDto"
      maxOccurs="unbounded" minOccurs="0" />
  </xs:sequence>
</xs:complexType>

```

```

    </xs:sequence>
    <xs:attribute name="provisioned" type="xs:boolean" use="optional" />
    <xs:attribute name="contextId" type="xs:string" use="required" />
    <xs:attribute name="timestamp" type="xs:long" use="optional" />
    <xs:attribute name="generationNumber" type="xs:long" use="optional" />
  </xs:complexType>

  <xs:complexType name="ApplicationDto">
    <xs:choice>
      <xs:element name="applicationSetId" type="xs:string" />
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="DestinationDto" abstract="true">
    <xs:sequence>
      <xs:element name="address" type="AddressDto" minOccurs="0" />
      <!-- Only in response, not considered in request -->
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Layer2DestinationDto">
    <xs:complexContent>
      <xs:extension base="DestinationDto">
        <xs:element name="application" type="ApplicationDto" minOccurs="0" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="Layer3DestinationDto">
    <xs:sequence>
      <xs:element name="address" type="AddressDto" minOccurs="0" />
      <xs:element name="application" type="ApplicationDto" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Layer3SourceAddressDto">
    <xs:sequence>
      <xs:element name="address" type="AddressDto" minOccurs="0" />
      <xs:element name="portInfo" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="FirewallRuleDto" abstract="true">
    <xs:sequence>
      <xs:element name="action" type="ActionEnum" />
      <xs:element name="logged" type="xs:boolean" />
      <xs:element name="notes" type="xs:string" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:long" use="required" />
    <xs:attribute name="precedence" type="PrecedenceEnum" use="optional" />
    <xs:attribute name="disabled" type="xs:boolean" use="optional" />
  </xs:complexType>

  <xs:complexType name="Layer2FirewallRuleDto">
    <xs:complexContent>
      <xs:extension base="FirewallRuleDto">
        <xs:sequence>
          <xs:element name="source" type="AddressDto" minOccurs="0" />
          <xs:element name="destination" type="Layer2DestinationDto" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="Layer3FirewallRuleDto">
    <xs:complexContent>
      <xs:extension base="FirewallRuleDto">
    </xs:extension>
  </xs:complexContent>

```

```

        <xs:sequence>
          <xs:element name="source" type="Layer3SourceAddressDto" minOccurs="0" />
          <xs:element name="destination" type="Layer3DestinationDto" minOccurs="0" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AddressDto">
    <xs:choice>
      <xs:element name="containerId" type="xs:string" minOccurs="0">
      </xs:element>
    </xs:choice>
    <xs:attribute name="exclude" type="xs:boolean" use="optional" default="false" />
  </xs:complexType>

  <xs:simpleType name="ActionEnum">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="allow" />
      <xs:enumeration value="deny" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PrecedenceEnum">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="default" />
      <xs:enumeration value="none" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ConfigStateEnum">
    <xs:restriction base="xs:NCName">
      <!-- <xs:enumeration value="saved" /> -->
      <xs:enumeration value="published" />
      <xs:enumeration value="inprogress" />
      <xs:enumeration value="publishFailed" />
      <xs:enumeration value="deleted" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="DatacenterStatusEnum">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="upgrading" />
      <xs:enumeration value="backwardCompatible" />
      <xs:enumeration value="backwardCompatibleReadyForSwitch" />
      <xs:enumeration value="migrating" />
      <xs:enumeration value="regular" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ProtocolsTypeEnum">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="application" />
      <xs:enumeration value="ipv4" />
      <xs:enumeration value="icmp" />
      <xs:enumeration value="ethernet" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

## vShield App SpoofGuard Schema

The following schema details SpoofGuard configuration.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

```

```

<xs:element name="vshieldConfiguration">
  <xs:complexType>
    <xs:choice>
      <xs:element name="globalSettings" type="GlobalSettingsDto" />
      <xs:element name="ipAssignmentStatistic" type="IpAssignmentStatisticDto" />
      <xs:element name="vnicIdList" type="VnicIdListDto" />
      <xs:element name="ipAssignmentDetailsList" type="IpAssignmentDetailsListDto" />
      <xs:element name="pagedIpAssignmentDetailsList" type="PagedIpAssignmentDetailsListDto" />
      <xs:element name="approveIpInfo" type="VnicInfoDto" />
    </xs:choice>
  </xs:complexType>
</xs:element>

<xs:complexType name="PagedIpAssignmentDetailsListDto">
  <xs:sequence>
    <xs:element name="ipAssignmentDetails" type="IpAssignmentDetailsDto" maxOccurs="unbounded" />
    <xs:element name="pagingDetails" type="PagingInfoDto" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PagingInfoDto">
  <xs:sequence>
    <xs:element name="pageSize" type="xs:int" />
    <xs:element name="startIndex" type="xs:int" />
    <xs:element name="totalCount" type="xs:int" />
    <xs:element name="sortOrderAscending" type="xs:boolean" />
    <xs:element name="sortBy" type="PagingSortByEnum" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IpAssignmentDetailsListDto">
  <xs:sequence>
    <xs:element name="ipAssignmentDetails" type="IpAssignmentDetailsDto" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

  <xs:complexType name="IpAssignmentDetailsDto">
    <xs:sequence>
      <xs:element name="vnicId" type="xs:string" />
      <xs:element name="macAddress" type="xs:string" />
      <xs:element name="ipAddress" type="xs:string" />
      <xs:element name="vnicName" type="xs:string" />
      <xs:element name="networkId" type="xs:string" />
      <xs:element name="vmId" type="xs:string" />
      <xs:element name="vmName" type="xs:string" />
      <xs:element name="approvedIpAddress" type="xs:string" />
      <xs:element name="approvedBy" type="xs:string" />
      <xs:element name="approvedOn" type="xs:long" />
      <xs:element name="publishedIpAddress" type="xs:string" />
      <xs:element name="publishedBy" type="xs:string" />
      <xs:element name="publishedOn" type="xs:long" />
      <xs:element name="reviewRequired" type="xs:boolean" />
      <xs:element name="duplicateCount" type="xs:int" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="IpAssignmentStatisticDto">
    <xs:sequence>
      <xs:element name="contextId" type="xs:string" />
      <xs:element name="inSync" type="xs:boolean" />
      <xs:element name="activeCount" type="xs:long" />
      <xs:element name="inactiveCount" type="xs:long" />
    </xs:sequence>
  </xs:complexType>

```

```

        <xs:element name="activeSinceLastPublishedCount" type="xs:long" />
        <xs:element name="requireReviewCount" type="xs:long" />
        <xs:element name="duplicateCount" type="xs:long" />
        <xs:element name="unpublishedCount" type="xs:long" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="VnicIdListDto">
    <xs:sequence>
        <xs:element name="vnicId" type="xs:string" maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="VnicInfoDto">
    <xs:sequence>
        <xs:element name="vnicId" type="xs:string" />
        <xs:element name="ipAddress" type="xs:string" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="GlobalSettingsDto">
    <xs:sequence>
        <xs:element name="status" type="OperationStatusEnum" />
        <xs:element name="mode" type="OperationModeEnum" />
        <!-- optional parameters will be part of response only -->
        <xs:element name="timestamp" type="xs:long" minOccurs="0" />
        <xs:element name="publishedBy" type="xs:string" minOccurs="0" />
    </xs:sequence>
</xs:complexType>

<xs:simpleType name="OperationStatusEnum">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="enabled" />
        <xs:enumeration value="disabled" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="OperationModeEnum">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="trustOnFirstUse" />
        <xs:enumeration value="manual" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PagingSortByEnum">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="VM_NAME" />
        <xs:enumeration value="MAC" />
        <xs:enumeration value="APPROVED_IP" />
        <xs:enumeration value="CURRENT_IP" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

## vShield App Namespace Schema

The following schema details namespace configuration.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="vmware.vshield.global.20.namespace"
    xmlns:vsns="vmware.vshield.global.20.namespace"
    elementFormDefault="qualified">

    <xs:element name="vshieldConfiguration">
        <xs:complexType>
            <xs:choice>

```

```

        <xs:element maxOccurs="unbounded" name="namespace"
            type="vsns:NamespaceDto" />
        <xs:element maxOccurs="3" name="namespaceType"
            type="vsns:NamespaceTypeEnum" />
    </xs:choice>
</xs:complexType>
</xs:element>

<xs:complexType name="NamespaceDto">
    <xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="namespacePortGroup"
    type="vsns:PortGroupDto" />
</xs:sequence>
<xs:attribute name="type" use="required" type="vsns:NamespaceTypeEnum" />
<xs:attribute name="id" use="optional" type="xs:long" />
</xs:complexType>

<xs:complexType name="PortGroupDto">
<xs:sequence>
<xs:element maxOccurs="1" name="Id" type="xs:string" />
</xs:sequence>
</xs:complexType>

<xs:simpleType name="NamespaceTypeEnum">
<xs:restriction base="xs:NCName">
<xs:enumeration value="DEFAULT" />
<xs:enumeration value="PORTGROUP" />
<xs:enumeration value="NONE" />
</xs:restriction>
</xs:simpleType>

</xs:schema>Retrieved from
    "https://wiki.eng.vmware.com/NS_DEV/vShieldManager/nsxmgr30/App/ipad/xsd"

```

## Error Message Schema

This schema details error messages.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

    <xs:element name="Errors">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="Error" type="ErrorType"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:complexType name="ErrorType">
        <xs:sequence>
            <xs:element name="code" type="xs:unsignedInt"/>
            <xs:element name="description" type="xs:string"/>
            <xs:element minOccurs="0" name="detailedDescription" type="xs:string"/>
            <xs:element minOccurs="0" name="index" type="xs:int"/>
            <xs:element minOccurs="0" name="resource" type="xs:NMTOKEN"/>
            <xs:element minOccurs="0" name="requestId" type="xs:NMTOKEN"/>
            <xs:element minOccurs="0" name="module" type="xs:NMTOKEN"/>
        </xs:sequence>
    </xs:complexType>

</xs:schema>

```

If a REST API call results in an error, the HTTP reply contains the following information.

- An XML error document as the response body
- Content-Type: application/xml
- An appropriate 2xx, 4xx, or 5xx HTTP status code

**Table 22-1.** Error Message Status Codes

<b>Code</b>	<b>Description</b>
200 OK	The request was valid and has been completed. Generally, this response is accompanied by a body document (XML).
201 Created	The request was completed and new resource was created. The Location header of the response contains the URI of newly created resource.
204 No Content	Same as 200 OK, but the response body is empty (No XML).
400 Bad Request	The request body contains an invalid representation or the representation of the entity is missing information. The response is accompanied by Error Object (XML).
401 Unauthorized	An authorization header was expected. Request with invalid or no vShield Manager Token.
403 Forbidden	The user does not have enough privileges to access the resource.
404 Not Found	The resource was not found. The response is accompanied by Error Object (XML).
500 Internal Server Error	Unexpected error with the server. The response is accompanied by Error Object (XML).
503 Service Unavailable	Cannot proceed with the request, because some of the services are unavailable. Example: vShield Edge is Unreachable. The response is accompanied by Error Object (XML).